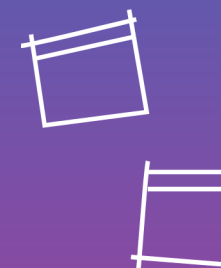




 **VEM SER**
DBC



Banco de Dados Oracle

Modelagem e Seleções



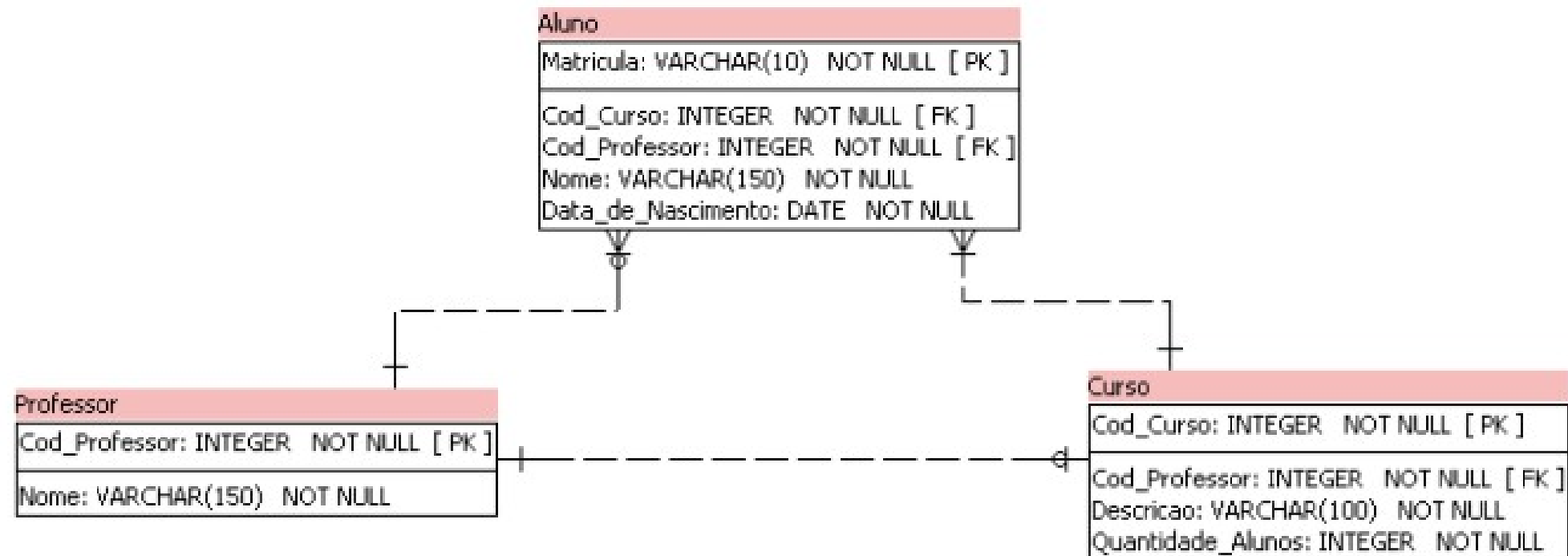
Sumário

- Modelagem de dados
- Ferramentas de modelagem
- Criar Foreign Key
- Alterar Tabelas
- Comandos de Seleção

Modelo Entidade Relacionamento (MER)

- O **MER** (Modelo Entidade Relacionamento) é utilizado para descrever os objetos do mundo real através de entidades, com suas propriedades que são os atributos e os seus relacionamentos.
- As entidades representam um **objeto** do mundo real e que possuem uma existência independente, como: pessoas, empresa, carro, casa, entre outras coisas que podem ser representadas por uma entidade.







Modelo Entidade Relacionamento (MER)



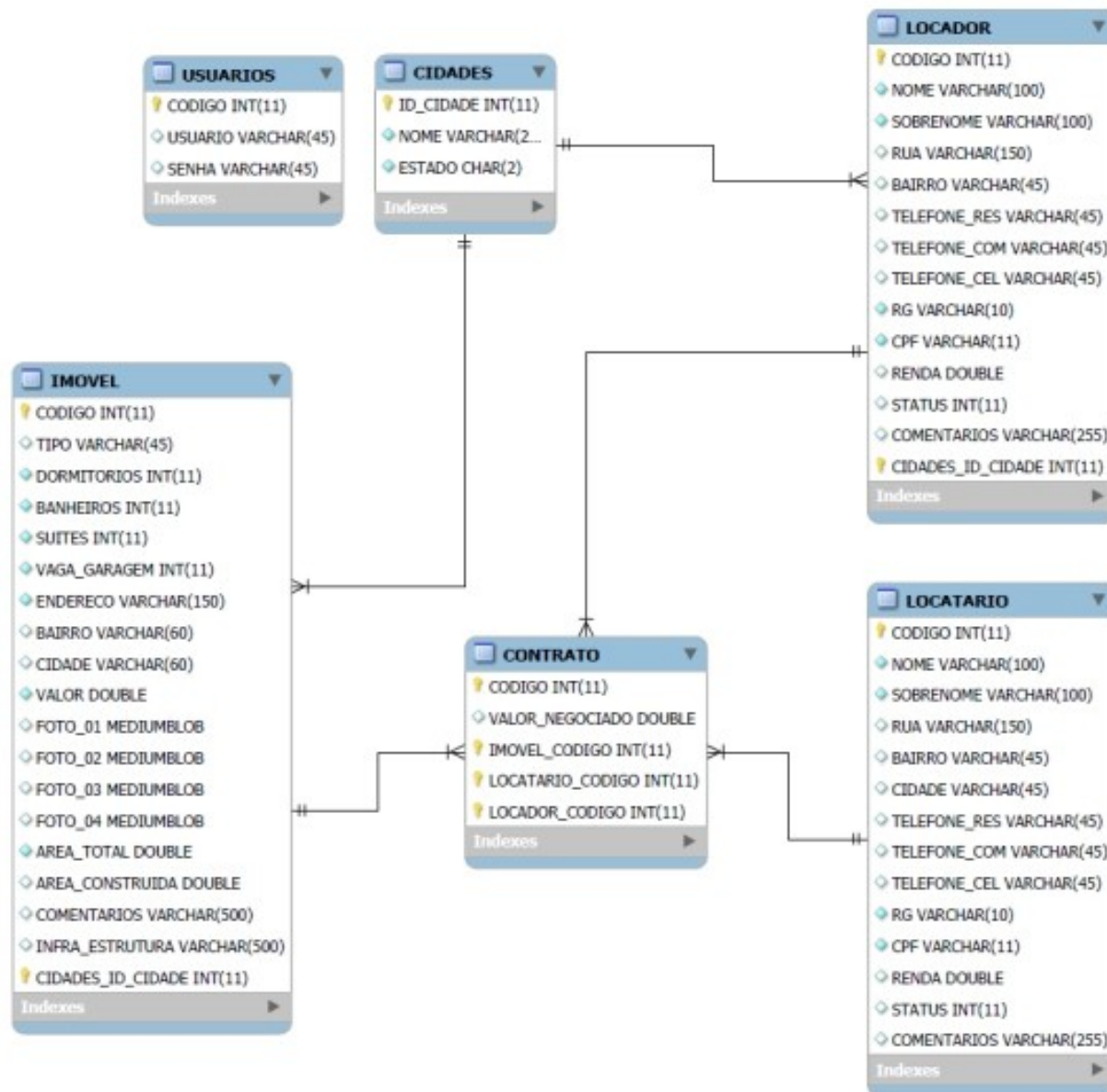
Cardinalidade ou Ordinalidade

- Cardinalidade refere-se ao número máximo de vezes que a instância em uma entidade pode ser relacionada a instâncias de outra entidade.
- Ordinalidade é o número mínimo de vezes que uma instância em uma entidade pode ser associada a uma instância em uma entidade relacionada.
- A cardinalidade e a ordinalidade são representadas pelo estilo de uma linha e sua extremidade, de acordo com o estilo de notação escolhido.

Cardinalidade ou Ordinalidade

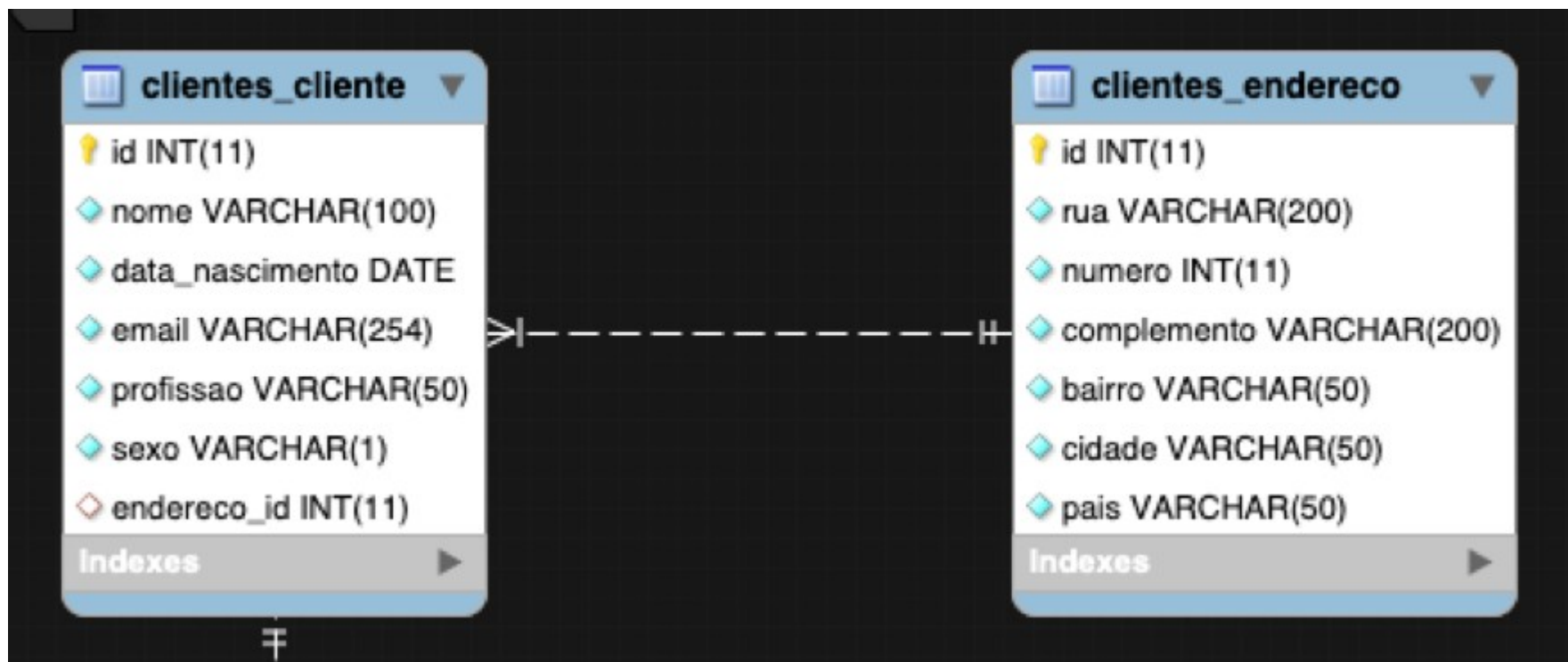
	One
	Many
	One (and only one)
	Zero or one
	One or many
	Zero or many

Cardinalidade ou Ordinalidade



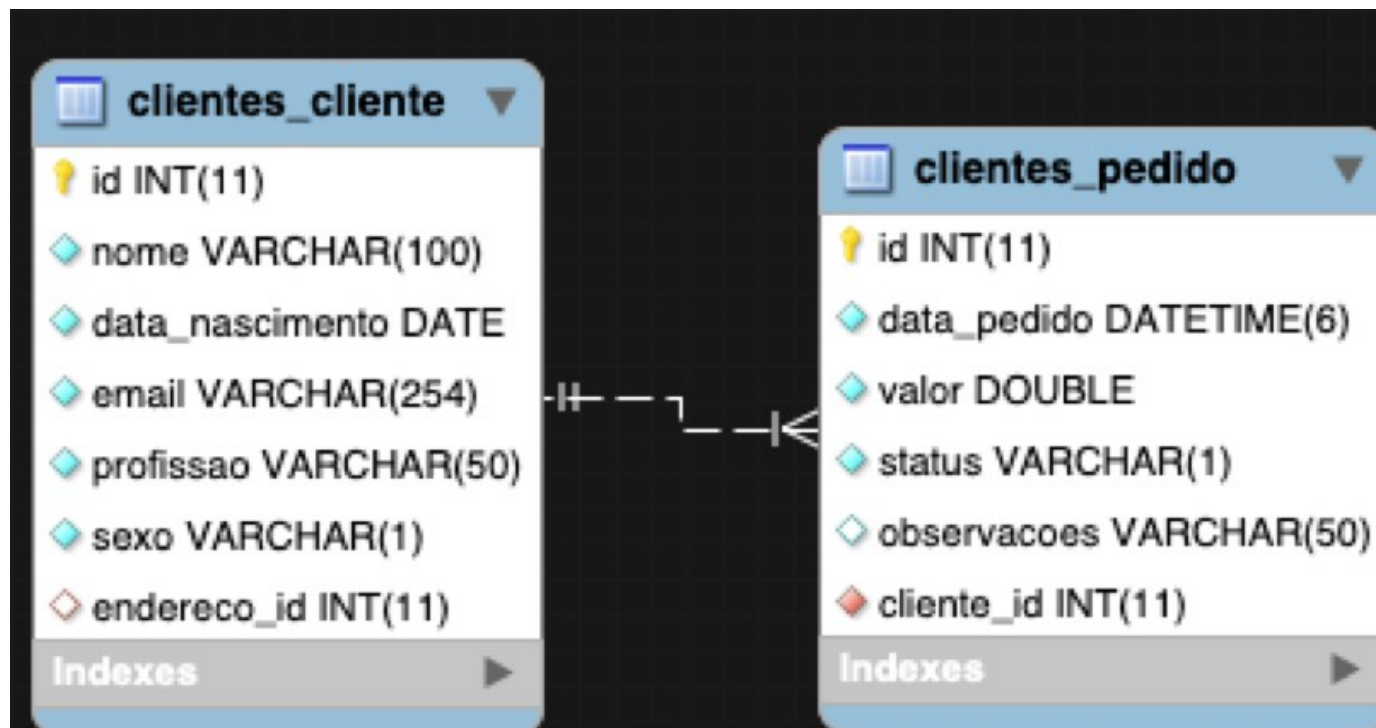
Tipos de Ligação

- **UM PARA UM (1:1):** Onde uma entidade X se associa unicamente a uma ocorrência da entidade Y.



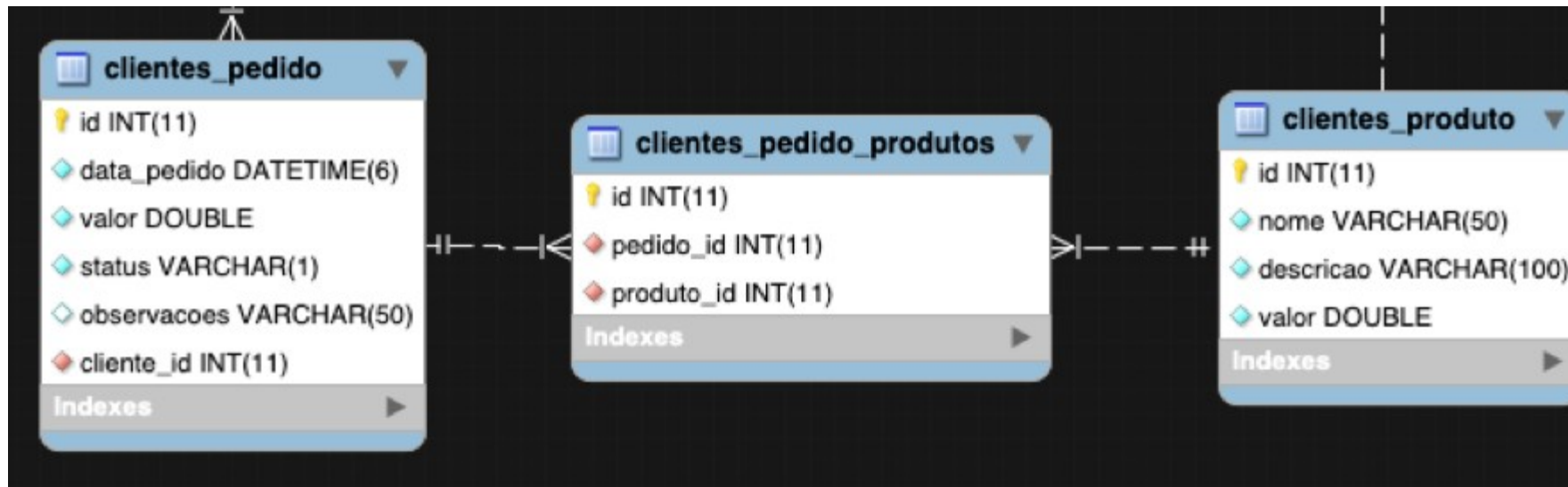
Tipos de Ligação

- **UM PARA MUITOS (1:N):** Onde uma entidade X se associa a várias ocorrências da entidade Y, porém, a entidade Y pode apenas se associar a uma ocorrência da entidade X.



Tipos de Ligação

- **MUITOS PARA MUITOS (N:N):** Onde a entidade X o pode se associar a várias ocorrências da entidade Y e a entidade Y pode também se associar a várias ocorrências da entidade X.





Ambiente

- DB Designer
 - <https://dbdesigner.softonic.com.br/download?ex=BB-1958.0>
- Lucid App
 - <https://lucid.app/>

Criando Tabelas com FK

```
CREATE TABLE VEM_SER.PAIS(
    ID_PAIS INT PRIMARY KEY NOT NULL,
    NOME VARCHAR2(50) UNIQUE NOT NULL
);
```

```
CREATE TABLE VEM_SER.ESTADO(
    ID_ESTADO INT PRIMARY KEY NOT NULL,
    ID_PAIS INT NOT NULL,
    NOME VARCHAR2(50) UNIQUE NOT NULL,
    CONSTRAINT FK_ID_PAIS FOREIGN KEY ( ID_PAIS ) REFERENCES PAIS( ID_PAIS )
);
```

Alterando Tabelas

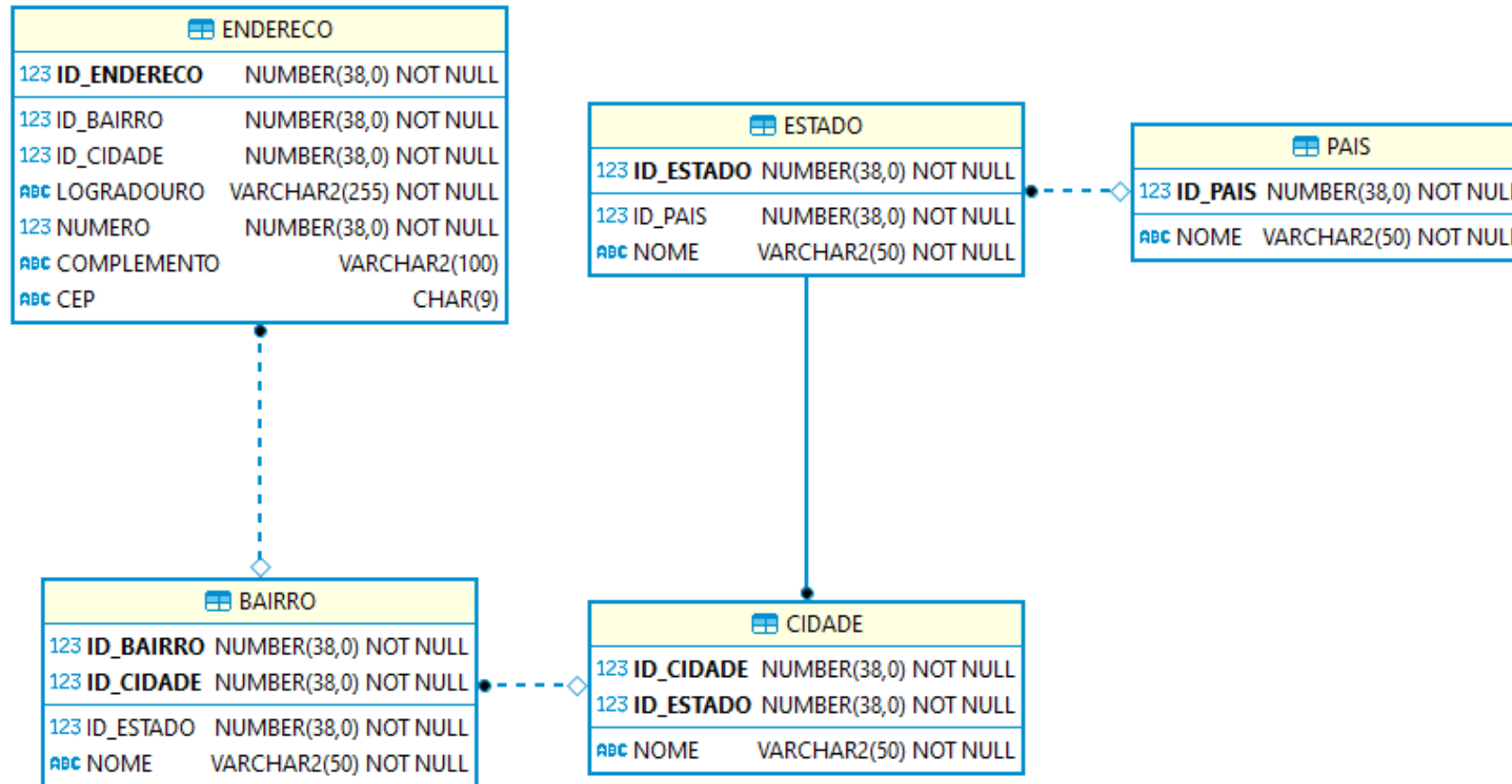
```
CREATE TABLE VEM_SER.ESTADO(
    ID_ESTADO NUMBER PRIMARY KEY NOT NULL,
    NOME VARCHAR2(50) UNIQUE NOT NULL
);
```

```
ALTER TABLE VEM_SER.ESTADO
ADD ID_PAIS INT NOT NULL;
```

```
ALTER TABLE VEM_SER.ESTADO
ADD CONSTRAINT FK_ID_PAIS FOREIGN KEY ( ID_PAIS ) REFERENCES PAIS( ID_PAIS );
```

Exercício #1

- Crie os scripts de CREATE TABLE para o seguinte diagrama de entidade e relacionamento



Exercício #2

- Criar sequences para as tabelas do exercício anterior
- Criar inserts para as tabelas do exercício anterior conforme especificação abaixo:
 - 2 países
 - 2 estados para cada país
 - 2 cidades para cada estado
 - 2 bairros para cada cidade
 - 2 endereços para cada bairro

Select – Seleção de Dados

```
SELECT <lista_de_campos>
FROM <nome_da_tabela></nome_da_tabela></lista_de_campos>
```

```
SELECT CODIGO, NOME FROM CLIENTES
```

```
SELECT * FROM CLIENTES
```

Where – Condicional

- A cláusula Where permite ao comando SQL passar condições de filtragem.

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE CODIGO = 10;
```

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE UF = 'RJ';
```

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE CODIGO >= 100 AND CODIGO <= 500;
```

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE UF = 'MG' OR UF = 'SP';
```

Where – Condicional

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE UF = 'RJ' OR (UF = 'SP' AND ATIVO = 'N')
```

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE (ENDERECO IS NULL) OR (CIDADE IS NULL)
```

Where – Filtro de Texto

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE NOME LIKE 'MARIA%'
```

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE NOME LIKE '%MARIA%'
```

```
SELECT CODIGO, NOME FROM CLIENTES
WHERE UPPER(NOME) LIKE 'MARIA %SILVA%'
```

Where – Between (entre valores)

```
SELECT coluna1, coluna2,... colunaN  
  FROM nome_tabela  
 WHERE nome_coluna BETWEEN valor_inicial AND valor_final;
```

```
SELECT nome_aluno, idade FROM alunos  
 WHERE idade NOT BETWEEN 20 AND 30;
```

Where – In (na lista de valores)

```
SELECT coluna1, coluna2,... colunaN
FROM nome_tabela
WHERE nome_coluna IN (1,2,3,9,...)
```

```
SELECT nome_aluno, idade FROM alunos
WHERE idade IN (20,21,18)
```

Group By – Agrupar

Employee

EmployeeID	Ename	DeptID	Salary
1001	John	2	4000
1002	Anna	1	3500
1003	James	1	2500
1004	David	2	5000
1005	Mark	2	3000
1006	Steve	3	4500
1007	Alice	3	3500

SELECT DeptID, AVG(Salary)
FROM Employee
GROUP BY DeptID;

GROUP BY
Employee Table
using DeptID

DeptID	AVG(Salary)
1	3000.00
2	4000.00
3	4250.00

- <https://www.datacamp.com/community/tutorials/group-by-having-clause-sql>

Having – Contém

Employee

EmployeeID	Ename	DeptID	Salary
1001	John	2	4000
1002	Anna	1	3500
1003	James	1	2500
1004	David	2	5000
1005	Mark	2	3000
1006	Steve	3	4500
1007	Alice	3	3500

SELECT DeptID, AVG(Salary)
FROM Employee
GROUP BY DeptID;

GROUP BY
Employee Table
using DeptID

DeptID	AVG(Salary)
1	3000.00
2	4000.00
3	4250.00

SELECT DeptID, AVG(Salary)
FROM Employee
GROUP BY DeptID
HAVING AVG(Salary) > 3000;

HAVING

DeptID	AVG(Salary)
2	4000.00
3	4250.00

- <https://www.datacamp.com/community/tutorials/group-by-having-clause-sql>

Funções de Agregação

- SUM(): Retorna a soma ou total de cada grupo.
- COUNT(): Retorna o número de linhas de cada grupo.
- AVG(): Retorna a média e a média de cada grupo.
- MIN(): Retorna o valor mínimo de cada grupo.
- MAX(): Retorna o valor máximo de cada grupo.

- <https://www.datacamp.com/community/tutorials/group-by-having-clause-sql>

Order By – Ordenação

```
SELECT CODIGO, NOME FROM CLIENTES  
ORDER BY NOME
```

```
SELECT CODIGO, NOME FROM CLIENTES  
ORDER BY UF, NOME DESC
```

Homework #1

- Na pasta “modulo2”, adicionar um script “homework2.sql” com a especificação abaixo:
- Utilizando os dados inseridos nas tabelas, crie os scripts abaixo:
 - Selecionar todos os países ordenados por nome decrescente;
 - Selecionar logradouro e cep dos endereços. Porém, somente os logradouros que comecem com a letra ‘a’ (maiúsculo ou minúsculo);
 - Selecionar todos os endereços que tenham cep com final ‘0’;
 - Selecionar todos os endereços que tenham números entre 1 e 100;
 - Selecionar todos os endereços que comecem por “RUA” e ordenar pelo cep de forma decrescente ;
 - Selecionar a quantidade de endereços cadastrados na tabela;
 - Selecionar a quantidade de endereços cadastrados agrupados pelo id da cidade;

Homework #2

- Utilizando uma ferramenta para desenhar diagramas de entidade e relacionamento, desenhe as tabelas e campos relacionadas ao tema da equipe.

Obrigado!

DBC

DIGITAL BUSINESS COMPANY®



 /dbc.company

 /dbccompany

 /dbccompany.com.br

 /company/dbc-company