# Banco de dados

## SQL – Linguagem de consulta estruturada

# Criar uma base de dados(banco de dados)

No MySQL**: create** **database** locadora;

No Postgresql**: createdb** locadora;

# Criar uma tabela chamada clientes

**create table clientes**

(

cli\_Codigo integer,

cli\_Nome varchar(30),

cli\_Cidade varchar(20),

cli\_Sexo char(1)

);

**create table filmes**

(

fil\_Codigo integer,

fil\_Nome varchar(30),

fil\_Genero varchar(15),

fil\_Preco numeric(4,2)

);

**Inserir dados na tabela clientes[INSERT];**

**insert into clientes**(cli\_Nome,cli\_Cidade,cli\_Sexo,cli\_DataCad,cli\_Salario,cli\_Comissao) **values**(’celio’,1,’M’,’2012-05-01’,1250,65);

**ou**

**insert into** clientes **values**(5,’carine’,‘ilopolis’,’F’);

Visualizar,consultar,pesquisar,selecionar[**SELECT**]:

Seqüência no comando "**Select**":

**SELECT** coluna(s)

**FROM** tabela(s)

**WHERE** condição(ões) da(s) tupla(s)

**GROUP BY** condição(ões) do(s) grupo(s) de tupla(s)

**HAVING** condição(ões) do(s) grupo(s) de tupla(s)

**ORDER BY** coluna(s);

A "sql" fará a seguinte avaliação**:**

**WHERE,** para estabelecer tuplas individuais candidatas (não pode conterfunções de grupo)

**GROUP BY**, para fixar grupos.

**HAVING**, para selecionar grupos para exibição

**SELECT** cli\_Nome,cli\_Cidade,cli\_Salario **FROM** clientes **HAVING** cli\_Salario **>** 500;

**select** cli\_Codigo,cli\_Nome,cli\_Cidade,cli\_Sexo from clientes;

**ou**

**select \* from clientes**;

# Clausula [ORDER BY];

Classifica a tabela em ordem **ascendente** ou **descendente**.

**select \* from** clientes **order by** cli\_Nome;

# ou

**select \* from** clientes **order by** cli\_Nome **asc**;

# ou

**select \* from** clientes **order by** cli\_Nome **desc**;

# ou

**select \* from** clientes **order by** cli\_Nome **desc,** cli\_Cidade **asc**;

**Restricao [WHERE];**

Restringe os dados(tuplas) que serao listados, ou seja, efetua uma condicao.

Operadores lógicos

= igual a

> maior que

>= maior que ou igual a

< menor que

<= menor que ou igual a

Ex.: **select \* from** clientes **where** cli\_codigo >= 2;

**select \* from** clientes **where** cli\_codigo >= 2 **and** cli\_codigo <= 5;

**select \* from** clientes **where** cli\_codigo >= 2 **and** cli\_codigo <= 5 **and** cli\_Sexo = ‘M’; **select \* from** clientes **where** cli\_codigo >= 2 **and** cli\_codigo <= 5 **and** cli\_Cidade = ‘anta gorda’ **order by** cli\_Nome;

**select \* from** clientes **where** cli\_Nome **=** ‘yuri’;

ou

**select \* from** clientes **where** cli\_Nome **like** ‘yuri’;

ou

//seleciona todos os registros que comecem com a letra **n**

**select \* from** clientes **where** cli\_Nome **like** ‘n%’;

ou

//seleciona todos os registros que contenham a letra **n** no seu contexto

**select \* from** clientes **where** cli\_Nome **like** ‘%n%’;

ou

//seleciona todos os registros que terminem com a palavra “dro”

**select \* from** clientes **where** cli\_Nome **like** ‘%dro’;

ou

**select \* from** clientes **where** cli\_Nome **like** '%n%'**order by** cli\_Codigo **desc**;

//seleciona o campo comissao cujo valor nao seja nulo

**select \* from** clientes **where** cli\_Comissao **is not NULL**;

//seleciona o campo comissao cujo valor seja nulo

**select \* from** clientes **where** cli\_Comissao **is NULL**;

//converte em minusculas

**select LOWER**(cli\_Nome) **from** clientes;

//converte em maiusculas

**select UPPER**(cli\_Nome) **from** clientes;

Mais Operadores

**like =** com padrão de caracteres

**between ...and...=**lista de 2 valores

**In =** lista de valor pre-definidos

# Restricao [WHERE] com [BETWEEN];

**select \* from** clientes **where** cli\_Codigo **Between** 2 **and** 5;

e a mesma coisa que--->

**select \* from** clientes **where** cli\_codigo >= 2 **and** cli\_codigo <= 5;

# Restricao [WHERE] com [IN];

**select \* from** clientes **where** cli\_Codigo **in(2,4,6)**;

**select \* from** clientes **where** cli\_Codigo **=** 2 **or** cli\_Sexo **=** ‘M’;

**Restricao [NOT BETWEEN] com [NOT IN];**

//seleciona todos os registros que não estejam entre 2 e 5

**select \* from** clientes **where** cli\_Codigo **not** **Between** 2 **and** 5;

//seleciona todos os registros que que não sejam 2,4,6

**select \* from** clientes **where** cli\_Codigo **not in**(2,4,6);

**Manipulando colunas das tabelas[RENOMEANDO COLUNAS]**

**select** cli\_Codigo’Codigo do Cliente’,cli\_Nome’Nome do Cliente’,cli\_Cidade’Cidade’,cli\_Sexo’Sexo’ from clientes;

Multiplicando 2 ao codigo

**select** cli\_Codigo \* 2’Codigo do Cliente’,cli\_Nome’Nome do Cliente’,cli\_Cidade’Cidade’,cli\_Sexo’Sexo’ from clientes;

**DISTINCT**

Elimina duplicidade de dados nas pesquisas

**Select distinct** cli\_Cidade **from** clientes;

**PRIMARY KEY**

Criando a chave principal

**create table clientes**

(

cli\_Codigo integer not null,

cli\_Nome varchar(30) not null,

cli\_Cidade varchar(20),

cli\_Sexo char(1)

primary key(cli\_Codigo)

);

**DROP**

Elimina uma tabela,dados ou banco de dados

Ex.: **drop table** clientes;

**ALTER**

Permite alterar ,modificar,inserir atributos na tabela existente

Ex.: **alter table** clientes **add** cli\_Email **varchar**(30);//adiciona

Ex.: **alter table** clientes **drop** cli\_Email;//exclui

**Modificar estrutura das colunas**

No **MySQL**= **alter table** clientes **modify** cli\_Email **varchar**(20); //modifica

No **Firebird**= **alter table** clientes **alter column** cli\_Email **type** **varchar**(20);

**alter table** clientes **alter column** cli\_Email **to** cli\_Correio; //troca o nome da coluna

**alter table** clientes

**add** cli\_polegar **varchar**(20),

**add** cli\_indicador **varchar**(20),

**add** cli\_medio **varchar**(20),

**add** cli\_anelar **varchar**(20),

**add** cli\_minimo **varchar**(20);

**DESCRIBE**

Permite visualizar a estrutura da tabela

Ex.: **describe** clientes;

**UPDATE**

Permite alterar os dados da tabela

Ex.: **update** clientes **set** cli\_Sexo **=** ‘F’ **where** cli\_Cidade **=** ‘ilopolis’;

**update** clientes **set** cli\_Cidade **=** ‘guapore’ **where** cli\_Codigo **=** 2;

**atualizacao**

**update** clientes **set** cli\_Salario **=** cli\_Salario \* 0.2 + cli\_Salario **where** cli\_Sexo **=** 'F'

**DELETE**

Elimina registros de uma tabela

Ex.: **delete from** clientes;

**TRUNCATE**

Elimina de forma irreversível todos os dados de uma tabela

Ex.: **truncate from** clientes;

**Funcoes de agregacao**

**AVG**

Calcula a media de uma coluna

Ex.: **select avg(**cli\_Salario**)as** Media **from** clientes;

**COUNT**

Conta os registros de uma Tabela

Ex.: **select count(**\***)as** Contagem **from** clientes;

**SUM**

Soma uma coluna de uma Tabela

Ex.: **select sum(**cli\_Salario**)as** Soma **from** clientes;

**select sum(**cli\_Salario**)as** Soma **from** clientes **where** cli\_Codigo **>** 2;

**MIN**

Calcula o valor mínimo de uma coluna

Ex.: **select min(**cli\_Salario**)as** Minimo **from** clientes;

**MAX**

Calcula o valor maximo de uma coluna

Ex.: **select max(**cli\_Salario**)as** Maximo **from** clientes;

**Funcoes escalares de texto**

**UPPER**

Converte para maiúsculas todos os caracteres

Ex.: **select upper(**cli\_Nome**)as** Maiusculas **from** clientes;

**LOWER**

Converte para Minusculas todos os caracteres

Ex.: **select lower(**cli\_Nome**)as** Minusculas **from** clientes;

**CONCAT(no MYSQL)**

Reune em uma única sequencia um ou mais atributos de tabela

Ex.: **select concat (**cli\_Nome,’ - ’,cli\_Salario**)as** ‘Nome e Salario’ **from** clientes;

**||(no FIREBIRD)**

Ex.: **select** cli\_Nome**||' - '||**cli\_Salario **as "**NomeeSalario**"**

**from** clientes;

**Indices**

**INDICE**

Indice e usado para agilizar as pesquisas de seleção de dados nas tabelas

O key e um sinonimo de índice

**Show index from** Clientes;

**Drop index** ind\_nome\_cliente **on** clientes;

**create index** ind\_nome\_cliente **on** clientes(cli\_Nome);

No MySQL = **Drop index** ind\_nome\_cliente **on** clientes;

No Firebird = **Drop index** ind\_nome\_cliente;

**AUTO INCREMENTO**

Preenhe automaticamente um campo inteiro toda vez que um conteúdo for inserido na tabela.

No MySQL = auto\_increment

Ex.: **alter table** clientes **modify** cli\_Codigo **integer not null auto\_increment**;

**GROUP BY**

Tem a função de agrupar dados

Ex.: **select count(**cli\_Sexo) **from** clientes **group by** cli\_Sexo;

**OU**

//Retorna a contagem dos registros agrupando por sexo

**select** **concat(**‘Do sexo ’,cli\_Sexo,’ tem ‘ ,

**count(**cli\_Sexo**) )as** Quantidade

**from** clientes

**group by** cli\_Sexo;

**OU**

//Retorna a media salarial de cada cidade agrupando por cidade

**select** **concat(**‘Na cidade de’,cli\_Cidade,’ a media salarial e ‘ ,

**avg(**cli\_Salario**),**’ reais ’ **)as** Saldo

**from** clientes

**group by** cli\_Cidade;

**OU**

//Retorna a media e a soma salarial de cada cidade agrupando por sexo

**select avg(**cli\_Salario**),sum(**cli\_Salario**) from** clientes **group by** cli\_Sexo;

**OU**

//Retorna a media e a soma salarial e a contagem de pessoas de cada cidade agrupando por sexo

**select avg**(cli\_Salario) **as** Media,

**sum**(cli\_Salario) **as** Soma,

**count**(cli\_Sexo) **as** 'Quantidade de pessoas'

**from** clientes

**group by** cli\_Sexo;

**GROUP BY COMPOSTO**

Tem a função de agrupar dados por duas condições

**select** cli\_Cidade **as** Cidade,

cli\_Sexo **as** Sexo,

**avg**(cli\_Salario) **as** Media,

**sum**(cli\_Salario) **as** Soma,

**count**(cli\_Sexo) **as** 'Quantidade de pessoas'

**from** clientes

**group by** cli\_Cidade,cli\_Sexo;

**TABELA CIDADES**

**create table** cidades

**(**

cid\_Codigo **integer not NULL auto\_increment,**

cid\_Nome **varchar(35),**

**PRIMARY key(**cid\_Codigo**)**

**);**

**CHAVE ESTRANGEIRA (foreign key)**

//Cria a chave estrangeira

**alter table** clientes

**add** **foreign key**(cli\_CodCid) **references** cidades(cid\_Codigo);

//Cria a chave estrangeira com constraint(Da o nome)

**alter table** clientes

**add CONSTRAINT** chave\_estrangeira\_cidade

**foreign key**(cli\_CodCid)

**references** cidades(cid\_Codigo)

**on delete RESTRICT**

**on update CASCADE;**

//Apaga a chave estrangeira

**alter table** clientes

**drop** **foreign key**(chave\_estrang\_cidade);

**UNION**

Usado para unir o conteudo de uma ou mais tabelas

**select** cid\_Codigo, cid\_Nome **from** cidades

**union**

**select** cli\_Codigo, cli\_Nome **from** clientes

**union**

**select** fil\_Codigo, fil\_Nome **from** filmes;

//usado para unir dados de uma mesma tabela

**select** cli\_Codigo, cli\_Nome, cli\_polegar **as** Digital\_cliente

**from** clientes **where** cli\_polegar **is** **not** **null**

**union** **all**

**select** cli\_Codigo, cli\_Nome, cli\_indicador **as** Digital\_cliente

**from** clientes **where** cli\_indicador **is** **not** **null**

**union** **all**

**select** cli\_Codigo, cli\_Nome, cli\_medio **as** Digital\_cliente

**from** clientes **where** cli\_medio **is** **not** **null**

**union** **all**

**select** cli\_Codigo, cli\_Nome, cli\_anelar **as** Digital\_cliente

**from** clientes **where** cli\_anelar **is** **not** **null**

**union** **all**

**select** cli\_Codigo, cli\_Nome, cli\_minimo **as** Digital\_cliente

**from** clientes **where** cli\_minimo **is** **not** **null**;