TI TOTAL

ÁREA FISCAL E CONTROLE

Professor
Ramon Souza

Tecnologia da Informação

SQL (DDL)

Questões Comentadas

SUMÁRIO

1. QUESTÕES COMENTADAS

1.1 CESPE/CEBRASPE

1- (CESPE / CEBRASPE - 2021 - APEX Brasil - Analista - Tecnologia da Informação e Comunicação)

create database pessoa;

O comando SQL apresentado anteriormente criará

a) um banco de dados denominado pessoa;.

b) uma tabela denominada pessoa;.

c) um tipo de dados denominado pessoa;.

d) um esquema denominado pessoa;.

Resolução:

O comando CREATE DATABASE é usado para criar bancos de dados. Assim, o comando trazido na questão, irá realizar a criação de um banco de dados chamado "pessoa".

Gabarito: **Letra A**.

2- (CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Ciência de Dados) Julgue o item a seguir, a respeito de conceitos de SQL.

O comando CREATE DATABASE TAB é utilizado para criar uma tabela em um banco de dados.

Resolução:

O comando para criar uma tabela é CREATE TABLE e não CREATE DATABASE.

O comando CREATE DATABASE é usado para criar bancos de dados. Assim, o comando trazido na questão, irá realizar a criação de um banco de dados chamado "TAB".

Para criar uma tabela, o comando correto seria:

CREATE TABLE TAB

Gabarito: **Errado**.

3- (CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Desenvolvimento de Software)

Tendo como referência o diagrama de entidade relacionamento precedente, julgue o próximo item, a respeito de linguagem de definição de dados e SQL.

A expressão SQL a seguir permite excluir as notas do aluno de nome Fulano.

truncate from matricula where aluno='Fulano'

Resolução:

TRUNCATE é para deletar todos os dados de uma tabela. Caso se deseje excluir somente registros específicos, então deve se usar o comando DELETE.

Ademais, o nome do aluno não está na tabela matrícula, mas sim na tabela aluno, então para deletar é precisa buscar o id desse aluno. O correto seria:

DELETE FROM matricula WHERE aluno = (SELECT id FROM aluno WHERE nome = 'Fulano');

Gabarito: **Errado**.

4- (CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Desenvolvimento de Software)

Tendo como referência o diagrama de entidade relacionamento precedente, julgue o próximo item, a respeito de linguagem de definição de dados e SQL.

As expressões DDL a seguir permitem a criação das tabelas presentes no diagrama apresentado.

```
create table aluno (
id integer primary key,
nome varchar(40) );
create table disciplina (
id integer primary key,
descricao varchar(60)
);

create table matricula (
aluno integer,
disciplina integer,
ano integer,
nota numeric,
```

constraint pk_matricula primary key (aluno,

disciplina, ano),

constraint fk_matricula_aluno foreign key

(aluno)

references aluno,

constraint fk_matricula_disciplina foreign

key (disciplina)

references disciplina);

Resolução:

Vamos analisar os comandos por partes:

Criação da tabela aluno com os atributos id e nome, sendo id definido como chave primária:

CREATE TABLE aluno (id integer **PRIMARY KEY**, nome varchar(40));

Criação da tabela disciplina com os atributos id e descrição, sendo id definido como chave primária:

CREATE TABLE disciplina (id integer **PRIMARY KEY**, descrição varchar(60));

Criação da tabela matrícula com os atributos aluno, disciplina, ano e nota:

CREATE TABLE matricula (aluno integer, disciplina integer, ano integer, nota numeric,

Definição das restrições na tabela matrícula, sendo a definição da chave primária formada pelos atributos aluno, disciplina e ano:

CONSTRAINT pk_matricula **PRIMARY KEY** (aluno, disciplina, ano),

A chave estrangeira no campo aluno apontando para a tabela aluno:

CONSTRAINT fk_matricula_aluno **FOREING KEY** (aluno) **REFERENCES** aluno,

A chave estrangeira no campo disciplina apontando para a tabela disciplina:

CONSTRAINT fk_matricula_disciplina **FOREING KEY** (disciplina) **REFERENCES** disciplina);

Logo, o comando está sintaticamente correto e é capaz de criar as tabelas representadas no modelo.

Gabarito: **Certo**.

5- (CESPE - 2018 - FUB - Técnico de Tecnologia da Informação) Julgue o item subsecutivo, a respeito de linguagem de definição e manipulação de dados.

O comando DROP TABLE permite excluir do banco de dados a definição de uma tabela e de todos os seus dados.

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

DROP TABLE nome_da_tabela;

Gabarito: **Certo**.

6- (CESPE - 2018 - BNB - Especialista Técnico - Analista de Sistema) Acerca de bancos de dados, julgue o item que segue.

O código a seguir, criado no SQL Server 2017, apresenta uma visão materializada, especificamente devido ao argumento

SCHEMABINDING.

```
CREATE VIEW VwTeste
WITH SCHEMABINDING
AS
SELECT campo1 FROM tabela WHERE campo1 > 17;
```

Resolução:

Para se criar uma visão materializada, deve ser utilizada a instrução CREATE MATERIALIZED VIEW.

A opção avançada SCHEMABINDING serve para associar a exibição ao esquema da tabela ou tabelas subjacentes. Quando SCHEMABINDING for especificado, a tabela ou tabelas base não poderão ser modificadas de um modo que possam afetar a definição da exibição.

Gabarito: **Errado**.

7- (CESPE - 2018 - STJ - Técnico Judiciário - Desenvolvimento de Sistemas)

Julgue o item a seguir, referente à modelagem de dados.

A DDL (data definition language) é usada para a definição da estrutura do banco de dados ou do esquema. São comandos DDL: CREATE, TRUNCATE, GRANT e ROLLBACK.

Resolução:

Os comandos da DDL (Data Definition Language) são CREATE, ALTER e DROP (ou TRUNCATE). GRANT e REVOKE são comandos da DCL (Data Control Language).

Gabarito: **Errado**.

8- (CESPE - 2017 - TRE-PE - Analista Judiciário - Análise de Sistemas)

Tabela 3A6AAA

dados da tabela:

ID; nome; idtipo; preco

25; creme; 3; 11,50

31; arroz; 4; 12,50

34; leite; 1; 14,00

42; sabão; 5; 11,00

46; carne; 1; 12,75

48; shampoo; 5; 12,30

58; azeite; 1; 13,25

Assinale a opção que apresenta o comando SQL correto para se incluir um novo campo idcategoria do tipo INT nos dados da tabela 3A6AAA, denominada tbproduto.

a) ALTER TABLE tbproduto INSERT idcategoria INT;

b) ALTER TABLE tbproduto ADD COLUMN idcategoria INT;

c) UPDATE TABLE tbproduto ADD COLUMN idcategoria INT;

d) ADD COLUMN idcategoria INT IN TABLE tbprodut;

e) UPDATE TABLE ADD COLUMN idcategoria INT IN tbproduto;

Resolução:

Como queremos inserir um novo campo em uma tabela, então devemos usar o comando ALTER TABLE. Logo, eliminamos c), d) e e).

Para **adicionar uma coluna**, usamos a cláusula **ADD ou ADD COLUMN**:

ALTER TABLE nome_da_tabela

ADD nome_da_coluna tipo_de_dado;

Assim, a sintaxe correta está na letra b).

Gabarito: **Letra B**.

9- (CESPE - 2016 - FUB - Técnico de Tecnologia da Informação) A respeito das principais instruções da linguagem SQL, julgue o item subsecutivo.

A instrução create assertion <nome-asserção> check <predicado> é utilizada para definir restrições de integridade.

Resolução:

A instrução CREATE ASSERTION é usada para definir restrições de integridade.

A sintaxe create assertion <nome-asserção> check <predicado> é válida para criar a restrição, sendo que no predicado, você deve colocar qual a condição que deve ser observada.

Gabarito: **Certo**.

10- (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Suporte) No que concerne à linguagem SQL, julgue o item seguinte.

O comando create table pode ser utilizado para criar tanto uma tabela vazia quanto uma com dados de outra tabela.

Resolução:

Perfeitamente, o comando CREATE TABLE pode definir uma tabela sem dados ou usar alguma outra tabela como base.

A sintaxe básica dessa instrução é:

```
CREATE TABLE nome_da_tabela (
   coluna1 tipo_de_dado,
   coluna2 tipo_de_dado,
   ....
);
```

Ao utilizar a DDL em conjunto com a DML é possível criar uma tabela a partir de outra tabela existente. Para isso basta usar o auxílio da cláusula AS conforme sintaxe a seguir:

```
CREATE TABLE nome_da_nova_tabela AS
   SELECT coluna1, coluna2,...
   FROM nome_da_tabela_existente
   WHERE ....;
```

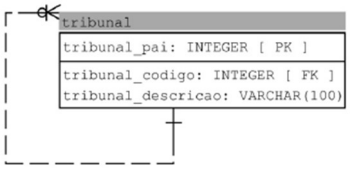
Gabarito: **Certo**.

11- (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Sistema) Julgue o próximo item, relativo à linguagem de definição de dados (DDL).

A expressão DDL abaixo cria a tabela referente ao diagrama de entidade e relacionamento apresentado a seguir.

```
create table tribunal(
tribunal_codigo integer ,
tribunal_descricao varchar(100),
tribunal_pai integer primary key,
constraint fk_tribunal
foreign key (tribunal_codigo )
references tribunal
```

)



Resolução:

Vamos analisar o comando por partes:

Criação da tabela tribunal:

CREATE TABLE tribunal (

Com os atributos tribunal_codigo, tribunal_descricao e tribunal_pai, sendo este definido como chave primária:

tribunal_codigo integer,

tribunal_descricao varchar(100),

tribunal_pai integer **PRIMARY KEY**,

Com uma restrição de chave estrangeira que faz referência a própria tabela, isto é, declara um autorrelacionamento:

CONSTRAINT fk_tribunal

FOREIGN KEY (tribunal_codigo)

REFERENCES tribunal

)

Gabarito: **Certo**.

12- (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Suporte) No que concerne à linguagem SQL, julgue o item seguinte.

Ao se criar uma view, não é necessário que os nomes dos atributos da view sejam os mesmos dos atributos da tabela.

Resolução:

Os nomes dos atributos nas visões podem ser definidos com bases nos alias ou apelidos. Sendo assim, não é necessário que o nome dos atributos da visão sejam os mesmos.

No comando a seguir, os atributos da visão serão renomeados.

```
CREATE VIEW [Mexicanos] AS
  SELECT Nome_Cliente AS Meu_Cliente, Cidade AS Município
  FROM Clientes
  WHERE Pais="Mexico";
```

Gabarito: **Certo.**

13- (CESPE - 2016 - POLÍCIA CIENTÍFICA - PE - Perito Criminal - Ciência da Computação) Em SQL, para alterar a estrutura de uma tabela do banco de dados e incluir nela uma nova foreign key, é correto utilizar o comando

a) convert

b) group by.

c) alter table.

d) update.

e) insert.

Resolução:

A instrução **ALTER TABLE** é usada para **adicionar, deletar ou modificar colunas em uma tabela existente**. Essa instrução também pode ser utilizada para adicionar ou deletar restrições a esta tabela.

Para adicionar uma restrição **FOREIGN KEY** em uma cláusula ALTER:

- **Para uma coluna**:

ALTER TABLE nome_da_tabela

ADD FOREIGN KEY(coluna) **REFERENCES** tabela_referenciada (chave)**;**

- **Para múltiplas colunas**:

ALTER TABLE nome_da_tabela

ADD CONSTRAINT nome_da_restricao **FOREIGN KEY**(coluna1, coluna2) **REFERENCES** tabela_referenciada (chave1, chave2);

Gabarito: **Letra C**.

14- (CESPE - 2016 - POLÍCIA CIENTÍFICA - PE - Perito Criminal - Ciência da Computação) Na linguagem SQL, o comando create table é usado para criar uma tabela no banco de dados; enquanto o relacionamento entre duas tabelas pode ser criado pela declaração

a) null. b) primary key. c) constraint.

d) auto_increment. e) not null.

Resolução:

O relacionamento é realizado através da chave estrangeira ou foreign key. A chave estrangeira pode ser definida em uma cláusula CONSTRAINT da seguinte forma:

CONSTRAINT nome_da_restricao **FOREIGN KEY** (coluna1, coluna2)) **REFERENCES** tabela_referenciada (chave1, chave2);

Gabarito: **Letra C**.

15- (CESPE - 2015 - MEC – Desenvolvedor) Com relação à linguagem de definição de dados (DDL) e à linguagem de manipulação de dados (DML), julgue o próximo item.

A DML utiliza o comando CREATE para inserir um novo registro na tabela de dados.

Resolução:

Para inserir dados em uma tabela, o comando utilizado é o INSERT INTO, que faz parte da DML.

O comando CREATE faz parte da DDL e é usado para criar as estruturas do banco de dados, como as tabelas, visões e outros elementos.

Gabarito: **Errado.**

16- (CESPE - 2015 - MEC – Desenvolvedor)

```
CREATE TABLE PESSOA (
ID INTEGER NOT NULL,
NOME CHAR(50) NOT NULL UNIQUE,
CPF DECIMAL (11,0) NULL,
NACIONALIDADE INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (NACIONALIDADE)
REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)
);
```

Com base no comando SQL apresentado, julgue o item subsequente.

A cláusula NULL na coluna CPF indica que o conteúdo dessa coluna pode ser zero, já que ela é do tipo DECIMAL (11,0).

Resolução:

A cláusula NULL indica que o conteúdo de CPF pode ser nulo. Nulo é diferente de zero.

Ao comparar qualquer coisa com NULL usando os operadores lógicos comuns, será retornado um resultado desconhecido na comparação (UNKNOWN).

Gabarito: **Errado.**

17- (CESPE - 2015 - MEC – Desenvolvedor)

```
CREATE TABLE PESSOA (
ID INTEGER NOT NULL,
NOME CHAR(50) NOT NULL UNIQUE,
CPF DECIMAL (11,0) NULL,
NACIONALIDADE INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (NACIONALIDADE)
REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)
);
```

Com base no comando SQL apresentado, julgue o item subsequente.

A tabela TABELA_NACIONALIDADE deve ter uma coluna de nome DESCRICAO_NACIONALIDADE para obter o texto equivalente a cada código.

Resolução:

Não há obrigatoriedade quanto ao nome da coluna e nem mesmo quanto a sua existência.

O mais comum é que haja campos descritivos nas tabelas, mas isso não é obrigatório.

Gabarito: **Errado.**

18- (CESPE - 2015 - MEC – Desenvolvedor)

```
CREATE TABLE PESSOA (
ID INTEGER NOT NULL,
NOME CHAR(50) NOT NULL UNIQUE,
CPF DECIMAL (11,0) NULL,
NACIONALIDADE INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (NACIONALIDADE)
REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)
);
```

Com base no comando SQL apresentado, julgue o item subsequente.

Na tabela TABELA_NACIONALIDADE, CODIGO_NACIONALIDADE deve ser PRIMARY KEY.

Resolução:

Pessoal, essa é uma daquelas típicas divergências que costumam pegar na hora da prova e infelizmente não temos uma conclusão bem definida, pois as bancas costumam adotar ambos os posicionamentos.

Como a chave primária é uma escolha (mais ou menos arbitrária) dentre as chaves candidatas, basta que a chave estrangeira referencie uma chave **CANDIDATA** de outra tabela e não necessariamente uma chave primária. Porém, várias são as questões que dão como correta a afirmação de que a chave candidata referencia ou deve referenciar a chave **PRIMÁRIA** da outra tabela.

Esse é o caso dessa questão. Geralmente a linha a seguir é:

- Se a questão afirmar que a chave estrangeira pode referenciar a chave candidata, aceite como **CORRETO**.
- Se a questão afirmar que a chave estrangeira deve referenciar a chave primária ou deve fazer parte da chave primária, aceite como **CORRETO**.
- Se a questão for mais explícita e afirmar que a chave estrangeira referencia somente a chave primária ou indicar que não pode referenciar uma chave candidata, marque **ERRADO**.

Gabarito: **Certo.**

19- (CESPE - 2015 - MEC – Desenvolvedor) CREATE TABLE PESSOA (

ID INTEGER NOT NULL,

NOME CHAR(50) NOT NULL UNIQUE,

CPF DECIMAL (11,0) NULL,

NACIONALIDADE INTEGER NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (NACIONALIDADE)

REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)

);

Com base no comando SQL apresentado, julgue o item subsequente.

Mais de uma PESSOA pode ter o mesmo NOME e a mesma NACIONALIDADE.

Resolução:

O NOME da PESSOA está definido como UNIQUE, logo não pode ser repetido para mais de um registro.

A restrição **UNIQUE** garante que **todos os valores em uma coluna sejam diferentes**.

Gabarito: **Errado.**

20- (CESPE - 2015 - MEC – Desenvolvedor)

CREATE TABLE PESSOA (

ID INTEGER NOT NULL,

NOME CHAR(50) NOT NULL UNIQUE,

CPF DECIMAL (11,0) NULL,

NACIONALIDADE INTEGER NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (NACIONALIDADE)

REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)

);

Com base no comando SQL apresentado, julgue o item subsequente.

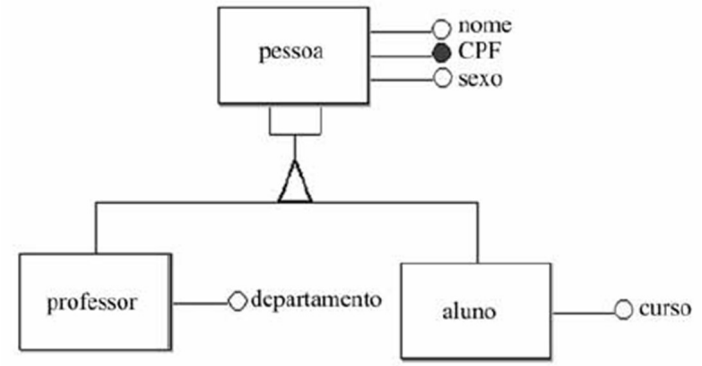
A tabela criada terá quatro colunas.

Resolução:

A tabela criada terá as seguintes colunas: ID, NOME, CPF e NACIONALIDADE.

Gabarito: **Certo.**

21- (CESPE - 2015 - TJ-DFT - Técnico Judiciário - Programação de Sistemas)



Com base no diagrama de entidade e relacionamento (DER) apresentado, julgue o item que se segue a respeito de modelagem de dados e linguagem de definição dos dados.

A seguir é apresentado o comando SQL correto para gerar o esquema físico do DER.

CREATE TABLE Pessoa (

Nome VARCHAR(50),

Sexo VARCHAR(1),

CPF VARCHAR(11),

Inheritance (aluno, pessoa))

CREATE TABLE Aluno (

Curso VARCHAR(20),

CPF VARCHAR(11) PRIMARY KEY)

CREATE TABLE Professor (

Departamento VARCHAR(30),

CPF VARCHAR(11) PRIMARY KEY)

Resolução:

Galera, aqui temos uma declaração interessante: Inheritance.

Esse comando serve para indicar que aluno e professor herdam de pessoa. Logo, todos os atributos e relacionamentos de PESSOA já vão também para ALUNO e PROFESSOR.

Desse modo, o comando está errado, pois não é válido declarar CPF novamente nas tabelas ALUNO e PROFESSOR, pois este atributo já foi declarado em PESSOA e é herdado por estas entidades. Como não é possível ter dois atributos com o mesmo nome em uma mesma tabela, então o comando é inválido.

Gabarito: **Errado.**

22- (CESPE / CEBRASPE - 2015 - TRE-MT - Técnico Judiciário - Programação de Sistemas) create table departamento (

coddepto integer primary key, nomedepto varchar(100)

);

create table professor (

codprof integer primary key,

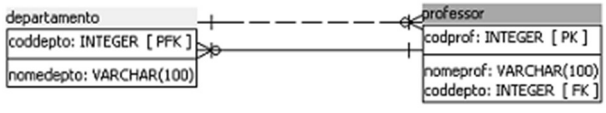
nomeprof varchar(100),

coddepto integer,

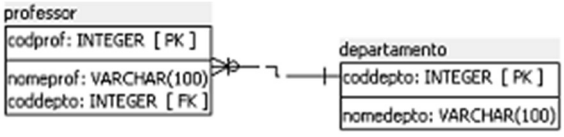
constraint fkprofessor foreign key (coddepto) references departamento);

Assinale a opção correta acerca do diagrama de entidade e relacionamento que representa a declaração relativa ao código SQL apresentado acima.

a)



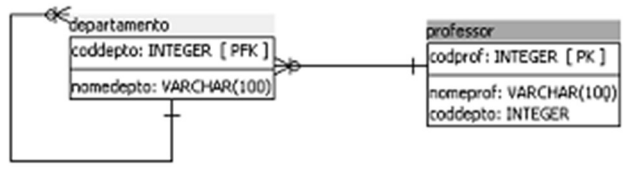
b)



c)

d)

e)



Resolução:

Vamos analisar o comando por partes:

Criação da tabela departamento com os atributos coddepto e nomedepto, sendo coddepto a chave primária:

create table departamento (coddepto integer **primary key**, nomedepto varchar(100));

Criação da tabela professor com os atributos codprof, nomeprof e coddepto, sendo codprof a chave primária:

create table professor (codprof integer **primary key**, nomeprof varchar(100), coddepto integer,

Com uma restrição de chave estrangeira em coddepto que aponta para a tabela departamento:

constraint fkprofessor foreign key (coddepto) references departamento);

Logo, teremos duas tabelas, sendo que professor possui uma chave estrangeira para o departamento a que está ligado.

Gabarito: **Letra B.**

23- (CESPE - 2015 - MEC - Administrador de Dados) Julgue o item que se segue, com relação às definições e aos problemas de execução de comandos nas linguagens SQL.

Uma operação DELETE do SQL não é realizada se sua chave primária for referida por chaves estrangeiras em registros de outras tabelas no banco de dados. Assim, a fim de garantir a existência de chaves primárias para cada chave estrangeira nos bancos de dados relacionais, o SQL não apresenta nenhuma cláusula ou opção adicional que permita tal operação ocorrer nessa situação.

Resolução:

Existe a opção ON DELETE CASCATE.

Ao usar a opção **ON DELETE CASCADE**, quando um registro da tabela que possui a chave primária associada a esta chave estrangeira for excluído, então os registros associados também são excluídos. Ex.: ao excluir um determinado País, todos os Estados que estão associados àquele País são também deletados (a associação é verificada pela chave estrangeira).

CONSTRAINT nome_da_restricao **FOREIGN KEY** (coluna1, coluna2)) **REFERENCES** tabela_referenciada (chave1, chave2) **ON DELETE CASCADE**;

Gabarito: **Errado.**

24- (CESPE - 2014 - ANATEL - Analista Administrativo - Tecnologia da Informação e Comunicação) Nos comandos em linguagem de consulta estruturada (SQL) apresentados a seguir, as chaves primárias estão sublinhadas e apenas horas_gastas é do tipo numérico, os demais campos são do tipo caractere. Em uma tarefa, com a utilização de um mesmo veículo, pode haver a participação de mais de um motorista na função de auxiliar.

MOTORISTA (cod_mot, nome, cnh)

TAREFA (cod_tarefa, cod_mot, cod_mot_auxiliar, placa, descricao, horas_gastas)

VEÍCULO (placa, ano, modelo)

Tendo como base as informações acima, julgue o item a seguir.

O comando a seguir exclui a chave primária, cod_tarefa, da tabela TAREFA.

Alter table tarefa drop primary key cod_tarefa;

Resolução:

O comando exclui apenas a restrição, mas não a coluna em si.

Para a coluna em si, seria necessário usar o comando DROP COLUMN.

Gabarito: **Errado.**

25- (CESPE - 2014 - ANATEL - Analista Administrativo - Desenvolvimento de Sistemas) Julgue os itens seguintes, a respeito das linguagens de banco de dados.

A DDL (data definition language) é responsável pela especificação da instância do banco de dados e também pode ser usada para especificar propriedades adicionais dos dados, como restrições de consistência.

Resolução:

A DDL não especifica a instância, mas sim o esquema do banco de dados.

Gabarito: **Errado.**

26- (CESPE - 2014 - SUFRAMA - Analista de Sistemas) No que se refere a linguagem de implementação de banco de dados, julgue os itens subsequentes.

O comando drop table remove toda a tabela da base de dados. Um exemplo de utilização desse comando é o seguinte: drop table examplo_timestamp;

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

DROP TABLE nome_da_tabela;

Essa instrução irá deletar todos os dados da tabela, bem como a própria tabela.

Gabarito: **Certo.**

27- (CESPE - 2013 - BACEN - Analista - Análise e Desenvolvimento de Sistemas) Julgue os itens seguintes, a respeito das linguagens de banco de dados.

```
CREATE TABLE Pessoa
(
Id int NULL,
Matricula int NOT NULL,
Nome varchar(255) NOT NULL,
DataNascimento date NULL)

CREATE TABLE EnderecoPessoa
(Id int NOT NULL,
TipoEndereco char (1) NOT NULL,
Endereco varchar(255),
Cidade char(55),
UF varchar (2)
)
```

Considerando os scripts acima para criação das Tabelas Pessoa e EnderecoPessoa, julgue o item seguinte.

Considerando que o campo Id na tabela Pessoa esteja corretamente configurado como chave primária simples, para se criar uma chave estrangeira entre as tabelas Pessoa e EnderecoPessoa, deve-se executar o comando a seguir.

```
ALTER TABLE EnderecoPessoa
ADD CONSTRAINT fk_Endereco_Pessoa FOREIGN KEY (P_id) REFERENCES Pessoa (Id)
```

Resolução:

O comando está incorreto, pois está tentando definir uma chave estrangeira para um campo inexistente: P_id. Note que este campo não foi definido na criação da tabela PESSOA.

Gabarito: **Errado.**

28- (CESPE - 2013 - BACEN - Analista - Análise e Desenvolvimento de Sistemas) Julgue os itens seguintes, a respeito das linguagens de banco de dados.

```
CREATE TABLE Pessoa
(
Id int NULL,
Matricula int NOT NULL,
Nome varchar(255) NOT NULL,
DataNascimento date NULL)

CREATE TABLE EnderecoPessoa
(Id int NOT NULL,
TipoEndereco char (1) NOT NULL,
Endereco varchar(255),
Cidade char(55),
UF varchar (2)
)
```

Considerando os scripts acima para criação das Tabelas Pessoa e EnderecoPessoa, julgue o item seguinte.

Para criar uma chave primária composta na Tabela Pessoa, deve-se executar o seguinte comando.

```
ALTER TABLE Pessoa ADD CONSTRAINT pk_PessoaID PRIMARY KEY (Id, Matricula)
```

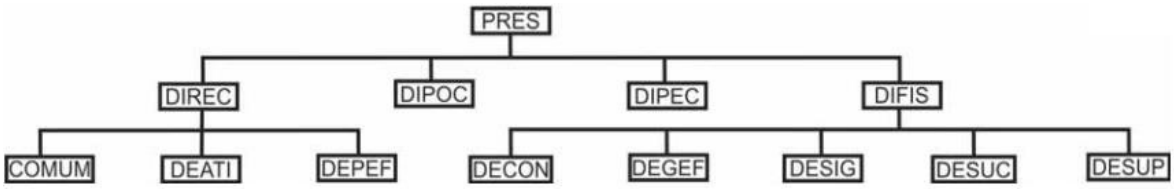
Resolução:

Ao usar o comando ALTER TABLE para adicionar uma chave primária, os campos devem ter sido declarados como NOT NULL.

Logo, o campo Id não pode ser chave primária, pois é definido como NULL.

Gabarito: **Errado.**

29- (CESPE - 2013 - BACEN - Analista - Suporte à Infraestrutura de Tecnologia da Informação)



A expressão SQL abaixo cria uma tabela com estrutura que permite armazenar informações acerca dos órgãos e sua hierarquia.

create table orgao (

codigo integer,

sigla char(10),

codigo_pai integer,

constraint pk_orgao primary key (codigo),

constraint fk_orgao foreign key (codigo_pai) references orgao

)

Resolução:

Vamos analisar o comando:

Criação da tabela órgão:

create table orgao (

Com os campos codigo, sigla e codigo_pai:

codigo integer,

sigla char(10),

codigo_pai integer,

Sendo a chave primária codigo:

constraint pk_orgao primary key (codigo),

E a chave estrageira codigo_pai:

constraint fk_orgao foreign key (codigo_pai) references orgao

)

Logo, há o estabelecimento de um autorrelacionamento entre as ocorrências da tabela órgão, podendo assim um órgão se relacionar a outro, formando o organograma.

Gabarito: **Certo.**

30- (CESPE / CEBRASPE - 2013 - CNJ - Técnico Judiciário - Programação de Sistemas) No que se refere ao conceito de banco de dados relacional, julgue os itens seguintes.

Na linguagem de consulta estruturada (SQL), é correto utilizar o comando TRUNCATE TABLE, com a finalidade de excluir todos os dados de uma tabela.

Resolução:

Para **excluir apenas os dados da tabela, sem excluir a estrutura dessa tabela**. Para isso, poderá usar o comando **TRUNCATE**:

TRUNCATE TABLE nome_da_tabela;

Gabarito: **Certo.**

31- (CESPE - 2013 - TRT - 10ª REGIÃO (DF e TO) - Técnico Judiciário - Tecnologia da Informação) Em relação aos comandos da linguagem SQL, julgue os itens seguintes.

O comando abaixo permite adicionar a tabela disciplinas a uma chave estrangeira com o nome fk_curso, do campo id_curso que pertence à tabela cursos.

alter table disciplinas

alter column fk_curso references cursos (id_curso);

Resolução:

Pessoal, muita atenção, pois não há de se falar em adicionar uma tabela a uma chave estrangeira. Muito pelo contrário, o que se pode é adicionar uma chave estrangeira a uma tabela.

A sintaxe para adicionar uma restrição **FOREIGN KEY** em uma cláusula ALTER:

- **Para uma coluna**:

ALTER TABLE nome_da_tabela

ADD FOREIGN KEY(coluna) **REFERENCES** tabela_referenciada (chave);

- **Para múltiplas colunas**:

ALTER TABLE nome_da_tabela

ADD CONSTRAINT nome_da_restricao **FOREIGN KEY**(coluna1, coluna2) **REFERENCES** tabela_referenciada (chave1, chave2);

Gabarito: **Errado.**

32- (CESPE - 2008 - STJ - Técnico Judiciário - Informática) Acerca da linguagem SQL, usada para fazer a manipulação e a definição de dados em sistemas gerenciadores de banco de dados, julgue os itens subsequentes.

O comando CREATE INDEX, usado para criar um parâmetro relacionado com uma tabela para buscar dados mais rapidamente, é considerado como DDL.

Resolução:

Sim, o comando CREATE INDEX é comando DDL. Esse comando permite criar um índice e possui a seguinte sintaxe básica.

```
CREATE INDEX nome_do_indice
ON nome_da_tabela (coluna1, coluna2, ...);
```

Gabarito: **Certo**.

1.2 FCC

33- (FCC - 2019 - SANASA Campinas - Analista de Tecnologia da Informação - Suporte de DBA-Banco de Dados) Atenção: Para responder à questão, considere os dados abaixo.

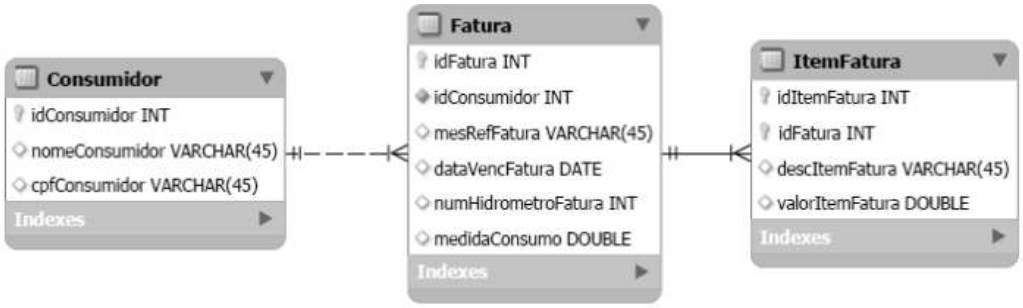


Tabela `Consumidor`:

idConsumidor	nomeConsumidor	cpfConsumidor
1	Paulo Vieira Lima	156.167.178-2
2	Marcos Santana Silva	234.156.765-12
3	Maria de Fátima Caetano Rosa	187.198.056-7
4	Zoraide Pereira Mota	238.765.234-12

Tabela `Fatura`:

idFatura	idConsumidor	mesRefFatura	DataVencFatura	numHidrometroFatura	medidaConsumo
3365693	1	05/2019	2019-05-20	344036	80
3366691	2	05/2019	2019-05-10	345681	120
3367690	1	04/2019	2019-04-28	344036	89
3390871	2	03/2019	2019-03-18	345681	100

Tabela `ItemFatura`:

idItemFatura	idFatura	descItemFatura	valorItemFatura
1	3365693	Captação de água bruta	0.1
1	3366691	Tratamento de água	90
1	3367690	Tratamento de água	61.23
2	3365693	Tratamento de água	60.19
2	3366691	Tratamento de esgoto	67.67
3	3365693	Coleta de esgoto	47.35
3	3366691	Afastamento de esgoto	80
4	3365693	Tratamento de Esgoto	25

Considere que não há nenhum registro cadastrado além dos mostrados nas tabelas acima.

Para excluir da tabela Fatura o campo mesRefFatura deve-se utilizar o comando

a) DROP COLUMN mesRefFatura FROM Fatura;

b) ALTER TABLE Fatura DROP COLUMN mesRefFatura;

c) DELETE COLUMN mesRefFatura FROM Fatura;

d) ALTER TABLE Fatura DELETE COLUMN mesRefFatura;

e) DROP COLUMN mesRefFatura WHERE TABLE= 'Fatura';

Resolução:

Para **deletar uma coluna**, usamos a cláusula **DROP COLUMN**:

ALTER TABLE nome_da_tabela

DROP COLUMN nome_da_coluna;

Gabarito: **Letra B**.

34- (FCC - 2019 - SABESP - Estagiário Ensino Médio Regular) Considere os comandos SQL abaixo.

```
I ..... (
   nome_Manancial varchar(50) PRIMARY KEY,
   volume number NOT NULL,
   media_historica number,
   data date
);

II ..... ('Cantareira', 58.0, , '12-05-2019');
```

Para que o primeiro comando crie a tabela MANANCIAIS e o segundo comando insira dados nesta tabela, deve-se preencher as lacunas I e II, correta e respectivamente, com

a) CREATE MANANCIAIS AS TABLE - INSERT INTO MANANCIAIS

b) CREATE TABLE MANANCIAIS - INSERT INTO MANANCIAIS VALUES

c) CREATE TABLE MANANCIAIS - INSERT IN TABLE MANANCIAIS

d) CREATE TABLE NAMED MANANCIAIS - INSERT INTO TABLE MANANCIAIS VALUES

e) CREATE MANANCIAIS AS TABLE - INSERT IN MANANCIAIS

Resolução:

O comando para criação de tabelas é CREATE TABLE nome_da_tabela, logo para criar uma tabela MANANCIAIS, devemos ter o comando;

CREATE TABLE MANANCIAIS

Para inserir dados em uma tabela, usa-se o comando INSERT INTO tabela VALUES (valor1, ...). Logo,

INSERTO INTO MANANCIAIS **VALUES** ('Cantareira', '58.0', , '12-05-2019');

Gabarito: **Letra B**.

35- (FCC - 2019 - SEFAZ-BA - Auditor Fiscal - Tecnologia da Informação - Prova II) Em um banco de dados aberto e em condições ideais há uma tabela chamada Contribuinte cuja chave primária é idContribuinte. Há também uma tabela chamada Imposto cuja chave primária é idimposto. Para criar uma tabela de associação chamada Contribuinte_imposto cuja chave primária é composta pelos campos idContribuinte e idImposto, que são chaves estrangeiras resultantes da relação dessa tabela com as tabelas Contribuinte e Imposto, utiliza-se a instrução SQL

a) CREATE TABLE Contribuinte_Imposto(idContribuinte INT, idImposto INT, PRIMARY KEY (idContribuinte), FOREIGN KEY (idContribuinte) REFERENCES Contribuinte (idContribuinte), PRIMARY KEY (idImposto), FOREIGN KEY (idContribuinte) REFERENCES Contribuinte (idContribuinte));

b) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), CONSTRAINT fk1 FOREIGN KEY (idContribuinte) REFERENCES Contribuinte (idContribuinte), CONSTRAINT fk2 FOREIGN KEY (idImposto) REFERENCES Imposto (idImposto));

c) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), FOREIGN KEY (idContribuinte) SOURCE Contribuinte (idContribuinte), FOREIGN KEY (idImposto) SOURCE Imposto (idImposto));

d) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), FOREIGN KEY (idContribuinte, idImposto) REFERENCES (Contribuinte!idContribuinte, Imposto!idImposto));

e) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), FOREIGN KEY (idContribuinte, idImposto) REFERENCES all parents);

Resolução:

Vamos analisar cada um dos itens:

a) **Incorreto**: a restrição PRIMARY KEY deve ser única na tabela. Caso a chave fosse simples, poderia ser adotada a definição PRIMARY KEY (atributo).

b) **Correto**: esse comando realiza a criação da tabela conforme solicitado no item. Vamos explicar o comando por partes:

Criação da tabela com os dois atributos solicitados:

CREATE TABLE Contribuinte_Imposto(idContribuinte **INT NOT NULL,** idImposto **INT NOT NULL**,

Definição da chave primária formada pelos dois atributos:

PRIMARY KEY (idContribuinte, idImposto),

Definição das chaves estrangeiras:

CONSTRAINT fk1 **FOREIGN KEY** (idContribuinte) **REFERENCES** Contribuinte (idContribuinte), **CONSTRAINT** fk2 **FOREIGN KEY** (idImposto) **REFERENCES** Imposto (idImposto));

Nesse caso temos a definição de duas restrições fk1 e fk2, uma para cada parte da chave estrangeira.

c) **Incorreto**: para definir uma chave primária com mais de um atributo, deve-se inserir uma **CONSTRAINT**. Ademais, para relacionar uma chave estrangeira com a tabela referenciada deve-se utilizar a palavra **REFERENCES** e não **SOURCE**.

d) **Incorreto**: para definir uma chave primária com mais de um atributo, deve-se inserir uma **CONSTRAINT**. Ademais, como os atributos relacionados por meio da chave estrangeira são de tabelas diferentes, deve-se utilizar mais de uma cláusula **FOREIGN KEY**.

e) **Incorreto**: mesma justificativa do item d, além de não existir essa referência a all parents, pois devem ser informados as tabelas e atributos sendo referenciados de acordo com a sintaxe **nome_da_tabela (atributo)**.

Gabarito: **Letra B**.

36- (FCC - 2019 - SANASA Campinas - Analista de Tecnologia da Informação - Suporte de DBA-Banco de Dados) Considere o código SQL abaixo, que gerou a tabela ItemFatura.

```
CREATE TABLE ItemFatura(
idItemFatura INT NOT NULL,
idFatura INT NOT NULL,
descItemFatura VARCHAR(45),
valorItemFatura DOUBLE,
..I..
);
```

Considerando que a tabela ItemFatura possui chave primária composta pelos campos idItemFatura e idFatura, e que se uma fatura for excluída, automaticamente serão excluídos todos os seus itens, a lacuna I deve ser preenchida corretamente por

a) PRIMARY KEY (idItemFatura, FOREIGN KEY(idFatura)) REFERENCES Fatura(idFatura) ON DELETE CASCADE

b) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) WITH DELETE CASCADE

c) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY (idFatura) REFERENCES Fatura(idFatura) ON DELETE CASCADE

d) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) REFERENCES Fatura(idFatura)

e) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) ON DELETE CASCADE

Resolução:

Ao usar a opção ON DELETE CASCADE, quando um registro da tabela que possui a chave primária associada a esta chave estrangeira for excluído, então os registros associados também são excluídos.

Dito isto, vamos analisar cada um dos itens:

a) **Incorreto**: PRIMARY KEY (idItemFatura**, idFatura)**, FOREIGN KEY(idFatura~~)~~ REFERENCES Fatura(idFatura) ON DELETE CASCADE

b) **Incorreto**: PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) ~~WITH~~ **ON** DELETE CASCADE

c) **Correto**: PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY (idFatura) REFERENCES Fatura(idFatura) ON DELETE CASCADE está de acordo com a sintaxe aplicável.

d) **Incorreto**: PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) REFERENCES Fatura(idFatura) **ON DELETE CASCADE**

e) **Incorreto**: PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) ~~FROM~~ **REFERENCES** Fatura(idFatura) ON DELETE CASCADE

Gabarito: **Letra C**.

37- (FCC - 2019 - TRF - 4ª REGIÃO - Analista Judiciário - Sistemas de Tecnologia da Informação) Uma Analista digitou o comando TRUNCATE TABLE processos; em um banco de dados SQL aberto em condições ideais para

a) excluir os dados da tabela, mas não a tabela em si.

b) excluir a estrutura da tabela e os dados nela contidos.

c) juntar a tabela aberta na memória com a tabela processos.

d) bloquear a tabela processos para uso exclusivo de seu usuário.

e) editar a estrutura da tabela em modo gráfico.

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

DROP TABLE nome_da_tabela;

Essa instrução irá deletar todos os dados da tabela, bem como a própria tabela.

Contudo, você pode desejar **excluir apenas os dados da tabela, sem excluir a estrutura dessa tabela**. Para isso, poderá usar o comando **TRUNCATE**:

TRUNCATE TABLE nome_da_tabela;

Gabarito: **Letra A**.

38- (FCC - 2018 - SABESP - Estagiário - Nível Médio Técnico) Considere que foi criada uma tabela em um banco de dados relacional capaz de armazenar os dados de clientes da SABESP, usando o comando SQL: CREATE TABLE Clientes_Sabesp (Cliente VARCHAR (50), MedidorAnt INTEGER NOT NULL, MedidorAtual INTEGER NOT NULL, Periodo DATE NOT NULL, Conta DECIMAL (10,2), PRIMARY KEY (Cliente));

É correto afirmar que

a) a instrução PRIMARY KEY define que o campo Cliente será a chave primária simples da tabela e este campo não pode ser nulo nem repetido.

b) com exceção do campo Cliente todos os outros campos são do tipo numérico inteiro.

c) para criar uma chave estrangeira, basta acrescentar FOREIGN KEY (Cliente) após a definição da PRIMARY KEY.

d) a chave estrangeira de um banco de dados relacional é usada para criar relacionamentos com as demais tabelas do banco de dados, por isso deve ter o mesmo nome da chave primária seguido de _FK. Nesta tabela seria Cliente_FK.

e) há um erro neste comando: todos os campos da tabela devem ser NOT NULL e os campos Cliente e Conta não têm esta restrição.

Resolução:

Vamos analisar cada um dos itens:

a) **Correto**: A restrição **PRIMARY KEY identifica exclusivamente cada registro em uma tabela**. As chaves primárias devem conter valores UNIQUE e não podem conter valores NULL.

b) **Incorreto**: com exceção do campo Cliente**, Período e Conta,** todos os outros campos são do tipo numérico inteiro.

c) **Incorreto**: para criar uma chave estrangeira, basta acrescentar FOREIGN KEY (Cliente) após a definição ~~da PRIMARY KEY~~ **de uma CONSTRAINT**.

d) **Incorreto**: a chave estrangeira não precisa ter o mesmo nome da chave primária relacionada.

e) **Incorreto**: podem existir campos nulos.

Gabarito: **Letra A**.

39- (FCC - 2018 - DPE-AM - Analista em Gestão Especializado de Defensoria - Analista de Banco de Dados) O comando SQL–ANSI para criar um procedimento chamado P1, que selecione os atributos A e B, de uma tabela T é:

a) PROCEDURE P1 IS

SELECT A, B

FROM T;

b) INSERT PROCEDURE P1 INTO DATABASE AS

SELECT A, B

FROM T

c) CREATE PROCEDURE P1( )

SELECT A, B

FROM T;

d) MAKE PROCEDURE P1 (SELECT A, B

FROM T);

e) PROCEDURE P1 AS

SELECT A, B

FROM T

Resolução:

Para criar uma **PROCEDURE**, basta utilizar a seguinte sintaxe:

CREATE PROCEDURE nome_da_procedure

AS

declaracoes_SQL

GO;

Nesse caso, como queremos criar a PROCEDURE P1 com base na seleção dos atributos A e B da tabela T, usamos:

CREATE PROCEDURE P1()

AS

SELECT A, B

FROM T;

GO;

Os () vazios indicam que a PROCEDURE não recebe nenhum parâmetro.

Gabarito: **Letra C**.

40- (FCC - 2017 - TRF - 5ª REGIÃO - Técnico Judiciário - Informática) Após constatar que todos os dados em uma tabela estavam incorretos, foi solicitado ao Técnico em Informática para limpar os registros desta tabela mantendo sua estrutura, para que os dados corretos fossem posteriormente inseridos. Para realizar este trabalho o Técnico terá que utilizar a instrução SQL

a) DROP TABLE table_name.

b) REDO * FROM table_name.

c) DELETE TABLE table_name.

d) ERASE * FROM table_name.

e) TRUNCATE TABLE table_name.

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

DROP TABLE nome_da_tabela;

Essa instrução irá deletar todos os dados da tabela, bem como a própria tabela. Contudo, você pode desejar **excluir apenas os dados da tabela, sem excluir a estrutura dessa tabela**. Para isso, poderá usar o comando **TRUNCATE**:

TRUNCATE TABLE nome_da_tabela;

Gabarito: **Letra E**.

41- (FCC - 2017 - DPE-RS - Analista - Banco de Dados) O comando SQL para criar uma visão V1, a partir de uma tabela T1, obtendo os atributos A1, A2 e A3 e os renomeando para C1, C2 e C3 é:

a) CREATE VIEW V1.C1, V1.C2. V1.C3

SELECT T1.A1, T1.A2, T1.A3;

b) CREATE VIEW V1 (C1, C2, C3)

AS SELECT A1, A2, A3

FROM T1;

c) CREATE VIEW C1, C2, C3 IN V1

FROM A1, A2, A3 OF T1;

d) CREATE VIEW V1

FROM T1

SELECT A1 → C1, A2 → C2, A3 → C3;

e) CREATE VIEW V1 (C1, C2, C3)

AS PART OF T1 (A1, A2, A3);

Resolução:

A sintaxe a seguir é utilizada para **criar uma view** em SQL:

```
CREATE VIEW [Nome da View] AS
  SELECT Coluna1, Coluna2,...
  FROM nome_da_tabela
  WHERE...;
```

Assim, vamos definir o comando para o solicitado na questão por partes:

Definição do nome da visão:

```
CREATE VIEW V1 AS
```

Seleção dos atributos A1, A2 e A3 da tabela T1:

```
SELECT A1, A2, A3
FROM T1;
```

Perceba que para essa seleção não foi definida nenhuma condição e, portanto, não precisamos utilizar a cláusula WHERE.

Contudo, para o comando ficar correto precisamos renomear os atributos na definição da visão. Para isso basta informar os novos nomes entre parênteses após o nome da visão.

```
CREATE VIEW V1(C1, C2, C3) AS
```

Gabarito: **Letra B**.

42- (FCC - 2017 - DPE-RS - Técnico - Informática) Um Técnico está criando uma tabela filha chamada funcionario, que será relacionada a uma tabela pai chamada departamento, por meio da chave estrangeira. Como parte do comando CREATE TABLE, usado para criar a tabela filha, ele deseja estabelecer uma restrição de chave estrangeira chamada emp_dept_fk para o campo department_id, que fará referência ao campo department_id que é chave primária na tabela departamento. Esta restrição será criada corretamente se for utilizada, na criação da tabela funcionario, a instrução SQL

a) RESTRICTION emp_dept_fk FOREIGN KEY (department_id) PRIMARY KEY departmento(department_id)

b) FOREIGN KEY emp_dept_fk FIELD(department_id) REFERENCES departmento(department_id)

c) CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) REFERENCES departmento(department_id)

d) DEFINE CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) WITH REFERENCES departmento department_id)

e) CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) PRIMARY KEY departmento(department_id)

Resolução:

A definição de uma chave estrangeira pode ser realizada por meio da adição de uma CONSTRAINT conforme sintaxe a seguir:

```
CREATE TABLE nome_da_tabela (
  coluna1 tipo_de_dado,
  coluna2 tipo_de_dado,
  coluna3 tipo_de_dado,
CONSTRAINT nome_da_restricao FOREIGN KEY (coluna1, coluna2...)) REFERENCES tabela_referenciada (chave1, chave2...);
);
```

Gabarito: **Letra C**.

43- (FCC - 2017 - DPE-RS - Analista - Banco de Dados) O comando SQL para criar uma tabela denominada Natural, contendo os campos ID, Nome, Cidade e País, sendo todos do tipo caractere e ID a chave primária é:

a) CREATE TABLE Natural

(ID, Nome, Cidade, País, (CHAR(15), CHAR(30), CHAR(20), CHAR(20));

b) CREATE TABLE Natural

(ID PK, Nome CHAR(30), Cidade CHAR(20), País CHAR(20));

c) CREATE TABLE Natural

(ID PK, Nome, Cidade, País, (CHAR(15, 30, 20, 20));

d) CREATE TABLE Natural

(ID Char(15) PRIMARY KEY, Nome CHAR(30), Cidade CHAR(20), País (CHAR(20));

e) CREATE TABLE Natural

(ID PRIMARY KEY CHAR(15), Nome AND Cidade AND País CHAR(30, 20, 20));

Resolução:

A instrução **CREATE TABLE** é usada para **criar uma nova tabela**. A sintaxe básica dessa instrução é:

```
CREATE TABLE nome_da_tabela (
  coluna1 tipo_de_dado,
  coluna2 tipo_de_dado,
  coluna3 tipo_de_dado,
  ....
);
```

Com base nessa sintaxe, vamos analisar cada um dos itens:

a) **Incorreto**: os tipos de dados devem ser declarados imediatamente após cada um dos atributos.

b) **Incorreto**: o tipo de dados de ID não foi declarado e não existe a restrição PK, mas sim PRIMARY KEY para indicar que o campo é chave primária.

c) **Incorreto**: tipos de dados devem seguir cada um dos campos e não existe a restrição PK.

d) **Correto**: criação da tabela Natural com campos ID, Nome, Cidade e País, tendo ID como chave primária.

CREATE TABLE Natural (ID Char(15) **PRIMARY KEY**, Nome CHAR(30), Cidade CHAR(20), País (CHAR(20));

e) **Incorreto**: a definição PRIMARY KEY deve vir após o tipo de dados. Ademais, o separador dos atributos é vírgula (,) e não AND.

Gabarito: **Letra D**.

44- (FCC - 2017 - TST - Técnico Judiciário – Programação) Para criar um banco de dados relacional chamado Tribunal e excluir uma tabela chamada Consulta, um Programador deverá escrever corretamente as expressões SQL

a) CREATE DATABASE Tribunal; e DELETE TABLE Consulta;

b) INSERT DATABASE Tribunal; e DELETE TABLE = Consulta;

c) CREATE DATABASE Tribunal; e DROP TABLE Consulta;

d) INSERT DATABASE Tribunal; e DROP TABLE Consulta;

e) INSERT DATABASE Tribunal; e DROP TABLE = Consulta;

Resolução:

A instrução **CREATE DATABASE** é usada para **criar um novo banco de dados** SQL. A sintaxe básica dessa instrução é:

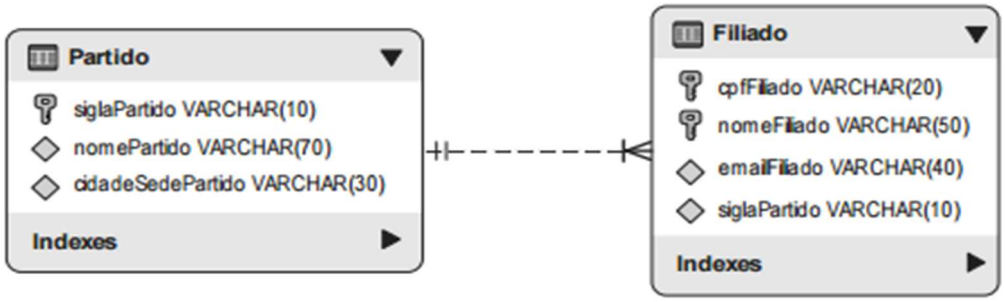
CREATE DATABASE nome_do_banco;

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**. A sintaxe para esse comando é:

DROP TABLE nome_da_tabela;

Gabarito: **Letra C**.

45- (FCC - 2016 - AL-MS - Técnico de Informática) Para responder à questão, considere o modelo mostrado na imagem abaixo, oriundo de uma situação hipotética:



Após criadas as tabelas Partido e Filiado, foram incluídos, respectivamente, os seguintes registros:

siglaPartido	nomePartido	cidadeSedePartido
PDT	Partido Democrático Trabalhista	Brasília
PMDB	Partido do Movimento Democrático Brasileiro	Brasília
PSDB	Partido da Social Democracia Brasileira	São Paulo

cpfFiliado	nomeFiliado	emailFiliado	siglaPartido
124.179.156-10	André Braga	braga@hotmail.com	PMDB
147.189.237-18	Marcos Pereira	mpereira@hotmail.com	PDT
154.496.172-14	Pedro Silva	pedro@gmail.com	PDT
192.345.176-01	Maria Souza	maria@ig.com.br	PSDB

Após a tabela Partido ser criada, para criar a tabela Filiado foi utilizada a instrução abaixo:

```
CREATE TABLE IF NOT EXISTS Filiado (
 cpfFiliado VARCHAR(20) NOT NULL,
 nomeFiliado VARCHAR(50),
 emailFiliado VARCHAR(40),
 siglaPartido VARCHAR(10) NOT NULL,
 PRIMARY KEY (cpfFiliado),
 FOREIGN KEY (siglaPartido)
I Partido (siglaPartido)
);
```

A lacuna I deverá ser corretamente preenchida por

a) CASCADE CONSTRAINT

b) REFERENCES

c) REFERENCE CONSTRAINT

d) EXTENDS

e) IMPLEMENTS

Resolução:

O que está faltando na instrução é a palavra chave REFERENCES que é utilizada para complementar o comando de definição da chave estrangeira da tabela, conforme seguinte sintaxe básica:

CREATE TABLE nome_da_tabela (

coluna1 tipo_de_dado,

coluna2 tipo_de_dado,

coluna3 tipo_de_dado,

CONSTRAINT nome_da_restricao **FOREIGN KEY** (coluna1, coluna2...)) **REFERENCES** tabela_referenciada (chave1, chave2...);

);

Gabarito: **Letra B**.

46- (FCC - 2016 - SEGEP-MA - Técnico da Receita Estadual - Tecnologia da Informação - Conhecimentos Específicos) Atenção: Para responder às questões, considere a figura abaixo.

Considere que na tabela Contribuinte estão cadastrados os seguintes dados:

IDContribuinte	NomeContribuinte	CPF_CNPJ
1	Paulo da Silva	154.246.037-12
2	Maria Pereira	143.172.129-50

Após as tabelas Imposto e Contribuinte terem sido criadas, para criar a tabela Contribuinte_Imposto deve ser utilizada a seguinte instrução SQL:

```
CREATE TABLE Contribuinte_Imposto (IDContribuinte INT NOT NULL, SiglaImposto VARCHAR(10) NOT
NULL, Valor_Imposto DOUBLE, PRIMARY KEY (IDContribuinte, SiglaImposto), I );
```

A lacuna I é corretamente preenchida por:

a) FOREIGN KEY (IDContribuinte) EXTENDS Contribuinte (IDContribuinte), FOREIGN KEY (SiglaImposto) EXTENDS Imposto (SiglaImposto)

b) FOREIGN_KEY (IDContribuinte) CONSTRAINT Contribuinte (IDContribuinte), FOREIGN_KEY (SiglaImposto) CONSTRAINT Imposto (SiglaImposto)

c) FOREIGN KEY (IDContribuinte) REFERENCES Contribuinte (IDContribuinte), FOREIGN KEY(SiglaImposto) REFERENCES Imposto (SiglaImposto)

d) FOREIGN KEY (IDContribuinte, Contribuinte), FOREIGN KEY (SiglaImposto, Imposto)

e) REFERENCES Contribuinte (IDContribuinte) FOREIGN KEY, REFERENCES Imposto (SiglaImposto) FOREIGN KEY

Resolução:

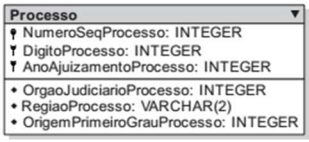
Como os atributos relacionados por meio da chave estrangeira são de tabelas diferentes, deve-se utilizar mais de uma cláusula **FOREIGN KEY**.

Assim, para referenciar a ID_contribuinte entre as tabelas Contribuinte e Contribuinte_Imposto usamos **FOREIGN KEY** (IDContribuinte) **REFERENCES** Contribuinte (IDContribuinte)

E para referenciar SiglaImposto entre as tabelas Imposto e Contribuinte_Imposto usamos **FOREIGN KEY** (SiglaImposto) **REFERENCES** Imposto (SiglaImposto).

Gabarito: **Letra C**.

47- (FCC - 2016 - TRF - 3ª REGIÃO - Técnico Judiciário - Informática) Para responder a questão, considere as informações abaixo.



Registros cadastrados:

NumeroSeqProcesso	DigitoProcesso	AnoAjuizamentoProcesso	OrgaoJudiciarioProcesso	RegiaoProcesso	OrigemPrimeiroGrauProcesso
15472	49	2002	4	3	3300
16592	44	2014	4	3	4500
17543	45	1999	4	3	3300
24535	23	2002	4	3	3300
29670	41	2012	4	2	2200
36535	45	2000	4	1	3400
44672	40	2012	4	2	3400
45891	43	2007	4	1	4400
67234	39	1997	4	1	3500

Considere que a tabela Processo foi criada sem chave primária. Nesse caso, para definir a chave primária, antes de serem inseridos registros, deve-se utilizar a instrução SQL

a) ADD TO Processo PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

b) INSERT INTO Processo PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

c) ALTER TABLE Processo ADD PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

d) ADD CONTRAINT PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso) from Processo;

e) UPDATE TABLE Processo ADD PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

Resolução:

Como desejamos definir uma chave primária com uma tabela que já foi criada, podemos utilizar a instrução ALTER TABLE para modificar a tabela, adicionando a chave desejada. A sintaxe básica é:

- **Para uma coluna**:

  ALTER TABLE nome_da_tabela

  ADD PRIMARY KEY(coluna)**;**

- **Para múltiplas colunas**:

  ALTER TABLE nome_da_tabela

  ADD CONSTRAINT nome_da_restricao **PRIMARY KEY** (coluna1, coluna2));

No caso da questão, foi utilizada a primeira sintaxe, que embora seja mais aplicada para chaves de uma única coluna, pode ser usada em alguns SGBDs para mais de uma coluna.

Gabarito: **Letra C**.

48- (FCC - 2016 - TRT - 23ª REGIÃO (MT) - Técnico Judiciário - Tecnologia da Informação)

Atenção: Para responder a questão, considere a informação abaixo.

Um Técnico está participando da modelagem de um banco de dados utilizando o Modelo Entidade-Relacionamento – MER e se deparou, dentre outras, com a entidade Processo, que contém os seguintes atributos:

NumeroProcesso – inteiro (PK)
DigitoProcesso – inteiro (PK)
AnoProcesso – inteiro (PK)
NumeroOABAdvogadoProcesso – cadeia de caracteres
NomeAdvogadoProcesso – cadeia de caracteres
NumeroOrgaoJudiciarioProcesso – inteiro (FK)
NumeroTribunal – inteiro (FK)
NumeroUnidadeOrigemProcesso – inteiro (FK)

Após criar a tabela Processo no Sistema Gerenciador de Banco de Dados SQL Server, para definir uma restrição que especifica que o campo AnoProcesso só poderá receber números inteiros maiores do que 2014, o Técnico deve utilizar a instrução

a) ADD CONSTRAINT Processo CHECK (AnoProcesso>2014);

b) ALTER TABLE Processo ADD CHECK (AnoProcesso>2014);

c) ADD CONSTRAINT (AnoProcesso>2014) FROM Processo;

d) CREATE CONSTRAINT Chk_Processo FROM Processo CHECK (AnoProcesso>2014);

e) ALTER TABLE Processo ADD CONSTRAINT (AnoProcesso>2014);

Resolução:

Como desejamos definir uma restrição para uma tabela que já foi criada, podemos utilizar a instrução ALTER TABLE para modificar a tabela, adicionando a verificação desejada. A sintaxe básica é:

- **Para uma coluna**:

ALTER TABLE nome_da_tabela

ADD CHECK (condicao)**;**

- **Para múltiplas colunas**:

ALTER TABLE nome_da_tabela

ADD CONSTRAINT nome_da_restricao **CHECK** (condicao1 **AND** condicao2));

Gabarito: **Letra B**.

49- (FCC - 2015 - TRT - 9ª REGIÃO (PR) - Analista Judiciário - Área Apoio Especializado - Tecnologia da Informação)

A tabela relativa a Débitos Trabalhistas a seguir deve ser utilizada para responder à questão.

Considere que a tabela já está criada, os dados iniciais já foram inseridos e o banco de dados a ser utilizado está aberto e funcionando em condições ideais.

Tabela DebTrab

NroProcesso	Principal	Juros	FGTS	Honor Periciais
111/15	25345.00	3801.75	7933.00	4755.00
777/15	125800.00	18870.00	57966.87	7543.00
333/15	8844.50	1326.67	4233.55	1781.00
555/15	327631.00	65526.20	104863.78	11523.00
444/15	5072.00	1014.40	895.14	700.00

Um Analista da área de TI trabalha em uma organização que possui aplicações que utilizam os SGBDs Oracle 11g e SQL Server. Ele identificou que o comando SQL que está correto e pode ser aplicado em ambas as plataformas é

a) ALTER TABLE DebTrab ALTER COLUMN NroProcesso integer;

b) ALTER TABLE DebTrab MODIFY NroProcesso int;

c) ALTER TABLE DebTrab ADD DataPartida data;

d) ALTER TABLE DebTrab ADD IndiceAtualiz float;

e) ALTER TABLE DebTrab DROP COLUMN DataPartida;

Resolução:

Vamos analisar cada um dos itens:

a) **Incorreto**: **ALTER COLUMN** pode ser usado no SQL Server, mas não no Oracle.

b) **Incorreto**: **MODIFY** é usado somente a partir da versão 10g do Oracle, mas não no SQL SERVER.

c) **Incorreto**: o tipo de dados "data" não existe. O correto seria **"date"**.

d) **Correto**: para **adicionar uma coluna**, usamos a cláusula **ADD**:

```
ALTER TABLE nome_da_tabela
ADD nome_da_coluna tipo_de_dado;
```

e) **Incorreto**: no modelo apresentado, a tabela DebTrab não possui nenhuma coluna DataPartida e, portanto, não é possível deletar algo que não existe.

Gabarito: **Letra D**.

50- (FCC - 2015 - DPE-SP - Programador)

Considere que os comandos foram executados na sequência da numeração indicada:

```
[1] CREATE TABLE Tab_InfJuv
    (ID integer UNIQUE, Unidade varchar (50), Defensor varchar (50), Tipo integer);
[2] INSERT INTO Tab_InfJuv VALUES ( 11, 'Chacara Santo Antonio-Sao Paulo', 'Jorge da Silva', 1);
[3] INSERT INTO Tab_InfJuv VALUES ( 14, 'Parque Monteiro Soares- Sao Paulo', 'Maria Joana Santos', 2);
[4] INSERT INTO Tab_InfJuv VALUES ( 12, 'Centro-São Paulo', 'Jorge da Silva', 2);
[5] INSERT INTO Tab_InfJuv VALUES ( 32, 'Centro-Diadema', 'Ana Maria da Silva', 1);
```

O comando SQL correto é:

a) ALTER TABLE Tab_InfJuv ADD FOREIGN KEY (ID_Unidade) REFERENCES Tab_InfJuv (ID);

b) ALTER TABLE Tab_InfJuv ADD Nome_Menor varchar(100);

c) CREATE VIEW V_Tab_InfJuv AS SELECT Unidade, Defensor, ID_Unidade FROM Tab_InfJuv;

d) ALTER TABLE TabInfJuv DROP Nome_Menor;

e) DELETE FROM TabInfJuv WHERE Defensor = 'Jorge da Silva';

Resolução:

Vamos analisar cada um dos itens:

a) **Incorreto**: o comando está referenciando a própria tabela, isto é, definindo um autorrelacionamento. Contudo, não existe nenhum atributo ID_Unidade nesta tabela para ser usado como chave estrangeira.

b) **Correto**: para **adicionar uma coluna**, usamos a cláusula **ADD**:

```
ALTER TABLE nome_da_tabela
ADD nome_da_coluna tipo_de_dado;
```

c) **Incorreto**: não existe coluna ID_Unidade na tabela e, sendo assim, não pode ser criada view com base nesse campo.

d) **Incorreto**: não existe coluna Nome_Menor e, portanto, ela não pode ser excluída.

e) **Incorreto**: o nome da tabela é Tab_InfJuv e não TabInfJuv. Perceba que existe um _ no nome.

Gabarito: **Letra B**.

51- (FCC - 2015 - TRE-AP - Técnico Judiciário - Programação de Sistemas) Um Técnico do TREAP – Tribunal Regional Eleitoral do Amapá ficou responsável por criar uma tabela no Banco de Dados denominada Tab_PAA que seja capaz de armazenar o código da auditoria (um identificador único, não nulo), o nome do auditor responsável e o órgão sendo auditado. Para isso, ele utilizou, corretamente, o seguinte comando SQL:

a) CREATE TABLE Tab_PAA (Cod_Audit integer PRIMARY KEY, Nome_Auditor varchar(50), Orgao_Audit varchar(50));

b) CREATE_TABLE (Cod_Audit integer, Nome_Auditor varchar(50), Orgao_Audit varchar(50)) Tab_PAA;

c) CREATE_TABLE Tab_PAA (Cod_Audit integer UNIQUE; Nome_Auditor varchar(50); Orgao_Audit varchar(50));

d) CREATE TABLE Tab_PAA (varchar(50) Nome_Auditor, varchar(50), Orgao_Audit PRIMARY KEY integer Cod_Audit);

e) CREATE TABLE (UNIQUE NOT NULL Cod_Audit, VARCHAR(50) Nome_Auditor, VARCHAR(50) Orgao_Audit) Tab_PAA;

Resolução:

Vamos analisar cada um dos itens:

a) **Correto**: esse comando realiza a criação da tabela conforme solicitado no item. A instrução **CREATE TABLE** é usada para **criar uma nova tabela** no banco de dados. A sintaxe básica dessa instrução é:

```
CREATE TABLE nome_da_tabela (
    coluna1 tipo_de_dado,
    coluna2 tipo_de_dado,
    coluna3 tipo_de_dado,
    ....
);
```

As restrições podem ser adicionadas após os tipos de dados. No caso do item, é adicionada a restrição de chave primária ao atributo Cod_Audit.

b) **Incorreto**: o nome da tabela deve vir imediatamente após o comando CREATE_TABLE e não no final.

c) **Incorreto**: o separador para os atributos é vírgula (,) e não ponto e vírgula (;)

d) **Incorreto**: a ordem correta para declaração dos atributos é coluna tipo_de_dados e restrições. Nesse item, os tipos de dados e as restrições estão precedendo o nome do atributo.

e) **Incorreto**: mesmo caso do item d.

Gabarito: **Letra A**.

52- (FCC - 2015 - TRT - 15ª Região (SP) - Técnico Judiciário - Tecnologia da Informação) Deseja-se criar uma tabela chamada Departamento contendo os seguintes campos:

idDep – inteiro, chave primária, não nulo, auto numeração.

nomeDep – cadeia de caracteres com, no máximo, 50 caracteres, não nulo.

telefoneDep – cadeia de caracteres com, no máximo, 15 caracteres.

Considerando-se que o banco de dados está aberto e em condições ideais, o comando SQL que deverá ser utilizado é:

a) CREATE TABLE Departamento (
idDep INT NOT NULL AUTO_NUMBERING,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15),
PRIMARY KEY (idDep));

b) CREATE TABLE Departamento (
idDep INT NOT NULL AUTOINCREMENT,
nomeDep VARCHAR2(50) NOT NULL,
telefoneDep VARCHAR2(15) NULLABLE,
PRIMARY KEY (idDep));

c) CREATE TABLE Departamento (
idDep INT NOT NULL AUTO_INCREMENT,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15),
PRIMARY KEY (idDep));

d) CREATE TABLE Departamento (
idDep INT NOT NULL AUTOINCREMENT,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15)
PRIMARY_KEY (idDep));

e) CREATE TABLE Departamento (
idDep INT NOT NULL AUTO_INCREMENT PRIMARY_KEY,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15));

Resolução:

Vamos comentar cada uma das assertivas:

a) **Incorreto**: não existe a restrição AUTO_NUMBERING, mas sim **AUTO_INCREMENT** que é responsável por automaticamente definir uma numeração sequencial para os registros inseridos no campo.

b) **Incorreto**: a restrição correta seria **AUTO_INCREMENT** e não AUTOINCREMENT. Ademais não existe a restrição NULLABLE.

c) **Correto**: A instrução **CREATE TABLE** é usada para **criar uma nova tabela** no banco de dados. A sintaxe básica dessa instrução é:

```
CREATE TABLE nome_da_tabela (
  coluna1 tipo_de_dado,
  coluna2 tipo_de_dado,
  coluna3 tipo_de_dado,
  ....
);
```

Para adicionar uma chave é possível utilizar a seguinte sintaxe:

```
PRIMARY KEY (coluna1)
```

Logo, esse item está em total conformidade com a sintaxe do SQL.

d) **Incorreto**: a restrição correta seria **AUTO_INCREMENT** e não AUTOINCREMENT.

e) **Incorreto**: a restrição correta seria **PRIMARY KEY** e não PRIMARY_KEY. Correto sem o _.

Gabarito: **Letra C**.

53- (FCC - 2015 - TRT - 15ª Região (SP) - Técnico Judiciário - Tecnologia da Informação) No SQL há dois comandos que podem eliminar completamente os registros de uma tabela. A diferença entre eles é que o comando I irá eliminar apenas os dados (registros), enquanto o comando II irá eliminar também a tabela.

Os comandos I e II são, respectivamente,

a) TRUNCATE TABLE e DROP TABLE.

b) DROP TABLE e DELETE FROM.

c) DELETE RECORD e DELETE TABLE.

d) DROP TABLE e TRUNCATE TABLE.

e) REMOVE RECORD e DROP TABLE.

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

DROP TABLE nome_da_tabela;

Essa instrução irá deletar todos os dados da tabela, bem como a própria tabela. Contudo, você pode desejar **excluir apenas os dados da tabela, sem excluir a estrutura dessa tabela**. Para isso, poderá usar o comando **TRUNCATE**:

TRUNCATE TABLE nome_da_tabela;

Gabarito: **Letra A**.

54- (FCC - 2014 - TJ-AP - Analista Judiciário - Área Apoio Especializado - Tecnologia da Informação - Banco de Dados- DBA) Considere o seguinte trecho de um comando SQL ANSI para a criação de uma tabela:

```
CREATE TABLE Teste
  (Comando 1 . . .
     . . . )
```

Deseja-se declarar um atributo de nome Item, do tipo caractere, com 20 posições e que seja chave primária da tabela. Para tanto, o Comando 1 deve ser substituído por

a) PK Item CHAR (20)

b) Item CHAR (20) PRIMARY KEY

c) CHAVE PRIMÁRIA Item CHAR: 20

d) Item CHAVE PRIMÁRIA CHAR (20)

e) Item PK CHAR: 20

Resolução:

A ordem correta para a definição de colunas em uma instrução CREATE TABLE é: coluna1 tipo_de_dado restrições

Sendo assim, vamos analisar cada um dos itens:

a) **Incorreto**: não existe restrição PK.

b) **Correto**: item correto, pois informa o nome da coluna (Item), o tipo de dados (CHAR(20)) e a restrição (PRIMARY KEY).

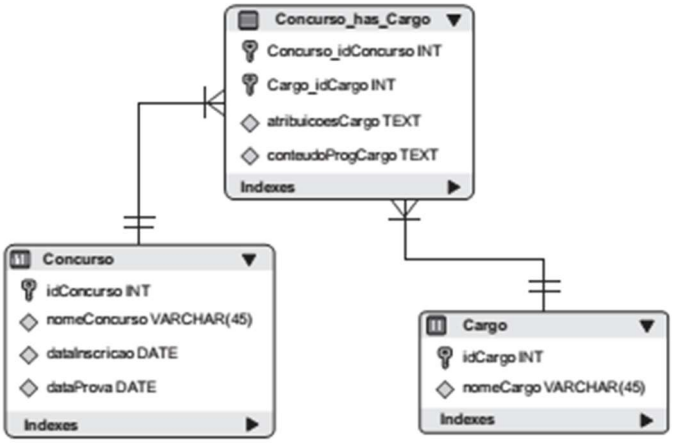
c) **Incorreto**: CHAVE PRIMÁRIA não é restrição válida.

d) **Incorreto**: CHAVE PRIMÁRIA não é restrição válida.

e) **Incorreto**: não existe restrição PK e tamanho do tipo de dados CHAR vem entre () e não após :.

Gabarito: **Letra B**.

55- (FCC - 2014 - TRF - 4ª REGIÃO - Analista Judiciário - Especialidade Informática) Atenção: Para responder à questão, considere o modelo de banco de dados abaixo.



As atribuições do cargo (atribuicoesCargo) e o conteúdo programático (conteudoProgCargo) normalmente são diferentes, dependendo do cargo e do concurso. O atributo dataInscricao refere-se à data de início das inscrições.

Para incluir na tabela Concurso o campo dataResultado para entrada de dados de tipo data, de forma que não aceite valores nulos, utiliza-se a instrução

a) ALTER COLUMN Concurso ADD dataResultado DATE NOT NULL;

b) INCLUDE COLUMN dataResultado FROM concurso DATE NOT NULL;

c) ALTER TABLE concurso ADD dataResultado DATE NOT NULL;

d) ALTER TABLE Concurso INCLUDE dataResultado (DATE) NOT NULL;

e) INSERT COLUMN dataResultado FROM concurso DATATYPE DATE NOT NULL;

Resolução:

A instrução **ALTER TABLE** é usada para **adicionar, deletar ou modificar colunas em uma tabela existente**. Essa instrução também pode ser utilizada para adicionar ou deletar restrições a esta tabela.

Para **adicionar uma coluna**, usamos a cláusula **ADD**:

```
ALTER TABLE nome_da_tabela
ADD nome_da_coluna tipo_de_dado;
```

Gabarito: **Letra C**.

56- (FCC - 2014 - AL-PE - Analista Legislativo - Sistemas) Em um banco de dados há duas tabelas, departamento e funcionario, cujos campos são:

Departamento:

DepNo - int - primary key - not null

DNome - varchar(50) - not null

DLocal - varchar(40)

Funcionario:

FunNo - int - primary key - not null

DepNo - int - foreign key - not null

FNome - varchar(50) - not null

FCargo - varchar(40)

FSal - double

Ambas as tabelas possuem muitos registros cadastrados, que não incluem valores nulos. Considere as instruções SQL a seguir:

I. select funcionario.FunNo, funcionario.FNome, departamento.DNome from funcionario INNER JOIN departamento ON funcionario.DepNo = departamento.DepNo;

II. select funcionario.FunNo, funcionario.FNome, departamento.DNome from funcionario, departamento where funcionario.DepNo = departamento.DepNo;

III. select f.FunNo, f.FNome, d.DNome from (select departamento.DepNo, departamento.DNome from departamento) as d, funcionario as f where d.DepNo = f.DepNo;

IV. select DISTINCT f.FunNo, f.FNome, d.DNome from funcionario f, departamento d;

Para criar uma nova tabela física chamada funcionario2, apenas com os campos FunNo e FNome da tabela funcionario, incluindo os dados cadastrados nestes campos, em ordem alfabética crescente pelo campo FNome, utiliza-se a instrução:

a) COPY f.FunNo, f.FNome FROM funcionario as f TO funcionario2 ORDER BY f.FNome;

b) CREATE TABLE funcionario2 FIELDS f.FunNo, f.FNome FROM funcionario as f TO ORDER BY f.FNome;

c) COPY FunNo, FNome FROM funcionario INTO funcionario2 ORDER BY f.FNome ASC;

d) CREATE TABLE funcionario2 AS SELECT f.FunNo, f.FNome FROM funcionario as f ORDER BY FNome;

e) CREATE VIEW funcionario2 AS SELECT FunNo, FNome FROM funcionario ORDER BY FNome;

Resolução:

Ao utilizar a DDL em conjunto com a DML é possível criar uma tabela a partir de outra tabela existente. Para isso basta usar o auxílio da cláusula AS conforme sintaxe a seguir:

```
CREATE TABLE nome_da_nova_tabela AS
  SELECT coluna1, coluna2,...
  FROM nome_da_tabela_existente
  WHERE ....;
```

Com essa sintaxe, é possível criar uma nova tabela a partir de uma instrução SELECT. Tanto a estrutura da seleção quando os dados selecionados serão armazenados na nova tabela.

Assim, o comando a seguir cria a tabela funcionario2 a partir da selecao de FunNo e FNome da tabela funcionario de forma ordenada pelo nome:

```
CREATE TABLE funcionario2 AS SELECT f.FunNo, f.FNome FROM funcionario as f ORDER BY FNome;
```

Vale ressaltar que poderia ser criada uma visão para isso conforme trazido no item e). Contudo, a questão pede a criação de uma tabela física e a view é uma tabela virtual.

Gabarito: **Letra D**.

57- (FCC - 2013 - AL-RN - Técnico Legislativo - Programador) A restrição DEFAULT em SQL é utilizada para

a) introduzir um valor padrão em uma coluna.

b) determinar uma operação padrão em uma Trigger.

c) atribuir um nome padrão para uma tabela.

d) determinar o valor padrão de retorno para uma procedure.

e) determinar a visualização (view) padrão que será utilizada.

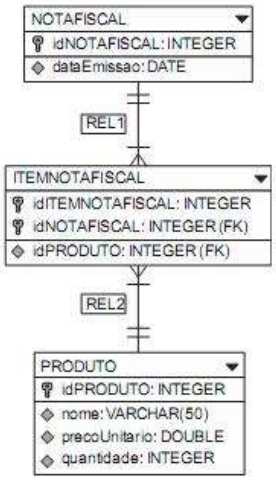
Resolução:

A restrição **DEFAULT** é usada para **fornecer um valor padrão para uma coluna**. O valor padrão será adicionado a todos os novos registros SE nenhum outro valor for especificado.

Gabarito: **Letra A**.

58- (FCC - 2013 - DPE-SP - Programador de computador)

Para responder às questões de números 41 e 42, utilize a figura abaixo, parte de um modelo Entidade-Relacionamento de um banco de dados.



Uma instrução SQL correta para criar a tabela NOTAFISCAL apresentada no modelo é:

a) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NOT NULL, dataEmissao DATE NULL, PRIMARY KEY(idNOTAFISCAL));

b) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NULL AUTOINCREMENT, dataEmissao DATE NULL, PRIMARY KEY(idNOTAFISCAL));

c) CREATE SCHEMA NOTAFISCAL (idNOTAFISCAL INTEGER NOT NULL AUTO_INCREMENT, dataEmissao DATE NOT NULL, PRIMARY KEY(idNOTAFISCAL));

d) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NULL, dataEmissao DATE NOT NULL, CONSTRAINT UNIQUE KEY(idNOTAFISCAL));

e) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NULL CONSTRAINT PRIMARY KEY, dataEmissao DATE NULL);

Resolução:

Vamos analisar cada um dos comandos:

a) **Correto**: sintaxe correta do comando CREATE TABLE para a tabela NOTAFISCAL, com a definição dos campos idNOTAFISCAL como inteiro e não nulo e dataEmissao como data e podendo ser nula. Além disso, definição de idNOTAFISCAL como chave primária.

b) **Incorreto**: idNOTAFISCAL é definida como chave primária e, portanto, não poderia ser NULL.

c) **Incorreto**: NOTAFISCAL é uma tabela e não um esquema.

d) **Incorreto**: faltou definir a chave primária.

e) **Incorreto**: se a PRIMARY KEY é definida logo após o campo, então não se deve utilizar a cláusula CONSTRAINT.

Gabarito: **Letra A**.

59- **(FCC - 2013 - MPE-AM - Agente de Apoio - Programador)** O comando SQL utilizado para adicionar, modificar ou remover colunas em uma tabela existente é chamado

a) INSERT INTO.

b) DROP TABLE.

c) CREATE TABLE.

d) ALTER TABLE.

e) TRUNCATE.

Resolução:

A instrução **ALTER TABLE** é usada para **adicionar, deletar ou modificar colunas em uma tabela existente**. Essa instrução também pode ser utilizada para adicionar ou deletar restrições a esta tabela.

Gabarito: **Letra D**.

60- **(FCC - 2012 - MPE-AP - Técnico Ministerial - Informática)** Em linguagem SQL, o comando utilizado para remover uma tabela de um banco de dados é

a) DROP TABLE.

b) DELETE TABLE.

c) REMOVE TABLE.

d) DELETE FROM.

e) ROLLBACK.

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

```
DROP TABLE nome_da_tabela;
```

Essa instrução irá deletar todos os dados da tabela, bem como a própria tabela.

Gabarito: **Letra A**.

61- (FCC - 2011 - TRT - 19ª Região (AL) - Técnico Judiciário - Tecnologia da Informação) É um comando do tipo DDL (Data Definition Language) no SQL:

a) SELECT.

b) DELETE.

c) INSERT.

d) UPDATE.

e) CREATE.

Resolução:

Os comandos da DDL são CREATE, ALTER e DROP (ou TRUNCATE).

Gabarito: **Letra E**.

62- (FCC - 2009 - TRT - 15ª Região - Analista Judiciário - Tecnologia da Informação) As constraints para as tabelas de um BD relacional podem ser especificadas quando de um

a) ALTER, somente.

b) CREATE, somente.

c) CREATE ou quando de um ALTER.

d) CREATE ou quando de um INSERT.

e) INSERT ou quando de um ALTER.

Resolução:

As **restrições SQL (constraints)** são usadas para **especificar regras para os dados em uma tabela**.

As **restrições** são usadas para **limitar o tipo de dados que podem ser colocados em uma tabela**. Isso garante a precisão e a confiabilidade dos dados na tabela. Se houver alguma violação entre a restrição e a ação de dados, a ação será abortada.

As restrições podem ser definidas na criação da tabela ou na alteração destas.

Gabarito: **Letra C**.

1.3 FGV

63- (FGV - 2022 – CGU – Auditor Federal de Finanças e Controle – Tecnologia da Informação) No contexto do SQL Server 2019, considere o script SQL a seguir.

```
create table XPTO (
        C1 int identity,
        C2 varchar(16))
insert into XPTO (C2) values ('A')
insert into XPTO (C2) values ('B')
insert into XPTO (C2) values ('C')
truncate table XPTO
insert into XPTO (C2) values ('D')
insert into XPTO (C2) values ('E')
delete from XPTO
truncate table XPTO
insert into XPTO (C2) values ('F')
truncate table XPTO
```

Após a execução desse script, o resultado exibido pelo comando select max(C1) from XPTO é:

a) 0

b) NULL

c) 1

d) 3

e) 6

Resolução:

De modo geral, nesse script nós temos primeiramente a criação de uma tabela XPTO com os atributos C1 e C2 no comando CREATE TABLE.

Após a criação da tabela há uma série de manipulações dos dados. E por fim, uma consulta para selecionar o valor máximo de C1. O segredo para resolver a questão de forma rápida é observar que o último comando antes da consulta é um TRUNCATE TABLE, que é responsável por eliminar todas as linhas da tabela. Sendo assim, na hora da consulta não haverá dados na tabela, portanto não existirá um valor máximo e o retorno será NULL.

Gabarito: **Letra B**.

64- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Desenvolvimento de Sistemas) Analise o comando de criação da tabela teste, com cinco constraints, exibido a seguir.

create table teste (

a int null,

b varchar (40) not null,

c int,

constraint pk_a primary key (a),

constraint fk_a_c foreign key (c) references teste,

constraint uq_b unique (b),

constraint up_c unique (c),

constraint ch_1 check (a >= 0.5 * c))

Assinale a constraint desse script que seria rejeitada no MS SQL Server.

a) A primeira.

b) A segunda.

c) A terceira.

d) A quarta.

e) A quinta.

Resolução:

Vamos analisar cada uma das Constraints:

a) **Incorreto**: não é possível definir a como chave primária, pois é atributo definido como NULL.

b) **Correto**: definição de uma chave estrangeira c que faz referência a tabela teste.

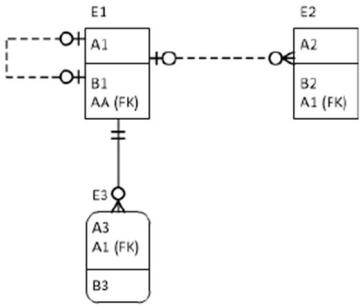
c) **Correto**: definição de uma restrição de exclusividade para o atributo b.

d) **Correto**: definição de uma restrição de exclusividade para o atributo c.

e) **Correto**: definição de uma restrição de check em que a deve ser maior ou igual a 0,5 vezes o valor de c.

Gabarito: **Letra A**.

65- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Administrador de Banco de dados) Considere o diagrama IDEF1X a seguir.



Assinale a opção que apresenta o script SQL que cria a tabela E1 corretamente.

a) create table E1 (

A1 int not null primary key,

B1 int null,

AA int not null,

constraint FK_X foreign key (AA)

references E1(A1))

b) create table E1 (

A1 int not null primary key,

B1 int null,

AA int not null,

constraint FK_X foreign key (AA)

references E1(A1))

c) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null,

constraint FK_X foreign key (AA)

references E1(A1))

d) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (AA)

references E1(A1))

e) create table E1 (

A1 int not null primary key,

B1 int null,

AA int not null unique,

constraint FK_X foreign key (AA)

references E1(A1))

Resolução:

Galera, essa é a típica questão da banca que exige o conhecimento da necessidade de definir a chave estrangeira como UNIQUE para implementar um relacionamento 1:1.

Perceba que o autorrelacionamento de E1 é 1:1, logo a chave estrangeira AA deve ser definida como UNIQUE.

Se o atributo AA permitisse a repetição, não haveria nenhum problema teórico, mas não se implementaria um relacionamento 1:1, mas sim 1:N.

Logo,

a) **Incorreto**: não define a chave estrangeira como UNIQUE.

b) **Incorreto**: não define a chave estrangeira como UNIQUE.

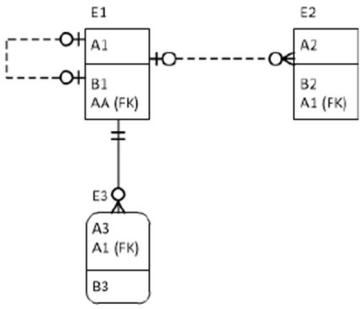
c) **Incorreto**: não define a chave estrangeira como UNIQUE.

d) **Correto**: AA pode ser NULL, pois a relação é opcional (0:1).

e) **Incorreto**: AA não é NOT NULL, pois a relação não é obrigatória.

Gabarito: **Letra D**.

66- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Administrador de Banco de dados) Considere o diagrama IDEF1X a seguir.



Assinale a opção que apresenta o script SQL que cria a tabela E2 corretamente.

a) create table E2 (

A2 int null primary key,

B2 int null,

A1 int,

constraint FK_X foreign key (A1)

references E1(A1))

b) create table E2 (

A2 int not nul,

B2 int null,

A1 int null,

constraint PK primary key(A2, A1),

constraint FK_X foreign key (A1)

references E1(A1))

c) create table E2 (

A2 int not nul,

B2 int null,

constraint PK primary key(A2, A1),

constraint FK_X foreign key (A1)

references E1(A1))

d) create table E2 (

A2 int not null,

B2 int null,

A1 int null,

constraint FK_X foreign key (A1)

references E1(A1))

e) create table E2 (

A2 int not null primary key,

B2 int null,

A1 int null,

constraint FK_X foreign key (A1)

references E1(A1))

Resolução:

Vamos analisar cada um dos comandos:

a) **Incorreto**: A2 está sendo definido como NULL e PRIMARY KEY ao mesmo tempo.

b) **Incorreto**: A1 está sendo definida como um dos atributos da chave primária para a tabela, em desconformidade com o diagrama que apresenta somente A2 como chave da tabela E2.

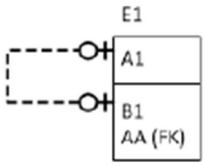
c) **Incorreto**: além do motivo indicado em b), não é definido o atributo A1.

d) **Incorreto**: não foi definida a chave primária da tabela.

e) **Correto**: sintaxe correta. Chave primária definida por A2 e chave estrangeira sendo A1 referenciando o A1 da tabela E1.

Gabarito: **Letra E**.

67- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Desenvolvimento de Sistemas) Analise o diagrama IDEF1X a seguir.



Assinale a opção que indica o script SQL que cria a tabela E1 corretamente.

a) create table E1 (

A1 int null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (AA)

references E1(A1))

b) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null,

constraint FK_X foreign key (A1)

references E1(AA))

c) create table E1 (

A1 int not null unique,

B1 int null,

AA int null primary key,

constraint FK_X foreign key (AA)

references E1(A1))

d) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (A1)

references E1(AA))

e) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (AA)

references E1(A1))

Resolução:

Galera, essa é a típica questão da banca que exige o conhecimento da necessidade de definir a chave estrangeira como UNIQUE para implementar um relacionamento 1:1.

Perceba que o autorrelacionamento de E1 é 1:1, logo a chave estrangeira AA deve ser definida como UNIQUE.

Se o atributo AA permitisse a repetição, não haveria nenhum problema teórico, mas não se implementaria um relacionamento 1:1, mas sim 1:N.

Logo,

a) **Incorreto**: A1 está sendo definida como PRIMARY KEY e NULL ao mesmo tempo.

b) **Incorreto**: a chave estrangeira é AA e não A1. Logo, o correto seria:

constraint FK_X foreign key (AA) references E1(A1))

c) **Incorreto**: A1 está sendo definida como PRIMARY KEY e NULL ao mesmo tempo.

d) **Incorreto**: a chave estrangeira é AA e não A1. Logo, o correto seria:

constraint FK_X foreign key (AA) references E1(A1))

e) **Correto**: sintaxe correta para criação de E1. AA pode ser NULL, pois o relacionamento é opcional (0:1). Deve ser UNIQUE, pois o relacionamento é 1:1.

Gabarito: **Letra E**.

68- (FGV - 2015 - Câmara Municipal de Caruaru - PE - Analista Legislativo - Informática) Em geral, a definição de chaves estrangeiras em bancos de dados relacionais pode vir acompanhada da especificação de procedimentos adicionais a serem adotados quando da exclusão/alteração de valores nos registros da tabela estrangeira.

create table R2 (

a int not null primary key,

b int null,

x int not null,

constraint FK foreign key (x) references

R1(x))

Considerando o script acima, as opções complementares compatíveis para a definição da chave estrangeira FK são:

a) on delete cascade

on update set null

b) on delete cascade

on update cascade

c) on delete no action

on update set null

d) on delete set null

on update restrict

e) on delete restrict

on update set null

Resolução:

As chaves estrangeiras podem ser criadas com a definição de cláusulas de exclusão ou atualização em cascata. Vejamos:

Ao usar a opção **ON DELETE CASCADE**, quando um registro da tabela que possui a chave primária associada a esta chave estrangeira for excluído, então os registros associados também são excluídos. Ex.: ao excluir um determinado País, todos os Estados que estão associados àquele País são também deletados (a associação é verificada pela chave estrangeira).

CONSTRAINT nome_da_restricao **FOREIGN KEY** (coluna1, coluna2)) **REFERENCES** tabela_referenciada (chave1, chave2) **ON DELETE CASCADE**;

Ao usar a opção **ON UPDATE CASCADE**, quando um registro da tabela que possui a chave primária associada a esta chave estrangeira for alterado, então os registros associados também são alterados.

Gabarito: **Letra B**.

69- (FGV - 2015 - TCM-SP - Agente de Fiscalização - Tecnologia da Informação) Analise os três comandos a seguir e as afirmativas a respeito de seus efeitos no âmbito do MS SQL Server.

delete from x

truncate table x

drop table x

I. O comando delete e o comando truncate removem o mesmo conjunto de registros da tabela X.

II. O comando drop, quando usado com a opção "with no removal", produz exatamente o mesmo efeito do comando truncate.

III. Devido às suas características operacionais, o comando delete é usualmente executado muito mais rapidamente que o comando truncate.

Está correto o que se afirma em:

a) somente I;

b) somente III;

c) somente I e II;

d) somente II e III;

e) I, II e III.

Resolução:

Vamos analisar cada um dos itens:

I. **Correto**: o comando DELETE remove registros de uma tabela. Como não foi indicada nenhuma condição, isto é, não foi definida a cláusula WHERE, então serão excluídos todos os registros.

O comando TRUNCATE é para deletar todos os dados de uma tabela.

Logo, os dois comandos executam a remoção dos mesmos registros.

II. **Incorreto**: O comando drop deleta a estrutura em si e não os registros. Logo, a tabela será removida.

III. **Incorreto**: Devido às suas características operacionais, o comando ~~delete~~ **truncate** é usualmente executado muito mais rapidamente que o comando ~~truncate~~ **delete**.

A operação TRUNCATE não pode ser revertida, pois não registra logs dos registros removidos. Logo é mais rápida.

Gabarito: **Letra A**.

70- (FGV - 2014 - Câmara Municipal do Recife-PE - Programador) Em geral, a definição de chaves estrangeiras em bancos de dados relacionais pode vir acompanhada da especificação de procedimentos adicionais a serem adotados quando da exclusão/alteração de valores nos registros da tabela estrangeira.

Considere o seguinte script SQL para a criação das tabelas T1 e T2:

```
create table t1(
a int not null primary key,
b varchar(50) not null)

create table t2 (
c int not null primary key,
a int not null unique,
constraint fk_1 foreign key (a)
references t1(a))
```

É correto concluir que as tabelas T1 e T2 têm entre si:

a) um relacionamento 1:n

b) um relacionamento n:1

c) um relacionamento n:m

d) um relacionamento 1:1

e) qualquer tipo de relacionamento, dependendo da instância de cada uma

Resolução:

Galera, essa é a típica questão da banca que exige o conhecimento da necessidade de definir a chave estrangeira como UNIQUE para implementar um relacionamento 1:1.

Perceba que o a é definido em t2 como unique e é a chave estrangeira desta tabela. Logo, como não pode se repetir, então termos um relacionamento 1:1.

Gabarito: **Letra D**.

71- (FGV - 2014 - DPE-RJ - Técnico Superior Especializado - Administração de Dados) Considere os seguintes comandos de criação para as tabelas R e S no MS SQL Server.

create table R (A int not null primary key, B int)

create table S (A int not null unique, C char(3),

constraint S_R foreign key (A) references R (A))

É correto afirmar que as entidades representadas pelas tabelas R e S, respectivamente, têm entre si um relacionamento

a) 1:0

b) 1:1

c) 1:N

d) N:1

e) N:M

Resolução:

Galera, essa é a típica questão da banca que exige o conhecimento da necessidade de definir a chave estrangeira como UNIQUE para implementar um relacionamento 1:1.

Perceba que o A é definido em S como unique e é a chave estrangeira desta tabela. Logo, como não pode se repetir, então termos um relacionamento 1:1.

Gabarito: **Letra B**.

72- (FGV - 2012 - Senado Federal - Analista Legislativo - Análise de Suporte de Sistemas) A DDL da SQL descreve como as tabelas e outros objetos Oracle podem ser definidos, alterados e removidos. De um modo geral, é a parte utilizada pelo DBA. O comando que elimina um índice já criado é

a) REMOVE INDEX

b) DELETE INDEX

c) PURGE INDEX

d) ERASE INDEX

e) DROP INDEX

Resolução:

O comando DROP é o único DDL dentre os itens e é o comando utilizado para excluir estruturas de bancos de dados. Para excluir índices, então usamos DROP INDEX.

Gabarito: **Letra E**.

73- (FGV - 2010 - DETRAN-RN - Assessor Técnico - Administração de Banco de Dados) Sobre o comando "drop table pedido;" assinale a alternativa correta:

a) Cria a tabela pedido.

b) Elimina a tabela pedido.

c) Duplica a tabela pedido.

d) Cria uma chave primária na tabela pedido.

e) Extrai dados da tabela pedido.

Resolução:

A instrução **DROP TABLE** é usada para **deletar uma tabela existente**.

A sintaxe para esse comando é:

```
DROP TABLE nome_da_tabela;
```

Gabarito: **Letra B**.

1.4 VUNESP

74- (VUNESP - 2019 - Prefeitura de Itapevi - SP - Analista em Tecnologia da Informação e Comunicação) Considere a sintaxe SQL básica do comando para a criação de gatilhos:

CREATE TRIGGER <nome do trigger>

<instante de execução> <evento disparador>

ON <nome da tabela> ...

Nesse comando, as opções corretas do

a) evento disparador são: Insert, Delete e Update.

b) evento disparador são: Select, Group By e Inner Join.

c) evento disparador são: Commit, Rollback e Savepoint.

d) instante de execução são: First e Last.

e) instante de execução são: One e All.

Resolução:

Triggers ou gatilhos são **programas armazenados que são executados ou disparados automaticamente quando alguns eventos ocorrem**.

Os triggers são, de fato, escritos para serem executados em resposta a qualquer um dos seguintes eventos:

- Uma instrução de manipulação de banco de dados (DML) (DELETE, INSERT ou UPDATE)
- Uma instrução de definição de banco de dados (DDL) (CREATE, ALTER ou DROP).
- Uma operação de banco de dados (SERVERERROR, LOGON, LOGOFF, STARTUP ou SHUTDOWN).

Gabarito: **Letra A.**

75- (VUNESP - 2019 - Prefeitura de Campinas - SP - Analista de Tecnologia da Informação) O correto comando DDL do SQL para criar uma visão denominada White, tendo como origem os atributos One e Two da tabela Color, é:

a) CREATE VIEW White AS

(SELECT One, Two

FROM Color)

b) CREATE VIEW White

FROM Color (One, Two)

c) CREATE VIEW White USING

(Color(One, Two))

d) CREATE VIEW White

FROM TABLE (Color (One, Two))

e) CREATE VIEW White

(FROM Color

USING (One, Two))

Resolução:

A sintaxe a seguir é utilizada para **criar uma view** em SQL:

```
CREATE VIEW [Nome da View] AS
  SELECT Coluna1, Coluna2,...
  FROM nome_da_tabela
  WHERE...;
```

Nos itens b) a e) não temos a cláusula AS.

Gabarito: **Letra A.**

76- (VUNESP - 2018 - IPSM - Analista de Gestão Municipal - Informática TI) No sistema gerenciador de bancos de dados PostgreSQL (v. 9.5), o comando para renomear uma coluna chamada Alfa para a denominação Beta, da tabela denominada Produto, é:

a) ALTER TABLE Produto RENAME COLUMN Alfa TO Beta

b) MODIFY TABLE Produto COLUMN NAME Alfa TO Beta

c) CHANGE TABLE Produto COLUMN Alfa INTO Beta

d) TRANSFORM TABLE Produto COLUMN NAME Alfa TO Beta

e) REVERSE TABLE Produto SET COLUMN Alfa INTO Beta

Resolução:

A instrução **ALTER TABLE** é usada para **adicionar, deletar ou modificar colunas em uma tabela existente**. Essa instrução também pode ser utilizada para adicionar ou deletar restrições a esta tabela.

Gabarito: **Letra A.**

77- (VUNESP - 2016 - Prefeitura de Presidente Prudente - SP - Analista de Tecnologia da Informação) Considere a seguinte tabela de um banco de dados relacional:

Item (ID, Nome, Categoria, Valor)

O comando SQL para criar uma visão chamada Alfa, tendo por base a tabela Item, apenas para itens de valor maior do que R$ 250,00 é:

a) CREATE AUX_TAB Alfa WHERE Valor > 250,00 FROM Item

b) CREATE INT_TAB Alfa WITH Valor > 250,00 STARTING FROM item

c) CREATE VISION Alfa ← → Item AS SELECT * FROM Item HAVING Valor > 250,00

d) CREATE VISION Alfa HAVING SELECT ALL FROM Item WITH Valor > 250,00

e) CREATE VIEW Alfa AS SELECT * FROM Item WHERE Valor > 250,00

Resolução:

A sintaxe a seguir é utilizada para **criar uma view** em SQL:

```
CREATE VIEW [Nome da View] AS
  SELECT Coluna1, Coluna2,...
  FROM nome_da_tabela
  WHERE...;
```

Gabarito: **Letra E.**

78- (VUNESP - 2014 - EMPLASA - Analista Administrativo - Tecnologia da Informação) Considerando o SQL, o formato geral do comando de criação de gatilhos é:

CREATE TRIGGER < nome do trigger>

< tempo de ação do trigger>

< evento para acionar o trigger>

ON < nome da tabela>

< ação>

O parâmetro < tempo de ação do trigger > possui as seguintes opções válidas:

a) BEFORE e AFTER.

b) BEGIN e END.

c) FIRST e LAST

d) SAME e DIFFERENT.

e) START e FINISH.

Resolução:

Em uma TRIGGER:

{BEFORE | AFTER | INSTEAD OF}: especifica quando o trigger será executado. BEFORE executa a trigger antes do evento. AFTER executa depois do evento. A cláusula INSTEAD OF é usada para criar uma trigger em uma VIEW.

Gabarito: **Letra A.**

79- (VUNESP - 2014 - PRODEST-ES - Analista de Tecnologia da Informação - Desenvolvimento de Sistemas) Considere a tabela T de um banco de dados relacional:

T (ID, Nome, Fone)

Indique a alternativa que contém a consulta SQL correta para criar uma visão V, a partir da tabela T, apenas para os Nomes começando pela letra J.

a) CREATE VIEW V FOR

(SELECT T.ID, T.Nome,T. Fone

FOR Nome NEXT 'J%')

b) CREATE VIEW V →

(SELECT T(ID, Nome, Fone)

WHERE Nome NEAR 'J%')

c) CREATE VIEW V

(SELECT ID, Nome, Fone

FROM T

WHERE Nome = 'J%')

d) CREATE VIEW V AS

(SELECT ID, Nome, Fone

FROM T

WHERE Nome LIKE 'J%')

e) CREATE VIEW V FROM

(SELECT ID, Nome, Fone

OF T

WHERE Nome IN 'J%')

Resolução:

A sintaxe a seguir é utilizada para **criar uma view** em SQL:

```
CREATE VIEW [Nome da View] AS
  SELECT Coluna1, Coluna2,...
```

FROM nome_da_tabela

WHERE...;

Somente o item d) respeita essa sintaxe.

Gabarito: **Letra D.**

80- **(VUNESP - 2013 - MPE-ES - Agente Especializado - Analista de Sistemas)** O comando do SQL para criar uma tabela de nome 'Convenio', com os atributos 'ID', 'Nome' e 'Tipo', sendo 'ID' o atributo chave,

a) CREATE TAB Convenio (ID, Nome, Tipo : Integer PK, Char (30) Char (15));

b) CREATE TAB Convenio (ID Integer PK, Nome, Tipo Char (30,15));

c) CREATE TAB Convenio (PRIMARY KEY ID Integer, Nome Char(30) NPK, Tipo Char (15) NPK);

d) CREATE TABLE Convenio (ID Integer PRIMARY KEY, Nome Char (30), Tipo Char 15));

e) CREATE TABLE Convenio (1. ID Integer PK, 2. Nome Char (30) AND 3. Tipo Char (15));

Resolução:

A sintaxe básica da Instrução **CREATE TABLE** é:

CREATE TABLE nome_da_tabela (
coluna1 tipo_de_dado,
....);

A definição de uma coluna como **PRIMARY KEY** varia de acordo com o SGBD:

- **No SQL Server / Oracle / MS Access**:

coluna1 tipo_de_dado **PRIMARY KEY**,

- **No MySQL**:

PRIMARY KEY (coluna1)

- **No MySQL / SQL Server / Oracle / MS Access para múltiplas colunas**:

CONSTRAINT nome_da_restricao **PRIMARY KEY** (coluna1, coluna2));

Logo,

a) **Incorreto**: os tipos dos dados devem ser informados logo após cada um dos atributos e não após :.

b) **Incorreto**: para definir uma chave primária se usa PRIMARY KEY e não PK, além disso não foi definido o tipo para o campo Nome.

c) **Incorreto**: a ordem de definição dos atributos é coluna1 tipo_de_dado restrição, assim o correto seria ID Integer PRIMARY KEY. Ademais, NPK não é uma restrição válida.

d) **Correto**: totalmente conforme a sintaxe.

e) **Incorreto**: a sintaxe do CREATE TABLE não prevê a definição de números para os campos, assim não são válidos 1., 2. e 3.

Gabarito: **Letra D.**

81- (VUNESP - 2013 - COREN-SP - Administrador de Banco de Dados) Considere o seguinte comando de definição de dados do SQL, para um banco de dados relacional:

CREATE TABLE Aluno

(Nome Char(30) PRIMARY KEY,

Tipo Char(20) NOT NULL,

'comando de restrição');

O 'comando de restrição' para que o Tipo fique restrito aos valores 'Comum' e 'Especial' é

a) Tipo = ('Comum' OR 'Especial')

b) Tipo IS ONLY IN ('Comum', 'Especial')

c) DESIGN Tipo AS ('Comum', 'Especial')

d) RESERVE Tipo IN ('Comum', 'Especial')

e) CONSTRAINT C1 CHECK (Tipo IN ('Comum', 'Especial'))

Resolução:

A restrição **CHECK** é **usada para limitar o intervalo de valores que pode ser colocado em uma coluna**. A definição restrição **CHECK** varia:

- **No SQL Server / Oracle / MS Access**:

coluna1 tipo_de_dado **CHECK** (condicao)

- **No MySQL**:

coluna1 tipo_de_dado,

CHECK coluna1 (condicao)

- **No MySQL / SQL Server / Oracle / MS Access para múltiplas colunas**:

coluna1 tipo_de_dado,

coluna2 tipo_de_dado,

CONSTRAINT nome_da_restricao **CHECK** (condicao1 **AND** condicao2));

Gabarito: **Letra E.**

82- (VUNESP - 2013 - CTA - Tecnologista Pleno - Computação) O comando do SQL para remover uma visão de nome Green é

a) DROP VIEW Green

b) DELETE SIGHT Green

c) REMOVE ALL Green

d) CASCADE Green

e) ERASE Green

Resolução:

Uma visão é **apagada** com o comando **DROP VIEW**:

DROP VIEW [Nome da View]**;**

Gabarito: **Letra A.**

83- (VUNESP - 2012 - SPTrans - Analista de Informática) Para excluir a chave primária da tabela, é necessário utilizar o comando

a) ALTER TABLE post EXCLUDE primary key;

b) ALTER TABLE post DROP primary key;

c) ALTER TABLE post DELETE primary key(id_post);

d) DELETE TABLE post DROP primary key;

e) DROP TABLE post LESS primary key(id_post);

Resolução:

Para excluir uma restrição, basta utilizar a cláusula DROP no ALTER TABLE:

- **No MySQL**:

ALTER TABLE nome_da_tabela

DROP PRIMARY KEY;

- **No SQL Server / Oracle / MS Access**:

ALTER TABLE nome_da_tabela

DROP CONSTRAINT nome_da_restricao;

Gabarito: **Letra B.**

1.5 QUADRIX

84- **(Quadrix - 2021 - CRBM - 4 - Técnico em Informática)** Quanto aos sistemas de bancos de dados e à linguagem de consulta estruturada (SQL), julgue o item.

Em um banco de dados MySQL, para se criar um banco de dados de nome dbEmpresa, é suficiente executar o comando a seguir. CREATE DATABASE dbEmpresa;

Resolução:

O comando CREATE DATABASE é usado para criar bancos de dados. Assim, o comando

CREATE DATABASE dbEmpresa

Irá criar um banco de dados chamado dbEmpresa.

Gabarito: **Certo**.

85- **(Quadrix - 2021 - CRBM - 4 - Técnico em Informática)** Quanto aos sistemas de bancos de dados e à linguagem de consulta estruturada (SQL), julgue o item.

A linguagem DDL (Data Definition Language) é um conjunto de comandos responsáveis pela consulta e pela atualização dos dados armazenados em um banco de dados.

Resolução:

A DDL não é a linguagem responsável pela consulta e atualização, essa seria a DML.

A **DDL (Data Definition Language)** é a sublinguagem do SQL que permite ao utilizador **definir tabelas novas e elementos associados**.

Gabarito: **Errado**.

86- **(Quadrix - 2021 - CRECI - 14ª Região (MS) - Analista de TI)**

```
1.  CREATE TABLE corretor
2.   ( corretor_id NUMBER(6),
3.     nome VARCHAR2(30) CONSTRAINT cor_nome_nn NOT NULL,
4.     email VARCHAR2(35) CONSTRAINT cor_email_nn NOT NULL,
5.     celular VARCHAR2(20),
6.     data_nascimento DATE CONSTRAINT cor_data_nascimento_nn NOT NULL,
7.     salario NUMBER(8,2),
8.     comissao NUMBER(2,2),
9.     gerente_id NUMBER(8),
10.    imobiliaria_id NUMBER(6),
11.    CONSTRAINT cor_salario_min CHECK (salario > 0),
12.    CONSTRAINT cor_email_uk UNIQUE (email)
13.  );
14.  CREATE UNIQUE INDEX cor_id_pk ON corretor (corretor_id);
15.  ALTER TABLE corretor ADD
16.  ( CONSTRAINT cor_id_pk PRIMARY KEY (corretor_id),
17.    CONSTRAINT cor_imob_fk FOREIGN KEY (imobiliaria_id) REFERENCES imobiliaria,
18.    CONSTRAINT cor_gerente_fk FOREIGN KEY (gerente_id) REFERENCES corretor
19.  );
```

Com relação ao código SQL acima, julgue o item.

A coluna data_nascimento, especificada na linha 6, não aceita valores nulos.

Resolução:

A coluna data_nascimento é definida como NOT NULL, logo não aceita valores nulos.

Gabarito: **Certo**.

87- (Quadrix - 2021 - CRECI - 14ª Região (MS) - Analista de TI)

```
1.  CREATE TABLE corretor
2.   ( corretor_id NUMBER(6),
3.     nome VARCHAR2(30) CONSTRAINT cor_nome_nn NOT NULL,
4.     email VARCHAR2(35) CONSTRAINT cor_email_nn NOT NULL,
5.     celular VARCHAR2(20),
6.     data_nascimento DATE CONSTRAINT cor_data_nascimento_nn NOT NULL,
7.     salario NUMBER(8,2),
8.     comissao NUMBER(2,2),
9.     gerente_id NUMBER(8),
10.    imobiliaria_id NUMBER(6),
11.    CONSTRAINT cor_salario_min CHECK (salario > 0),
12.    CONSTRAINT cor_email_uk UNIQUE (email)
13.  );
14.  CREATE UNIQUE INDEX cor_id_pk ON corretor (corretor_id);
15.  ALTER TABLE corretor ADD
16.  ( CONSTRAINT cor_id_pk PRIMARY KEY (corretor_id),
17.    CONSTRAINT cor_imob_fk FOREIGN KEY (imobiliaria_id) REFERENCES imobiliaria,
18.    CONSTRAINT cor_gerente_fk FOREIGN KEY (gerente_id) REFERENCES corretor
19.  );
```

Com relação ao código SQL acima, julgue o item.

A criação da chave primária está especificada na linha 16.

Resolução:

Perfeitamente. Na linha 16, temos a criação de uma CONSTRAINT do tipo PRIMARY KEY, que define correto_id como chave primária da tabela.

Gabarito: **Certo**.

88- (Quadrix - 2019 - CRESS - SC - Assistente de Comunicação e Tecnologia

Código 1:

```
CREATE TABLE Assistente_Social
(
   ID_Func      NUMERIC(4)    NOT NULL,
   NomeFunc     VARCHAR(30)   NOT NULL,
   Endereco     VARCHAR(50)   NOT NULL,
   DataNasc     DATE          NOT NULL,
   Sexo         CHAR(1)       NOT NULL,
   Salario      NUMERIC(8,2)  NOT NULL,
   ID_Depto     NUMERIC(2)    NOT NULL,
   CONSTRAINT pk_func PRIMARY KEY (ID_Func),
   CONSTRAINT ck_sexo CHECK (Sexo='M' or Sexo='F')
);
```

Código 2:

```
SELECT ID_Depto, AVG(Salario)
FROM Assistente_Social
WHERE AVG(Salario) > 5000
GROUP BY ID_Depto;
```

No que diz respeito aos códigos 1 e 2 da linguagem SQL acima apresentados, julgue o item.

A instrução CREATE TABLE está especificada da forma correta no Código 1.

Resolução:

O comando do Código 1 está totalmente correto. Através desse comando, há a criação da tabela Assistente_Social com os atributos ID_Func, NomeFunc, Endereco, DataNasc, Sexo, Salario e ID_Depto.

Foram definidas também duas restrições:

- CONSTRAINT pk_func PRIMARY KEY (ID_Func) define uma restrição de chave primária, sendo o ID_Func o campo utilizado como chave.
- CONSTRAINT ck_seco CHECK (Sexo='M' or Sexo='F') define uma restrição de checagem, na qual o atributo sexo deve sempre um caractere 'M' ou 'F'.

Gabarito: Certo.

1.6 FUNCAB

89- (FUNCAB - 2012 - MPE-RO - Analista de Sistemas) Na criação de uma tabela em um banco de dados MySQL, o parâmetro UNIQUE do comando CREATE INDEX:

a) define a chave estrangeira.

b) define a chave primária.

c) garante a unicidade de um registro.

d) determina a ordem física das linhas correspondentes em uma tabela.

e) determina a direção de classificação de uma determinada coluna.

Resolução:

O parâmetro UNIQUE em um índice garante que não haverá duplicidade nos registros.

Para isso, utiliza-se CREATE UNIQUE INDEX:

```
CREATE UNIQUE INDEX nome_do_indice
ON nome_da_tabela (coluna1, coluna2, ...);
```

Gabarito: **Letra C**.

90- (FUNCAB - 2014 - PRODAM-AM - Analista de Banco de Dados) A diferença básica dos conceitos de trigger e stored procedure é que, respectivamente:

a) são executadas de acordo com um evento, mas não são inclusas no banco de dados.

b) é executada de acordo com um evento; é chamada para ser executada e são inclusas no banco de dados.

c) são executadas após serem chamadas, porém a primeira não é inclusa no banco de dados.

d) são executadas após serem chamadas, porém a segunda não é inclusa no banco de dados.

e) é chamada para ser executada; é executada de acordo com um evento e não são inclusas no banco de dados.

Resolução:

Uma trigger é disparada com base em um evento e a procedure pode ser executada com uma chamada.

Gabarito: Letra B.

1.7 UNIRIO

91- (UNIRIO - 2014 - UNIRIO - Analista Tecnologia da Informação - Desenvolvimento de Sistemas) Na criação de uma tabela em um banco de dados MySQL, o parâmetro UNIQUE do comando CREATE INDEX:

Considere a seguinte especificação de índice na linguagem SQL:

CREATE UNIQUE INDEX ORD_PROD_IDX ON ORDERS (MFR, PRODUTO);

É CORRETO afirmar que a especificação constrói

a) um índice para a tabela PRODUTO baseado na coluna MFR desta tabela, sendo que todos os elementos do índice são únicos.

b) um índice para a tabela MFR baseado na coluna PRODUTO desta tabela, sendo que todos os elementos do índice são únicos.

c) um índice para a tabela ORDERS baseado nas colunas MFR e PRODUTO, sendo que a combinação das duas colunas tem valor único no índice.

d) um índice único para a tabela ORDERS baseado na coluna PRODUTO com critério de ordenação MFR.

e) um índice único para a tabela PRODUTO baseado na coluna ORDERS com critério de ordenação MFR.

Resolução:

A sintaxe de criação de um índice único ou exclusivo é:

```
CREATE UNIQUE INDEX nome_do_indice
ON nome_da_tabela (coluna1, coluna2, ...);
```

Dessa forma, o comando

CREATE UNIQUE INDEX ORD_PROD_IDX

ON ORDERS (MFR, PRODUTO);

Cria um índice exclusivo chamado ORD_PROD_IDX para a tabela ORDERS usando as colunas MFR e PRODUTO. Sendo um índice UNIQUE, então os valores para as duas colunas não devem se repetir para mais de uma linha.

Gabarito: **Letra C**.

2. LISTA DE QUESTÕES

2.1 CESPE/CEBRASPE

1- (CESPE / CEBRASPE - 2021 - APEX Brasil - Analista - Tecnologia da Informação e Comunicação)

create database pessoa;

O comando SQL apresentado anteriormente criará

a) um banco de dados denominado pessoa;.

b) uma tabela denominada pessoa;.

c) um tipo de dados denominado pessoa;.

d) um esquema denominado pessoa;.

2- (CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Ciência de Dados) Julgue o item a seguir, a respeito de conceitos de SQL.

O comando CREATE DATABASE TAB é utilizado para criar uma tabela em um banco de dados.

3- (CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Desenvolvimento de Software)

Tendo como referência o diagrama de entidade relacionamento precedente, julgue o próximo item, a respeito de linguagem de definição de dados e SQL.

A expressão SQL a seguir permite excluir as notas do aluno de nome Fulano.

truncate from matricula where aluno='Fulano'

4- (CESPE / CEBRASPE - 2020 - Ministério da Economia - Tecnologia da Informação - Desenvolvimento de Software)

Tendo como referência o diagrama de entidade relacionamento precedente, julgue o próximo item, a respeito de linguagem de definição de dados e SQL.

As expressões DDL a seguir permitem a criação das tabelas presentes no diagrama apresentado.

```
create table aluno (
id integer primary key,
nome varchar(40) );
create table disciplina (
id integer primary key,
descricao varchar(60)
);

create table matricula (
aluno integer,
disciplina integer,
ano integer,
nota numeric,
constraint pk_matricula primary key (aluno,
disciplina, ano),
constraint fk_matricula_aluno foreign key
(aluno)
references aluno,
constraint fk_matricula_disciplina foreign
key (disciplina)
references disciplina );
```

5- (CESPE - 2018 - FUB - Técnico de Tecnologia da Informação) Julgue o item subsecutivo, a respeito de linguagem de definição e manipulação de dados.

O comando DROP TABLE permite excluir do banco de dados a definição de uma tabela e de todos os seus dados.

6- (CESPE - 2018 - BNB - Especialista Técnico - Analista de Sistema) Acerca de bancos de dados, julgue o item que segue.

O código a seguir, criado no SQL Server 2017, apresenta uma visão materializada, especificamente devido ao argumento

SCHEMABINDING.

```
CREATE VIEW VwTeste
WITH SCHEMABINDING
AS
SELECT campo1 FROM tabela WHERE campo1 > 17;
```

7- (CESPE - 2018 - STJ - Técnico Judiciário - Desenvolvimento de Sistemas)

Julgue o item a seguir, referente à modelagem de dados.

A DDL (data definition language) é usada para a definição da estrutura do banco de dados ou do esquema. São comandos DDL: CREATE, TRUNCATE, GRANT e ROLLBACK.

8- (CESPE - 2017 - TRE-PE - Analista Judiciário - Análise de Sistemas)

Tabela 3A6AAA

dados da tabela:

ID; nome; idtipo; preco

25; creme; 3; 11,50

31; arroz; 4; 12,50

34; leite; 1; 14,00

42; sabão; 5; 11,00

46; carne; 1; 12,75

48; shampoo; 5; 12,30

58; azeite; 1; 13,25

Assinale a opção que apresenta o comando SQL correto para se incluir um novo campo idcategoria do tipo INT nos dados da tabela 3A6AAA, denominada tbproduto.

a) ALTER TABLE tbproduto INSERT idcategoria INT;

b) ALTER TABLE tbproduto ADD COLUMN idcategoria INT;

c) UPDATE TABLE tbproduto ADD COLUMN idcategoria INT;

d) ADD COLUMN idcategoria INT IN TABLE tbprodut;

e) UPDATE TABLE ADD COLUMN idcategoria INT IN tbproduto;

9- (CESPE - 2016 - FUB - Técnico de Tecnologia da Informação) A respeito das principais instruções da linguagem SQL, julgue o item subsecutivo.

A instrução create assertion <nome-asserção> check <predicado> é utilizada para definir restrições de integridade.

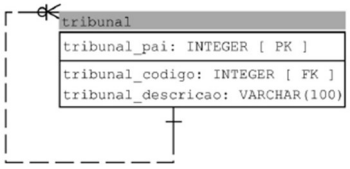
10- (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Suporte) No que concerne à linguagem SQL, julgue o item seguinte.

O comando create table pode ser utilizado para criar tanto uma tabela vazia quanto uma com dados de outra tabela.

11- (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Sistema) Julgue o próximo item, relativo à linguagem de definição de dados (DDL).

A expressão DDL abaixo cria a tabela referente ao diagrama de entidade e relacionamento apresentado a seguir.

```
create table tribunal(
tribunal_codigo integer ,
tribunal_descricao varchar(100),
tribunal_pai integer primary key,
constraint fk_tribunal
foreign key (tribunal_codigo )
references tribunal
)
```



12- (CESPE - 2016 - TCE-PA - Auditor de Controle Externo - Área Informática - Analista de Suporte) No que concerne à linguagem SQL, julgue o item seguinte.

Ao se criar uma view, não é necessário que os nomes dos atributos da view sejam os mesmos dos atributos da tabela.

13- (CESPE - 2016 - POLÍCIA CIENTÍFICA - PE - Perito Criminal - Ciência da Computação) Em SQL, para alterar a estrutura de uma tabela do banco de dados e incluir nela uma nova foreign key, é correto utilizar o comando

a) convert

b) group by.

c) alter table.

d) update.

e) insert.

14- (CESPE - 2016 - POLÍCIA CIENTÍFICA - PE - Perito Criminal - Ciência da Computação) Na linguagem SQL, o comando create table é usado para criar uma tabela no banco de dados; enquanto o relacionamento entre duas tabelas pode ser criado pela declaração

a) null.

b) primary key.

c) constraint.

d) auto_increment.

e) not null.

15- (CESPE - 2015 - MEC – Desenvolvedor) Com relação à linguagem de definição de dados (DDL) e à linguagem de manipulação de dados (DML), julgue o próximo item.

A DML utiliza o comando CREATE para inserir um novo registro na tabela de dados.

16- (CESPE - 2015 - MEC – Desenvolvedor)

```
CREATE TABLE PESSOA (
ID INTEGER NOT NULL,
NOME CHAR(50) NOT NULL UNIQUE,
CPF DECIMAL (11,0) NULL,
NACIONALIDADE INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (NACIONALIDADE)
REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)
);
```

Com base no comando SQL apresentado, julgue o item subsequente.

A cláusula NULL na coluna CPF indica que o conteúdo dessa coluna pode ser zero, já que ela é do tipo DECIMAL (11,0).

17- (CESPE - 2015 - MEC – Desenvolvedor)

CREATE TABLE PESSOA (

ID INTEGER NOT NULL,

NOME CHAR(50) NOT NULL UNIQUE,

CPF DECIMAL (11,0) NULL,

NACIONALIDADE INTEGER NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (NACIONALIDADE)

REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)

);

Com base no comando SQL apresentado, julgue o item subsequente.

A tabela TABELA_NACIONALIDADE deve ter uma coluna de nome DESCRICAO_NACIONALIDADE para obter o texto equivalente a cada código.

18- (CESPE - 2015 - MEC – Desenvolvedor)

CREATE TABLE PESSOA (

ID INTEGER NOT NULL,

NOME CHAR(50) NOT NULL UNIQUE,

CPF DECIMAL (11,0) NULL,

NACIONALIDADE INTEGER NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (NACIONALIDADE)

REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)

);

Com base no comando SQL apresentado, julgue o item subsequente.

Na tabela TABELA_NACIONALIDADE, CODIGO_NACIONALIDADE deve ser PRIMARY KEY.

19- (CESPE - 2015 - MEC – Desenvolvedor) CREATE TABLE PESSOA (

ID INTEGER NOT NULL,

NOME CHAR(50) NOT NULL UNIQUE,

CPF DECIMAL (11,0) NULL,

NACIONALIDADE INTEGER NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (NACIONALIDADE)

REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)

);

Com base no comando SQL apresentado, julgue o item subsequente.

Mais de uma PESSOA pode ter o mesmo NOME e a mesma NACIONALIDADE.

20- (CESPE - 2015 - MEC – Desenvolvedor)

CREATE TABLE PESSOA (

ID INTEGER NOT NULL,

NOME CHAR(50) NOT NULL UNIQUE,

CPF DECIMAL (11,0) NULL,

NACIONALIDADE INTEGER NOT NULL,

PRIMARY KEY (ID),

FOREIGN KEY (NACIONALIDADE)

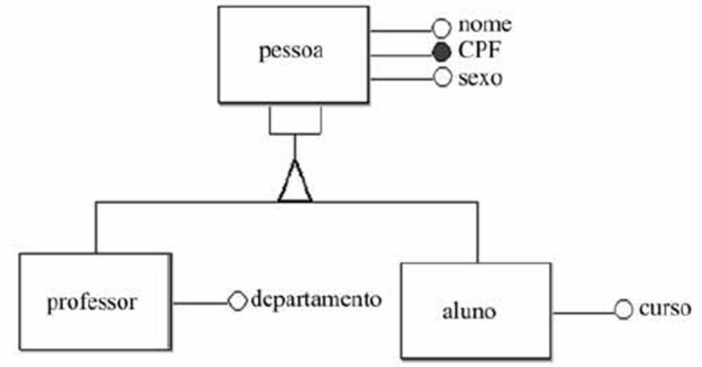
REFERENCES TABELA_NACIONALIDADE(CODIGO_NACIONALIDADE)

);

Com base no comando SQL apresentado, julgue o item subsequente.

A tabela criada terá quatro colunas.

21- (CESPE - 2015 - TJ-DFT - Técnico Judiciário - Programação de Sistemas)



Com base no diagrama de entidade e relacionamento (DER) apresentado, julgue o item que se segue a respeito de modelagem de dados e linguagem de definição dos dados.

A seguir é apresentado o comando SQL correto para gerar o esquema físico do DER.

CREATE TABLE Pessoa (

Nome VARCHAR(50),

Sexo VARCHAR(1),

CPF VARCHAR(11),

Inheritance (aluno, pessoa))

CREATE TABLE Aluno (

Curso VARCHAR(20),

CPF VARCHAR(11) PRIMARY KEY)

CREATE TABLE Professor (

Departamento VARCHAR(30),

CPF VARCHAR(11) PRIMARY KEY)

22- (CESPE / CEBRASPE - 2015 - TRE-MT - Técnico Judiciário - Programação de Sistemas) create table departamento (

coddepto integer primary key, nomedepto varchar(100)

);

create table professor (

codprof integer primary key,

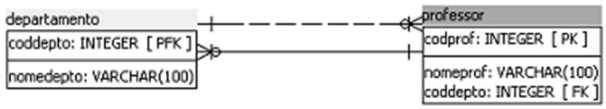
nomeprof varchar(100),

coddepto integer,

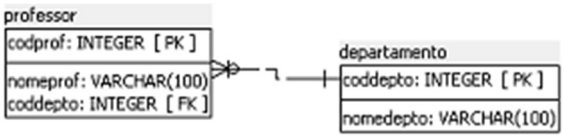
constraint fkprofessor foreign key (coddepto) references departamento);

Assinale a opção correta acerca do diagrama de entidade e relacionamento que representa a declaração relativa ao código SQL apresentado acima.

a)



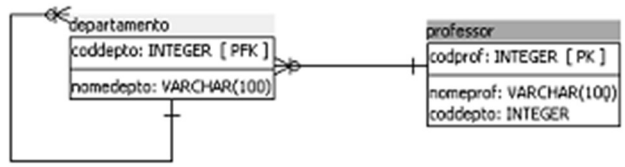
b)



c)

d)

e)



23- (CESPE - 2015 - MEC - Administrador de Dados) Julgue o item que se segue, com relação às definições e aos problemas de execução de comandos nas linguagens SQL.

Uma operação DELETE do SQL não é realizada se sua chave primária for referida por chaves estrangeiras em registros de outras tabelas no banco de dados. Assim, a fim de garantir a existência de chaves primárias para cada chave estrangeira nos bancos de dados relacionais, o SQL não apresenta nenhuma cláusula ou opção adicional que permita tal operação ocorrer nessa situação.

24- (CESPE - 2014 - ANATEL - Analista Administrativo - Tecnologia da Informação e Comunicação) Nos comandos em linguagem de consulta estruturada (SQL) apresentados a seguir, as chaves primárias estão sublinhadas e apenas horas_gastas é do tipo numérico, os demais campos são do tipo caractere. Em uma tarefa, com a utilização de um mesmo veículo, pode haver a participação de mais de um motorista na função de auxiliar.

MOTORISTA (cod_mot, nome, cnh)

TAREFA (cod_tarefa, cod_mot, cod_mot_auxiliar, placa, descricao, horas_gastas)

VEÍCULO (placa, ano, modelo)

Tendo como base as informações acima, julgue o item a seguir.

O comando a seguir exclui a chave primária, cod_tarefa, da tabela TAREFA.

Alter table tarefa drop primary key cod_tarefa;

25- (CESPE - 2014 - ANATEL - Analista Administrativo - Desenvolvimento de Sistemas) Julgue os itens seguintes, a respeito das linguagens de banco de dados.

A DDL (data definition language) é responsável pela especificação da instância do banco de dados e também pode ser usada para especificar propriedades adicionais dos dados, como restrições de consistência.

26- **(CESPE - 2014 - SUFRAMA - Analista de Sistemas)** No que se refere a linguagem de implementação de banco de dados, julgue os itens subsequentes.

O comando drop table remove toda a tabela da base de dados. Um exemplo de utilização desse comando é o seguinte: drop table examplo_timestamp;

27- **(CESPE - 2013 - BACEN - Analista - Análise e Desenvolvimento de Sistemas)** Julgue os itens seguintes, a respeito das linguagens de banco de dados.

CREATE TABLE Pessoa

(

Id int NULL,

Matricula int NOT NULL,

Nome varchar(255) NOT NULL,

DataNascimento date NULL)

CREATE TABLE EnderecoPessoa

(Id int NOT NULL,

TipoEndereco char (1) NOT NULL,

Endereco varchar(255),

Cidade char(55),

UF varchar (2)

)

Considerando os scripts acima para criação das Tabelas Pessoa e EnderecoPessoa, julgue o item seguinte.

Considerando que o campo Id na tabela Pessoa esteja corretamente configurado como chave primária simples, para se criar uma chave estrangeira entre as tabelas Pessoa e EnderecoPessoa, deve-se executar o comando a seguir.

ALTER TABLE EnderecoPessoa

ADD CONSTRAINT fk_Endereco_Pessoa FOREIGN KEY (P_id) REFERENCES Pessoa (Id)

28- (CESPE - 2013 - BACEN - Analista - Análise e Desenvolvimento de Sistemas)
Julgue os itens seguintes, a respeito das linguagens de banco de dados.

CREATE TABLE Pessoa

(

Id int NULL,

Matricula int NOT NULL,

Nome varchar(255) NOT NULL,

DataNascimento date NULL)

CREATE TABLE EnderecoPessoa

(Id int NOT NULL,

TipoEndereco char (1) NOT NULL,

Endereco varchar(255),

Cidade char(55),

UF varchar (2)

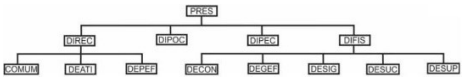
)

Considerando os scripts acima para criação das Tabelas Pessoa e EnderecoPessoa, julgue o item seguinte.

Para criar uma chave primária composta na Tabela Pessoa, deve-se executar o seguinte comando.

ALTER TABLE Pessoa ADD CONSTRAINT pk_PessoaID PRIMARY KEY (Id, Matricula)

29- (CESPE - 2013 - BACEN - Analista - Suporte à Infraestrutura de Tecnologia da Informação)



A expressão SQL abaixo cria uma tabela com estrutura que permite armazenar informações acerca dos órgãos e sua hierarquia.

create table orgao (

codigo integer,

sigla char(10),

codigo_pai integer,

constraint pk_orgao primary key (codigo),

constraint fk_orgao foreign key (codigo_pai) references orgao)

30- (CESPE / CEBRASPE - 2013 - CNJ - Técnico Judiciário - Programação de Sistemas) No que se refere ao conceito de banco de dados relacional, julgue os itens seguintes.

Na linguagem de consulta estruturada (SQL), é correto utilizar o comando TRUNCATE TABLE, com a finalidade de excluir todos os dados de uma tabela.

31- (CESPE - 2013 - TRT - 10ª REGIÃO (DF e TO) - Técnico Judiciário - Tecnologia da Informação) Em relação aos comandos da linguagem SQL, julgue os itens seguintes.

O comando abaixo permite adicionar a tabela disciplinas a uma chave estrangeira com o nome fk_curso, do campo id_curso que pertence à tabela cursos.

alter table disciplinas

alter column fk_curso references cursos (id_curso);

32- (CESPE - 2008 - STJ - Técnico Judiciário - Informática) Acerca da linguagem SQL, usada para fazer a manipulação e a definição de dados em sistemas gerenciadores de banco de dados, julgue os itens subsequentes.

O comando CREATE INDEX, usado para criar um parâmetro relacionado com uma tabela para buscar dados mais rapidamente, é considerado como DDL.

2.2 FCC

33- (FCC - 2019 - SANASA Campinas - Analista de Tecnologia da Informação - Suporte de DBA-Banco de Dados) Atenção: Para responder à questão, considere os dados abaixo.

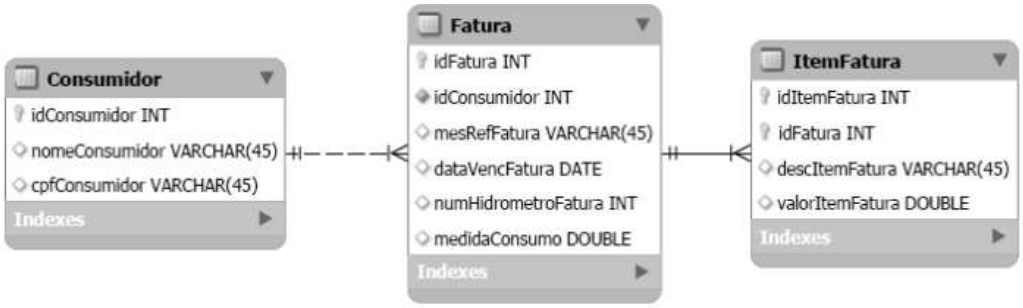


Tabela `Consumidor`:

idConsumidor	nomeConsumidor	cpfConsumidor
1	Paulo Vieira Lima	156.167.178-2
2	Marcos Santana Silva	234.156.765-12
3	Maria de Fátima Caetano Rosa	187.198.056-7
4	Zoraide Pereira Mota	238.765.234-12

Tabela `Fatura`:

idFatura	idConsumidor	mesRefFatura	DataVencFatura	numHidrometroFatura	medidaConsumo
3365693	1	05/2019	2019-05-20	344036	80
3366691	2	05/2019	2019-05-10	345681	120
3367690	1	04/2019	2019-04-28	344036	89
3390871	2	03/2019	2019-03-18	345681	100

Tabela `ItemFatura`:

idItemFatura	idFatura	descItemFatura	valorItemFatura
1	3365693	Captação de água bruta	0.1
1	3366691	Tratamento de água	90
1	3367690	Tratamento de água	61.23
2	3365693	Tratamento de água	60.19
2	3366691	Tratamento de esgoto	67.67
3	3365693	Coleta de esgoto	47.35
3	3366691	Afastamento de esgoto	80
4	3365693	Tratamento de Esgoto	25

Considere que não há nenhum registro cadastrado além dos mostrados nas tabelas acima.

Para excluir da tabela Fatura o campo mesRefFatura deve-se utilizar o comando

a) DROP COLUMN mesRefFatura FROM Fatura;

b) ALTER TABLE Fatura DROP COLUMN mesRefFatura;

c) DELETE COLUMN mesRefFatura FROM Fatura;

d) ALTER TABLE Fatura DELETE COLUMN mesRefFatura;

e) DROP COLUMN mesRefFatura WHERE TABLE= 'Fatura';

34- (FCC - 2019 - SABESP - Estagiário Ensino Médio Regular) Considere os comandos SQL abaixo.

```
I  (
   nome_Manancial varchar(50) PRIMARY KEY,
   volume number NOT NULL,
   media_historica number,
   data date
);

II  ('Cantareira', 58.0, , '12-05-2019');
```

Para que o primeiro comando crie a tabela MANANCIAIS e o segundo comando insira dados nesta tabela, deve-se preencher as lacunas I e II, correta e respectivamente, com

a) CREATE MANANCIAIS AS TABLE - INSERT INTO MANANCIAIS

b) CREATE TABLE MANANCIAIS - INSERT INTO MANANCIAIS VALUES

c) CREATE TABLE MANANCIAIS - INSERT IN TABLE MANANCIAIS

d) CREATE TABLE NAMED MANANCIAIS - INSERT INTO TABLE MANANCIAIS VALUES

e) CREATE MANANCIAIS AS TABLE - INSERT IN MANANCIAIS

35- (FCC - 2019 - SEFAZ-BA - Auditor Fiscal - Tecnologia da Informação - Prova II) Em um banco de dados aberto e em condições ideais há uma tabela chamada Contribuinte cuja chave primária é idContribuinte. Há também uma tabela chamada Imposto cuja chave primária é idimposto. Para criar uma tabela de associação chamada Contribuinte_imposto cuja chave primária é composta pelos campos idContribuinte e idImposto, que são chaves estrangeiras resultantes da relação dessa tabela com as tabelas Contribuinte e Imposto, utiliza-se a instrução SQL

a) CREATE TABLE Contribuinte_Imposto(idContribuinte INT, idImposto INT, PRIMARY KEY (idContribuinte), FOREIGN KEY (idContribuinte) REFERENCES Contribuinte (idContribuinte), PRIMARY KEY (idImposto), FOREIGN KEY (idContribuinte) REFERENCES Contribuinte (idContribuinte));

b) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), CONSTRAINT fk1 FOREIGN KEY (idContribuinte) REFERENCES Contribuinte (idContribuinte), CONSTRAINT fk2 FOREIGN KEY (idImposto) REFERENCES Imposto (idImposto));

c) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), FOREIGN KEY (idContribuinte) SOURCE Contribuinte (idContribuinte), FOREIGN KEY (idImposto) SOURCE Imposto (idImposto));

d) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), FOREIGN KEY (idContribuinte, idImposto) REFERENCES (Contribuinte!idContribuinte, Imposto!idImposto));

e) CREATE TABLE Contribuinte_Imposto(idContribuinte INT NOT NULL, idImposto INT NOT NULL, PRIMARY KEY (idContribuinte, idImposto), FOREIGN KEY (idContribuinte, idImposto) REFERENCES all parents);

36- (FCC - 2019 - SANASA Campinas - Analista de Tecnologia da Informação - Suporte de DBA-Banco de Dados) Considere o código SQL abaixo, que gerou a tabela ItemFatura.

CREATE TABLE ItemFatura(

idItemFatura INT NOT NULL,

idFatura INT NOT NULL,

descItemFatura VARCHAR(45),

valorItemFatura DOUBLE,

..I..

);

Considerando que a tabela ItemFatura possui chave primária composta pelos campos idItemFatura e idFatura, e que se uma fatura for excluída, automaticamente serão excluídos todos os seus itens, a lacuna I deve ser preenchida corretamente por

a) PRIMARY KEY (idItemFatura, FOREIGN KEY(idFatura)) REFERENCES Fatura(idFatura) ON DELETE CASCADE

b) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) WITH DELETE CASCADE

c) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY (idFatura) REFERENCES Fatura(idFatura) ON DELETE CASCADE

d) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) REFERENCES Fatura(idFatura)

e) PRIMARY KEY (idItemFatura, idFatura), FOREIGN KEY(idFatura) FROM Fatura(idFatura) ON DELETE CASCADE

37- (FCC - 2019 - TRF - 4ª REGIÃO - Analista Judiciário - Sistemas de Tecnologia da Informação) Uma Analista digitou o comando TRUNCATE TABLE processos; em um banco de dados SQL aberto em condições ideais para

a) excluir os dados da tabela, mas não a tabela em si.

b) excluir a estrutura da tabela e os dados nela contidos.

c) juntar a tabela aberta na memória com a tabela processos.

d) bloquear a tabela processos para uso exclusivo de seu usuário.

e) editar a estrutura da tabela em modo gráfico.

38- (FCC - 2018 - SABESP - Estagiário - Nível Médio Técnico) Considere que foi criada uma tabela em um banco de dados relacional capaz de armazenar os dados de clientes da SABESP, usando o comando SQL: CREATE TABLE Clientes_Sabesp (Cliente VARCHAR (50), MedidorAnt INTEGER NOT NULL, MedidorAtual INTEGER NOT NULL, Periodo DATE NOT NULL, Conta DECIMAL (10,2), PRIMARY KEY (Cliente));

É correto afirmar que

a) a instrução PRIMARY KEY define que o campo Cliente será a chave primária simples da tabela e este campo não pode ser nulo nem repetido.

b) com exceção do campo Cliente todos os outros campos são do tipo numérico inteiro.

c) para criar uma chave estrangeira, basta acrescentar FOREIGN KEY (Cliente) após a definição da PRIMARY KEY.

d) a chave estrangeira de um banco de dados relacional é usada para criar relacionamentos com as demais tabelas do banco de dados, por isso deve ter o mesmo nome da chave primária seguido de _FK. Nesta tabela seria Cliente_FK.

e) há um erro neste comando: todos os campos da tabela devem ser NOT NULL e os campos Cliente e Conta não têm esta restrição.

39- (FCC - 2018 - DPE-AM - Analista em Gestão Especializado de Defensoria - Analista de Banco de Dados) O comando SQL–ANSI para criar um procedimento chamado P1, que selecione os atributos A e B, de uma tabela T é:

a) PROCEDURE P1 IS

SELECT A, B

FROM T;

b) INSERT PROCEDURE P1 INTO DATABASE AS

SELECT A, B

FROM T

c) CREATE PROCEDURE P1( )

SELECT A, B

FROM T;

d) MAKE PROCEDURE P1 (SELECT A, B

FROM T);

e) PROCEDURE P1 AS

SELECT A, B

FROM T

40- (FCC - 2017 - TRF - 5ª REGIÃO - Técnico Judiciário - Informática) Após constatar que todos os dados em uma tabela estavam incorretos, foi solicitado ao Técnico em Informática para limpar os registros desta tabela mantendo sua estrutura, para que os dados corretos fossem posteriormente inseridos. Para realizar este trabalho o Técnico terá que utilizar a instrução SQL

a) DROP TABLE table_name.

b) REDO * FROM table_name.

c) DELETE TABLE table_name.

d) ERASE * FROM table_name.

e) TRUNCATE TABLE table_name.

41- (FCC - 2017 - DPE-RS - Analista - Banco de Dados) O comando SQL para criar uma visão V1, a partir de uma tabela T1, obtendo os atributos A1, A2 e A3 e os renomeando para C1, C2 e C3 é:

a) CREATE VIEW V1.C1, V1.C2. V1.C3

SELECT T1.A1, T1.A2, T1.A3;

b) CREATE VIEW V1 (C1, C2, C3)

AS SELECT A1, A2, A3

FROM T1;

c) CREATE VIEW C1, C2, C3 IN V1

FROM A1, A2, A3 OF T1;

d) CREATE VIEW V1

FROM T1

SELECT A1 → C1, A2 → C2, A3 → C3;

e) CREATE VIEW V1 (C1, C2, C3)

AS PART OF T1 (A1, A2, A3);

42- (FCC - 2017 - DPE-RS - Técnico - Informática) Um Técnico está criando uma tabela filha chamada funcionario, que será relacionada a uma tabela pai chamada departamento, por meio da chave estrangeira. Como parte do comando CREATE TABLE, usado para criar a tabela filha, ele deseja estabelecer uma restrição de chave estrangeira chamada emp_dept_fk para o campo department_id, que fará referência ao campo department_id que é chave primária na tabela departamento. Esta restrição será criada corretamente se for utilizada, na criação da tabela funcionario, a instrução SQL

a) RESTRICTION emp_dept_fk FOREIGN KEY (department_id) PRIMARY KEY departmento(department_id)

b) FOREIGN KEY emp_dept_fk FIELD(department_id) REFERENCES departmento(department_id)

c) CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) REFERENCES departmento(department_id)

d) DEFINE CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) WITH REFERENCES departmento department_id)

e) CONSTRAINT emp_dept_fk FOREIGN KEY (department_id) PRIMARY KEY departmento(department_id)

43- (FCC - 2017 - DPE-RS - Analista - Banco de Dados) O comando SQL para criar uma tabela denominada Natural, contendo os campos ID, Nome, Cidade e País, sendo todos do tipo caractere e ID a chave primária é:

a) CREATE TABLE Natural

(ID, Nome, Cidade, País, (CHAR(15), CHAR(30), CHAR(20), CHAR(20));

b) CREATE TABLE Natural

(ID PK, Nome CHAR(30), Cidade CHAR(20), País CHAR(20));

c) CREATE TABLE Natural

(ID PK, Nome, Cidade, País, (CHAR(15, 30, 20, 20));

d) CREATE TABLE Natural

(ID Char(15) PRIMARY KEY, Nome CHAR(30), Cidade CHAR(20), País (CHAR(20));

e) CREATE TABLE Natural

(ID PRIMARY KEY CHAR(15), Nome AND Cidade AND País CHAR(30, 20, 20));

44- (FCC - 2017 - TST - Técnico Judiciário – Programação) Para criar um banco de dados relacional chamado Tribunal e excluir uma tabela chamada Consulta, um Programador deverá escrever corretamente as expressões SQL

a) CREATE DATABASE Tribunal; e DELETE TABLE Consulta;

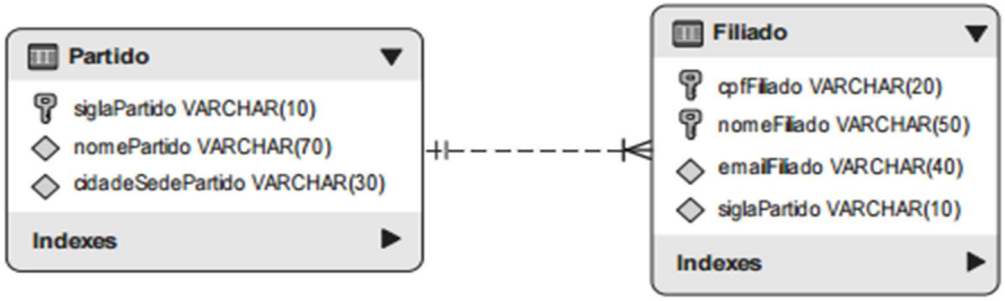
b) INSERT DATABASE Tribunal; e DELETE TABLE = Consulta;

c) CREATE DATABASE Tribunal; e DROP TABLE Consulta;

d) INSERT DATABASE Tribunal; e DROP TABLE Consulta;

e) INSERT DATABASE Tribunal; e DROP TABLE = Consulta;

45- (FCC - 2016 - AL-MS - Técnico de Informática) Para responder à questão, considere o modelo mostrado na imagem abaixo, oriundo de uma situação hipotética:



Após criadas as tabelas Partido e Filiado, foram incluídos, respectivamente, os seguintes registros:

siglaPartido	nomePartido	cidadeSedePartido
PDT	Partido Democrático Trabalhista	Brasília
PMDB	Partido do Movimento Democrático Brasileiro	Brasília
PSDB	Partido da Social Democracia Brasileira	São Paulo

cpfFiliado	nomeFiliado	emailFiliado	siglaPartido
124.179.156-10	André Braga	braga@hotmail.com	PMDB
147.189.237-18	Marcos Pereira	mpereira@hotmail.com	PDT
154.496.172-14	Pedro Silva	pedro@gmail.com	PDT
192.345.176-01	Maria Souza	maria@ig.com.br	PSDB

Após a tabela Partido ser criada, para criar a tabela Filiado foi utilizada a instrução abaixo:

```
CREATE TABLE IF NOT EXISTS Filiado (
 cpfFiliado VARCHAR(20) NOT NULL,
 nomeFiliado VARCHAR(50),
 emailFiliado VARCHAR(40),
 siglaPartido VARCHAR(10) NOT NULL,
 PRIMARY KEY (cpfFiliado),
 FOREIGN KEY (siglaPartido)
I Partido (siglaPartido)
);
```

A lacuna I deverá ser corretamente preenchida por

a) CASCADE CONSTRAINT

b) REFERENCES

c) REFERENCE CONSTRAINT

d) EXTENDS

e) IMPLEMENTS

46- (FCC - 2016 - SEGEP-MA - Técnico da Receita Estadual - Tecnologia da Informação - Conhecimentos Específicos) Atenção: Para responder às questões, considere a figura abaixo.

Considere que na tabela Contribuinte estão cadastrados os seguintes dados:

IDContribuinte	NomeContribuinte	CPF_CNPJ
1	Paulo da Silva	154.246.037-12
2	Maria Pereira	143.172.129-50

Após as tabelas Imposto e Contribuinte terem sido criadas, para criar a tabela Contribuinte_Imposto deve ser utilizada a seguinte instrução SQL:

```
CREATE TABLE Contribuinte_Imposto (IDContribuinte INT NOT NULL, SiglaImposto VARCHAR(10) NOT
NULL, Valor_Imposto DOUBLE, PRIMARY KEY (IDContribuinte, SiglaImposto), I );
```

A lacuna I é corretamente preenchida por:

a) FOREIGN KEY (IDContribuinte) EXTENDS Contribuinte (IDContribuinte), FOREIGN KEY (SiglaImposto) EXTENDS Imposto (SiglaImposto)

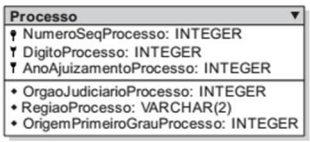
b) FOREIGN_KEY (IDContribuinte) CONSTRAINT Contribuinte (IDContribuinte), FOREIGN_KEY (SiglaImposto) CONSTRAINT Imposto (SiglaImposto)

c) FOREIGN KEY (IDContribuinte) REFERENCES Contribuinte (IDContribuinte), FOREIGN KEY(SiglaImposto) REFERENCES Imposto (SiglaImposto)

d) FOREIGN KEY (IDContribuinte, Contribuinte), FOREIGN KEY (SiglaImposto, Imposto)

e) REFERENCES Contribuinte (IDContribuinte) FOREIGN KEY, REFERENCES Imposto (SiglaImposto) FOREIGN KEY

47- (FCC - 2016 - TRF - 3ª REGIÃO - Técnico Judiciário - Informática) Para responder a questão, considere as informações abaixo.



Registros cadastrados:

NumeroSeqProcesso	DigitoProcesso	AnoAjuizamentoProcesso	OrgaoJudiciarioProcesso	RegiaoProcesso	OrigemPrimeiroGrauProcesso
15472	49	2002	4	3	3300
16592	44	2014	4	3	4500
17543	45	1999	4	3	3300
24535	23	2002	4	3	3300
29670	41	2012	4	2	2200
36535	45	2000	4	1	3400
44672	40	2012	4	2	3400
45891	43	2007	4	1	4400
67234	39	1997	4	1	3500

Considere que a tabela Processo foi criada sem chave primária. Nesse caso, para definir a chave primária, antes de serem inseridos registros, deve-se utilizar a instrução SQL

a) ADD TO Processo PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

b) INSERT INTO Processo PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

c) ALTER TABLE Processo ADD PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

d) ADD CONTRAINT PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso) from Processo;

e) UPDATE TABLE Processo ADD PRIMARY KEY(NumeroSeqProcesso, DigitoProcesso, AnoAjuizamentoProcesso);

48- (FCC - 2016 - TRT - 23ª REGIÃO (MT) - Técnico Judiciário - Tecnologia da Informação)

Atenção: Para responder a questão, considere a informação abaixo.

Um Técnico está participando da modelagem de um banco de dados utilizando o Modelo Entidade-Relacionamento – MER e se deparou, dentre outras, com a entidade Processo, que contém os seguintes atributos:

NumeroProcesso – inteiro (PK)
DigitoProcesso – inteiro (PK)
AnoProcesso – inteiro (PK)
NumeroOABAdvogadoProcesso – cadeia de caracteres
NomeAdvogadoProcesso – cadeia de caracteres
NumeroOrgaoJudiciarioProcesso – inteiro (FK)
NumeroTribunal – inteiro (FK)
NumeroUnidadeOrigemProcesso – inteiro (FK)

Após criar a tabela Processo no Sistema Gerenciador de Banco de Dados SQL Server, para definir uma restrição que especifica que o campo AnoProcesso só poderá receber números inteiros maiores do que 2014, o Técnico deve utilizar a instrução

a) ADD CONSTRAINT Processo CHECK (AnoProcesso>2014);

b) ALTER TABLE Processo ADD CHECK (AnoProcesso>2014);

c) ADD CONSTRAINT (AnoProcesso>2014) FROM Processo;

d) CREATE CONSTRAINT Chk_Processo FROM Processo CHECK (AnoProcesso>2014);

e) ALTER TABLE Processo ADD CONSTRAINT (AnoProcesso>2014);

49- (FCC - 2015 - TRT - 9ª REGIÃO (PR) - Analista Judiciário - Área Apoio Especializado - Tecnologia da Informação)

A tabela relativa a Débitos Trabalhistas a seguir deve ser utilizada para responder à questão.

Considere que a tabela já está criada, os dados iniciais já foram inseridos e o banco de dados a ser utilizado está aberto e funcionando em condições ideais.

Tabela DebTrab

NroProcesso	Principal	Juros	FGTS	Honor Periciais
111/15	25345.00	3801.75	7933.00	4755.00
777/15	125800.00	18870.00	57966.87	7543.00
333/15	8844.50	1326.67	4233.55	1781.00
555/15	327631.00	65526.20	104863.78	11523.00
444/15	5072.00	1014.40	895.14	700.00

Um Analista da área de TI trabalha em uma organização que possui aplicações que utilizam os SGBDs Oracle 11g e SQL Server. Ele identificou que o comando SQL que está correto e pode ser aplicado em ambas as plataformas é

a) ALTER TABLE DebTrab ALTER COLUMN NroProcesso integer;

b) ALTER TABLE DebTrab MODIFY NroProcesso int;

c) ALTER TABLE DebTrab ADD DataPartida data;

d) ALTER TABLE DebTrab ADD IndiceAtualiz float;

e) ALTER TABLE DebTrab DROP COLUMN DataPartida;

50- (FCC - 2015 - DPE-SP - Programador)

Considere que os comandos foram executados na sequência da numeração indicada:

```
[1] CREATE TABLE Tab_InfJuv
    (ID integer UNIQUE, Unidade varchar (50), Defensor varchar (50), Tipo integer);
[2] INSERT INTO Tab_InfJuv VALUES ( 11, 'Chacara Santo Antonio-Sao Paulo', 'Jorge da Silva', 1);
[3] INSERT INTO Tab_InfJuv VALUES ( 14, 'Parque Monteiro Soares- Sao Paulo', 'Maria Joana Santos', 2);
[4] INSERT INTO Tab_InfJuv VALUES ( 12, 'Centro-São Paulo', 'Jorge da Silva', 2);
[5] INSERT INTO Tab_InfJuv VALUES ( 32, 'Centro-Diadema', 'Ana Maria da Silva', 1);
```

O comando SQL correto é:

a) ALTER TABLE Tab_InfJuv ADD FOREIGN KEY (ID_Unidade) REFERENCES Tab_InfJuv (ID);

b) ALTER TABLE Tab_InfJuv ADD Nome_Menor varchar(100);

c) CREATE VIEW V_Tab_InfJuv AS SELECT Unidade, Defensor, ID_Unidade FROM Tab_InfJuv;

d) ALTER TABLE TabInfJuv DROP Nome_Menor;

e) DELETE FROM TabInfJuv WHERE Defensor = 'Jorge da Silva';

51- (FCC - 2015 - TRE-AP - Técnico Judiciário - Programação de Sistemas) Um Técnico do TREAP − Tribunal Regional Eleitoral do Amapá ficou responsável por criar uma tabela no Banco de Dados denominada Tab_PAA que seja capaz de armazenar o código da auditoria (um identificador único, não nulo), o nome do auditor responsável e o órgão sendo auditado. Para isso, ele utilizou, corretamente, o seguinte comando SQL:

a) CREATE TABLE Tab_PAA (Cod_Audit integer PRIMARY KEY, Nome_Auditor varchar(50), Orgao_Audit varchar(50));

b) CREATE_TABLE (Cod_Audit integer, Nome_Auditor varchar(50), Orgao_Audit varchar(50)) Tab_PAA;

c) CREATE_TABLE Tab_PAA (Cod_Audit integer UNIQUE; Nome_Auditor varchar(50); Orgao_Audit varchar(50));

d) CREATE TABLE Tab_PAA (varchar(50) Nome_Auditor, varchar(50), Orgao_Audit PRIMARY KEY integer Cod_Audit);

e) CREATE TABLE (UNIQUE NOT NULL Cod_Audit, VARCHAR(50) Nome_Auditor, VARCHAR(50) Orgao_Audit) Tab_PAA;

52- (FCC - 2015 - TRT - 15ª Região (SP) - Técnico Judiciário - Tecnologia da Informação) Deseja-se criar uma tabela chamada Departamento contendo os seguintes campos:

idDep − inteiro, chave primária, não nulo, auto numeração.

nomeDep − cadeia de caracteres com, no máximo, 50 caracteres, não nulo.

telefoneDep − cadeia de caracteres com, no máximo, 15 caracteres.

Considerando-se que o banco de dados está aberto e em condições ideais, o comando SQL que deverá ser utilizado é:

a) CREATE TABLE Departamento (

idDep INT NOT NULL AUTO_NUMBERING,

nomeDep VARCHAR(50) NOT NULL,

telefoneDep VARCHAR(15),
PRIMARY KEY (idDep));

b) CREATE TABLE Departamento (
idDep INT NOT NULL AUTOINCREMENT,
nomeDep VARCHAR2(50) NOT NULL,
telefoneDep VARCHAR2(15) NULLABLE,
PRIMARY KEY (idDep));

c) CREATE TABLE Departamento (
idDep INT NOT NULL AUTO_INCREMENT,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15),
PRIMARY KEY (idDep));

d) CREATE TABLE Departamento (
idDep INT NOT NULL AUTOINCREMENT,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15)
PRIMARY_KEY (idDep));

e) CREATE TABLE Departamento (
idDep INT NOT NULL AUTO_INCREMENT PRIMARY_KEY,
nomeDep VARCHAR(50) NOT NULL,
telefoneDep VARCHAR(15));

53- (FCC - 2015 - TRT - 15ª Região (SP) - Técnico Judiciário - Tecnologia da Informação) No SQL há dois comandos que podem eliminar completamente os registros de uma tabela. A diferença entre eles é que o comando I irá eliminar apenas os dados (registros), enquanto o comando II irá eliminar também a tabela.

Os comandos I e II são, respectivamente,

a) TRUNCATE TABLE e DROP TABLE.

b) DROP TABLE e DELETE FROM.

c) DELETE RECORD e DELETE TABLE.

d) DROP TABLE e TRUNCATE TABLE.

e) REMOVE RECORD e DROP TABLE.

54- (FCC - 2014 - TJ-AP - Analista Judiciário - Área Apoio Especializado - Tecnologia da Informação - Banco de Dados- DBA) Considere o seguinte trecho de um comando SQL ANSI para a criação de uma tabela:

```
CREATE TABLE Teste
  (Comando 1 . . .
    . . . )
```

Deseja-se declarar um atributo de nome Item, do tipo caractere, com 20 posições e que seja chave primária da tabela. Para tanto, o Comando 1 deve ser substituído por

a) PK Item CHAR (20)

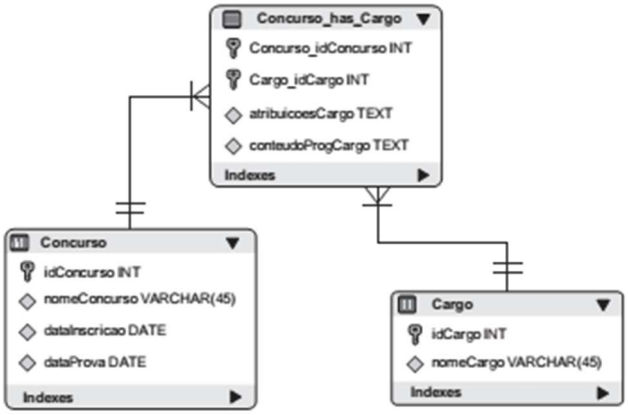
b) Item CHAR (20) PRIMARY KEY

c) CHAVE PRIMÁRIA Item CHAR: 20

d) Item CHAVE PRIMÁRIA CHAR (20)

e) Item PK CHAR: 20

55- (FCC - 2014 - TRF - 4ª REGIÃO - Analista Judiciário - Especialidade Informática) Atenção: Para responder à questão, considere o modelo de banco de dados abaixo.



As atribuições do cargo (atribuicoesCargo) e o conteúdo programático (conteudoProgCargo) normalmente são diferentes, dependendo do cargo e do concurso. O atributo dataInscricao refere-se à data de início das inscrições.

Para incluir na tabela Concurso o campo dataResultado para entrada de dados de tipo data, de forma que não aceite valores nulos, utiliza-se a instrução

a) ALTER COLUMN Concurso ADD dataResultado DATE NOT NULL;

b) INCLUDE COLUMN dataResultado FROM concurso DATE NOT NULL;

c) ALTER TABLE concurso ADD dataResultado DATE NOT NULL;

d) ALTER TABLE Concurso INCLUDE dataResultado (DATE) NOT NULL;

e) INSERT COLUMN dataResultado FROM concurso DATATYPE DATE NOT NULL;

56- (FCC - 2014 - AL-PE - Analista Legislativo - Sistemas) Em um banco de dados há duas tabelas, departamento e funcionario, cujos campos são:

Departamento:

DepNo - int - primary key - not null

DNome - varchar(50) - not null

DLocal - varchar(40)

Funcionario:

FunNo - int - primary key - not null

DepNo - int - foreign key - not null

FNome - varchar(50) - not null

FCargo - varchar(40)

FSal - double

Ambas as tabelas possuem muitos registros cadastrados, que não incluem valores nulos. Considere as instruções SQL a seguir:

I. select funcionario.FunNo, funcionario.FNome, departamento.DNome from funcionario INNER JOIN departamento ON funcionario.DepNo = departamento.DepNo;

II. select funcionario.FunNo, funcionario.FNome, departamento.DNome from funcionario, departamento where funcionario.DepNo = departamento.DepNo;

III. select f.FunNo, f.FNome, d.DNome from (select departamento.DepNo, departamento.DNome from departamento) as d, funcionario as f where d.DepNo = f.DepNo;

IV. select DISTINCT f.FunNo, f.FNome, d.DNome from funcionario f, departamento d;

Para criar uma nova tabela física chamada funcionario2, apenas com os campos FunNo e FNome da tabela funcionario, incluindo os dados cadastrados nestes campos, em ordem alfabética crescente pelo campo FNome, utiliza-se a instrução:

a) COPY f.FunNo, f.FNome FROM funcionario as f TO funcionario2 ORDER BY f.FNome;

b) CREATE TABLE funcionario2 FIELDS f.FunNo, f.FNome FROM funcionario as f TO ORDER BY f.FNome;

c) COPY FunNo, FNome FROM funcionario INTO funcionario2 ORDER BY f.FNome ASC;

d) CREATE TABLE funcionario2 AS SELECT f.FunNo, f.FNome FROM funcionario as f ORDER BY FNome;

e) CREATE VIEW funcionario2 AS SELECT FunNo, FNome FROM funcionario ORDER BY FNome;

57- (FCC - 2013 - AL-RN - Técnico Legislativo - Programador) A restrição DEFAULT em SQL é utilizada para

a) introduzir um valor padrão em uma coluna.

b) determinar uma operação padrão em uma Trigger.

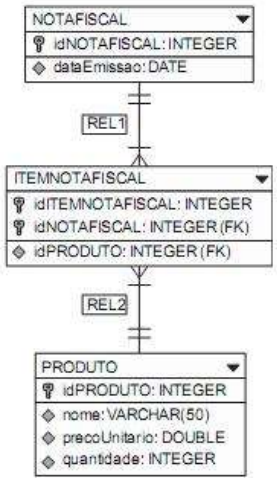
c) atribuir um nome padrão para uma tabela.

d) determinar o valor padrão de retorno para uma procedure.

e) determinar a visualização (view) padrão que será utilizada.

58- (FCC - 2013 - DPE-SP - Programador de computador)

Para responder às questões de números 41 e 42, utilize a figura abaixo, parte de um modelo Entidade-Relacionamento de um banco de dados.



Uma instrução SQL correta para criar a tabela NOTAFISCAL apresentada no modelo é:

a) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NOT NULL, dataEmissao DATE NULL, PRIMARY KEY(idNOTAFISCAL));

b) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NULL AUTOINCREMENT, dataEmissao DATE NULL, PRIMARY KEY(idNOTAFISCAL));

c) CREATE SCHEMA NOTAFISCAL (idNOTAFISCAL INTEGER NOT NULL AUTO_INCREMENT, dataEmissao DATE NOT NULL, PRIMARY KEY(idNOTAFISCAL));

d) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NULL, dataEmissao DATE NOT NULL, CONSTRAINT UNIQUE KEY(idNOTAFISCAL));

e) CREATE TABLE NOTAFISCAL (idNOTAFISCAL INTEGER NULL CONSTRAINT PRIMARY KEY, dataEmissao DATE NULL);

59- **(FCC - 2013 - MPE-AM - Agente de Apoio - Programador)** O comando SQL utilizado para adicionar, modificar ou remover colunas em uma tabela existente é chamado

a) INSERT INTO.

b) DROP TABLE.

c) CREATE TABLE.

d) ALTER TABLE.

e) TRUNCATE.

60- **(FCC - 2012 - MPE-AP - Técnico Ministerial - Informática)** Em linguagem SQL, o comando utilizado para remover uma tabela de um banco de dados é

a) DROP TABLE.

b) DELETE TABLE.

c) REMOVE TABLE.

d) DELETE FROM.

e) ROLLBACK.

61- **(FCC - 2011 - TRT - 19ª Região (AL) - Técnico Judiciário - Tecnologia da Informação)** É um comando do tipo DDL (Data Definition Language) no SQL:

a) SELECT.

b) DELETE.

c) INSERT.

d) UPDATE.

e) CREATE.

62- **(FCC - 2009 - TRT - 15ª Região - Analista Judiciário - Tecnologia da Informação)** As constraints para as tabelas de um BD relacional podem ser especificadas quando de um

a) ALTER, somente.

b) CREATE, somente.

c) CREATE ou quando de um ALTER.

d) CREATE ou quando de um INSERT.

e) INSERT ou quando de um ALTER.

2.3 FGV

63- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Desenvolvimento de Sistemas) Analise o comando de criação da tabela teste, com cinco constraints, exibido a seguir.

create table teste (

a int null,

b varchar (40) not null,

c int,

constraint pk_a primary key (a),

constraint fk_a_c foreign key (c) references teste,

constraint uq_b unique (b),

constraint up_c unique (c),

constraint ch_1 check (a >= 0.5 * c))

Assinale a constraint desse script que seria rejeitada no MS SQL Server.

a) A primeira.

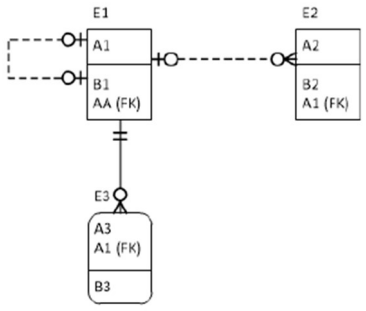
b) A segunda.

c) A terceira.

d) A quarta.

e) A quinta.

64- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Administrador de Banco de dados) Considere o diagrama IDEF1X a seguir.



Assinale a opção que apresenta o script SQL que cria a tabela E1 corretamente.

a) create table E1 (

A1 int not null primary key,

B1 int null,

AA int not null,

constraint FK_X foreign key (AA)

references E1(A1))

b) create table E1 (

A1 int not null primary key,

B1 int null,

AA int not null,

constraint FK_X foreign key (AA)

references E1(A1))

c) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null,

constraint FK_X foreign key (AA)

references E1(A1))

d) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (AA)

references E1(A1))

e) create table E1 (

A1 int not null primary key,

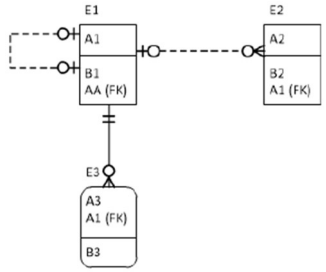
B1 int null,

AA int not null unique,

constraint FK_X foreign key (AA)

references E1(A1))

65- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Administrador de Banco de dados) Considere o diagrama IDEF1X a seguir.



Assinale a opção que apresenta o script SQL que cria a tabela E2 corretamente.

a) create table E2 (

A2 int null primary key,

B2 int null,

A1 int,

constraint FK_X foreign key (A1)

references E1(A1))

b) create table E2 (

A2 int not nul,

B2 int null,

A1 int null,

constraint PK primary key(A2, A1),

constraint FK_X foreign key (A1)

references E1(A1))

c) create table E2 (

A2 int not nul,

B2 int null,

constraint PK primary key(A2, A1),

constraint FK_X foreign key (A1)

references E1(A1))

d) create table E2 (

A2 int not null,

B2 int null,

A1 int null,

constraint FK_X foreign key (A1)

references E1(A1))

e) create table E2 (

A2 int not null primary key,

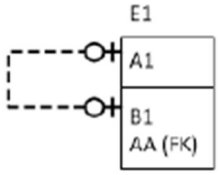
B2 int null,

A1 int null,

constraint FK_X foreign key (A1)

references E1(A1))

66- (FGV - 2018 - MPE-AL - Analista do Ministério Público - Desenvolvimento de Sistemas) Analise o diagrama IDEF1X a seguir.



Assinale a opção que indica o script SQL que cria a tabela E1 corretamente.

a) create table E1 (

A1 int null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (AA)

references E1(A1))

b) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null,

constraint FK_X foreign key (A1)

references E1(AA))

c) create table E1 (

A1 int not null unique,

B1 int null,

AA int null primary key,

constraint FK_X foreign key (AA)

references E1(A1))

d) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (A1)

references E1(AA))

e) create table E1 (

A1 int not null primary key,

B1 int null,

AA int null unique,

constraint FK_X foreign key (AA)

references E1(A1))

67- (FGV - 2015 - Câmara Municipal de Caruaru - PE - Analista Legislativo - Informática) Em geral, a definição de chaves estrangeiras em bancos de dados relacionais pode vir acompanhada da especificação de procedimentos adicionais a serem adotados quando da exclusão/alteração de valores nos registros da tabela estrangeira.

create table R2 (

a int not null primary key,

b int null,

x int not null,

constraint FK foreign key (x) references

R1(x))

Considerando o script acima, as opções complementares compatíveis para a definição da chave estrangeira FK são:

a) on delete cascade

on update set null

b) on delete cascade

on update cascade

c) on delete no action

on update set null

d) on delete set null

on update restrict

e) on delete restrict

on update set null

68- (FGV - 2015 - TCM-SP - Agente de Fiscalização - Tecnologia da Informação) Analise os três comandos a seguir e as afirmativas a respeito de seus efeitos no âmbito do MS SQL Server.

delete from x

truncate table x

drop table x

I. O comando delete e o comando truncate removem o mesmo conjunto de registros da tabela X.

II. O comando drop, quando usado com a opção "with no removal", produz exatamente o mesmo efeito do comando truncate.

III. Devido às suas características operacionais, o comando delete é usualmente executado muito mais rapidamente que o comando truncate.

Está correto o que se afirma em:

a) somente I;

b) somente III;

c) somente I e II;

d) somente II e III;

e) I, II e III.

69- (FGV - 2014 - Câmara Municipal do Recife-PE - Programador) Em geral, a definição de chaves estrangeiras em bancos de dados relacionais pode vir acompanhada da especificação de procedimentos adicionais a serem adotados quando da exclusão/alteração de valores nos registros da tabela estrangeira.

Considere o seguinte script SQL para a criação das tabelas T1 e T2:

```
create table t1(
a int not null primary key,
b varchar(50) not null)

create table t2 (
c int not null primary key,
a int not null unique,
constraint fk_1 foreign key (a)
references t1(a))
```

É correto concluir que as tabelas T1 e T2 têm entre si:

a) um relacionamento 1:n

b) um relacionamento n:1

c) um relacionamento n:m

d) um relacionamento 1:1

e) qualquer tipo de relacionamento, dependendo da instância de cada uma

70- (FGV - 2014 - DPE-RJ - Técnico Superior Especializado - Administração de Dados) Considere os seguintes comandos de criação para as tabelas R e S no MS SQL Server.

```
create table R ( A int not null primary key, B int )
create table S ( A int not null unique, C char(3),
        constraint S_R foreign key (A) references R (A))
```

É correto afirmar que as entidades representadas pelas tabelas R e S, respectivamente, têm entre si um relacionamento

a) 1:0

b) 1:1

c) 1:N

d) N:1

e) N:M

71- (FGV - 2012 - Senado Federal - Analista Legislativo - Análise de Suporte de Sistemas) A DDL da SQL descreve como as tabelas e outros objetos Oracle podem ser definidos, alterados e removidos. De um modo geral, é a parte utilizada pelo DBA. O comando que elimina um índice já criado é

a) REMOVE INDEX

b) DELETE INDEX

c) PURGE INDEX

d) ERASE INDEX

e) DROP INDEX

72- (FGV - 2010 - DETRAN-RN - Assessor Técnico - Administração de Banco de Dados) Sobre o comando "drop table pedido;" assinale a alternativa correta:

a) Cria a tabela pedido.

b) Elimina a tabela pedido.

c) Duplica a tabela pedido.

d) Cria uma chave primária na tabela pedido.

e) Extrai dados da tabela pedido.

2.4 VUNESP

73- (VUNESP - 2019 - Prefeitura de Itapevi - SP - Analista em Tecnologia da Informação e Comunicação) Considere a sintaxe SQL básica do comando para a criação de gatilhos:

CREATE TRIGGER <nome do trigger>

<instante de execução> <evento disparador>

ON <nome da tabela> ...

Nesse comando, as opções corretas do

a) evento disparador são: Insert, Delete e Update.

b) evento disparador são: Select, Group By e Inner Join.

c) evento disparador são: Commit, Rollback e Savepoint.

d) instante de execução são: First e Last.

e) instante de execução são: One e All.

74- (VUNESP - 2019 - Prefeitura de Campinas - SP - Analista de Tecnologia da Informação) O correto comando DDL do SQL para criar uma visão denominada White, tendo como origem os atributos One e Two da tabela Color, é:

a) CREATE VIEW White AS

(SELECT One, Two

FROM Color)

b) CREATE VIEW White

FROM Color (One, Two)

c) CREATE VIEW White USING

(Color(One, Two))

d) CREATE VIEW White

FROM TABLE (Color (One, Two))

e) CREATE VIEW White

(FROM Color

USING (One, Two))

75- (VUNESP - 2018 - IPSM - Analista de Gestão Municipal - Informática TI) No sistema gerenciador de bancos de dados PostgreSQL (v. 9.5), o comando para renomear uma coluna chamada Alfa para a denominação Beta, da tabela denominada Produto, é:

a) ALTER TABLE Produto RENAME COLUMN Alfa TO Beta

b) MODIFY TABLE Produto COLUMN NAME Alfa TO Beta

c) CHANGE TABLE Produto COLUMN Alfa INTO Beta

d) TRANSFORM TABLE Produto COLUMN NAME Alfa TO Beta

e) REVERSE TABLE Produto SET COLUMN Alfa INTO Beta

76- (VUNESP - 2016 - Prefeitura de Presidente Prudente - SP - Analista de Tecnologia da Informação) Considere a seguinte tabela de um banco de dados relacional:

Item (ID, Nome, Categoria, Valor)

O comando SQL para criar uma visão chamada Alfa, tendo por base a tabela Item, apenas para itens de valor maior do que R$ 250,00 é:

a) CREATE AUX_TAB Alfa WHERE Valor > 250,00 FROM Item

b) CREATE INT_TAB Alfa WITH Valor > 250,00 STARTING FROM item

c) CREATE VISION Alfa ← → Item AS SELECT * FROM Item HAVING Valor > 250,00

d) CREATE VISION Alfa HAVING SELECT ALL FROM Item WITH Valor > 250,00

e) CREATE VIEW Alfa AS SELECT * FROM Item WHERE Valor > 250,00

77- (VUNESP - 2014 - EMPLASA - Analista Administrativo - Tecnologia da Informação) Considerando o SQL, o formato geral do comando de criação de gatilhos é:

CREATE TRIGGER < nome do trigger>

< tempo de ação do trigger>

< evento para acionar o trigger>

ON < nome da tabela>

< ação>

O parâmetro < tempo de ação do trigger > possui as seguintes opções válidas:

a) BEFORE e AFTER.

b) BEGIN e END.

c) FIRST e LAST

d) SAME e DIFFERENT.

e) START e FINISH.

78- (VUNESP - 2014 - PRODEST-ES - Analista de Tecnologia da Informação - Desenvolvimento de Sistemas) Considere a tabela T de um banco de dados relacional:

T (ID, Nome, Fone)

Indique a alternativa que contém a consulta SQL correta para criar uma visão V, a partir da tabela T, apenas para os Nomes começando pela letra J.

a) CREATE VIEW V FOR

(SELECT T.ID, T.Nome,T. Fone

FOR Nome NEXT 'J%')

b) CREATE VIEW V →

(SELECT T(ID, Nome, Fone)

WHERE Nome NEAR 'J%')

c) CREATE VIEW V

(SELECT ID, Nome, Fone

FROM T

WHERE Nome = 'J%')

d) CREATE VIEW V AS

(SELECT ID, Nome, Fone

FROM T

WHERE Nome LIKE 'J%')

e) CREATE VIEW V FROM

(SELECT ID, Nome, Fone

OF T

WHERE Nome IN 'J%')

79- (VUNESP - 2013 - MPE-ES - Agente Especializado - Analista de Sistemas) O comando do SQL para criar uma tabela de nome 'Convenio', com os atributos 'ID', 'Nome' e 'Tipo', sendo 'ID' o atributo chave,

a) CREATE TAB Convenio (ID, Nome, Tipo : Integer PK, Char (30) Char (15));

b) CREATE TAB Convenio (ID Integer PK, Nome, Tipo Char (30,15));

c) CREATE TAB Convenio (PRIMARY KEY ID Integer, Nome Char(30) NPK, Tipo Char (15) NPK);

d) CREATE TABLE Convenio (ID Integer PRIMARY KEY, Nome Char (30), Tipo Char 15));

e) CREATE TABLE Convenio (1. ID Integer PK, 2. Nome Char (30) AND 3. Tipo Char (15));

80- **(VUNESP - 2013 - COREN-SP - Administrador de Banco de Dados)** Considere o seguinte comando de definição de dados do SQL, para um banco de dados relacional:

CREATE TABLE Aluno

(Nome Char(30) PRIMARY KEY,

Tipo Char(20) NOT NULL,

'comando de restrição');

O 'comando de restrição' para que o Tipo fique restrito aos valores 'Comum' e 'Especial' é

a) Tipo = ('Comum' OR 'Especial')

b) Tipo IS ONLY IN ('Comum', 'Especial')

c) DESIGN Tipo AS ('Comum', 'Especial')

d) RESERVE Tipo IN ('Comum', 'Especial')

e) CONSTRAINT C1 CHECK (Tipo IN ('Comum', 'Especial'))

81- **(VUNESP - 2013 - CTA - Tecnologista Pleno - Computação)** O comando do SQL para remover uma visão de nome Green é

a) DROP VIEW Green

b) DELETE SIGHT Green

c) REMOVE ALL Green

d) CASCADE Green

e) ERASE Green

82- **(VUNESP - 2012 - SPTrans - Analista de Informática)** Para excluir a chave primária da tabela, é necessário utilizar o comando

a) ALTER TABLE post EXCLUDE primary key;

b) ALTER TABLE post DROP primary key;

c) ALTER TABLE post DELETE primary key(id_post);

d) DELETE TABLE post DROP primary key;

e) DROP TABLE post LESS primary key(id_post);

2.5 QUADRIX

83- **(Quadrix - 2021 - CRBM - 4 - Técnico em Informática)** Quanto aos sistemas de bancos de dados e à linguagem de consulta estruturada (SQL), julgue o item.

Em um banco de dados MySQL, para se criar um banco de dados de nome dbEmpresa, é suficiente executar o comando a seguir. CREATE DATABASE dbEmpresa;

84- **(Quadrix - 2021 - CRBM - 4 - Técnico em Informática)** Quanto aos sistemas de bancos de dados e à linguagem de consulta estruturada (SQL), julgue o item.

A linguagem DDL (Data Definition Language) é um conjunto de comandos responsáveis pela consulta e pela atualização dos dados armazenados em um banco de dados.

85- **(Quadrix - 2021 - CRECI - 14ª Região (MS) - Analista de TI)**

```
1.  CREATE TABLE corretor
2.   ( corretor_id NUMBER(6),
3.     nome VARCHAR2(30) CONSTRAINT cor_nome_nn NOT NULL,
4.     email VARCHAR2(35) CONSTRAINT cor_email_nn NOT NULL,
5.     celular VARCHAR2(20),
6.     data_nascimento DATE CONSTRAINT cor_data_nascimento_nn NOT NULL,
7.     salario NUMBER(8,2),
8.     comissao NUMBER(2,2),
9.     gerente_id NUMBER(8),
10.    imobiliaria_id NUMBER(6),
11.    CONSTRAINT cor_salario_min CHECK (salario > 0),
12.    CONSTRAINT cor_email_uk UNIQUE (email)
13.  );
14.  CREATE UNIQUE INDEX cor_id_pk ON corretor (corretor_id);
15.  ALTER TABLE corretor ADD
16.  ( CONSTRAINT cor_id_pk PRIMARY KEY (corretor_id),
17.    CONSTRAINT cor_imob_fk FOREIGN KEY (imobiliaria_id) REFERENCES imobiliaria,
18.    CONSTRAINT cor_gerente_fk FOREIGN KEY (gerente_id) REFERENCES corretor
19.  );
```

Com relação ao código SQL acima, julgue o item.

A coluna data_nascimento, especificada na linha 6, não aceita valores nulos.

86- **(Quadrix - 2021 - CRECI - 14ª Região (MS) - Analista de TI)**

```
1.  CREATE TABLE corretor
2.   ( corretor_id NUMBER(6),
3.     nome VARCHAR2(30) CONSTRAINT cor_nome_nn NOT NULL,
4.     email VARCHAR2(35) CONSTRAINT cor_email_nn NOT NULL,
5.     celular VARCHAR2(20),
6.     data_nascimento DATE CONSTRAINT cor_data_nascimento_nn NOT NULL,
7.     salario NUMBER(8,2),
8.     comissao NUMBER(2,2),
9.     gerente_id NUMBER(8),
10.    imobiliaria_id NUMBER(6),
11.    CONSTRAINT cor_salario_min CHECK (salario > 0),
12.    CONSTRAINT cor_email_uk UNIQUE (email)
13.  );
14.  CREATE UNIQUE INDEX cor_id_pk ON corretor (corretor_id);
15.  ALTER TABLE corretor ADD
16.  ( CONSTRAINT cor_id_pk PRIMARY KEY (corretor_id),
17.    CONSTRAINT cor_imob_fk FOREIGN KEY (imobiliaria_id) REFERENCES imobiliaria,
18.    CONSTRAINT cor_gerente_fk FOREIGN KEY (gerente_id) REFERENCES corretor
19.  );
```

Com relação ao código SQL acima, julgue o item.

A criação da chave primária está especificada na linha 16.

87- (Quadrix - 2019 - CRESS - SC - Assistente de Comunicação e Tecnologia

Código 1:

```
CREATE TABLE Assistente_Social
(
   ID_Func       NUMERIC(4)     NOT NULL,
   NomeFunc   VARCHAR(30)  NOT NULL,
   Endereco     VARCHAR(50)  NOT NULL,
   DataNasc     DATE             NOT NULL,
   Sexo          CHAR(1)          NOT NULL,
   Salario       NUMERIC(8,2)  NOT NULL,
   ID_Depto     NUMERIC(2)    NOT NULL,
   CONSTRAINT pk_func PRIMARY KEY (ID_Func),
   CONSTRAINT ck_sexo CHECK (Sexo='M' or Sexo='F')
);
```

Código 2:

```
SELECT ID_Depto, AVG(Salario)
FROM Assistente_Social
WHERE AVG(Salario) > 5000
GROUP BY ID_Depto;
```

No que diz respeito aos códigos 1 e 2 da linguagem SQL acima apresentados, julgue o item.

A instrução CREATE TABLE está especificada da forma correta no Código 1.

2.6 FUNCAB

88- (FUNCAB - 2012 - MPE-RO - Analista de Sistemas) Na criação de uma tabela em um banco de dados MySQL, o parâmetro UNIQUE do comando CREATE INDEX:

a) define a chave estrangeira.

b) define a chave primária.

c) garante a unicidade de um registro.

d) determina a ordem física das linhas correspondentes em uma tabela.

e) determina a direção de classificação de uma determinada coluna.

89- (FUNCAB - 2014 - PRODAM-AM - Analista de Banco de Dados) A diferença básica dos conceitos de trigger e stored procedure é que, respectivamente:

a) são executadas de acordo com um evento, mas não são inclusas no banco de dados.

b) é executada de acordo com um evento; é chamada para ser executada e são inclusas no banco de dados.

c) são executadas após serem chamadas, porém a primeira não é inclusa no banco de dados.

d) são executadas após serem chamadas, porém a segunda não é inclusa no banco de dados.

e) é chamada para ser executada; é executada de acordo com um evento e não são inclusas no banco de dados.

2.7 UNIRIO

90- (UNIRIO - 2014 - UNIRIO - Analista Tecnologia da Informação - Desenvolvimento de Sistemas) Na criação de uma tabela em um banco de dados MySQL, o parâmetro UNIQUE do comando CREATE INDEX:

Considere a seguinte especificação de índice na linguagem SQL:

CREATE UNIQUE INDEX ORD_PROD_IDX ON ORDERS (MFR, PRODUTO);

É CORRETO afirmar que a especificação constrói

a) um índice para a tabela PRODUTO baseado na coluna MFR desta tabela, sendo que todos os elementos do índice são únicos.

b) um índice para a tabela MFR baseado na coluna PRODUTO desta tabela, sendo que todos os elementos do índice são únicos.

c) um índice para a tabela ORDERS baseado nas colunas MFR e PRODUTO, sendo que a combinação das duas colunas tem valor único no índice.

d) um índice único para a tabela ORDERS baseado na coluna PRODUTO com critério de ordenação MFR.

e) um índice único para a tabela PRODUTO baseado na coluna ORDERS com critério de ordenação MFR.

3. GABARITO

3.1 CESPE/CEBRASPE

1-	A	**8-**	B	**15-**	Errado	**22-**	B	**29-**	Certo
2-	Errado	**9-**	Certo	**16-**	Errado	**23-**	Errado	**30-**	Certo
3-	Errado	**10-**	Certo	**17-**	Errado	**24-**	Errado	**31-**	Errado
4-	Certo	**11-**	Certo	**18-**	Certo	**25-**	Errado	**32-**	Certo
5-	Certo	**12-**	Certo	**19-**	Errado	**26-**	Certo		
6-	Errado	**13-**	C	**20-**	Certo	**27-**	Errado		
7-	Errado	**14-**	C	**21-**	Errado	**28-**	Errado		

3.2 FCC

33-	B	**41-**	B	**49-**	D	**57-**	A
34-	B	**42-**	C	**50-**	B	**58-**	A
35-	B	**43-**	D	**51-**	A	**59-**	D
36-	C	**44-**	C	**52-**	C	**60-**	A
37-	A	**45-**	B	**53-**	A	**61-**	E
38-	A	**46-**	C	**54-**	B	**62-**	C
39-	C	**47-**	C	**55-**	C		
40-	E	**48-**	B	**56-**	D		

3.3 FGV

63-	A	**66-**	E	**69-**	D	**72-**	B
64-	D	**67-**	B	**70-**	B		
65-	E	**68-**	A	**71-**	E		

3.4 VUNESP

73-	A	**76-**	E	**79-**	D	**82-**	B
74-	A	**77-**	A	**80-**	E		
75-	A	**78-**	D	**81-**	A		

3.5 QUADRIX

83- Certo **84-** Errado **85-** Certo **86-** Certo **87-** Certo

3.6 FUNCAB

88- C **89-** B

3.7 UNIRIO

90- C