TI TOTAL

ÁREA FISCAL E CONTROLE

Professor
Ramon Souza

Tecnologia da Informação

TEORIA

SQL (DTL)

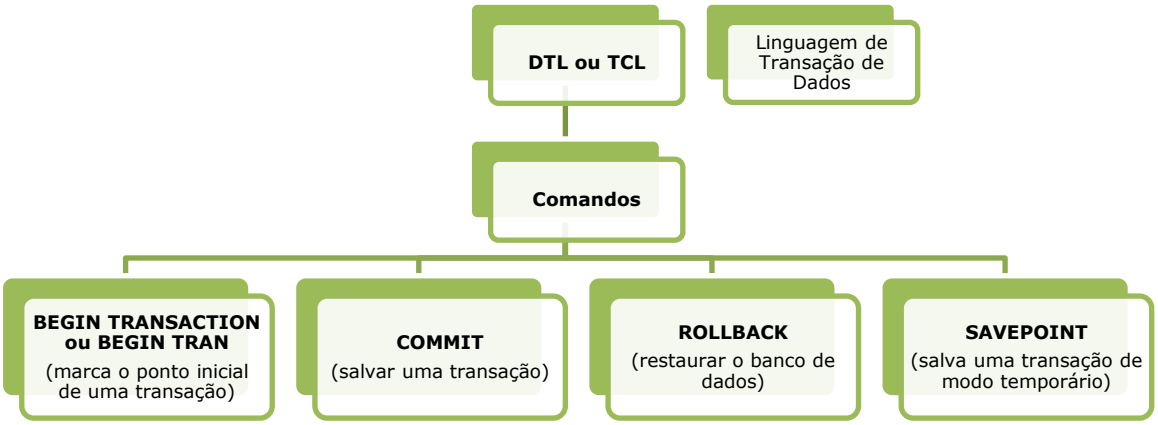
SUMÁRIO

A nossa aula é bem esquematizada, então para facilitar o seu acesso aos **esquemas**, você pode usar o seguinte índice:

1. SQL (DTL)

1.1 Introdução à DCL

A **DTL (Data Transaction Language) ou TCL (Transaction Control Language)** é a sublinguagem do SQL para **tratar as transações**. Os principais comandos desta linguagem são o **BEGIN TRANSACTION (ou BEGIN TRAN)**, o **COMMIT**, o **ROLLBACK** e o **SAVEPOINT**.



Esquema 1 – DTL.

1- (FUNDATEC - 2019 - Prefeitura de Gramado - RS - Técnico em Informática) Os comandos DTL são responsáveis por gerenciar diferentes transações ocorridas dentro de um Banco de dados. Ele é dividido em três comandos, quais sejam:

a) BEGIN TRAN, COMMIT e ROLLBACK.

b) BEGIN DEVTRAN, DEFINE e BACK.

c) BEGIN LIBRARY, FIND e ROLLBACK.

d) BEGIN, DEFINE LIBRARY e SCROLL.

e) TRAN, FIND e FINISH.

Resolução:

A **DTL (Data Transaction Language) ou TCL (Transaction Control Language)** é a sublinguagem do SQL para **tratar as transações**. Os principais comandos desta linguagem são o **BEGIN TRANSACTION (ou BEGIN TRAN)**, o **COMMIT** e o **ROLLBACK**.

Gabarito: **Letra A.**

1.2 Sintaxe básica do BEGIN TRANSACTION

O comando **BEGIN TRANSACTION ou BEGIN TRAN ou simplesmente BEGIN marca o início de uma transação**. As transações explícitas começam com a instrução BEGIN TRANSACTION e terminam com a instrução COMMIT ou ROLLBACK.

A sintaxe do comando BEGIN TRANSACTION é apresentada a seguir:

```
BEGIN TRANSACTION ou BEGIN TRAN ou BEGIN;
<instruções>
COMMIT ou ROLLBACK;
```

Vale ressaltar que se forem encontrados erros durante a execução das instruções após a cláusula BEGIN, todas as modificações poderão ser revertidas para voltar ao estado de consistência.

EXEMPLIFICANDO!!!

Iniciar uma transação de exclusão do Candidato cujo id é 13:

```
BEGIN TRANSACTION;
DELETE FROM HumanResources.JobCandidate
   WHERE JobCandidateID = 13;
COMMIT;
```

Nesse exemplo, o início da transação é marcado pela cláusula BEGIN TRANSACTION e, caso seja possível deletar o elemento de ID = 13, então a transação será executada com base no COMMIT.

1.3 Sintaxe básica do COMMIT

O **COMMIT** é o comando usado para **salvar permanentemente** qualquer **transação** em um banco de dados.

Quando usamos qualquer comando DML como INSERT, UPDATE ou DELETE, as alterações feitas por esses comandos não são permanentes, até que a sessão atual seja fechada, as alterações feitas por esses comandos podem ser revertidas. Para evitar isso, usamos o comando COMMIT para marcar as alterações como permanentes.

Para realizar um COMMIT, basta colocar essa palavra no fim de uma transação:

```
BEGIN TRANSACTION ou BEGIN TRAN ou BEGIN;
<instruções>
COMMIT;
```

EXEMPLIFICANDO!!!

A transação a seguir executa uma exclusão do um candidato de ID igual a 13.

```
BEGIN TRANSACTION;
DELETE FROM HumanResources.JobCandidate
   WHERE JobCandidateID = 13;
COMMIT;
```

Nesse exemplo, caso seja possível deletar o elemento de ID = 13, então a transação será executada com base no COMMIT.

2- **(NC-UFPR - 2017 - ITAIPU BINACIONAL - Profissional Nível Técnico I - Técnico em Eletrônica)** Assinale a alternativa que identifica corretamente o comando SQL usado para tornar permanentes as alterações realizadas desde o início de uma transação.

a) COMMIT b) SAVE c) SYNC d) FLUSH e) APPEND

Resolução:

O **COMMIT** é o comando usado para **salvar permanentemente** qualquer **transação** em um banco de dados.

Quando usamos qualquer comando DML como INSERT, UPDATE ou DELETE, as alterações feitas por esses comandos não são permanentes, até que a sessão atual seja fechada, as alterações feitas por esses comandos podem ser revertidas. Para evitar isso, usamos o comando COMMIT para marcar as alterações como permanentes.

Gabarito: **Letra A.**

1.4 Sintaxe básica do ROLLBACK

O **ROLLBACK** é o comando usado para **restaurar o banco de dados** para o último estado confirmado. Também é usado com o comando SAVEPOINT para ir para um ponto de salvamento em uma transação em andamento.

Se tivermos usado o comando UPDATE para fazer algumas alterações no banco de dados e percebermos que essas alterações não eram necessárias, poderemos usar o comando ROLLBACK para reverter essas alterações, se elas não foram confirmadas usando o comando COMMIT.

Para realizar um ROLLBACK, basta colocar essa palavra no fim de uma transação:

```
BEGIN TRANSACTION ou BEGIN TRAN ou BEGIN;
<instruções>
ROLLBACK;
```

EXEMPLIFICANDO!!!

O exemplo a seguir ilustra a criação de um a tabela ValueTable e a definição de uma transação para inserir valores nessa tabela.

```
CREATE TABLE ValueTable (id int);
BEGIN TRANSACTION;
    INSERT INTO ValueTable VALUES(1);
    INSERT INTO ValueTable VALUES(2);
ROLLBACK;
```

Neste exemplo, a instrução ROLLBACK reverterá a instrução INSERT, isto é, não serão inseridos os valores na tabela. Contudo, a tabela criada ainda continuará a existir, pois perceba que a instrução CREATE TABLE está fora do escopo da transação.

1.5 Sintaxe básica do SAVEPOINT

O **SAVEPOINT** é o comando usado **salvar temporariamente uma transação**, para que você possa reverter para esse ponto sempre que necessário.

Para definir um SAVEPOINT usamos:

```
SAVEPOINT <nome do ponto de salvamento>;
```

Para **retornar uma transação** até um ponto de salvamento, podemos usar **ROLLBACK TO SAVEPOINT** e para **excluir um ponto**, usamos **RELEASE SAVEPOINT**. É possível omitir a palavra SAVEPOINT, usando o nome do ponto de salvamento de forma direta.

EXEMPLIFICANDO!!!

O exemplo a seguir ilustra a criação de um a tabela ValueTable e a definição de uma transação para inserir valores nessa tabela.

```
BEGIN;
  INSERT INTO table1 VALUES (1);
  SAVEPOINT my_savepoint;
  INSERT INTO table1 VALUES (2);
  ROLLBACK TO SAVEPOINT my_savepoint;
  INSERT INTO table1 VALUES (3);
COMMIT;
```

Neste exemplo, ocorre o seguinte:

1. Insere-se o valor 1 na tabela.
2. Cria-se o SAVEPOINT my_savepoint.
3. Insere-se o valor 2 na tabela.
4. Recupera-se o estado salvo em my_savepoint. Nesse caso, tudo que foi executado entre o SAVEPOINT e o ROLLBACK TO SAVEPOINT foi desconsiderado. Logo, a tabela retorna ao estado de antes de ter sido inserido o valor 2.
5. Insere-se o valor 3 na tabela.
6. Conclui-se a transação.

Nesse exemplo, a tabela conterá os valores 1 e 3, mas não o valor 2.

3- **(FCC – 2017 – TRF – 5ª REGIÃO – Técnico Judiciário – Informática)** Um Técnico em informática utilizou, em um banco de dados aberto e em condições ideais, as instruções abaixo.

UPDATE funcionarios SET nome= 'Pedro' WHERE id=1;

SAVEPOINT altera;

INSERT INTO funcionarios VALUES (2,'Marcos');

Para descartar o que foi realizado após o SAVEPOINT, ou seja, a inserção do funcionário Marcos, utiliza-se a instrução PL/SQL

a) RESTORE TO altera WITH UNDO OPTION;

b) ROLLBACK TO altera;

c) UNDO TO altera;

d) COMMIT TO altera.

e) UNMAKE TO altera WITH ROLLBACK OPTION;

Resolução:

O **SAVEPOINT** é o comando usado **salvar temporariamente uma transação**, para que você possa reverter para esse ponto sempre que necessário.

Para **retornar uma transação** até um ponto de salvamento, podemos usar **ROLLBACK TO SAVEPOINT.** É possível omitir a palavra SAVEPOINT, usando o nome do ponto de salvamento de forma direta.

Assim, para desfazer todos os comandos que ocorreram após o ponto de salvamento altera, podemos usar:

ROLLBACK TO SAVEPOINT altera;

ou

ROLLBACK TO altera;

Gabarito: **Letra B.**

4- **(IBADE - 2019 - IF-RO - Analista de Tecnologia da Informação)** Em bancos de dados, uma transação é um conjunto de operações delimitadas por um início e um fim. Iniciando quando se executa o primeiro comando SQL e terminando de acordo com as seguintes situações:

(1) encerra a transação salvando permanentemente todas as alterações realizadas durante a transação.

(2) encerra a transação descartando todas as alterações realizadas durante a transação.

As operações em (1) e em (2) são conhecidas, respectivamente, por:

a) COMMIT e ROLLBACK.

b) COMMIT e REVOKE.

c) COMMIT e DROP.

d) SAVEPOINT e ROLLBACK.

e) SAVEPOINT e REVOKE.

Resolução:

A **DTL (Data Transaction Language) ou TCL (Transaction Control Language)** é a sublinguagem do SQL para **tratar as transações**. Os principais comandos desta linguagem são o **BEGIN TRANSACTION (ou BEGIN TRAN)**, o **COMMIT** e o **ROLLBACK**.

O **COMMIT** é o comando usado para **salvar permanentemente** qualquer **transação** em um banco de dados.

O **ROLLBACK** é o comando usado para **restaurar o banco de dados** para o último estado confirmado. Também é usado com o comando SAVEPOINT

Gabarito: **Letra A.**

5- **(UFMT - 2017 - UFSBA - Analista de Tecnologia da Informação)** Sobre transações em SQL, considere:

I - São uma sequência de operações num sistema gerenciador de banco de dados, que são tratadas como um bloco único e indivisível (atômico).

II - Os comandos COMMIT, ROLLBACK e END TRANSACTION fazem parte do controle de transações do SQL.

III - O comando COMMIT garante de forma permanente as mudanças ocorridas nos dados durante a transação. O comando ROLLBACK desfaz as mudanças ocorridas nos dados durante a transação. Independente do comando executado (COMMIT ou ROLLBACK), faz-se necessário encerrar a transação por meio do comando END TRANSACTION.

Está correto o que se afirma em

a) I, II e III.

b) I, apenas.

c) III, apenas.

d) II e III, apenas.

Resolução:

Vamos analisar cada um dos comandos:

I – **Correto**: São uma sequência de operações num sistema gerenciador de banco de dados, que são tratadas como um bloco único e indivisível (atômico).

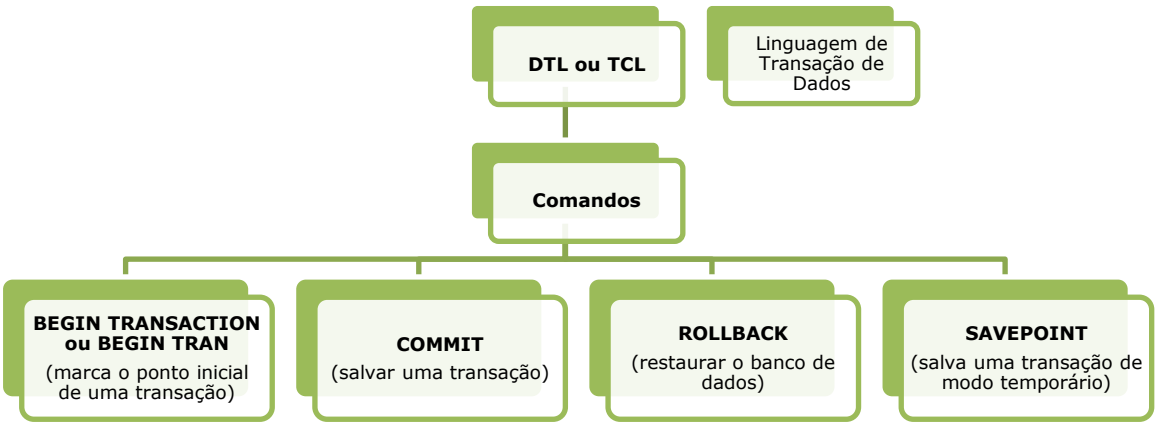
II – **Incorreto**: Os comandos COMMIT, ROLLBACK e ~~END~~ **BEGIN** TRANSACTION fazem parte do controle de transações do SQL.

III - **Incorreto**: O comando COMMIT garante de forma permanente as mudanças ocorridas nos dados durante a transação. O comando ROLLBACK desfaz as mudanças ocorridas nos dados durante a transação. ~~Independente do comando executado (COMMIT ou ROLLBACK), faz-se necessário encerrar a transação por meio do comando END TRANSACTION~~.

Gabarito: **Letra B.**

2. ESQUEMAS DE AULA

DTL



3. REFERÊNCIAS

MICROSFT. **BEGIN TRANSACTION (Transact-SQL)**. Disponível em: <https://docs.microsoft.com/pt-br/sql/t-sql/language-elements/begin-transaction-transact-sql?view=sql-server-ver15> Acesso em: 13 abr. 2020.

STUDYTONIGHT. **Commit, Rollback and Savepoint SQL commands**. Disponível em: <https://www.studytonight.com/dbms/tcl-command.php> Acesso em: 13 abr. 2020.