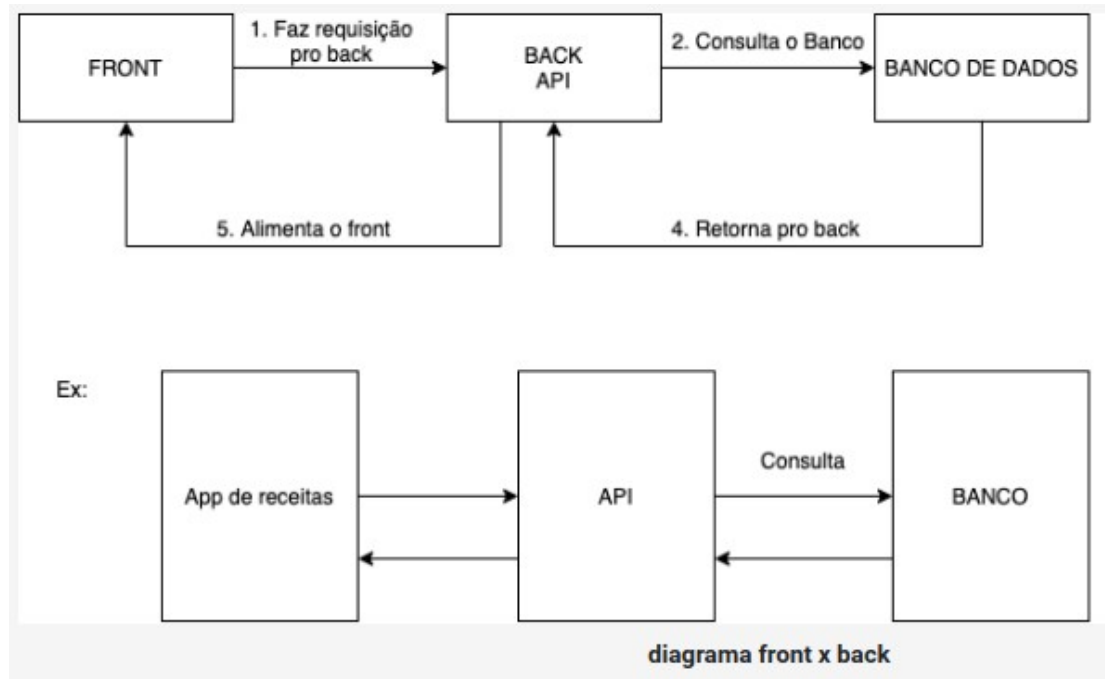


TRYBE

Modulo III – Back-end

Bloco 20 – Introdução

Dados → informações → conhecimentos.



1) Bancos de dado SQL

Banco de dado: o que é, tipos

SGBD: Sistema de gerenciamento de bancos de dados: para organizar, pesquisar, editar, gerar novos dados a partir dos primeiros, limitar acesso.

Dois tipos: relacional e não relacional.

RELACIONAIS E NÃO RELACIONAIS



TABELAS PRE-DEFINIDAS
SEM ALTERAÇÕES DINÂMICAS DA ESTRUTURA
PREVISÍVEL
RELACIONAMENTO ENTRE TABELAS
USA SQL

NoSQL

ESTRUTURA PRE-DEFINIDA NÃO É OBRIGATÓRIO
DADOS PODEM SER INSERIDOS DINÂMICAMENTE
NÃO POSSUI LINGUAGEM OFICIAL

Exemplos relacionais: MySQL, SQLServer, Oracle

O que é SQL (Structured Query Language)

Linguagem usada para criar, pesquisar, extrair e também manipular dados dentro de um banco de dados relacional.

Comandos principais: SELECT, UPDATE, DELETE, INSERT e WHERE.

Tabelas e Constraints

Pesquisas realizadas dentro de um banco de dados são feitas em tabelas.

Exemplos de comandos para manipular tabela: *CREATE, ALTER, USE, DROP*.

Constraints ou restrições: regras para delimitar modo de inserir dados.

Exemplos principais:

- NOT NULL: coluna não aceita valor nulo
- UNIQUE: impede repetir dados na coluna ou conjunto de colunas.
- PRIMARY KEY: Automaticamente unica, uma chave primária por tabela, sem valor nulo na coluna.
- FOREIGN KEY: campo que aponta para chave primária em outra tabela, conecta tabelas entre si.

```
CONSTRAINT fk_ID_Autor FOREIGN KEY (ID_Autor)
REFERENCES tbl_autores(ID_Autor)
```

Constraints(principais)

- * NOT NULL
- * UNIQUE
- * PRIMARY KEY
- * FOREIGN KEY
- * DEFAULT

- DEFAULT: valor padrão caso nenhum valor seja inserido.

TERMINAL: instalar e manipular MySQL Server

sudo apt update

sudo apt install mysql-server

Verificar

sudo systemctl status mysql

Ativar / Parar

systemctl start mysql / systemctl stop mysql

(Sair com ctrl+c)

Determinar quando inicializa

Manualmente: *sudo systemctl disable mysql + sudo service mysql start*

Automaticamente: *sudo systemctl enable mysql*

Desinstalar totalmente

sudo apt-get remove --purge mysql-server mysql-client mysql-common

sudo apt-get autoremove

sudo apt-get autoclean

sudo rm -rf /etc/mysql

Visualizar quais bancos de dados estão disponíveis

sudo mysql

show databases;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.02 sec)
```

Comandos mais comuns

(Úteis na ausência de interface gráfica):

Selecionar banco de dados:

use nome_do_banco_de_dados_que_quer_conectar;

Visualizar tabelas:

show tables;

Visualizar estrutura de tabela:

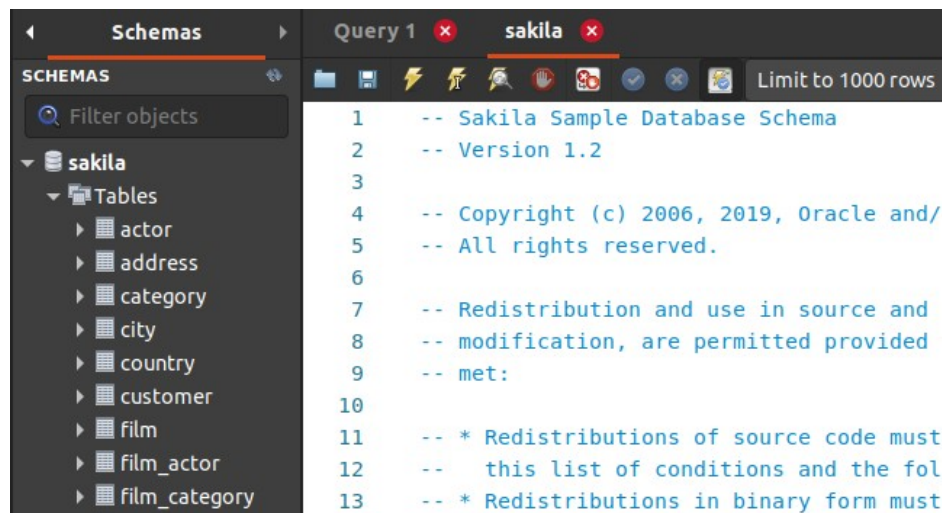
describe nome_da_tabela;

Criar um banco de dados:

create database nome_do_banco_de_dados;

Interface gráfica (MySQL [Workbench](#)) & banco de dados de prática ([sakila](#))

[Instalar.](#)



Primeiros conceitos

Entidades contêm propriedades.

Dados ligados de 4 jeitos:

- Um por Um
- Um para Muitos
- Muitos para Um
- Muitos para Muitos

Regra de negócio: determina a lógica das tabelas. Quando muda, refatora.

Primeiras manipulações

- VARCHAR(45): string de até 45 caracteres

```
42 CREATE TABLE actor (  
43     actor_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
44     first_name VARCHAR(45) NOT NULL,  
45     last_name VARCHAR(45) NOT NULL,  
46     last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
47     PRIMARY KEY (actor_id),  
48     KEY idx_actor_last_name (last_name)  
49 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

2) Encontrando dados em um banco de dados

Queries

Nome dado aos comandos que interagem com uma base de dados.

Tipos de queries: DDL, DML, DCL, TCL

DDL: Data Definition Language - todos os comandos que lidam com o esquema, a descrição e como os dados devem existir em um banco de dados:

CREATE - Para criação de bancos de dados, tabelas, índices, views, procedures, functions e triggers

ALTER - Para alteração da estrutura de qualquer objeto

DROP - Permite deletar objetos

TRUNCATE - Apenas esvazia os dados dentro de uma tabela, mas a mantém no banco de dados

DML: Data Manipulation Language - os comandos que são usados para manipular dados. São utilizados para armazenar, modificar, buscar e excluir dados em um banco de dados:

SELECT: Usado para buscar dados em um banco de dados

INSERT: Insere dados em uma tabela

UPDATE: Altera dados dentro de uma tabela

DELETE: Exclui dados de uma tabela

DCL: Data Control Language - Focado mais nos comandos que concedem direitos, permissões e outros tipos de controle ao sistema de banco de dados:

GRANT - Concede acesso a um usuário

REVOKE - Remove acessos concedidos através do comando GRANT

TCL: Transactional Control Language - Lida com as transações dentro de suas pesquisas:

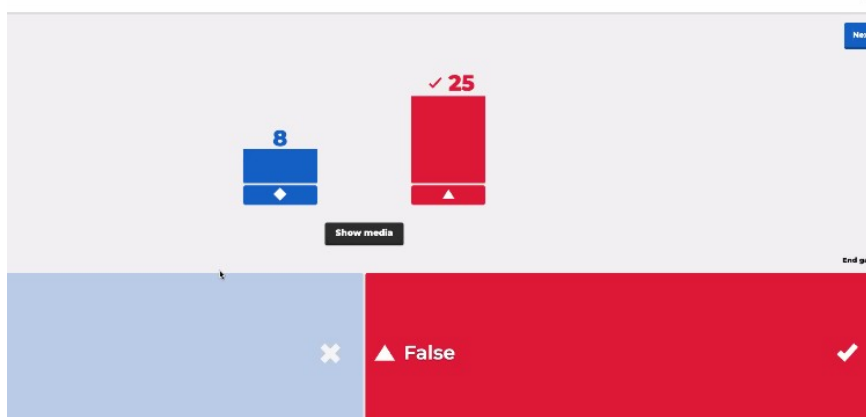
COMMIT - Muda suas alterações de temporárias para permanentes no seu banco de dados

ROLLBACK - Desfaz todo o impacto realizado por um comando

SAVEPOINT - Define pontos para os quais uma transação pode voltar. É uma maneira de voltar para pontos específicos de sua query

TRANSACTION - Comandos que definem onde, como e em que escopo suas transações são executadas

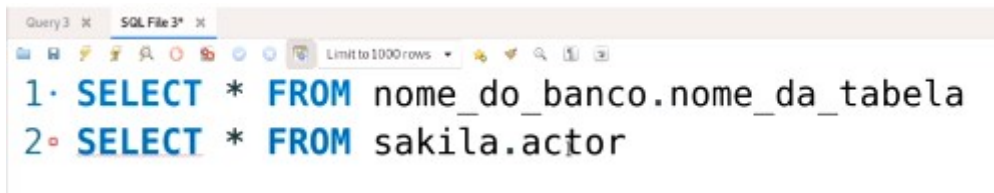
Há um limite de **comandos SQL** para serem usados em uma **query**



Encontrar dados com diversos comandos comuns

(Rodar sempre com ícone de raio ou com ctrl + enter)

SELECT



SELECT columnName1, columnName2 FROM bankName.tableName;

Ou pode para não repetir banco sempre:

use bankName;

*SELECT * FROM tableName;*

WHERE para consultar especificamente:

```
SELECT * FROM actor WHERE first_name='PENELOPE';
```

CONCAT

Para concatenar colunas em apenas uma.

SELECT CONCAT(first_name, ' ', last_name) AS 'Nome Completo' FROM sakila.actor;

DISTINCT

Remover dados duplicados da tabela.

SELECT DISTINCT etc

COUNT

Remover dados duplicados da tabela.

SELECT COUNT(column-to-count) FROM bank/table;

COUNT(*) para contar todas as linhas.

```
SELECT district, count(district) FROM sakila.address  
WHERE district = 'Alberta';
```

LIMIT

Limitar o nº de resultados.

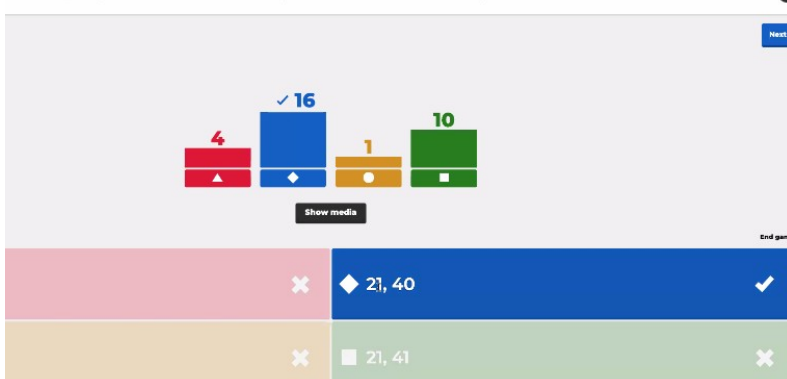
*SELECT * FROM sakila.rental LIMIT 10;*

LIMIT OFFSET

Pular linhas desnecessárias.

*SELECT * FROM sakila.rental LIMIT 10 OFFSET 3;*

A query abaixo retorna o registro de ID ____ até o registro de ID ____



// LIMIT 20 OFFSET 20

ORDER BY

```
1. SELECT * FROM sakila.address
2 ORDER BY address;
```

Ordem crescente é padrão: ASC.

ORDER BY address DESC; dà ordem decrescente.

Diferentes syntax SQL de bancos relacionais

PostgreSQL	MySQL
<i>datefield</i> + 6	DATE_ADD(<i>datefield</i> , INTERVAL 6 DAY)
<i>datefield</i> + INTERVAL '6 months'	DATE_ADD(<i>datefield</i> , INTERVAL 6 MONTH)
<i>datefield</i> - INTERVAL '6 months'	DATE_SUB(<i>datefield</i> , INTERVAL 6 MONTH)

PostgreSQL	MySQL
DATE_PART('year', <i>datefield</i>)	YEAR(<i>datefield</i>)
DATE_PART('month', <i>datefield</i>)	MONTH(<i>datefield</i>)
DATE_PART('day', <i>datefield</i>)	DAY(<i>datefield</i>)

Factors	MS SQL Server	MySQL
Length function	SELECT LEN(data_string) FROM TableName	SELECT CHARACTER_LENGTH(data_string) FROM TableName
Concatenation function	SELECT ('SQL' + 'SERVER')	SELECT CONCAT ('My', 'SQL')
Select top n records from a table	SELECT TOP 10 * FROM TableName WHERE id = 2	SELECT * FROM TableName WHERE id = 2 LIMIT 10
Generate GUID (Global Unique Identifier)	SELECT NEWID()	SELECT UUID()
Get current date and time	SELECT GETDATE()	SELECT NOW()

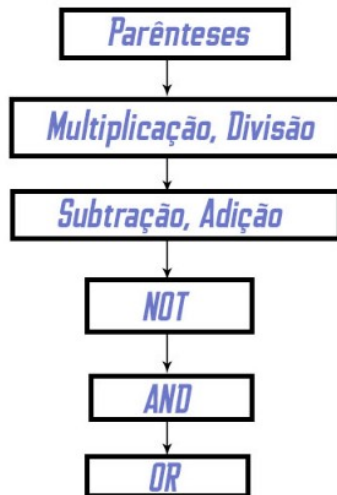
3) Filtrando dados de forma específica

WHERE

SELECT something FROM somewhere WHERE condition;

Prioridade

(Na dúvida, usar parênteses).



Operadores

```
-- OPERADOR - DESCRIÇÃO
=   IGUAL
>   MAIOR QUE
<   MENOR QUE
>=  MAIOR QUE OU IGUAL
<=  MENOR QUE OU IGUAL
<>  DIFERENTE DE
AND  OPERADOR LÓGICO E
OR   OPERADOR LÓGICO OU
NOT  NEGAÇÃO
IS   COMPARA COM VALORES BOOLEANOS (TRUE, FALSE, NULL)
```

Ordem dos operadores

LIKE

Quando quiser pesquisar algo tendo informação incompleta.

*SELECT * FROM sakila.film
WHERE title LIKE '%don';*

Uso de _ e %:

```
-- Encontra qualquer resultado finalizando com "don"
WHERE title LIKE '%don';

-- Encontra qualquer resultado iniciando com "plu"
WHERE title LIKE 'plu%';

-- Encontra qualquer resultado que contém "plu"
WHERE title LIKE '%plu%';

-- Encontra qualquer resultado que inicia com "p" e finaliza com "r"
WHERE title LIKE 'p%r';

-- Encontra qualquer resultado em que o segundo caractere da frase é "C"
WHERE title LIKE '_C%';

-- Encontra qualquer resultado em que o título possui exatamente 8 caracteres
WHERE title LIKE '____';

-- Encontra todas as palavras com no mínimo 3 caracteres e que iniciam com E
WHERE title LIKE 'E__';
```

Faixa de resultados: IN e BETWEEN

No caso do IN, você precisa especificar os valores que devem ser incluídos no resultado e, no caso do BETWEEN, você não precisa incluir os valores que estão entre o valor inicial e final.
Para escolher: tempo do Execution Plan.

IN

*SELECT * FROM banco_de_dados
WHERE coluna IN (expressão);*

```
SELECT * FROM sakila.actor  
WHERE first_name = 'PENELOPE'  
OR first_name = 'NICK'  
OR first_name = 'ED'  
OR first_name = 'JENNIFER';
```

=

```
SELECT * FROM sakila.actor  
WHERE first_name IN ('PENELOPE', 'NICK', 'ED', 'JENNIFER');
```

Pode ter Select dentro:

A query abaixo é válida ?

✓ 21

7

```
SELECT * FROM toot  
WHERE something IN (SELECT xablau FROM doot);
```

True ✓ False

13

Show media

True ✓ False

o resultado da busca inclui o valor inicial e final.

BETWEEN

*SELECT name, area FROM world
WHERE area BETWEEN 200000 AND 250000*

Pode usar também com strings e com datas (formato padrão **YYYY-MM-DD HH:MM:SS**).

Procurar por datas

- **DATE** - Possui apenas data, no formato **YYYY-MM-DD** na faixa de **1001-01-01** até **9999-12-31**
- **DATETIME** - Possui data e tempo, no formato **YYYY-MM-DD HH:MM:SS** com a faixa de **1000-01-01 00:00:00** até **9999-12-31 23:59:59**.

```
SELECT DATE(payment_date) FROM sakila.payment; -- YYYY-MM-DD  
SELECT YEAR(payment_date) FROM sakila.payment; -- Ano  
SELECT MONTH(payment_date) FROM sakila.payment; -- Mês  
SELECT DAY(payment_date) FROM sakila.payment; -- Dia  
SELECT HOUR(payment_date) FROM sakila.payment; -- Hora  
SELECT MINUTE(payment_date) FROM sakila.payment; -- Minuto  
SELECT SECOND(payment_date) FROM sakila.payment; -- Segundo
```


Exemplo:

```
SELECT * FROM rental
WHERE date(rental_date)='2005-05-26'
```

Dicas diversas

ROUND

Para decidir das decimais.

```
ROUND(7253.86, 0)    -> 7254
ROUND(7253.86, 1)    -> 7253.9
ROUND(7253.86, -3)    -> 7000
```

LENGTH

Para pegar o nº de caracteres de uma string.

```
LENGTH('Hello') -> 5
```

LEFT(s,n) para extrair da string s n caracteres: `LEFT('Hello world', 4) -> 'Hell'`

4) Manipulando tabelas

CRUD

Create Read Update Delete deve ser presente em todo software.

```
-- Create => INSERT algum_dado ON alguma_tabela VALUES (nosso_dados)
-- Read => SELECT algum_dado FROM alguma_tabela
-- Update => UPDATE algum_dado FROM alguma_tabela SET novo_dado
-- Delete => DELETE algum_dado FROM alguma_tabela
```

INSERT

Aspas: crase para palavras reservadas, classicas para string.

Para inserir novos valores numa tabela

```
INSERT INTO nome_da_tabela (coluna1, coluna2)
VALUES ('valor_coluna1', 'valor_coluna2');
```

Várias linhas de uma vez

```
INSERT INTO nome_da_tabela (coluna1, coluna2) VALUES
('valor_1','valor_2'),
('valor_3','valor_4'),
('valor_5','valor_6');
```

Pulando para evitar erros silenciosos (duplicar,...)

```
INSERT IGNORE INTO pessoas (id, name) VALUES
(4,'Gloria'), -- Sem o IGNORE, essa linha geraria um erro e o INSERT não continuaria.
```

(5,'Amanda');

Auto_increment: não precisa colocar no INSERT esse valor que vai ser criado automaticamente

Inserir dados de outra tabela: **INSERT SELECT**

```
INSERT INTO destination(columns)
  SELECT columns FROM source;
```

```
INSERT INTO sakila.actor (first_name, last_name)
  SELECT first_name, last_name FROM sakila.staff;
```

UPDATE

Para consertar erro

```
UPDATE nome_da_tabela
```

```
SET propriedade_a_ser_alterada = 'novo valor para coluna'
```

```
WHERE alguma_condicao; -- importantíssimo aplicar o WHERE para não alterar a tabela inteira!
```

Desativar safe mode (de não poder update ou delete sem id):

```
SET SQL_SAFE_UPDATES = 0;
```

Mais de uma coluna ao mesmo tempo :

```
UPDATE sakila.staff
```

```
SET first_name = 'Rannveig', last_name = 'Jordan'
```

```
WHERE staff_id = 4;
```

Alterar em massa usando WHERE ou CASE

```
-- Opção 1 - Incluindo a lista de condições fixas
UPDATE sakila.actor
SET first_name = 'JOE'
WHERE actor_id IN (1,2,3);

-- Opção 2 - Especificando como cada entrada será alterada individualmente
UPDATE sakila.actor
SET first_name = (
CASE actor_id WHEN 1 THEN 'JOE' -- se actor_id = 1, alterar first_name para 'JOE'
              WHEN 2 THEN 'DAVIS' -- se actor_id = 2, alterar first_name para 'DAVIS'
              WHEN 3 THEN 'CAROLINE' -- se actor_id = 3, alterar first_name para 'CAROLINE'
END);
```

Update Sequencial

```
UPDATE nome_da_tabela
```

```
SET coluna1 = valor1, coluna2 = valor2
```

```
[WHERE condições]
```

```
[ORDER BY expressao [ ASC | DESC ]] – resultados alterados nesta ordem encontrada
```

```
[LIMIT quantidade_resultados]; – limitar nº de update
```

DELETE

Excluir dados de uma tabela

```
DELETE FROM banco_de_dados.tabela
```

```
WHERE coluna = 'valor'
```

-- O WHERE é opcional. Porém, sem ele, todas as linhas da tabela seriam excluídas.

Desativar safe mode (de não poder update ou delete sem id):

```
SET SQL_SAFE_UPDATES = 0;
```

Exemplos de reações para rejeições/restrições do Delete

```
-- Rejeita o comando DELETE.  
ON DELETE NO ACTION  
  
-- Rejeita o comando DELETE.  
ON DELETE RESTRICT  
  
-- Permite a exclusão dos registros da tabela pai, e seta para NULL os registros da tabela filho.  
ON DELETE SET NULL  
  
-- Exclui a informação da tabela pai e registros relacionados.  
ON DELETE CASCADE
```

//Para conseguir deletar, deletar primeiro o id, no caso de chave estrangeira limitando o delete.

Você está tentando excluir uma empresa que está sendo usada no registro de um usuário. Qual comando vai impedir isso ?

▲ NO ACTION	✓	◆ RESTRICT e NO ACTION	✓
● SETNULL	✗	■ CASCADE	✗

// No action e Restrict e No Action dà a mesma coisa

Truncate para deletar os dados da tabela inteira rapidamente:

```
TRUNCATE banco_de_dados.tabela;
```

Dicas diversas

[CheatSheet](#) de comandos diversos SQL.