

SQL

Linguagem Estruturada de Consulta, fundada em 3 pilares:

- DDL - Data Definition Language, linguagem para criar o dicionário do meu banco de dados, para alterar as coisas fisicamente no banco de dados.
- DML - Data Manipulation Language, responsável pela manipulação dos dados fisicamente, alterando, incluindo e removendo os dados da tabela.
- DQL - Data Query Language, linguagem de consulta, extração e exibição de dados.
- Alguns autores colocam o DQL junto com o DML.

DDL

Create Table Cliente

(

Codigo number (10) Not Null Primary Key,

Nome varchar(50) Not Null,

Telefone varchar(15)

)

É possível criar uma tabela sem Primary Key, por mais que não seja muito aconselhado, e alguns bancos de dados não permitem isso, porém algumas situações se encaixam melhor com essa especificação.

DML

Insert into Cliente (Codigo, Nome, Telefone)

values (1, "Lorem ipsum", "(88) 999 9999")

Delete from Cliente

Where Codigo = 1

Update Cliente

set Nome = "Lorem Ipsum"

Where Codigo = 1

É importante a cláusula **Where**, pois sem especificar qual **tupla** vai ser alterada você pode acabar deletando ou modificando toda a tabela, sem o **Where** no **update**, todos os nomes seriam mudados, e no **delete**, todos os dados seriam apagados.

DQL

Select Codigo,

Nome

from Cliente

<Where> Codigo = 1

<Group by> Profissao

<Having> Count(1) > 0

<Order by> Nome, Codigo

É preciso uma equivalência nas colunas que você vai fazer a extração.

DIGITAL INNOVATION ONE

Query

Item_venda			
Venda	Cod_Produto	Quantidade	Valor
1	1	10	22,30
2	1	11	22,30
3	2	10	30,00

Produto		
Codigo	Descricao	Valor
1	Lapis	19,00
2	Caneta	35,00
3	Borracha	5,00

```
Select Quantidade  
      , Valor  
      , Descricao  
from Item_venda  
JOIN Produto  
ON Codigo = Cod_Produto  
Where Valor > 5
```



DIGITAL INNOVATION ONE

Funções de Conjunto

```
Select Sum(ven.Quantidade) as QTotal  
      , Sum(ven.Valor) as VTotal  
      , pro.Descricao  
from Item_venda ven  
JOIN Produto pro  
ON pro.Codigo = ven.Cod_Produto  
Where ven.Valor > 5  
Group by pro.Descricao  
Having Sum(ven.Valor) >= 10
```



Quando for usar uma função, toda coluna deve ter uma cláusula associada a ela.
A cláusula Having só pode ser usada quando tem um Group by.

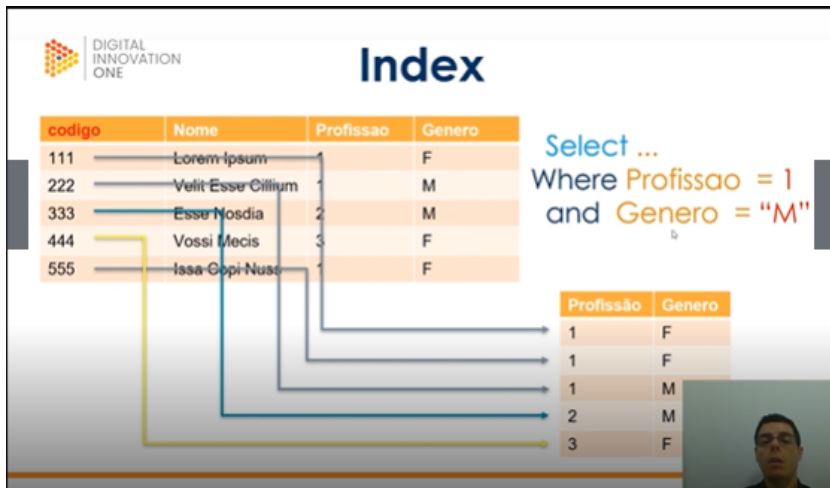


Table Access Full - Acesso Completo à Tabela.

Index - Indexadores, são usados para não haver muito uso do desempenho, são pequenas tabelas extraídas da tabela principal, que são físicas e ao mesmo tempo memória. Elas são mantidas por ordenação, ou seja, se você puxar dados referentes a uma tupla do index, você vai puxar os dados da tabela referente daquela tupla.

Deve ser analisando quantos index serão usados.

O Index afeta a memória dos comandos DML, pois cada modificação que você faz em sua tabela, você vai acabar atualizando o Index.