

BANCO DE DADOS

BANCO DE DADOS

Banco de dados é um conjunto de dados armazenados que podem estar interligados e geram informações importantes para uma determinada área.

Dados: Elementos isolados que não tem qualquer significado específico.

SGBD – Sistema gerenciador de banco de dados

Linguagem de exploração. Ex: **SQL**(structured Query Language)

Tabela é constituída por campos e registros.

Registros corresponde as linhas da tabela(tuplas)

Campos corresponde as colunas da tabela

Normalização é um processo que permite eliminar redundancias/informações repetidas na base de dados

1ªForma Normal

*Cada atributo contem apenas valores atômicos(valores indivisíveis).

*Não há conjuntos de atributos repetidos descrevendo a mesma característica.

2ªForma Normal

*Primeira forma Normal

*Todos atributos não chave dependem funcionalmente da chave.

3ªForma Normal

*Segunda forma Normal

*Os atributos não chave dependem funcionalmente um dos outros.

COMANDOS:

DDL(data definition language) – Comandos que mexem nas estrutura do banco de dados.

DML(data manipulation language) – Comandos que de manipulação de dados

DQL(data query linguagem) – Comando de pesquisa.

Criar banco de dados

```
=> create database NomeDoBancoDeDados  
    default character set utf8  
    default collate utf8_general_ci;
```

Usar banco de dados

```
=>use NomeDoBancoDeDados;
```

Mostrar todas tabelas do banco de dados

```
=>show tables;
```

Criar tabela

```
=> create table NomeDaTabela(  
    campo1 tipo,  
    campo2 tipo  
    )default charset=utf8;
```

Inserir dados na tabela

```
=> insert into NomeDaTabela(campo1, campo2) values  
("valor1" , "valor2"),  
("valor1" , "valor2");
```

Criar tabela com chave primária(pk)

```
=> create table NomeDaTabela(  
    campo1 tipo,  
    campo2 tipo,  
  
    primary key(campo1)  
    )default charset=utf8;
```

Tornar um campo auto incremento, adicionar um valor default

```
=> create table NomeDaTabela(  
    campo1 tipo auto_increment,  
    campo2 tipo default "valor",  
  
    primary key(campo1)  
    )default charset=utf8;
```

Ver estrutura da tabela

```
=>desc NomeDaTabela;
```

Adicionar coluna

```
=>alter table NomeDaTabela  
    add column NomeColuna tipo;
```

att: after, before;

Alterar tamanho da coluna

```
=>alter table NomeDaTabela  
    modify column NomeColuna tipo(tamanho);
```

Alterar nome da coluna

```
=>alter table NomeTabela  
    change column NomeAntigo NomeNovo tipo;
```

Eliminar coluna

=>alter table NomeTabela
drop column NomeColuna;

Eliminar tabela

=> drop table NomeTabela;

Alterar valor registo (linha/tupla)

=>update NomeTabela
set NomeCampo = “NovoValor”
where id = “numberPK”;

***Alterar mais de 1 valor no registo (linha/tupla) ***

=>update NomeTabela
set NomeCampo = “NovoValor”, NomeCampo= “NovoValoe”
where id = “numberPK”
limit 1;

Apagar Linha

=>delete from NomeTabela
where id = “numberPk”;

Apagar todas as linhas da tabela

=>Truncate table NomeTabela;

Selecionar todos os dados

=>select * from NomeTabela;

Selecionar todos os dados e ordenar

=>select * from NomeTabela
order by NomeCampo;

Selecionar todos com idade maior que 18

=>select * from NomeTabela
where idade>18;

Selecionar todos com idade diferente a 18

=>select * from NomeTabela
where idade<>18; /*where idade !=18*/

Selecionar todos com o idade no intervalo dos 18 a 21

```
=>select * from NomeTabela  
where idade between 18 and 21;
```

Selecionar todos com idade 18 e 21anos

```
=>select * from NomeTabela  
where idade in(18, 21);
```

Selecionar nome com a inicial A

```
=>select * from NomeTabela  
where nome like 'A%';
```

Selecionar nome com a inicial A e obrigatoriamente com letras após o A

```
=>select * from NomeTabela  
where nome like 'A_%';
```

Contar quantos alunos tem idade maior que 18

```
=>select count(*) from NomeTabela  
where idade > 18;
```

Selecionar maior de idade

```
=> select max(idade) from NomeTabela;
```

Selecionar menor de idade

```
=>select min(idade) from NomeTabela;
```

Calcular a média das idades

```
=>select avg(idade) from NomeTabela;
```

Selecionar idade de todos, sem considerar a quantidade de repetições

```
=>select distinct idade from NomeTabela;
```

Adicionar chave estrangeira

```
=>alter table NomeTabela  
add foreign key(NomeCampo)  
references NomeTabela2(idTabela2);
```

Consultar dados de duas tabelas

```
select table1.Campo, table2.Campo from Table1  
join Table2  
on Table1.NomeFk = Table2.NomePK;
```

Mostar todos dados da table1 (left) até os que não tem ligação com a table2

```
=>select CampoTable1, CampoTable2 from Table1 left  
join Table2  
on Table1.id = Table2.id;
```

att: o contrário é só substituir o left por right.

Consultar dados N – N

```
=> select table1.campo, table2.campo from Table1 join tableMeio  
on table1.id = tableMeio.id  
join table2  
on table2.id = tableMeio.id;
```