

Banco de Dados

CENTRO UNIVERSITÁRIO NEWTON PAIVA

Prof. Dr. João Paulo Aramuni





Banco de Dados

2º período

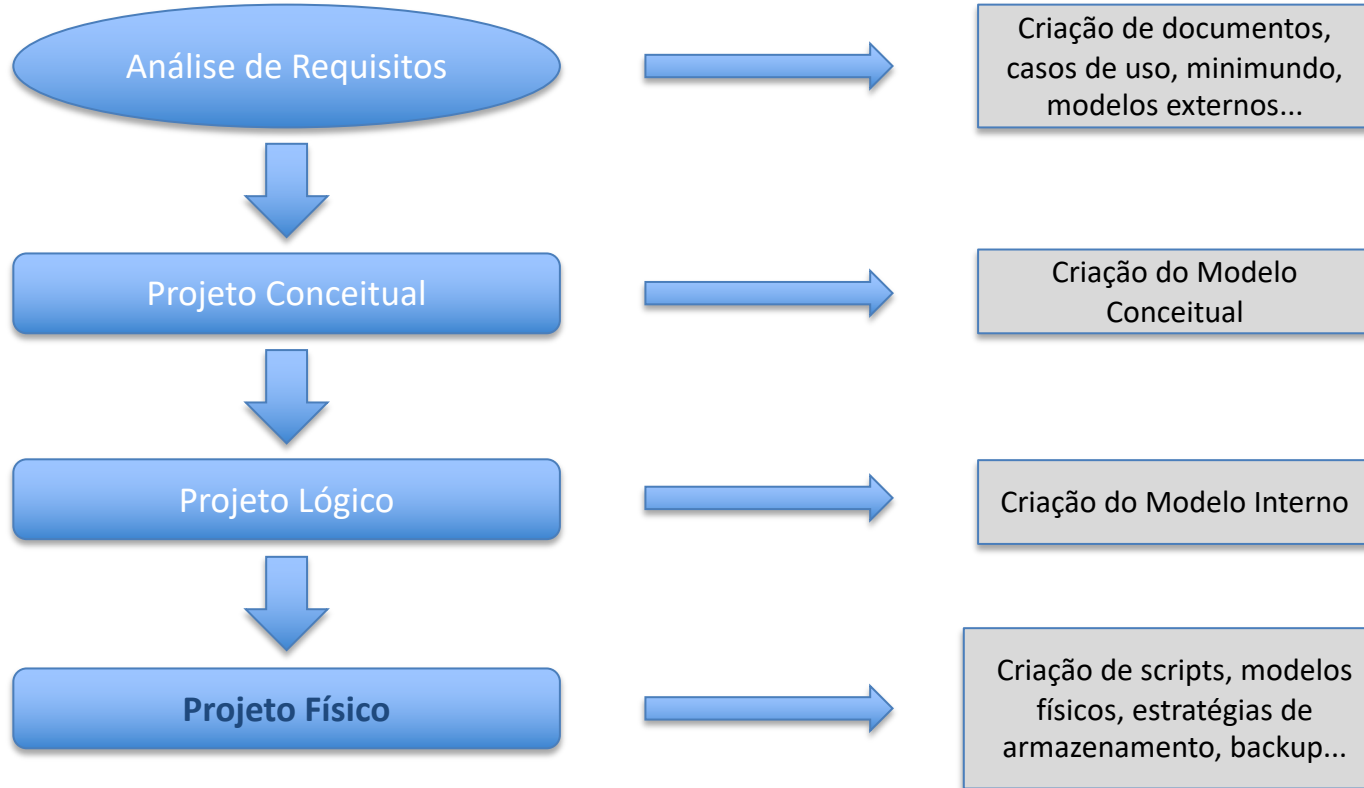
Prof. Dr. João Paulo Aramuni



Projeto Físico de Bando de Dados

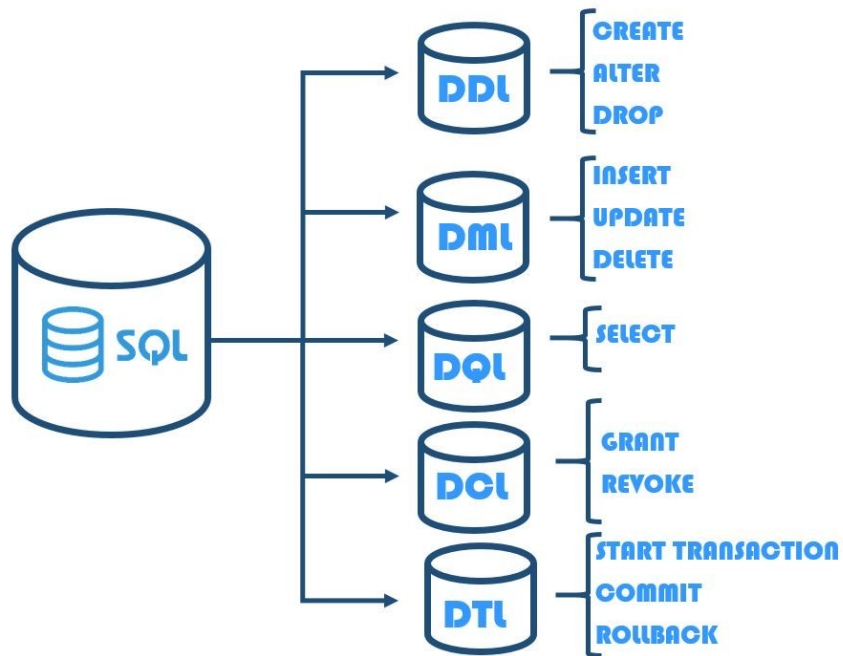
Aula 12

Projeto Físico de BD



Projeto Físico de BD

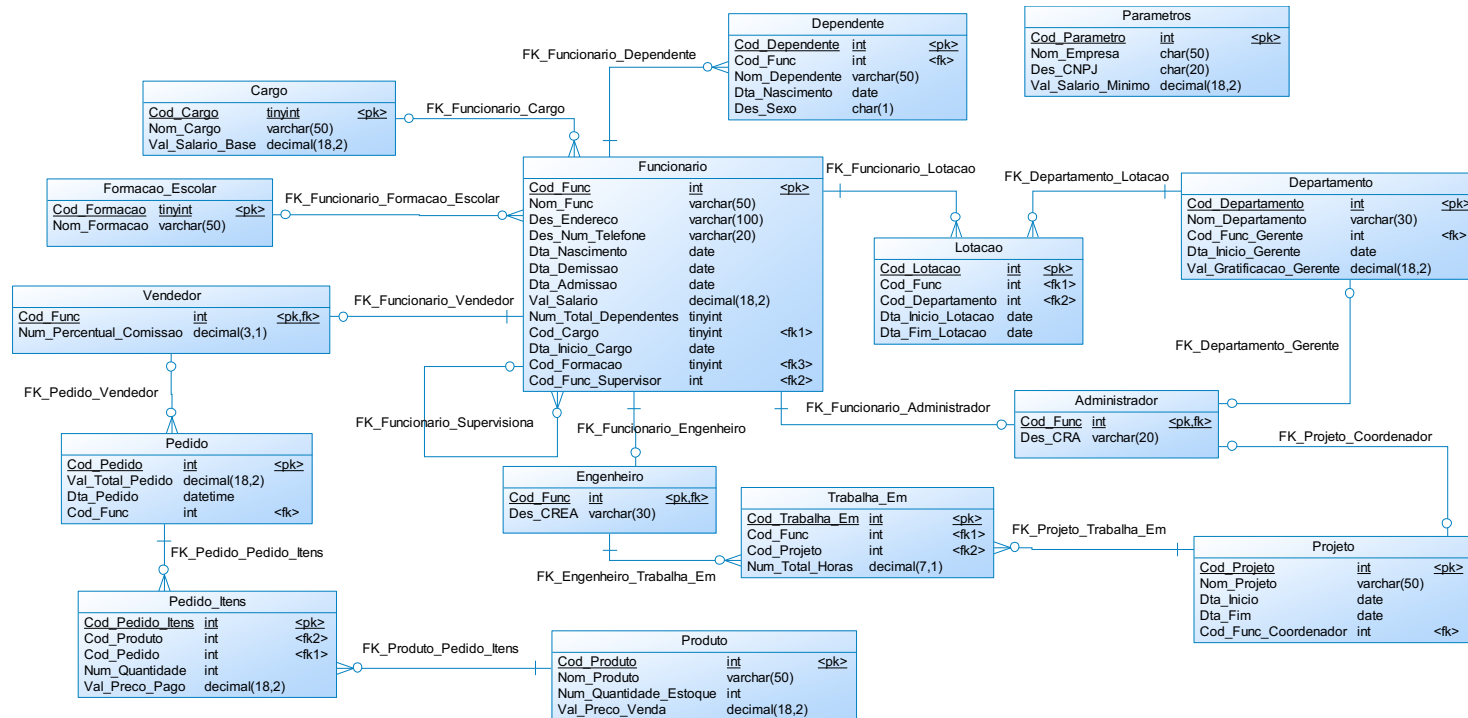
Divisão da Linguagem SQL



Projeto Físico de BD

Vamos trabalhar com o BD do DER abaixo.

Execute o script de criação do banco de dados BD_RH.



Projeto Físico de BD

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

-- Para recuperar/consultar dados

```
SELECT ...  
FROM ...  
WHERE ...  
ORDER BY ...
```

OPERADORES:

=, <, <=, >, >=, <>, AND, OR

DISTINCT/BETWEEN

IS NULL e IS NOT NULL

IN, NOT IN, LIKE

Função DATEDIFF

Projeto Físico de BD

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Revisão da última aula:

Recupere os funcionários não demitidos e que foram admitidos há mais de cinco anos e cujo salário seja maior ou igual a 3000 mil reais. Ordene os dados mostrando primeiro os funcionários com maior salário primeiro.

Projeto Físico de BD

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Funções:

YEAR()

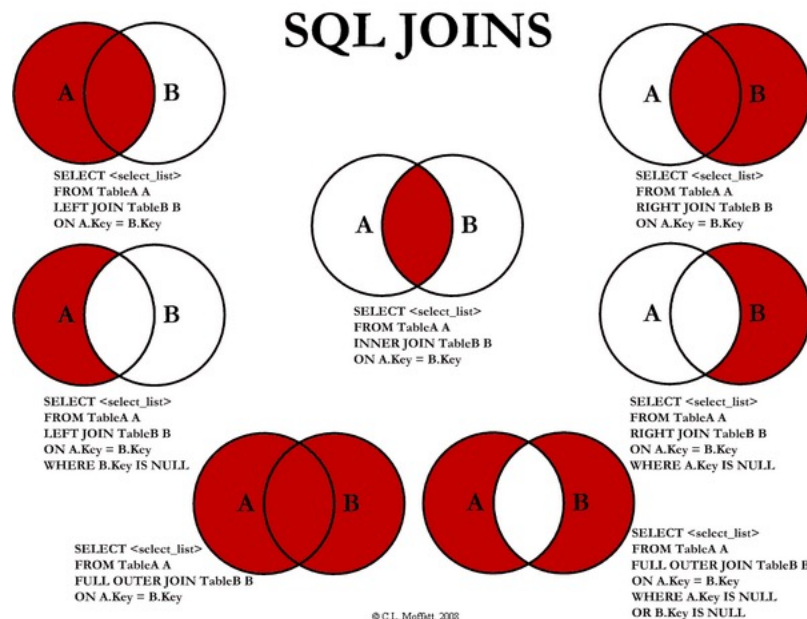
MONTH()

DAY()

Recupere os funcionários não demitidos que nasceram entre 1980 e 1999.

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:



© C.L. Moffatt, 2008

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

INNER JOIN

O INNER JOIN é um tipo de junção (join) em SQL que combina linhas de duas ou mais tabelas com base em uma condição de junção. Ele retorna apenas as linhas em que há correspondência nas duas tabelas.

Exemplo:

```
SELECT Clientes.Nome, Pedidos.Produto FROM Clientes  
INNER JOIN Pedidos ON Clientes.ID = Pedidos.ID_Cliente;
```

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

INNER JOIN

Recupere o nome dos funcionários não demitidos e o nome de seus dependentes. Ordene os dados em ordem alfabética do nome dos funcionários, seguido pelo nomes dos dependentes, também em ordem alfabética.

Recupere o número do pedido, o nome dos produtos associados ao pedido e a quantidade do produto no pedido.

Recupere o nome dos projetos e o nome dos funcionários que trabalham no projeto e o total de horas que o funcionário trabalha no projeto.

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

LEFT OUTER JOIN

O LEFT OUTER JOIN, também conhecido como LEFT JOIN, é outro tipo de junção (join) em SQL que retorna todas as linhas da tabela à esquerda (primeira tabela) e as linhas correspondentes da tabela à direita (segunda tabela) com base na condição de junção. Se não houver correspondência na tabela à direita, as colunas da tabela à direita serão preenchidas com valores nulos.

Exemplo:

```
SELECT Clientes.Nome, Pedidos.Produto FROM Clientes  
LEFT OUTER JOIN Pedidos ON Clientes.ID = Pedidos.ID_Cliente;
```

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

LEFT OUTER JOIN

Recupere o nome dos funcionários não demitidos nascidos após o ano 1960, seu cargo e o nome dos seus dependentes. Mostre todos os funcionários, mesmo que não tenha cargo definido ou não tenha dependentes.

Recupere o nome dos projetos e o nome dos funcionários que trabalham no projeto e a quantidade de horas que cada funcionário trabalhou no projeto. Mostre todos os projetos.

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

RIGHT OUTER JOIN

O RIGHT OUTER JOIN, também conhecido como RIGHT JOIN, é mais um tipo de junção (join) em SQL que retorna todas as linhas da tabela à direita (segunda tabela) e as linhas correspondentes da tabela à esquerda (primeira tabela) com base na condição de junção. Se não houver correspondência na tabela à esquerda, as colunas da tabela à esquerda serão preenchidas com valores nulos.

Exemplo:

SELECT Clientes.Nome, Pedidos.Produto **FROM** Clientes

RIGHT OUTER JOIN Pedidos **ON** Clientes.ID = Pedidos.ID_Cliente;

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

RIGHT OUTER JOIN

Recupere o nome dos funcionários e o nome de seus dependentes. Mostre todos os funcionários mesmo que não tenham dependentes. Use RIGHT OUTER JOIN.

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

FULL OUTER JOIN

O FULL OUTER JOIN, também conhecido como FULL JOIN, é outro tipo de junção (join) em SQL que retorna todas as linhas de ambas as tabelas (primeira e segunda tabelas), com base na condição de junção. Se não houver correspondência em uma das tabelas, as colunas da tabela sem correspondência serão preenchidas com valores nulos.

Exemplo:

SELECT Clientes.Nome, Pedidos.Produto **FROM** Clientes

FULL OUTER JOIN Pedidos **ON** Clientes.ID = Pedidos.ID_Cliente;

Projeto Físico de BD

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

OPERADORES DE JUNÇÃO DE TABELAS:

FULL OUTER JOIN

Recupere o nome dos projetos e o nome dos funcionários. Mostre todos os projetos e todos os funcionários cadastrados no BD.

Projeto Físico de BD

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Perceba a importância de conhecer como o BD foi projetado.

Mostre o nome dos departamentos e o nome dos funcionários lotados atualmente no departamento. Ordene o resultado pelo nome do departamento e em seguida pelo nome do funcionário.

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Desafio:

Recupere o nome do funcionário, o nome do cargo do funcionário, o nome da formação escolar do funcionário o nome do projeto e quantas horas o funcionário trabalha no projeto. Mostre todos os funcionários não demitidos e somente os projetos não encerrados.

- **DQL** (*Data Query Language*): Linguagem de Consulta de Dados.

Operador **UNION**

O **UNION** é um operador em SQL que combina os resultados de duas ou mais consultas em um único conjunto de resultados. Ele é usado para unir as linhas de resultados de consultas que possuem a mesma quantidade de colunas e tipos de dados compatíveis.

O conjunto de resultados combinado pelo **UNION** não contém duplicatas, ou seja, linhas duplicadas são eliminadas automaticamente. O **UNION** é diferente do operador **JOIN**, que é usado para combinar colunas de diferentes tabelas.

```
SELECT coluna1, coluna2, ... FROM tabela1 WHERE condição1  
UNION SELECT coluna1, coluna2, ... FROM tabela2 WHERE condição2;
```



Obrigado!

joao.aramuni@newtonpaiva.br