

Mysql

# Comandos básicos

- Definir um banco de dados para manipular :
  - `USE <banco> ;`
- Verificar informações da conexão ao banco :
  - `status ;`
- Executar comandos de um arquivo .sql :
  - `source nome_do_arquivo.sql ;`

# Adicionar

- CREATE DATABASE <banco> ;
- CREATE TABLE `<banco>`.`<tabela>` (  
    `<campos>` <tipos> <opcionais> ,  
    `<campos>` <tipos> <opcionais>  
);
- CREATE DOMAIN <novo\_tipo> <tipo> ;

# Criar Tabela

```
CREATE TABLE Venda (  
    codForn INTEGER NOT NULL,  
    codPeca INTEGER NOT NULL,  
    total DOUBLE NOT NULL,  
    data DATE NOT NULL,  
    PRIMARY KEY (codForn, codPeca),  
    FOREIGN KEY (codForn) REFERENCES Fornecedor(codigo),  
    FOREIGN KEY (codPeca) REFERENCES Peca(codPeca) ON  
    DELETE CASCADE ON UPDATE RESTRICT  
);
```

# Alterar

- ALTER TABLE <tabela> ADD <coluna> <tipo> ;
- ALTER TABLE <tabela> DROP COLUMN <coluna> ;
- ALTER TABLE <tabela> MODIFY COLUMN <coluna> <tipo> ;
- ALTER TABLE <tabela> CHANGE COLUMN <coluna> <novo\_nome> <tipo> ;

# Alterar Tabela

- **ALTER TABLE** Venda **ADD** Cliente **CHAR(10);**
- **ALTER TABLE** Venda **CHANGE COLUMN** Cliente Nome\_Cliente **VARCHAR(80);**
- **ALTER TABLE** Venda **MODIFY COLUMN** Cliente **VARCHAR(100);**
- **ALTER TABLE** Venda **DROP COLUMN** Cliente;

# Deletar

- DROP DATABASE <banco>;
- DROP DATABASE IF EXISTS <banco>;
- DROP TABLE <tabela>;

# Visualizar

- SHOW DATABASES ;
- SHOW TABLES ;
- SHOW TABLES FROM <banco>;
- DESCRIBE <tabela>;
- SHOW COLUMNS FROM <tabela>;



# Manipulação nos dados

- INSERT INTO
- UPDATE SET
- DELETE FROM
- SELECT FROM

# Inserção de dados

- INSERT INTO <tabela> (<colunas>) VALUES (<valores>)
- INSERT INTO <tabela> VALUES (<valores>)

# Inserir Registros

```
INSERT INTO Venda (codForn, codPeca, total, data)  
VALUES (1, 2, 34.78, CURDATE());
```

```
INSERT INTO Venda VALUES (1, 2, 34.78,  
'2018-2-01');
```

# Atualização de dados

- UPDATE <tabela> SET <campos> = '<valores>' WHERE <condições>
- UPDATE <tabela> SET <campos> = '<valores>' WHERE <coluna> in ( SELECT ... )

# Atualizar Registros

- **UPDATE Venda SET total = 100 WHERE total < 100;**
- **UPDATE Venda SET total = 0 WHERE data > CURDATE();**

# Exclusão de dados

- DELETE FROM <tabela> WHERE <condições>
- DELETE FROM <tabela>  
**OBS: Desse jeito todos os dados da tabela serão apagados**

# Consultas básicas

- **SELECT \* FROM Venda ;**
- **SELECT \* FROM Venda v, Fornecedor f ;**
- **SELECT codPeca, nome FROM Venda, Fornecedor ;**
- **SELECT v.codPeca, f.nome FROM Venda AS v, Fornecedor as f ;**
- **SELECT Venda.codPeca, Peca.descricao FROM Venda, Peca ;**

# Ordenando e sem duplicatas

- **SELECT DISTINCT cidade FROM Fornecedor ;**
- **SELECT \* FROM Venda ORDER BY total ASC;**
- **SELECT \* FROM Venda ORDER BY total, codForn ASC, data DESC;**
- **SELECT codPeca as Codigo FROM Peca ;**  
**OBS : A coluna codPeca sairá com o nome Codigo !**



# Operações matemáticas

- **SELECT COUNT(cidade) FROM Fornecedor ;**
- **SELECT SUM(total) FROM Venda ;**
- **SELECT MIN(total) FROM Venda ;**
- **SELECT MAX(total) FROM Venda ;**
- **SELECT AVG(total) FROM Venda ;**
- **SELECT SUM(total) - MAX(total) FROM Venda ;**
- **SELECT MIN(total) \* 10 FROM Venda ;**
- **SELECT MAX(total) / 2 FROM Venda ;**

# Agrupando resultados

- **SELECT SUM(total) FROM Venda GROUP BY codForn ;**
- **SELECT SUM(total) as Lucro FROM Venda GROUP BY codForn HAVING sum(total) > 50 ;**
- **SELECT nome as Fornecedor FROM Venda, Fornecedor GROUP BY nome HAVING avg(total) < 1000 ;**

# Especificando cláusulas

- **SELECT** nome **FROM** Peca as P, Venda as V  
**WHERE** P.codPeca = V.codPeca;
- **SELECT** \* **FROM** Venda, Fornecedor **WHERE**  
Venda.codForn = Fornecedor.codigo;
- **SELECT** nome **FROM** Fornecedor **WHERE** cidade  
**IS NULL**;

# Expressões Lógicas

- **SELECT DISTINCT** F.nome, total, data **FROM** Venda as V, Peca, Fornecedor as F **WHERE** F.codigo = V.codForn **AND** ( total > 0 **OR** data = CURTIME() );
- **SELECT \* FROM** Peca **WHERE** codPeca **LIMIT** 5;
- **SELECT \* FROM** Venda **WHERE** total **BETWEEN** 0 **AND** 1000;

# Operador IN

- **SELECT \* FROM Venda as V WHERE V.data IN (CURTIME() , '2017-12-31', '2018-1-01') ;**
- **SELECT \* FROM Peca WHERE codPeca NOT IN (7, 3, 2, 6);**
- **SELECT codigo, nome FROM Fornecedor WHERE nome IN ( SELECT nome FROM Venda, Fornecedor WHERE codForn = codigo);**

# Operador LIKE

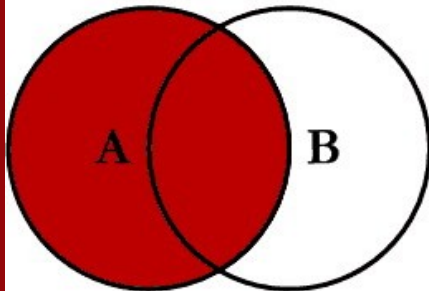
- **SELECT \* FROM** Fornecedor as f **WHERE** f.nome **LIKE** 'th%i%o';
- **SELECT \* FROM** Peca **WHERE** nome **LIKE** '%ul\_\_%';
- **SELECT** nome **FROM** Fornecedor **WHERE** nome **NOT LIKE** 'thuli\_';

# Operadores Extras

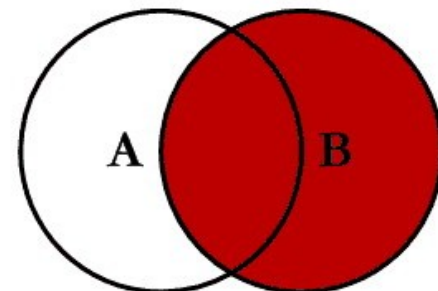
- **SELECT** nome **FROM** Peca **WHERE** codPeca = **ALL** (**SELECT** codPeca **FROM** Venda **WHERE** total = 0);
- **SELECT** nome **FROM** Fornecedor **WHERE** codigo = **ANY** (**SELECT** codForn **FROM** Venda **WHERE** total = 0);
- **SELECT** nome **FROM** Fornecedor **WHERE EXISTS** (**SELECT** codForn **FROM** Venda **WHERE** total = 0 **AND** Fornecedor.codigo = Venda.codForn);

# JOINS

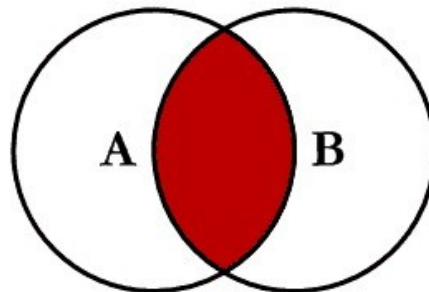
## SQL JOINS



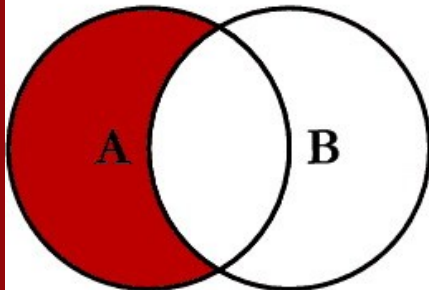
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



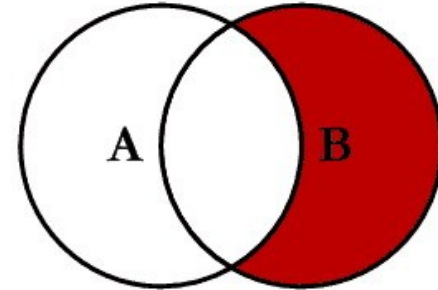
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



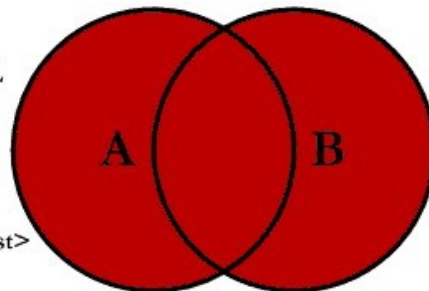
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



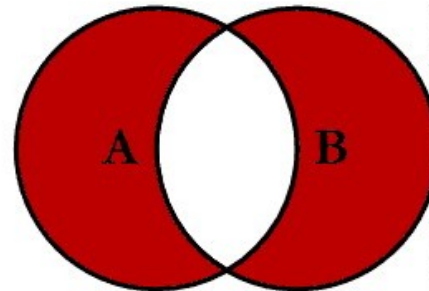
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```