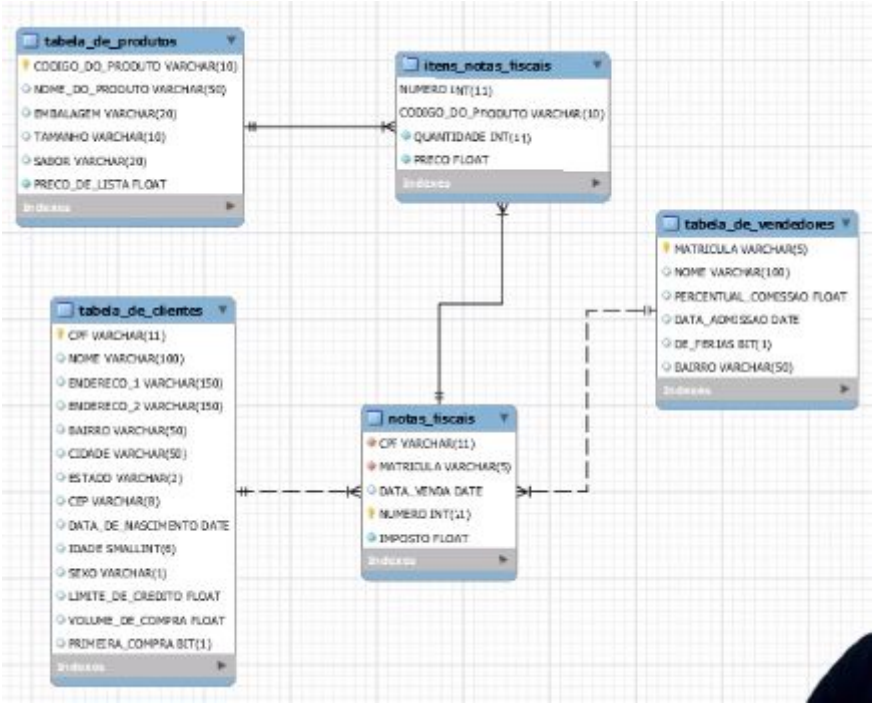
Consultas SQL

INSTALAÇÃO

Donwload versão free (*community edition*): https://dev.mysql.com/downloads/
No caso do curso, será utilizado a versão “MySQL on Windows (Installer & Tools)

CONHECENDO O BANCO DE DADOS

É possível gerar uma imagem visual das relações entre as tabelas clicando em “Databasase” -> “Reverse Engineer”



REVISANDO CONSULTAS

Use suco_vendas;

SELECT *CPF, NOME, END_1, END_2, BAIRRO, CIDADE, ESTADO, CEP, DATA_DE_NASC, IDADE, SEXO, LIM_CREDITO, VOL_COMPRA,PRIMEIRA_COMPRA* ***FROM*** *tabela_clientes;*
ou
SELECT * FROM *tabela_clientes;*
ou
SELECT *CPF, NOME* ***FROM*** *tabela_clientes;*
ou
SELECT CPF AS *IDENTIFICADOR, NOME* ***AS*** *CLIENTE* ***FROM*** *tabela_clientes;*
ou
SELECT * FROM tabela_produtos **WHERE** cod_produto = ‘550’;

Para selecionar um valor específico: (Pois caso se coloque por ex. 5.51, o sql pegará números próximos)
SELECT * FROM *tabela_produtos* ***WHERE*** *cod_produto* ***BETWEEN*** *19.50* ***AND*** *19.52;*

CONSULTAS CONDICIONAIS

SELECT * FROM *tabela_produtos* ***WHERE*** *cod_produto = '550'*
AND NOT *(tamanho = '470 ml');*

SELECT * FROM *tabela_produtos* ***WHERE NOT*** *(cod_produto = '550'*
OR tamanho = '470 ml');

SELECT * FROM *tabela_produtos* ***WHERE*** *sabor* ***IN*** *('Laranja','Manga');*
ou
SELECT * FROM *tabela_produtos* ***WHERE*** *sabor = 'Laranja'* ***OR*** *sabor = 'Manga';*

LIKE

SELECT * FROM *tabela_produtos* ***WHERE*** *CAMPO* ***LIKE*** *<CONDIÇÃO>*

SELECT * FROM *tabela_produtos* ***WHERE*** *cod_produto* ***LIKE*** *'%550';* //qualquer coisa antes do 5
ou
SELECT * FROM *tabela_produtos* ***WHERE*** *cod_produto* ***LIKE*** *'550%';* //qualquer coisa dps do 5
ou
SELECT * FROM *tabela_produtos* ***WHERE*** *cod_produto = '_550'* //um campo qualquer antes do 5

DISTINCT

SELECT DISTINCT * FROM *TABELA;* //retorna dados únicos diferentes

SELECT DISTINCT *** *EMBALAGEM, TAMANHO* ***FROM *tabela_produtos* ***WHERE*** *sabor = 'Laranja';*

LIMITANDO A SAÍDA DA CONSULTA

SELECT * FROM *tabela_produtos* ***LIMIT*** *4;* //limit fica sempre no final da consulta

SELECT * FROM *TABELA* ***LIMIT*** *2,3;* //a partir do segundo,seleciona os 3 proximos

ORDENANDO A SAÍDA DA CONSULTA

SELECT * FROM *tabela_produto* ***ORDER BY*** *campo;* //por default é ASC

SELECT * FROM *tabela_produto* ***ORDER BY*** *campo* ***DESC****;*

SELECT * FROM *tabela_produto* ***ORDER BY*** *campo, embalagem;* //primeiro ordenada por campo depois por embalagem (dentro da mesmal palavra do campo)

AGRUPANDO RESULTADOS

-Junta campos repetidos

SELECT *ESTADO,* ***SUM****(LIM_CREDITO)* ***AS*** *limite_total* ***FROM*** *tabela_clientes* ***GROUP BY*** *ESTADO;*

```
SELECT ESTADO, SUM(LIM_CREDITO) AS limite_total FROM tabela_clientes GROUP BY ESTADO;

SELECT EMBALAGEM, MAX(preco_lista) AS maior_preco FROM tabela_clientes GROUP BY
EMBALAGEM;

SELECT EMBALAGEM, COUNT(*) AS contador FROM tabela_produtos GROUP BY EMBALAGEM;

SELECT ESTADO, BAIRRO, SUM(LIM_DE_CREDITO) AS LIMITE FROM tabela_clientes
WHERE CIDADE = 'Rio de Janeiro'
GROUP BY ESTADO, BAIRRO
ORDER BY  BAIRRO;
```

HAVING

```
SELECT ESTADO, SUM(LIM_CREDITO) AS SOMA_LIMITE FROM tabela_clientes
GROUP BY ESTADO HAVING SUM(LIM_CREDITO) > 900000;
```

CLASSIFICAR RESULTADOS

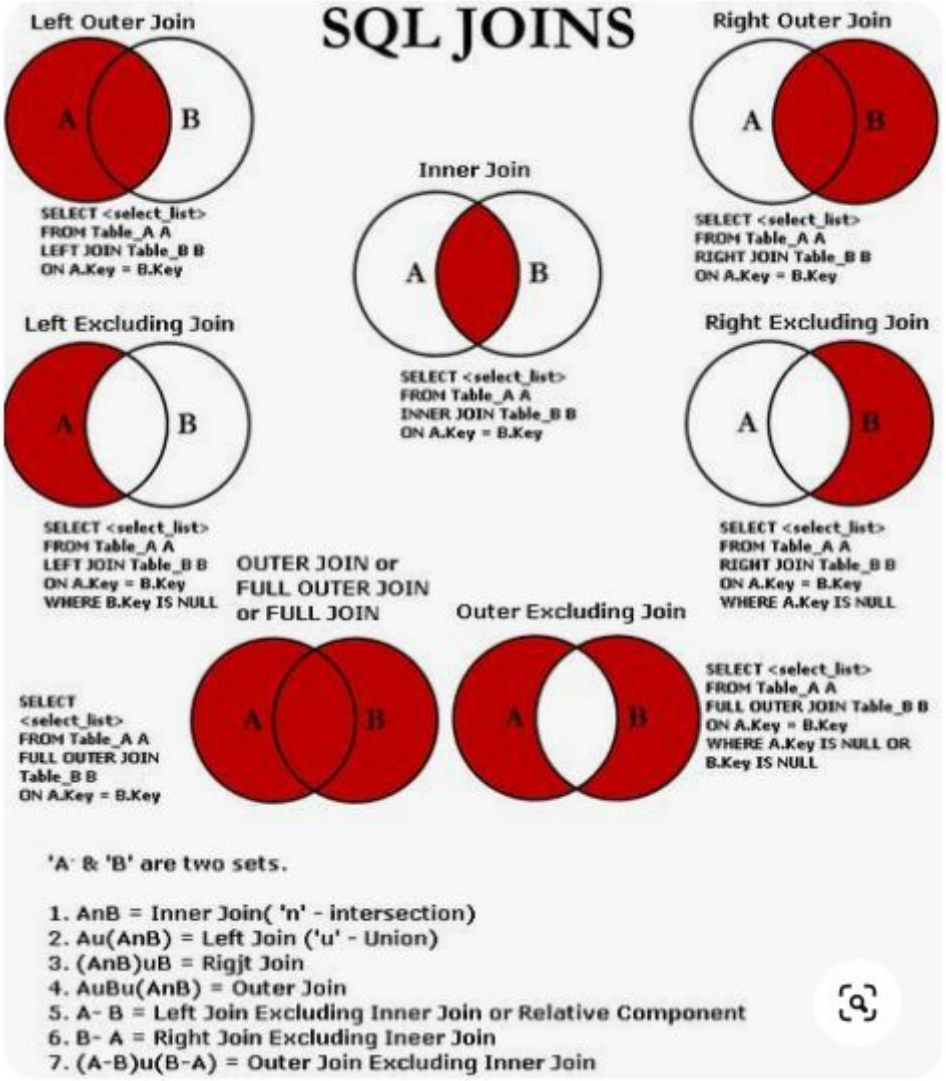
-É uma espécie de IF e ELSE

```
SELECT nome_produto, preco_lista,
CASE
        WHEN preco_lista >= 12 THEN 'PRODUTO CARO'
        WHEN preco_lista >= 7 AND preco_lista < 12 THEN 'PRODUTO EM CONTA'
        ELSE 'PRODUTO BARATO'
END AS status_preco
FROM tabela_produtos;
```

JOINS

-Une uma ou mais tabelas através de campos em comum

-INNER JOIN, LEFT JOIN , FULL JOIN, CROSS JOIN



***SELECT** * **FROM** tabela_vendedores A*
***INNER JOIN** notas_fiscais B*
***ON** A.MATRICULA = B.MATRICULA;*

SELECT** A.MATRICULA, A.NOME, **COUNT**(*) **FROM
tabela_vendedores A
***INNER JOIN** notas_fiscais B*
***ON** A.MATRICULA = B.MATRICULA*
***GROUP BY** A.MATRICULA, A.NOME;*

***SELECT YEAR**(DATA_VENDA), **SUM**(QUANTIDADE * PRECO) **AS** FATURAMENTO*
***FROM** notas_fiscais NF **INNER JOIN** itens_notas_fiscais INF*
***ON** NF.NUMERO = INF.NUMERO*
***GROUP BY** YEAR(DATA_VENDA)*

LEFT/RIGHT JOIN: Traz todo mundo da tabela da esquerda/direta e apenas os correspondentes da tabela da direita/esquerda.
ex. procurar quem nunca comprou na loja:
***SELECT DISTINCT** A.CPF, A.NOME, B.CPF **FROM** tabela_de_clientes A*
***LEFT JOIN** notas_fiscais B on A.CPF = B.CPF*

WHERE** B.CPF **IS NULL;

>>> 45919294831 FABIO CARVALHO NULL

INNER JOIN: Traz as colunas em comum para as duas tabelas.
ex.
SELECT tabela_de_vendedores.BAIRRO,
tabela_de_vendedores.NOME,
tabela_de_clientes.BAIRRO,
tabela_de_clientes.NOME **FROM** tabela_de_vendedores **INNER JOIN** tabela_de_clientes
ON tabela_de_vendedores.BAIRRO = tabela_de_clientes.BAIRRO;

CROSS JOIN:
ex.
SELECT tabela_de_vendedores.BAIRRO,
tabela_de_vendedores.NOME, DE_FERIAS,
tabela_de_clientes.NOME **FROM** tabela_de_vendedores, tabela_de_clientes;

JUNTANDO CONSULTAS

-O FULL JOIN é uma LEFT JOIN UNION RIGHT JOIN;
-UNION: Necessário ter o mesmo número de campos nas duas tabelas.
-Tem o DISTINCT por default, caso não se queira isso, usa-se UNION ALL
ex.
SELECT DISTINCT AIRRO **FROM** tabela_clientes
UNION
SELECT DISTINCT AIRRO **FROM** tabela_vendedores;

SUB CONSULTAS

-Chamamos uma consulta de X por ex, e depois utilizamos ela.
ex.
SELECT X.EMBALAGEM, X.PRECO_MAXIMO **FROM**
(**SELECT** EMBALAGEM, **MAX**(PRECO_DE_LISTA) **AS** PRECO_MAXIMO **FROM** tabela_de_produtos
GROUP BY EMBALAGEM) X **WHERE** X.PRECO_MAXIMO >= 10;

VIEW

-É uma tabela lógica, resultado de uma consulta, que pode ser usada em outra consulta qualquer;
-Uma view tem um custo pois é uma espécia de sub consulta.
ex.
USE `sucos_vendas`;
CREATE OR REPLACE VIEW 'VW_MAIORES_EMBALAGENS' **AS**
SELECT EMBALAGEM, **MA**(PRECO_DE_LISTA) **AS** MAIOR_PRECO **FROM** tabela_de_produtos
GROUP BY EMBALAGEM

FUNÇÕES DE STRING

-Sites para consulta:
dev.mysql.com/doc/
https://www.w3schools.com/sql/func_mysql_instr.asp

CONCAT()
-contatena strings
ex.
SELECT CONCAT(“SQL”, “ORACLE”); **AS** ConcatenatedString;

LTRIM()
-retira os espaços
ex.
SELECT LTRIM(“ SQL Tutorial”) **AS** LeftTrimmedString;

-Também existe a **RTRIM()** e a **TRIM()** [retira espaços da esquerda e direita]

UPPER() e **LOWER()**
-Transforma em maiúscula e minúscula

SUBSTRING()
-extraia uma substring de uma string
ex. inicie na posição 5 e extraia 3 caracteres:
SELECT SUBSTRING(“SQL Tutorial”, 5, 3) **AS** ExtractString;
>>>Tut

FUNÇÕES DE DATAS

ADDDATE()
-Adiciona um intervalo de hora / data a uma data e retorna a data
SELECT ADDDATE(“2017-06-15”, **INTERVAL** 10 **DAY**);
>> 2017-06-25

ADDTIME()
-Adiciona um intervalo de tempo a uma hora/data e retorna a hora/data e hora.
SELECT ADDTIME(“2017-06-15 09:34:21”, “2”);

CURDATE()
-Traz a data atual

CURRENT_TIMESTAMP()
-Traz a data e horário atual

DATEDIFF()
-Traz a diferença entre duas datas.
ex.
SELECT DATEDIFF(**CURRENT_TIMESTAMP(),** ‘2019-01-01’);
>>> 109

FUNÇÕES MATEMÁTICAS

https://www.w3schools.com/sql/func_mysql_sqrt.asp

ABS()
ex.
SELECT ABS(-243.5)
>>> 243.5

SELECT (23+((25-2)/2)*45) **AS** RESULTADO;

ROUND()
-Arredonda um número
-Caso se queira o próximo número inteiro usa-se **CEILING()**, para se pegar apenas o número inteiro usa-se **FLOOR()**
ex.
SELECT ROUND(12.3333333) **AS** RESULTADO;
>>> 12

RAND()
-Retorna um número aleatório.
ex.
SELECT RAND() AS RESULTADO;

CONVERSÃO DE DADOS

CONVERT()
ex. transformar um float em um char
SELECT CONVERT(23.3, CHAR) **AS** RESULTADO;