

COMANDOS DE INTRODUÇÃO AO AMBIENTE

- **SELECT** VERSION();
- **SELECT** CURRENT_DATE;
- **SELECT** CURRENT_TIME;

COMANDOS DE INTRODUÇÃO AO AMBIENTE

- **SELECT** CURDATE(), CURTIME();
- **SELECT** CURDATE() **AS** data_atual, CURTIME() **AS** hora_atual;
- **SELECT** NOW();

COMANDOS DE INTRODUÇÃO AO AMBIENTE

- **SELECT USER();**
- **SELECT CURRENT_USER;**

COMANDOS DE INTRODUÇÃO AO AMBIENTE

- **SELECT** 5 + 7.5, 5 - 7.5, 5 * 7.5, 5 / 7.5;
- **SELECT** 1 + 2 * 3;
- **SELECT** (1 + 2) * 3;

COMANDOS DE INTRODUÇÃO AO AMBIENTE

- **SELECT** POWER(2, 8);
- **SELECT** SQRT(144);
 - **SELECT** POWER(X, (1/N)); => Raiz de índice qualquer.
- **SELECT** 9 **DIV** 2; => Divisão com quociente inteiro.
- **SELECT** MOD(5, 2); => Resto da divisão.
 - **SELECT** 5 % 2; => Utilizando operador aritmético.

CRIAÇÃO DE BANCO DE DADOS

- Sintaxe do comando de criação de banco de dados:

CREATE DATABASE <nome>;

CREATE DATABASE [IF NOT EXISTS] <nome>;

- Para visualizar os bancos de dados existentes no sistema:

SHOW DATABASES;

USO DE BANCO DE DADOS

- Quando um banco de dados é criado, ele deve ser usado:

USE <nome>;

REMOÇÃO DE BANCO DE DADOS

- Sintaxe do comando de remoção de banco de dados:

DROP DATABASE <nome>;

TABELAS E REGISTROS

- As **tabelas** são os componentes mais importantes para o trabalho de manutenção dos bancos de dados, pois é o local onde uma coleção de dados é inserida.
- Essa coleção de dados recebe o nome de **registro**.

CRIAÇÃO DE TABELAS

- Sintaxe simplificada:

```
CREATE TABLE [IF NOT EXISTS] <tabela> (  
    <coluna> <tipo coluna> [NULL/NOT NULL]  
                                [DEFAULT <valor>]  
                                [AUTO_INCREMENT]  
                                [PRIMARY KEY]  
);
```

CRIAÇÃO DE TABELAS

- **NULL/NOT NULL**

Indicação do campo em aceitar ou não os valores nulos.

- **DEFAULT *valor***

Permite determinar um valor padrão para o campo definido.

CRIAÇÃO DE TABELAS

- **AUTO_INCREMENT**

Determina se o campo será preenchido automaticamente quando nele forem entrando valores do tipo zero ou **NULL**, sendo válido apenas para campos que aceitem valores inteiros.

- **PRIMARY KEY**

Determina que o campo é a chave primária da tabela.

CRIAÇÃO DE TABELAS

- **UNSIGNED**

Usado com tipos de dados numéricos.

Limita os dados armazenados a números positivos e zero.

CRIAÇÃO DE TABELAS

- **UNIQUE**

Garante que todos os valores em uma coluna são diferentes.

UNIQUE e **PRIMARY KEY** fornecem uma garantia de exclusividade para uma coluna ou conjunto de colunas.

Uma **PRIMARY KEY** é automaticamente **UNIQUE**.

É possível ter vários **UNIQUE** por tabela, mas apenas uma **PRIMARY KEY**.

TIPOS DE DADOS MAIS COMUNS

- <https://dev.mysql.com/doc/refman/5.7/en/data-types.html>

TINYINT, SMALLINT, INTEGER, BIGINT

FLOAT, DOUBLE

DECIMAL

CHAR, VARCHAR

ENUM

DATE, TIME, DATETIME

```
CREATE DATABASE virtual;
```

```
USE virtual;
```

```
CREATE TABLE funcionario (  
    id            INTEGER NOT NULL PRIMARY KEY  
                AUTO_INCREMENT,  
    nome          VARCHAR(40) NOT NULL,  
    departamento  CHAR(2),  
    funcao        VARCHAR(20),  
    salario       DECIMAL(6, 2)  
);
```


VISUALIZAÇÃO DE TABELAS

SHOW TABLES;

DESCRIBE <nome> [coluna];

DESCRIBE funcionario;

DESCRIBE funcionario id;

INSERÇÃO DE REGISTROS

- A partir do momento em que uma tabela está criada, ela pode receber entrada de dados.

```
INSERT INTO    <tabela (campo1, campo2, ..., campoN)>  
VALUES        <(valor1, valor2, ..., valorN)>;
```

INserção DE REGISTROS

INSERT INTO funcionario

(id, nome, departamento, funcao, salario)

VALUES

(12, 'Carlos Alberto', '3', 'Vendedor', 1530.00);

INSERÇÃO DE REGISTROS

INSERT INTO funcionario

VALUES

(15, 'Marcos Henrique', '2', 'Gerente', 1985.75);

A instrução pode ser simplificada, omitindo os nomes dos campos.

id	nome	departamento	funcao	salario
7	Aparecida Silva	3	Secretaria	1200.50
9	Solange Pacheco	5	Supervisora	1599.51
6	Marcelo Souza	3	Analista	2250.11
1	Celia Nascimento	2	Secretaria	1200.50
2	Wilson de Macedo	3	Programador	1050.00
5	Augusto Souza	3	Programador	1050.00
4	Carlos Bastos	5	Vendedor	1530.00
25	Pedro Silva	3	Supervisor	1599.51
3	Ana Bastos	5	Vendedora	1530.00
10	Paulo da Silva	2	Vendedor	1530.00

CONSULTA DE REGISTROS

- Para extrair listagens de registros de uma tabela, a linguagem SQL disponibiliza o comando **SELECT** que anexado a alguns parâmetros, fornece uma instrução de consulta:

```
SELECT [tipo] <campos> FROM <tabela> [condição];
```

CONSULTA DE REGISTROS

- Apresentar a listagem de todos os funcionários cadastrados:

```
SELECT * FROM funcionario;
```

CHAVES ESTRANGEIRAS E ÍNDICES

```
CREATE TABLE IF NOT EXISTS `mydb2`.`cliente` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

```
-- Table `mydb2`.`telefone`
```

```
CREATE TABLE IF NOT EXISTS `mydb2`.`telefone` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `numero` VARCHAR(20) NOT NULL,  
  `cliente_id` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `fk_telefone_cliente_idx` (`cliente_id` ASC),  
  CONSTRAINT `fk_telefone_cliente`  
    FOREIGN KEY (`cliente_id`)  
    REFERENCES `mydb2`.`cliente` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```