

Danilo

Quem sou eu ?





Olá pessoal, eu sou Danilo Aparecido.

Atuo na área de tecnologia por volta de 20 anos

Trabalhei profissionalmente com as staks

1. Delphi
2. ASP
3. PHP
4. C#
5. Java
6. Ruby

Antes de trabalhar com tecnologia, fui operador de máquinas e operador de empilhadeira em uma indústria de fabricação de velas

Após entrar na área de tecnologia em busca de mais renda encarei uma dupla jornada, trabalhando em dois lugares das 6 às 23h da noite.

Após alguns anos fundei um negócio de consultoria, criei um canal no youtube e virei sócio de uma startup no ramo de organização de corridas virtuais



O que é um banco de dados

Bancos de dados - são conjuntos de arquivos relacionados entre si com registros sobre pessoas, lugares ou coisas.



Alguns bancos de dados

Relacionais

- SQL Server
- MySQL
- PostgreSQL
- Oracle
- MariaDB

Não relacionais

- MongoDB
- CouchDB

Configuração



Download e configuração do MySQL



Configurações

1. Fazer o download do mysql server no link:
<https://www.mysql.com/downloads/>
2. Fazer download do mysql Workbench e instalar
<https://dev.mysql.com/downloads/workbench/>



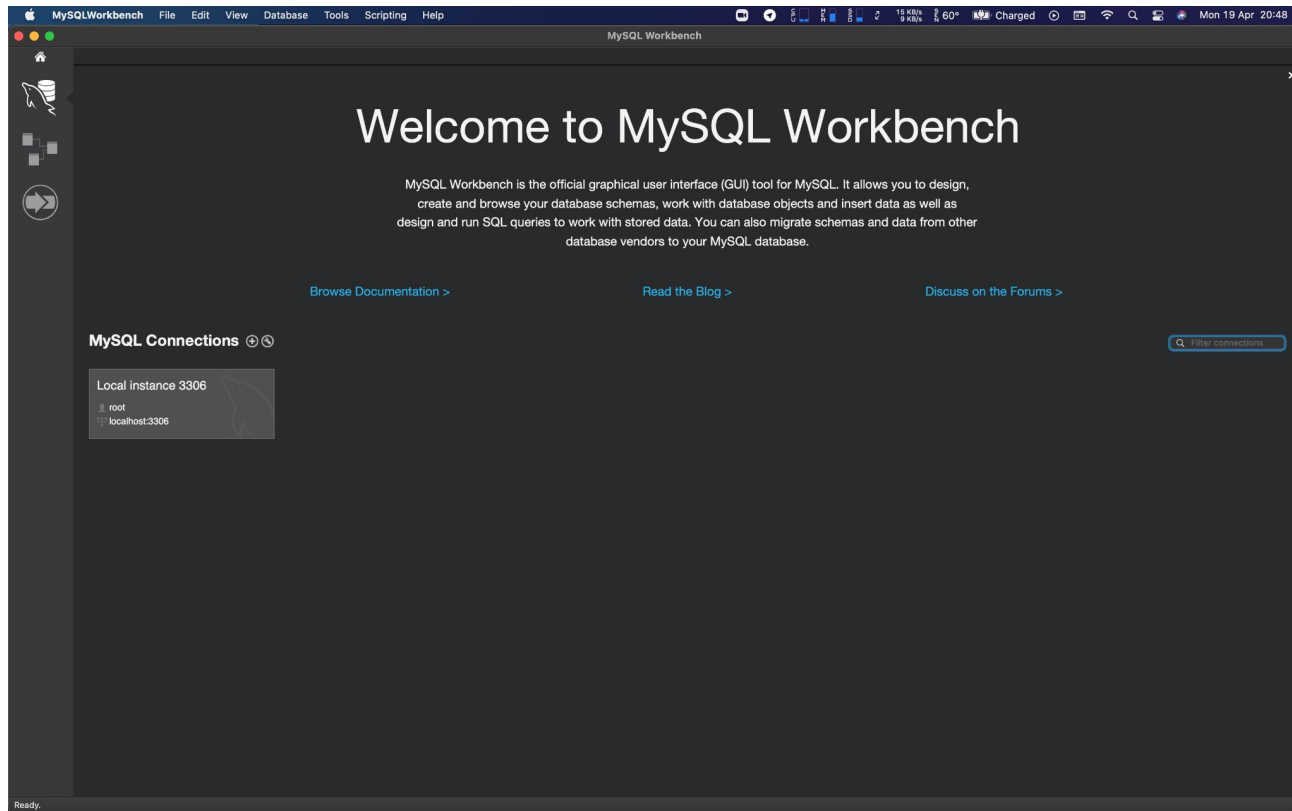
Conhecendo o Ambiente

O que é SGBD ?

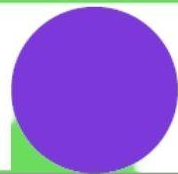
Como utilizar o MySQL
Workbench



Conhecendo o MySQL Workbench



Fazendo conexão

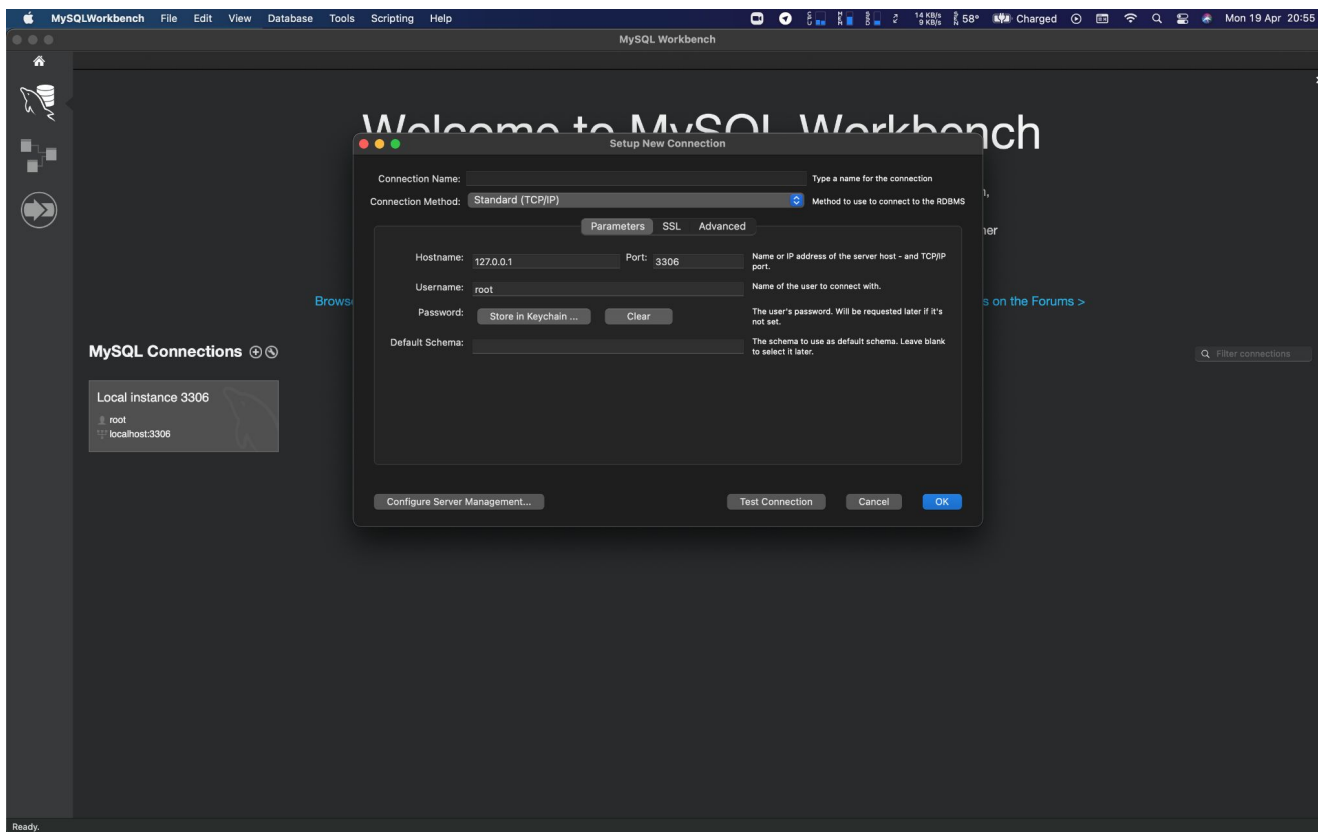


Com banco



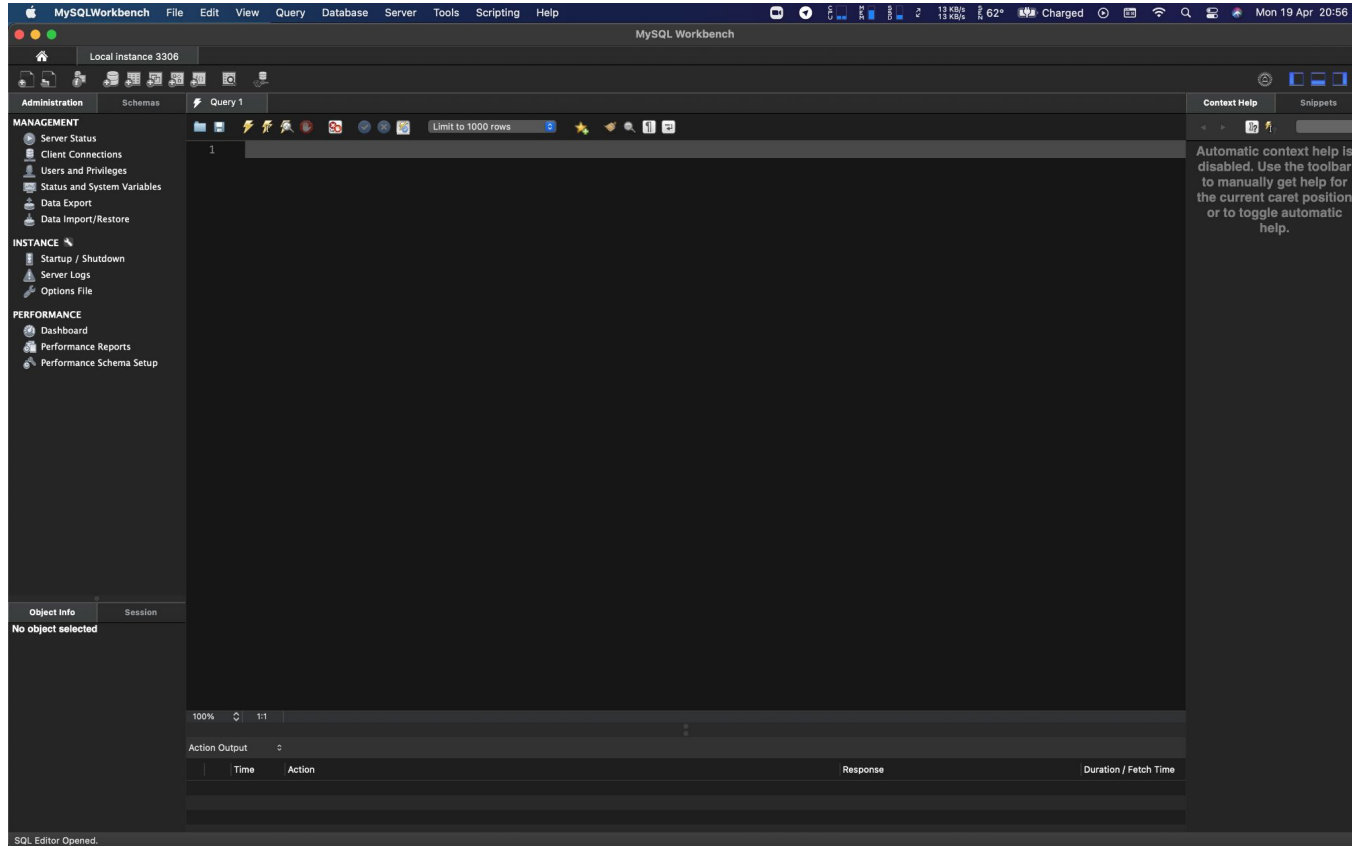


Conectando com database





Ambiente do banco de dados





Adicionando um novo banco de dados

The screenshot shows the MySQL Workbench interface with the Schema Editor open. The left sidebar contains a navigation tree with categories: MANAGEMENT (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), INSTANCE (Startup / Shutdown, Server Logs, Options File), and PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup). The main area is titled 'Schema Editor' and contains the following fields:

- Schema Name:** A text input field containing 'new_schema' and a 'Rename References' button.
- Character Set:** A dropdown menu set to 'Default Charset'.
- Collation:** A dropdown menu set to 'Default Collation'.

Below the main editor, there is a 'Schema' section with an 'Apply' button and a 'Revert' button. At the bottom, there is an 'Action Output' table with columns: Time, Action, Response, and Duration / Fetch Time. The table is currently empty.

On the right side of the interface, there is a 'Context Help' panel with the text: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

At the bottom left, a status bar indicates 'SQL Editor Opened.'



Script de criação de banco de dados

MySQL Workbench

Local Instance 3306

Administration Schemas Query 1 novo_banco_de_dados

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Object Info Session

No object selected

Schema Editor

Schema novo_banco_de_dados

Character Set

Collation

Apply SQL Script to Database

Review the SQL Script to be Applied on the Database

Please review the following SQL script that will be applied to the database.
Note that once applied, these statements may not be revertible without losing some of the data.
You can also manually change the SQL statements before execution.

Online DDL

Algorithm: Default Lock Type: Default

```
1 CREATE SCHEMA `novo_banco_de_dados` ;
2
```

100% 1/1

Go Back Apply

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Schema

Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
1	21:09:37	Apply changes to novo_banco_de_dados	Applying object changes ...	

SQL Editor Opened.



Schemas criados

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main window is titled 'MySQL Workbench' and shows the 'Local Instance 3306' connection. The 'Schemas' tab is active, displaying a list of databases: novo_banco_de_dados, stats_development, stats_test, and sys. The 'Query' tab is also visible, showing a query editor with a 'Limit to 1000 rows' option. The 'Object Info' tab at the bottom left shows 'No object selected'. The 'Context Help' tab on the right displays a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.' The bottom status bar indicates 'Added new scratch query editor'.

MySQL Workbench

Local Instance 3306

Administration Schemas Query 3

SCHEMAS

Filter objects

- novo_banco_de_dados
- stats_development
- stats_test
- sys

Object Info Session

No object selected

100% 1:1

Action Output

Time	Action	Response	Duration / Fetch Time
1 21:09:37	Apply changes to novo_banco_de_dados	Changes applied	

Added new scratch query editor

Context Help

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Criando tabelas

Modo Visual



Criando tabela

The screenshot displays the MySQL Workbench interface on a Mac. The 'Schemas' pane on the left shows the 'novos_banco_de_dados' schema selected, with a context menu open over the 'Tables' folder. The main workspace shows the 'new_table - Table' editor for the 'novos_banco_de_dados' schema. The 'Column details' tab is active, showing fields for 'Column Name', 'Datatype', 'CharSet/Collation', and 'Storage'. The 'Storage' section includes options for 'VIRTUAL' or 'STORED', 'Primary Key', 'Not NULL', 'Unique', 'Binary', 'Unsigned', 'ZeroFill', 'Auto Increment', and 'Generated'. The 'Action Output' pane at the bottom shows a successful execution of the 'Apply changes to novos_banco_de_dados' command.

MySQL Workbench

Local instance 3306

Administration Schemas Query 3

SCHEMAS

Filter objects

novos_banco_de_dados

Tables

Views

Stored Procedures

Functions

stats_development

stats_test

sys

Create Table...

Create Table Like...

Search Table Data...

Table Data Import Wizard

Refresh All

Object Info Session

Schema: novos_banco_de_dados

100% 1:1

Action Output

Time Action

1 21:09:37 Apply changes to novos_banco_de_dados

Added new scratch query editor

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

MySQL Workbench

Administration Schemas Query 3 new_table - Table

Name: new_table Schema: novos_banco_de_dados

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Column Datatype PK NN UQ BIN UN ZF AI O Default / Expression

<click to edit>

Column details

Column Name: Datatype: Default

CharSet/Collation: Default CharSet Default Collation

Comments:

Storage: ☒ VIRTUAL ☐ STORED

☐ Primary Key ☐ Not NULL ☐ Unique

☐ Binary ☐ Unsigned ☐ ZeroFill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Apply Revert

Action Output

Time Action Response Duration / Fetch Time

1 21:09:37 Apply changes to novos_banco_de_dados Changes applied

Added new scratch query editor



Adicionando campos a tabela

The screenshot shows the MySQL Workbench interface. The 'Schemas' pane on the left shows the 'novo_banco_de_dados' database. The 'Query 3' tab is active, showing the 'tabelas1' table. The 'Column' tab is selected, and a new column 'idtabelas1' is being added. The 'Datatype' dropdown is open, showing options: INT, VARCHAR(), DECIMAL(), DATETIME, and BLOB. The 'Column details' pane at the bottom shows the column name 'idtabelas1', character set 'Default Charset', and collation 'Default Collation'. The 'Storage' section is set to 'VIRTUAL'. The 'Primary Key' checkbox is checked. The 'Not NULL' checkbox is checked. The 'Unique' checkbox is checked. The 'ZeroFill' checkbox is checked. The 'Generated' checkbox is checked. The 'Apply' button is visible. The 'Action Output' pane at the bottom shows the action 'Apply changes to novo_banco_de_dados' with a response of 'Changes applied'.

MySQL Workbench

Local instance 3306

Schema: novo_banco_de_dados

Table: tabelas1

Column: idtabelas1

Datatype: INT

Column details 'idtabelas1'

Column Name: idtabelas1

Charset/Collation: Default Charset / Default Collation

Comments:

Datatype: Default

Storage: VIRTUAL

Primary Key

Not NULL

Unique

ZeroFill

Generated

Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
1	21:09:37	Apply changes to novo_banco_de_dados	Changes applied	

Added new scratch query editor



Script de criação de tabela

MySQL Workbench

Local Instance 3306

Administration Schemas Query 3 tabel1 - ...

SCHEMAS

Filter objects

- novο_banco_de_dados
 - Tables
 - Views
 - Stored Procedures
 - Functions
 - stats_development
 - stats_test
 - sys

Column Datatype

- id INT
- Nome VARCHAR
- Telefone VARCHAR

<click to edit>

Column details 'Nome'

Column Name: Nome

Charset/Collation: Default

Comments:

Object Info Session

Schema: novo_banco_de_dados

Apply SQL Script to Database

Review the SQL Script to be Applied on the Database

Please review the following SQL script that will be applied to the database.
Note that once applied, these statements may not be revertible without losing some of the data.
You can also manually change the SQL statements before execution.

Online DDL

Align: Default Lock Type: Default

```
1 CREATE TABLE `novo_banco_de_dados`.`tabela1` (  
2   `id` INT NOT NULL AUTO_INCREMENT,  
3   `Nome` VARCHAR(45) NOT NULL,  
4   `Telefone` VARCHAR(12) NOT NULL,  
5   PRIMARY KEY (`id`),  
6   UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE);  
7
```

100% 1:1

Go Back Apply

STORED

Not NULL Unique

Unsigned ZeroFill

Generated

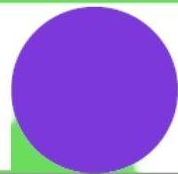
Columns Indexes Foreign Keys Triggers Partitioning Options Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
1	21:09:37	Apply changes to novo_banco_de_dados	Changes applied	
2	21:24:50	Apply changes to tabela1	Applying object changes ...	

Added new scratch query editor

Entidades e



Atributos





Entidade

Uma **entidade** (entity) é um objeto que existe e é distinguível dos outros objetos. Por exemplo, Paulo Silva com número de CPF 123.456.789-00 é uma entidade, visto que isso identifica unicamente uma pessoa particular do universo. Assim a conta número 40167-9 na agência Lapa é uma entidade que identifica unicamente uma conta corrente particular. Uma entidade pode ser **concreta**, como uma pessoa ou um livro, ou pode ser **abstrata**, como um feriado ou um conceito.

Um **conjunto de entidades** (*entity set*) é um conjunto de entidades do mesmo tipo. O conjunto de todas as pessoas com conta em um banco, por exemplo, pode ser definido como o conjunto de todas as entidades *cliente*. Similarmente, o conjunto de entidades *conta* pode representar o conjunto de todas as contas de um banco particular. É convenção adotar nomes de conjuntos de entidades no singular, mas não é obrigatório.

Conjuntos de entidades não precisam ser disjuntos. Por exemplo, é possível definir o conjunto de entidades de todos os funcionários de um banco (*funcionários*) e o conjunto de todos os clientes do banco (*clientes*). Uma entidade *pessoa* pode ser uma entidade *funcionário*, uma entidade *cliente*, ambas ou nenhuma delas.

Referência: Professor André Rodrigo Sanches - USP



Atributos

Uma entidade é representada por um conjunto de **atributos**. Possíveis atributos do conjunto de entidades cliente podem ser *nome-cliente*, *cpf*, *rua* e *cidade-cliente*. Possíveis atributos do conjunto de entidade conta são *número-conta* e *saldo*. Para cada atributo, existe um conjunto de valores permitidos chamado **domínio** daquele atributo. O domínio do atributo *nome-cliente* pode ser o conjunto de todas as cadeias de texto (*strings*) de um certo tamanho. Assim, o domínio do atributo *número-conta* pode ser o conjunto de todos os inteiros positivos. O atributo *idade* de uma entidade *pessoa* poderia ter como domínio os números inteiros entre 0 e 150.

Formalmente, um atributo é uma função que mapeia um conjunto de entidades em um domínio. Portanto, toda entidade é descrita por um conjunto de pares (atributo, valor do atributo), sendo um par para cada atributo do conjunto de entidades. Uma entidade *cliente* é descrita pelo conjunto $\{(nome, \text{Paulo Silva}), (cpf, 123.456.789-00), (rua, \text{XV de Novembro}), (cidade, \text{Campinas})\}$, que significa que a entidade descreve uma pessoa chamada Paulo Silva, cujo CPF é 123.456.789-00, residente à rua XV de Novembro, em Campinas.

Referência: Professor André Rodrigo Sanches - USP

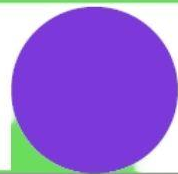


Nomenclatura

Uma boa prática para nomenclatura de uma entidades ou atributos é o uso do padrão camelCase ou PascalCase ou underscore.

Evitando caracteres como , . * & ^ e outros

Tipo de



Dados





Tipo de dados

- CHAR(size) - Campo com com espaços fixos
- VARCHAR(size) - Campo com comprimento variável o tamanho pode ser de 0 a 65535
- TINYTEXT - Contém uma string com comprimento máximo de 255 caracteres
- TEXT(size) - Contém uma string com comprimento máximo de 65.535 bytes
- BLOB (size) - Para BLOBs (objetos binários grandes). Armazena até 65.535 bytes de dados
- LONGTEXT - Armazena uma string com comprimento máximo de 4.294.967.295 caracteres
- BIT (size) - Um tipo de valor de bit. O número de bits por valor é especificado em tamanho. O parâmetro de tamanho pode conter um valor de 1 a 64. O valor padrão para tamanho é 1.
- TINYINT (size) - Um inteiro muito pequeno. O intervalo com sinal é de -128 a 127. O intervalo sem sinal é de 0 a 255. O parâmetro de tamanho especifica a largura máxima de exibição (que é 255)



Tipo de dados

- **BOOL** - Zero é considerado falso, valores diferentes de zero são considerados verdadeiros.
- **SMALLINT (size)** - Um número inteiro pequeno. O intervalo com sinal é de -32768 a 32767. O intervalo sem sinal é de 0 a 65535. O parâmetro de tamanho especifica a largura máxima da tela (que é 255)
- **INT (size)** - Um inteiro médio. O intervalo com sinal é de -2147483648 a 2147483647. O intervalo sem sinal é de 0 a 4294967295. O parâmetro de tamanho especifica a largura máxima da tela (que é 255)
- **BIGINT (size)** - Um inteiro grande. O intervalo com sinal é de -9223372036854775808 a 9223372036854775807. O intervalo sem sinal é de 0 a 18446744073709551615. O parâmetro de tamanho especifica a largura máxima da tela (que é 255)
- **FLOAT (size, d)** - Um número de ponto flutuante. O número total de dígitos é especificado em tamanho. O número de dígitos após o ponto decimal é especificado no parâmetro d. Esta sintaxe está obsoleta no MySQL 8.0.17 e será removida em versões futuras do MySQL



Tipo de dados

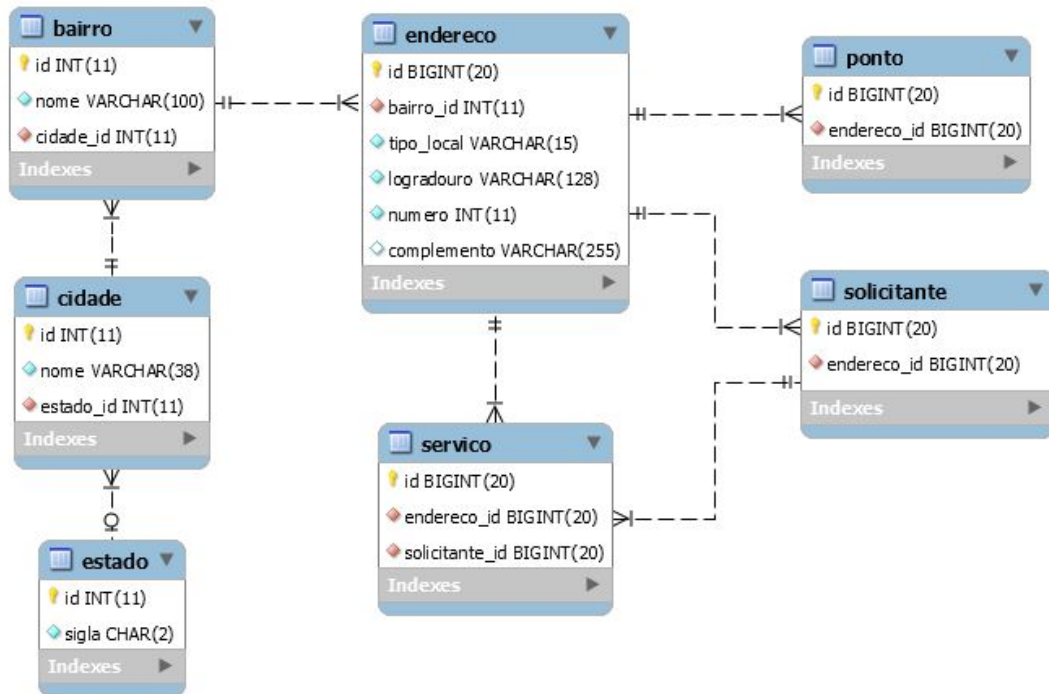
- **DOUBLE (size, d)** - Um número de ponto flutuante de tamanho normal. O número total de dígitos é especificado em tamanho. O número de dígitos após o ponto decimal é especificado no parâmetro d
- **DECIMAL (size, d)** - Um número de ponto fixo exato. O número total de dígitos é especificado em tamanho. O número de dígitos após o ponto decimal é especificado no parâmetro d. O número máximo para tamanho é 65. O número máximo para d é 30. O valor padrão para tamanho é 10. O valor padrão para d é 0.
- **DATA** - Uma data. Formato: AAAA-MM-DD. O intervalo compatível é de '1000-01-01' a '9999-12-31'
- **DATETIME (fsp)** - Uma combinação de data e hora. Formato: AAAA-MM-DD hh: mm: ss. O intervalo compatível é de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'. Adicionando **DEFAULT** e **ON UPDATE** na definição da coluna para obter inicialização automática e atualização para a data e hora atuais

Fazendo uma

Modelagem

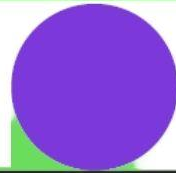


Modelagem



Normalização

De dados





Normalização

Normalização de banco de dados é um conjunto de regras que visa, principalmente, a organização de um projeto de **banco de dados** para reduzir a redundância de dados, aumentar a **integridade de dados** e o desempenho. Para normalizar o banco de dados, deve-se examinar as colunas (atributos) de uma entidade e as relações entre entidades (**tabelas**), com o objetivo de se evitar anomalias observadas na inclusão, exclusão e alteração de registros.

Referência: wikipedia

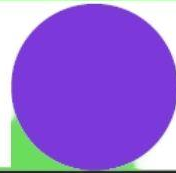


Integridade nos dados

Um banco de dados bem formatado precisa ser analisado entidade por entidade, para evitar duplicação de dados ou amarrações em chaves incoerentes

Uma normalização bem feita precisa evitar duplicação de dados e ter integridade

Chaves



Primárias





Chaves primárias

Chaves primárias (em inglês, *Primary keys* ou "PK"), sob o ponto de vista de um [banco de dados relacional](#), referem-se aos conjuntos de um ou mais campos, cujos valores, considerando a combinação de valores em caso de mais de uma chave primária, nunca se repetem na mesma tabela e, desta forma, podem ser usadas como um índice de referência para criar relacionamentos com as demais tabela do banco de dados (daí vem o nome [banco de dados relacional](#)). Portanto, uma chave primária nunca pode ter valor nulo, nem repetição.

Referência: wikipedia

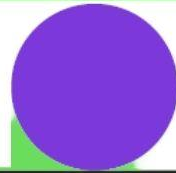


Chaves estrangeiras

Uma **chave estrangeira** é um campo, que aponta para a chave primária de outra tabela ou da mesma tabela. Ou seja, passa a existir uma relação entre duplas de duas tabelas ou de uma única tabela. A finalidade da chave estrangeira é garantir a integridade dos dados referenciais, pois apenas serão permitidos valores que supostamente vão aparecer na base de dados.

Referência: wikipedia

Linguagem



SQL





A linguagem SQL

Structured Query Language, ou **Linguagem de Consulta Estruturada** ou **SQL**, é a linguagem de pesquisa declarativa padrão para [banco de dados relacional](#) (base de dados relacional). Muitas das características originais do SQL foram inspiradas na [álgebra relacional](#).

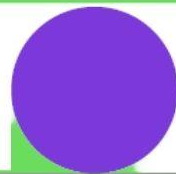
O **SQL** foi desenvolvido originalmente no início dos anos 70 nos laboratórios da [IBM](#) em San Jose, dentro do projeto [System R](#), que tinha por objetivo demonstrar a viabilidade da implementação do [modelo relacional](#) proposto por [E. F. Codd](#). O nome original da linguagem era *SEQUEL*, acrônimo para "*Structured English Query Language*" (Linguagem de Consulta Estruturada, em Inglês),^[1] vindo daí o facto de, até hoje, a sigla, em inglês, ser comumente pronunciada "síquel" ao invés de "és-kiú-él", letra a letra. No entanto, em português, a pronúncia mais corrente é letra a letra: "ésse-quê-éle".

A linguagem é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela é uma linguagem declarativa em oposição a outras linguagens procedurais. Isto reduz o ciclo de aprendizado daqueles que se iniciam na linguagem.

Embora o SQL tenha sido originalmente criado pela IBM, rapidamente surgiram vários "dialetos" desenvolvidos por outros produtores. Essa expansão levou à necessidade de ser criado e adaptado um padrão para a linguagem. Esta tarefa foi realizada pela [American National Standards Institute](#) (ANSI) em 1986 e ISO em 1987.

Referência: wikipedia

Inserção de



Dados





O comando insert

O comando insert é formado pela seguinte estrutura

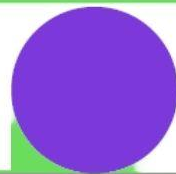
Insert into [Tabela]

(Atributos)

Values

(Valores)

Atualização de



Dados





O comando Update

O comando update é formado pela seguinte estrutura

update [Tabela]

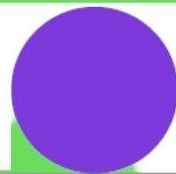
SET

Atributos = Valores

Where

Atributo = Valor

Exclusão de



Dados





O comando Delete

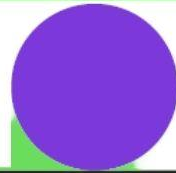
O comando delete é formado pela seguinte estrutura

`delete from [Tabela]`

`Where`

`Atributo = Valor`

Seleccionando



Dados





O comando Select

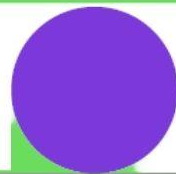
O comando select é formado pela seguinte estrutura

select (atributos) from [Tabela]

Where

Atributo = Valor

Junção de



Dados





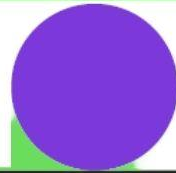
O comando join

Temos alguns tipos de joins

- **inner join** - Valida a integridade entre a tabela A com a tabela B
- **left join** - Retorna os registros da tabela da esquerda vinculando com a tabela da direita não validando a integridade, a prioridade de retorno é a tabela da esquerda
- **right join** - Retorna os registros da tabela da direita vinculando com a tabela da esquerda não validando a integridade, a prioridade de retorno é a tabela da direita
- **full join** - Retorna todos os registros da tabela independente da integridade
- **cross join** - A cláusula CROSS JOIN retorna todas as linhas das tabelas por cruzamento, ou seja, para cada linha da tabela esquerda queremos todos os linhas da tabelas direita ou vice-versa. Porém, para isso é preciso que ambas tenham o campo em comum, para que a ligação exista entre as duas tabelas. Para entender melhor, pense que temos um banco de dado, onde temos uma tabela CATEGORIA e uma tabela PRODUTO, assim poderíamos ter vários produtos para uma única CATEGORIA.

```
select t1.atributo, t2.atributo from t1 inner join t2 on t1.id = t2.t1Id where t2.nome = 'Danilo'
```

O comando



Union





O comando union

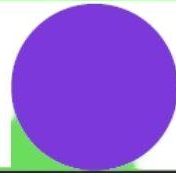
O comando union serve para unir dois selects em tabelas diferentes ou da mesma tabela, faz a união entre duas queries

```
SELECT column_name(s) FROM table1
```

```
UNION
```

```
SELECT column_name(s) FROM table2;
```


Paginando



Dados





O comando limit e offset

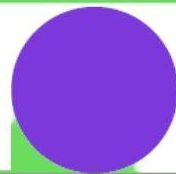
O limit serve para limitar uma quantidade de registros a ser apresentada

O offset representa a página dos itens limitados

SELECT * FROM Orders LIMIT 10 OFFSET 10

O OFFSET representa o próximo registro a ser apresentado ou próxima sequência de registros

Agrupamento de



Dados





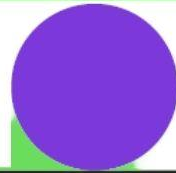
O comando group by

```
SELECT COUNT(CustomerID), Country
```

```
FROM Customers
```

```
GROUP BY Country;
```

Ordenando



Dados





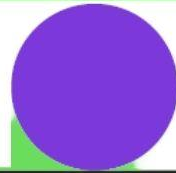
O comando order by

SELECT *

FROM Customers

ORDER BY name;

Indexando



Dados





Indexação de dados

O índice serve para dar agilidade nas buscas com registros grandes, utilizamos indexação quando queremos definir um dado que será buscado e precisará retornar com uma maior facilidade em nosso banco de dados

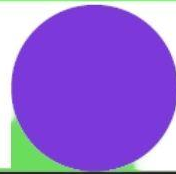
Comando de criação de índice

```
CREATE INDEX index_name ON table_name (column1, column2, ...);
```

Cria índice onde os valores não podem ser duplicados

```
CREATE UNIQUE INDEX index_name ON table_name (column1, column2, ...);
```


Criando



Tabelas

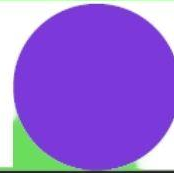




Criando tabelas

```
CREATE TABLE `novo_banco_de_dados`.`tabela1` (  
  
  `id` INT NOT NULL AUTO_INCREMENT,  
  
  `Nome` VARCHAR(45) NOT NULL,  
  
  `Telefone` VARCHAR(12) NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE  
  
);
```

Adicionando



Columnas

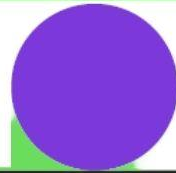




Adicionando columnas

```
ALTER TABLE table_name ADD column_name datatype;
```

Alterando



Colunas

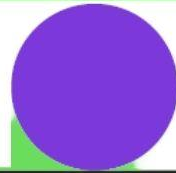




Adicionando colunas

```
alter table teste CHANGE column data_criacao data_criacao varchar(50)
```

Removendo



Colunas

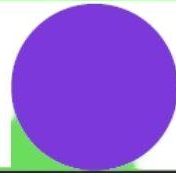




Removendo colunas

```
ALTER TABLE Customers DROP COLUMN Email;
```


Criando



Views





O que é views

View é uma forma de pré formatar um select compilado exemplo:

```
CREATE VIEW MinhaView AS
```

```
SELECT COLUNA2, COLUNA4, COLUNA6
```

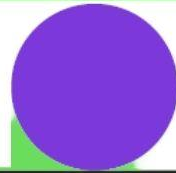
```
FROM TABELA
```

```
WHERE COLUNA1 = 1;
```

Como consultar

```
SELECT * FROM MinhaView;
```

Criando



Procedures





O que é procedures

Procedure é um conjunto de ações que visa calcular dados diretamente no banco de dados. Com procedure é possível programar algumas ações diretamente no banco de dados

delimiter //

CREATE PROCEDURE citycount (IN country CHAR(3), OUT cities INT)

BEGIN

SELECT COUNT(*) INTO cities FROM world.city

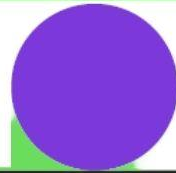
WHERE CountryCode = country;

END//

delimiter ;

CALL citycount('JPN', @cities);

Criando



Funções





O que é uma função

A função serve para formatar dados que você irá utilizar em um select customizado exemplo:

```
CREATE FUNCTION calc_nota(nota NUMERIC(15,2)) RETURNS NUMERIC(15,2)

BEGIN

DECLARE peso INT;

IF nota > 9.5 THEN

SET peso = 2;

ELSE

SET peso = 1;

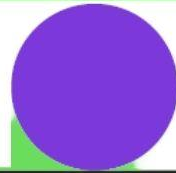
END IF;

RETURN (nota*peso) / 20;

END

SELECT a.nome, a.descricao, calcula_nota(a.valor_nota) AS nota_calculada FROM aluno
```

Subquery



Jobs

