

```
create database if not exists aula2;

use aula2;

show tables;

create table cliente(
    idCliente int primary key,
    nome varchar (35),
    email varchar (50),
    sexo varchar (1),
    valormensalidade double
);

desc cliente;

insert into cliente (email,idCliente,nome,sexo,valormensalidade)
    values ('porfirio@gmail.com',10,'porfirio','m',200);

#ou
#Inserindo em uma Ordem Pré-definida

insert into cliente values (20,'lu','lu@gmail.com','f',300);
insert into cliente values
(30,'marcelo','marcelo@gmail.com','m',500);
insert into cliente values
(40,'garra','garra@gmail.com','m',580);
insert into cliente values
(50,'sergio','sergio@gmail.com','m',380);

select * from cliente;

create table endereco(
    idEndereco int primary key,
    bairro varchar (35),
    cidade varchar (50),
    id_Cliente int,
    foreign key(id_cliente) references Cliente(idCliente));
```

desc endereco;

Field	Type	Null	Key	Default	Extra
idEndereco	int(11)	NO	PRI	NULL	
bairro	varchar(35)	YES		NULL	
cidade	varchar(50)	YES		NULL	
id_Cliente	int(11)	YES	MUL	NULL	

#Essa teoria È Chamada Chave Estrangeira (Último Numero)

```
insert into cliente values
(60, 'belem', 'belem@gmail.com', 'm', 350);
```

#Dados do Endereco

#Insere oS DADOS DO ENDEREÇO

```
insert into endereco values (1000, 'tijuca', 'Rio de janeiro', 40);
insert into endereco values (1500, 'Barra da tijuca', 'Rio de
janeiro', 10);
insert into endereco values (2000, null, 'Rio de janeiro', 30);
insert into endereco values (2500, 'nova iguacu', 'Rio de
janeiro', 50);
insert into endereco values (3000, 'Leblon', 'Rio de janeiro', 20);
insert into endereco values (3500, 'Recreio', 'Rio de janeiro', 60);
```

#Atualizar ...

commit;

#Quero mostrar o Cliente com Endereco

#UNiR DUAS TABELAS

ChavePrimaria=Chave_estrangeira

#mostra os dados dos cliente e do endereco

mostra o nome, email, bairro, cidade do cliente e do Endereco

#c não obrigado mais é um diferencial

só é obrigatorio se existirem dois campos iguais ...

mostra o nome do cliente, o email do cliente, bairro do Endereco

#A Cidade do Endereco onde chavePrimari= ChaveEstrangeira ...

```
select c.nome, c.email, e.bairro, e.cidade from cliente c,  
endereco e where c.idCliente= e.id_cliente;
```

```
select c.nome, c.email, e.bairro, e.cidade from cliente c,  
endereco e where c.idCliente= e.id_cliente;
```

nome	email	bairro	cidade
porfirio janeiro	porfirio@gmail.com	Barra da tijuca	Rio de Janeiro
lu janeiro	lu@gmail.com	Leblon	Rio de Janeiro
marcelo janeiro	marcelo@gmail.com	NULL	Rio de Janeiro
garra janeiro	garra@gmail.com	tijuca	Rio de Janeiro
sergio janeiro	sergio@gmail.com	nova iguacu	Rio de Janeiro
belem janeiro	belem@gmail.com	Recreio	Rio de Janeiro

Visão (banco de dados)

```
#View é chamada tabela União de junção de Tabelas ...  
#Geralmente a View $  
# View é um atalho para Consulta ...  
#União de Tabelas para ser atalho de consulta  
# fica sempre muito mais Prático ...
```

Em teoria de banco de dados, uma **visão**, ou **vista** (em inglês: **view**), é um conjunto resultado de uma consulta *armazenada* sobre os dados, em que os usuários do banco de dados podem consultar simplesmente como eles fariam em um objeto de coleção de banco de dados persistente. Este comando de consulta pré-estabelecido é armazenado no dicionário de banco de dados. Diferente das *tabelas base* ordinárias em um banco de dados relacional, uma visão não forma parte do esquema físico: como um conjunto de resultado, ele é uma tabela virtual computada ou coletada dinamicamente

dos dados no banco de dados quando o acesso àquela visão é solicitado. Alterações aplicadas aos dados em uma *tabela subjacente* relevante são refletidas nos dados mostrados em invocações subsequentes da visão. Em alguns bancos de dados Não SQL, visões são a única maneira de consultar dados. Também pode ser definida como um objeto que não armazena dados, e não uma relação, composto dinamicamente por uma consulta que é previamente analisada e otimizada. Isso significa que, diferentemente de tabelas, visões não são objetos físicos, ou seja, não ocupam espaço em disco. Alterações nos dados de tabelas que são acessadas por visões, conseqüentemente alteram os resultados gerados pelas consultas armazenadas nessas visões.

Entre as principais utilidades estão, a depender do SGBD utilizado, o aumento de segurança por propiciar uma visão limitada e controlada dos dados que podem ser obtidos da base e a performance por utilizar uma consulta previamente otimizada, tornando desnecessário este processo de otimização quando for realizada.

Fornece mecanismo de segurança, restringindo o acesso de usuários. Simplifica a interação entre usuário final e banco de dados.

#Criou Uma Visão para unir duas tabelas

```
drop view if exists V$cliente_Endereco;
```

```
create view V$Cliente_Endereco as  
select c.idCliente, c.nome, c.email, c.sexo,  
c.valormensalidade,  
e.bairro, e.cidade from cliente c, endereco e  
where c.idcliente = e.id_cliente;
```

```
select * from V$cliente_endereco;
```

idCliente	nome	email	cidade	sexo	valormensalidade	bairro
40	garra	garra@gmail.com	m		580	tijuca
10	porfirio	porfirio@gmail.com	m		200	Barra da tijuca
30	marcelo	marcelo@gmail.com	m		500	Recreio
NULL						
50	sergio	sergio@gmail.com	m		380	nova iguacu
20	lu	lu@gmail.com	f		300	Leblon
60	belem	belem@gmail.com	m		350	Recreio

#TABELAS UNIDAS POR VISÃO

```
show tables;
```

Tables_in_aula1
cliente
endereco
v\$cliente_endereco

#Comando de Condição if

```
# quando for m o sexo = masculino, quando for f imprimir
feminino
# Mostrar o nome, masculino ou feminino da tabela cliente
#if Somente como irei Visualizar ...
```

```
select nome, if(sexo='m','Masculino','Feminino') "sexo"
from V$cliente_Endereco;
```

nome	sexo
porfirio	Masculino
lu	Feminino
marcelo	Masculino
garra	Masculino
sergio	Masculino
belem	Masculino

#case Somente como irei Visualizar ...

```
select nome, case sexo
              when 'm' then 'Masculino'
              when 'f' then 'Feminino'
              else 'vazio'
            end "sexo" ,
email from cliente;
```

nome	sexo	email
porfirio	Masculino	porfirio@gmail.com
lu	Feminino	lu@gmail.com
marcelo	Masculino	marcelo@gmail.com
garra	Masculino	garra@gmail.com
sergio	Masculino	sergio@gmail.com
belem	Masculino	belem@gmail.com

case = if

```
select if(lower(bairro) = 'tijuca', 'centro',
         if(lower (bairro) = 'leblon', 'zona sul',
            if(lower (bairro) = 'barra da tijuca', 'centro',
               'nao centro'
            ))) as "Mensagem", bairro from endereco;
```

Mensagem	bairro
centro	tijuca
centro	Barra da tijuca
nao centro	NULL

```
| nao centro | nova iguacu |
| nao centro | Leblon      |
| nao centro | Recreio     |
+-----+
```

```
select lower(bairro) from endereco;
```

```
+-----+
| lower(bairro) |
+-----+
| tijuca        |
| barra da tijuca |
| NULL         |
| nova iguacu   |
| leblon        |
| recreio       |
+-----+
```

```
select now();
```

```
+-----+
| now()        |
+-----+
| 2016-04-11 16:18:09 |
+-----+
```

Formatando a Data

Não preciso de From porque não estou perguntando a nenhuma TABELA !!!!!

```
select date_format(now(), '%d') as "dia",
       date_format(now(), '%m') as "Mes",
       date_format(now(), '%Y') as "Ano",
       date_format(now(), '%d-%m-%Y %h:%i %P') as "Data
Formatada",
       date_format(now(), '%v') as "Numero Semana";
```

```
+-----+-----+-----+-----+-----+
| dia | Mes | Ano | Data Formatada | Numero Semana |
+-----+-----+-----+-----+-----+
| 11  | 04  | 2016 | 11-04-2016 04:18 P | 15             |
+-----+-----+-----+-----+-----+
```