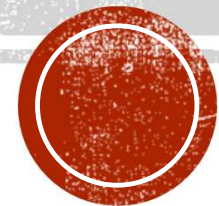


BANCO DE DADOS

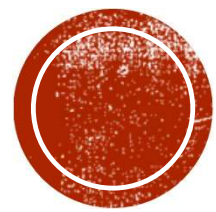


Elaborado pelos professores:

❖ *Leandro Vasconcelos*

❖ *Luciano Xiscatti*

❖ *Marcelo Fassina*



TRANSAÇÕES EM MYSQL



TRANSAÇÕES EM MYSQL

Uma transação em bancos de dados configura um conjunto de declarações SQL que são combinadas em uma única unidade de trabalho, sendo executadas como se fossem uma única operação. Essa abordagem é empregada para evitar diversos problemas que podem ocorrer em um banco de dados, além de aplicar as propriedades ACID.



ACID

- **Atomicidade:** se as operações forem confirmadas, elas acontecerão todas sem exceção. Não ocorre a execução de apenas parte das operações contidas na transação.
- **Consistência:** as alterações no banco de dados somente acontecem caso o novo estado do sistema for válido. Se uma mudança inválida for executada, ela irá falhar, e o sistema permanece no seu estado anterior, válido.
- **Isolamento:** a transação não é visível a outros processos até que ela seja confirmada e persistida no banco.
- **Durabilidade:** após a transação ter sido confirmada, ou seja, que suas operações tenham todas sido executadas com sucesso, as alterações são automaticamente persistidas (gravadas no banco), e não é necessário se preocupar em gravar as mudanças.

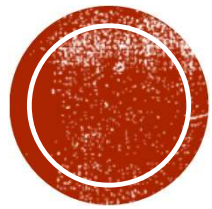


TRANSAÇÕES EM MYSQL

As transações buscam fornecer unidades de trabalho confiáveis e que permitam a recuperação correta dos dados em caso de falhas.

Uma transação deve ser atômica, ou seja, ela deve ser executada de forma completa ou não ter nenhum efeito nas tabelas envolvidas. Ou uma transação ocorre por completo ou não.





COMMIT E ROLLBACK



COMMIT E ROLLBACK

Quando uma transação é efetuada com sucesso (todas as suas operações são bem-sucedidas), o banco de dados é alterado de forma permanente, com os dados envolvidos na transação persistidos (ou seja, salvos em disco). A essa operação damos o nome de **COMMIT**.

Caso haja falha em qualquer uma das operações constituintes da transação, o banco de dados deverá ser retornado ao estado em que se encontrava anteriormente à execução da transação. A essa operação damos o nome de **ROLLBACK**.



CARACTERÍSTICAS DAS TRANSAÇÕES

- Todas as transações possuem início e fim.
- As transações podem ser salvas (COMMIT) ou desfeitas (ROLLBACK).
- Se transação falhar durante a execução de qualquer uma de suas operações constituintes, nenhuma destas operações será salva no banco de dados.



AUTOCOMMIT

Por padrão, o MySQL utiliza o modo AUTOCOMMIT, que significa que declarações INSERT, UPDATE ou DELETE sofrem COMMIT automaticamente quando executadas.

```
SELECT @@autocommit;
```

```
SET autocommit = 1;
```



DECLARAÇÃO `START TRANSACTION`

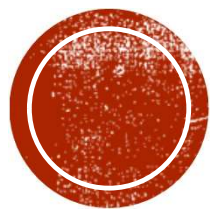
A declaração `START TRANSACTION` identifica o início de uma transação. Essa declaração desabilita temporariamente o modo `AUTO COMMIT`, de modo que você deverá programar o momento de efetivar a transação. O modo `AUTO COMMIT` é reativado logo após o término da transação (ao usar `COMMIT` ou `ROLLBACK`).



APLICAÇÃO DAS TRANSAÇÕES

Use transações sempre que existirem declarações INSERT, UPDATE ou DELETE que afetem dados que são relacionados e sempre que a falha em uma puder violar a integridade dos dados e incorrer em sua perda.





EXEMPLO



/* Verifica se autocommit está ativo ou não (1 para ativo, 0 para inativo. */

SELECT @@autocommit;

/* Força o MySQL a executar o commit automaticamente. */

SET autocommit = 1;

/* Cria o banco de dados "teste", caso o mesmo não exista. */

create database if not exists teste;

/* Seleciona o banco de dados para uso. */

use teste;

/* Cria a tabela de clientes. */

**create table clientes(
 id int not null auto_increment,
 nome varchar(40),
 cidade varchar(30),
 primary key(id)
);**

/* Mostra a estrutura da tabela de clientes. */

describe clientes;

/* Inclui 5 registros na tabela de clientes. */

**insert into clientes (nome, cidade) values
('Ana', 'Curitiba'),
('Beatriz', 'Maringá'),
('Carolina', 'Londrina'),
('Denise', 'Umuarama'),
('Elaine', 'São José dos Pinhais');**

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Exclui o registro de id = 1 da tabela de clientes. Não será possível recuperar o registro. */

delete from clientes where id = 1;

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Inicia uma transação. Esse comando desabilita temporariamente o método "commit". */

start transaction;

/* Exclui o registro de id = 2 da tabela de clientes. */

delete from clientes where id = 2;



/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Recupera o registro de id = 2 da tabela de clientes. */

rollback;

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Inicia uma transação. */

start transaction;

/* Altera a cidade do registro de id = 2 da tabela de clientes. */

update clientes set cidade = 'Campo Largo' where id = 2;

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Restaura o registro de id = 2 da tabela de clientes. */

rollback;

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Inicia uma transação. */

start transaction;

/* Exclui o registro de id = 2 da tabela de clientes. */

delete from clientes where id = 2;

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Confirma a transação. Após a execução do comando não há possibilidade de restauração. */

commit;

/* Mostra os registros da tabela de clientes. */

select * from clientes;

/* Inicia uma transação. */

start transaction;

/* Exclui o registro de id = 3 da tabela de clientes. */

delete from clientes where id = 3;

/* Mostra os registros da tabela de clientes. */

select * from clientes;



```
/* Atribui um ponto de restauração da transação. */  
savepoint ponto_restauracao;
```

```
/* Exclui o registro de id = 4 da tabela de clientes. */  
delete from clientes where id = 4;
```

```
/* Exclui o registro de id = 5 da tabela de clientes. */  
delete from clientes where id = 5;
```

```
/* Mostra os registros da tabela de clientes. */  
select * from clientes;
```

```
/* Desfaz as transações até o ponto de restauração. */  
rollback to ponto_restauracao;
```

```
/* Mostra os registros da tabela de clientes. */  
select * from clientes;
```

```
/* Inicia uma transação. */  
start transaction;
```

```
/* Exclui o registro de id = 4 da tabela de clientes. */  
delete from clientes where id = 4;
```

```
/* Exclui o registro de id = 5 da tabela de clientes. */  
delete from clientes where id = 5;
```

```
/* Mostra os registros da tabela de clientes. */  
select * from clientes;
```

```
/* Executa as transações de forma definitiva. */  
commit;
```

```
/* Mostra os registros da tabela de clientes. */  
select * from clientes;
```

```
/* Exclui a tabela de clientes. */  
drop table clientes;
```

```
/* Exclui o banco de dados. */  
drop database teste;
```



FONTES

Como trabalhar com Transações em MySQL

<http://www.bosontreinamentos.com.br/mysql/como-trabalhar-com-transacoes-em-mysql/>

START TRANSACTION, COMMIT, and ROLLBACK Statements

<https://dev.mysql.com/doc/refman/8.0/en/commit.html>

