

Programação em Banco de Dados

Faculdade Anhanguera – Unidade Betim/MG

AULA 03

Aula 03

- Processamento de Consultas
- Linguagem SQL DTL

Introdução



- No SGBD *MySQL* (e em outros SGBDs), um **transação** permite **agrupar** várias **operações** SQL para que sejam **executadas** como uma **unidade** atômica.
- Logo **todas** as **operações** devem ser **confirmadas** juntas (**COMMIT**) ou desfeitas (**ROLLBACK**) em caso de erro.
- Esta **característica** é **possível** devido as **propriedades** ACID presente na Linguagem SQL

SQL DTL

- Linguagem de **Transação de Dados**
- Conjunto de **instruções** para o **controle** de **transações lógicas** que são **agrupadas** e **executadas** pela DML.
 - *START TRANSACTION*
 - *SAVEPOINT*
 - *COMMIT*
 - *ROLLBACK [TO SAVEPOINT]*
 - *RELEASE SAVEPOINT*

Prática 1

1. *CREATE DATABASE* AULA_PRATICA;
2. *USE* AULA_PRATICA;
3. *CREATE TABLE* ESTUDANTE
(*MATRICULA INT AUTO_INCREMENT,*
NOME VARCHAR(50) NOT NULL,
EMAIL TEXT NOT NULL,
CURSO VARCHAR(50) NOT NULL,
PRIMARY KEY(*MATRICULA*));

Prática 1

4. *ALTER TABLE ESTUDANTE AUTO_INCREMENT = 1000;*
5. *INSERT INTO ESTUDANTE (NOME,EMAIL,CURSO) VALUES ('CMN', 'CMN@CMN.BR', 'CIÊNCIA DA COMPUTAÇÃO');*
6. *INSERT INTO ESTUDANTE (NOME,EMAIL,CURSO) VALUES ('ANA', 'ANA@ANA.BR', 'FISIOTERAPIA');*
7. *INSERT INTO ESTUDANTE (NOME,EMAIL,CURSO) VALUES ('BETO', 'BETO@ANA.BR', 'DIREITO');*
8. *SELECT * FROM ESTUDANTE;*

START TRANSACTION / ROLLBACK COMMIT



- Cria / inicia uma **nova transação** SQL no banco de dados.
- Após a **cláusula** *START TRANSACTION*, as **operações** executadas **não** são **salvas** no **banco de dados** até que um **COMMIT** seja **realizado**.
- A **cláusula** *ROLLBACK* **reverte** todas as **operações realizadas** desde o *START TRANSACTION*, ou até um *SAVEPOINT* **específico**.
- A **cláusula** *COMMIT* **finaliza** a **transação** SQL e confirma todas as **alterações** no banco de dados.

Prática 2

```
9. SELECT * FROM ESTUDANTE;  
10. SET AUTOCOMMIT = OFF;  
11. DELETE FROM ESTUDANTE;  
12. SELECT * FROM ESTUDANTE;  
13. ROLLBACK;  
14. DELETE FROM ESTUDANTE;  
15. COMMIT  
16. ROLLBACK;  
17. SELECT * FROM ESTUDANTE;
```


SAVEPOINT

- A **cláusula** *SAVEPOINT* cria um **ponto de restauração** dentro de uma **transação** SQL.
- Caso um **erro** ocorrer, podemos **reverter** apenas **até** o *SAVEPOINT*, sem **cancelar** toda a **transação** SQL.
- Uma **cláusula** *SAVEPOINT* **também** pode ser **excluída** de uma **transação** SQL.

Prática 3

18. *INSERT INTO* ESTUDANTE (NOME,EMAIL,CURSO) *VALUES* ('CMN', 'CMN@CMN.BR', 'CIÊNCIA DA COMPUTAÇÃO');

19. *INSERT INTO* ESTUDANTE (NOME,EMAIL,CURSO) *VALUES* ('ANA', 'ANA@ANA.BR', 'FISIOTERAPIA');

20. *INSERT INTO* ESTUDANTE (NOME,EMAIL,CURSO) *VALUES* ('BETO', 'BETO@ANA.BR', 'DIREITO');

21. *SELECT * FROM* ESTUDANTE;

Prática 3

19. START TRANSACTION;

SAVEPOINT SP0;

INSERT INTO ESTUDANTE (NOME,EMAIL,CURSO)

VALUES ('TESTE','TESTE@TESTE.BR','TESTE');

*20. SELECT * FROM ESTUDANTE;*

21. ROLLBACK TO SAVEPOINT SP0;

Prática 3

22. *SELECT* * *FROM* ESTUDANTE;

23. *START TRANSACTION*;

SAVEPOINT SP1;

UPDATE ESTUDANTE

SET NOME = 'CRISTIANO MARTINS NUNES',

EMAIL = 'CMN@ANHANGUERA.COM.BR',

CURSO = 'DESIGN DE INTERIORES'

WHERE MATRICULA = 1030;

24. *SELECT* * *FROM* ESTUDANTE;

RELEASE SAVEPOINT

- A cláusula *RELEASE SAVEPOINT* remove um SAVEPOINT dentro da transação SQL.
- Após a execução da cláusula não é mais possível fazer ROLLBACK para esse ponto.

Prática 4

25. START TRANSACTION;

SAVEPOINT SP99;

INSERT INTO ESTUDANTE (NOME,EMAIL,CURSO)

VALUES ('Anhanguera','betim@betim.BR','Betim/MG');

*26. SELECT * FROM ESTUDANTE;*

27. RELEASE SAVEPOINT SP99;

28. ROLLBACK TO SAVEPOINT SP99;

Resumo

Comando	Descrição
START TRANSACTION;	Inicia uma transação.
SAVEPOINT nome;	Cria um ponto de restauração.
COMMIT;	Confirma e aplica as mudanças no banco.
ROLLBACK;	Desfaz todas as mudanças da transação.
ROLLBACK TO SAVEPOINT nome;	Desfaz até o SAVEPOINT , sem reverter tudo.
RELEASE SAVEPOINT nome;	Remove um SAVEPOINT .

Prática 5 / Atividade



1. Crie o **SAVEPOINT** SP2 para um sentença SQL que execute um delete com cláusula Where.
2. Crie o **SAVEPOINT** SP3 para um sentença SQL que execute um delete s/cláusula Where.