

Banco de Dados

Prof. Anthony Ferreira La Marca

anthony@computacao.cua.ufmt.br

SQL

- O SQL utiliza os operadores
- IS NULL
 - Verifica se o atributo é NULO
 - Retorna TRUE
- IS NOT NULL
 - Verifica se o atributo é NÃO NULO
 - Retorna TRUE
- Exemplo
 - Retornar o nome de todos os funcionários que não tenham um supervisor

SQL

- Há ocasiões em que uma busca precisa ser feita a partir de outra busca
- Chamada consultas aninhadas
- Exemplo
 - Quero saber o número dos projetos na qual possuem um funcionário com o nome 'Silva' envolvido como gerente

SQL

- Operador IN
- Compara um valor de um atributo (v) com um conjunto de valores (V)
- E avalia como TRUE caso v for um dos elementos de V
- Exemplo
 - Quero saber o número dos projetos na qual possuem um funcionário com o nome 'Silva' envolvido como gerente e o número dos projetos que possuem o funcionário com o sobrenome 'Silva' envolvido como trabalhador

SQL

- Se a consulta retorna um único valor
- Pode-se utilizar o operador = ao invés do IN
- De modo geral, uma consulta aninhada retorna uma tabela

SQL

- Permite mais de um atributo nas comparações
- Exemplo
- ```
SELECT DISTINCT FCPF
FROM TRABALHA_EM
WHERE (PNR, HORAS) IN
 (SELECT PNR, HORAS
 FROM TRABALHA_EM
 WHERE FCPF = '35719293816');
```

# SQL

- Além do IN pode-se ter o operador ALL
- Estes podem ser combinados com os operadores >, >=, <, <= e !=
- Exemplo

# SQL

- ```
SELECT PNAME, UNOME
FROM FUNCIONARIO
WHERE SALARIO > ALL
(SELECT SALARIO
FROM FUNCIONARIO
WHERE DNR = 2);
```


SQL

- Consultas aninhadas correlacionadas
- Sempre que a consulta interna referencia algum atributo de uma relação declarada na consulta externa
- Ambas as consultas são consideradas correlacionadas
- Exemplo utilizando ALIAS
 - Recuperar o nome de todos os funcionários, cujo seu dependente tem o mesmo nome e sexo

SQL

- Operador EXISTS
- Seu valor é booleano
- Sendo TRUE se o resultado da consulta aninhada tiver pelo menos uma tupla
- FALSE se não tiver tuplas
- Exemplo

SQL

- SELECT F.PNOME, F.UNOME
FROM FUNCIONARIO F
WHERE EXISTS
(SELECT * FROM DEPENDENTE D
WHERE F.PNOME = D.NOME_DEPEN
DENTE
AND F.CPF = D.FCPF
AND F.SEXO =D.SEXO);

SQL

- Recuperar o nome e sobrenome dos funcionários que não tenham dependentes
- ```
SELECT F.PNOME, F.UNOME
FROM FUNCIONARIO F
WHERE NOT EXISTS
 (SELECT * FROM DEPENDENTE D
 WHERE F.CPF = D.FCPF);
```

# SQL

- Listar os nomes dos gerentes que possuem pelo menos um dependente
- ```
SELECT F.PNOME, F.UNOME  
FROM FUNCIONARIO F  
WHERE EXISTS  
      (SELECT * FROM DEPENDENTE D  
        WHERE D.FCPF = F.CPF) AND  
      EXISTS (SELECT * FROM DEPARTAMENTO DEP  
              WHERE F.CPF = DEP.CPF_GERENTE);
```

SQL –Exemplo

- Recuperar o nome de cada funcionário que trabalha em todos os projetos controlados pelo departamento 5
- ```
SELECT F.PNOME, F.UNOME
FROM FUNCIONARIO F WHERE NOT EXISTS
((SELECT P.PROJNUMERO FROM PROJETO P
WHERE P.DNUM =5)
EXCEPT (SELECT T.PNR FROM TRABALHA_EM T
WHERE F.CPF = T.FCPF));
```

OBS: NÃO RODA NA VERSÃO GRATUITA

# SQL

- Pode-se renomear o nome dos atributos que aparece no resultado
- Operador AS
- Recuperar o nome e sobrenome de todos os funcionários que trabalham nos projetos 10, 20 e 30
  - `SELECT F.PNOME AS NOME_FUNCIONARIO,  
F.UNOME AS SOBRENOME_FUNCIONARIO  
FROM FUNCIONARIO F, PROJETO P, TRABALHA_EM T  
WHERE F.CPF = T.FCPF AND T.PNR = P.PROJNUMERO  
AND T.PNR IN (10, 20, 30);`

# SQL

- Junção em SQL
- Permite realizar as junções das relações envolvidas na consulta na clausula FROM
- Eliminando a necessidade de mistura-las na clausula WHERE junto as condições de seleção
- ```
SELECT F.PNOME, F.UNOME, F.ENDERECO  
      FROM (FUNCIONARIO F INNER JOIN DEPAR  
TAMENTO D ON (F.DNR = D.DNUMERO))  
      WHERE D.DNOME LIKE '%PESQUISA%';
```


SQL

- Junção Natural
- ```
SELECT F.PNOME, F.UNOME, F.ENDERECO
FROM (FUNCIONARIO F NATURAL JOIN
 DEPARTAMENTO D)
WHERE D.DNOME LIKE '%PESQUISA%';
```
- Obs: Os atributos de junção devem ter o mesmo nome

# SQL

- Temos também o OUTER JOIN
  - LEFT OUTER JOIN
  - RIGHT OUTER JOIN
  - FULL OUTER JOIN

# Junção – Outro Exemplo

empregado

| nome_empregado | rua            | cidade        |
|----------------|----------------|---------------|
| José           | João Goulart   | Novo Hamburgo |
| Ana            | Assis Brasil   | Porto Alegre  |
| Pedro          | Getúlio Vargas | São Leopoldo  |
| Maria          | João Pessoa    | Porto Alegre  |

trabalhador\_integral

| nome_empregado | nome_agencia | salario |
|----------------|--------------|---------|
| José           | NOH-1        | 5000    |
| Ana            | POA-1        | 4800    |
| Flávia         | SAL-1        | 3200    |
| Maria          | POA-1        | 6500    |

# Junção

- Descobrir o nome, a rua, a cidade, a agência e o salário de todos os empregados
- Junção Natural

| nome_empregado | rua          | cidade        | nome_agencia | salario |
|----------------|--------------|---------------|--------------|---------|
| José           | João Goulart | Novo Hamburgo | NOH-1        | 5000    |
| Ana            | Assis Brasil | Porto Alegre  | POA-1        | 4800    |
| Maria          | João Pessoa  | Porto Alegre  | POA-1        | 6500    |

- Problema
  - Os dados de Pedro foram perdidos (nome, rua e cidade)
  - Os dados de Flávia também (nome, nome\_agencia e salario)

# Junção Externa à Esquerda

- Pega as tuplas da relação da esquerda que não se encontram par entre as tuplas da relação à direita
- Preenche com nulo os valores
- Adicionadas ao resultado da Junção Natural

| nome_empregado | rua            | cidade        | nome_agencia | salario |
|----------------|----------------|---------------|--------------|---------|
| José           | João Goulart   | Novo Hamburgo | NOH-1        | 5000    |
| Ana            | Assis Brasil   | Porto Alegre  | POA-1        | 4800    |
| Maria          | João Pessoa    | Porto Alegre  | POA-1        | 6500    |
| Pedro          | Getúlio Vargas | São Leopoldo  | Nulo         | Nulo    |

# Junção Externa à Direita

- Pega as tuplas da relação da direita que não se encontram par entre as tuplas da relação à esquerda
- Preenche com nulo os valores
- Adicionadas ao resultado da Junção Natural

| nome_empregado | rua          | cidade        | nome_agencia | salario |
|----------------|--------------|---------------|--------------|---------|
| José           | João Goulart | Novo Hamburgo | NOH-1        | 5000    |
| Ana            | Assis Brasil | Porto Alegre  | POA-1        | 4800    |
| Maria          | João Pessoa  | Porto Alegre  | POA-1        | 6500    |
| Flávia         | Nulo         | Nulo          | SAL-1        | 3200    |

# Junção Externa Total

- Faz ambas as operações

| nome_empregado | rua            | cidade        | nome_agencia | salario |
|----------------|----------------|---------------|--------------|---------|
| José           | João Goulart   | Novo Hamburgo | NOH-1        | 5000    |
| Ana            | Assis Brasil   | Porto Alegre  | POA-1        | 4800    |
| Maria          | João Pessoa    | Porto Alegre  | POA-1        | 6500    |
| Pedro          | Getúlio Vargas | São Leopoldo  | Nulo         | Nulo    |
| Flávia         | Nulo           | Nulo          | SAL-1        | 3200    |

# SQL

- Vamos imaginar essa consulta
- ```
SELECT F.PNOME, F.UNOME  
FROM FUNCIONARIO F, FUNCIONARIO S  
WHERE F.CPF_SUPERVISOR = S.CPF;
```
- Mostrar todos mesmo não tendo um supervisor
- ```
SELECT F.PNOME, F.UNOME
FROM FUNCIONARIO F LEFT OUTER JOIN
FUNCIONARIO S ON (F.CPF_SUPERVISOR =
S.CPF);
```



# SQL

- LEFT
  - Toda tupla da esquerda tem que aparecer, mesmo se ela não tiver uma tupla combinando, ela é preenchida com valores NULL para a tabela da direita
- RIGHT
  - Toda tupla da direita tem que aparecer, mesmo se ela não tiver uma tupla combinando, ela é preenchida com valores NULL para a tabela da esquerda
- FULL
  - Aparece todas as tuplas da tabela da esquerda e da direita, mesmo não tendo uma tupla combinando, elas devem ser preenchidas com valores NULL, ambas as tabelas

# Exercício

- Recuperar o nome dos alunos e curso que não tenham uma nota A em quaisquer disciplinas
- Recuperar o nome e curso de todos os alunos com nota A em todas as disciplinas

# SQL

- Funções agregadas
- Resume informações de várias tuplas em uma síntese de uma única tupla
  - COUNT
  - SUM
  - MAX
  - MIN
  - AVG
- O agrupamento é usado para criar subgrupos de tuplas antes do resumo
  - GROUP BY

# SQL

- Achar a soma dos salários de todos os funcionários que trabalham no departamento “pesquisa”, bem como o salário máximo, o salário mínimo e a média dos salário
- ```
SELECT SUM(SALARIO), MAX(SALARIO),  
       MIN(SALARIO), AVG(SALARIO)  
FROM (FUNCIONARIO JOIN DEPARTAMENTO  
      ON DNR = DNUMERO)  
WHERE DNOME LIKE '%PESQUISA%';
```

SQL

- Recuperar o número total de funcionários da empresa
- `SELECT COUNT(*) FROM FUNCIONARIO;`
- Quando utilizado a função COUNT com o *, retorna a quantidade de tuplas
- Pode-se utiliza também para contar os valores de um determinado atributo (coluna)

SQL

- Contar quantos salários distintos há na empresa
- `SELECT COUNT(DISTINCT SALARIO)`
`FROM FUNCIONARIO;`
- Com a cláusula `DISTINCT` os valores duplicados serão eliminados
- Valores `NULL` são descartados
- As funções agregadas também são utilizadas em consultas aninhadas

SQL

- Recuperar o nome dos funcionários que tem 2 ou mais dependentes
- ```
SELECT F.PNOME FROM FUNCIONARIO F
WHERE (SELECT COUNT(*)
FROM DEPENDENTE D WHERE F.CPF =
D.FCPF) >= 2;
```

# SQL

- Aplicar a agregação em subgrupos
- GROUP BY
- HAVING
- Encontrar a média salarial dos funcionários que trabalham em cada departamento
- Encontrar o número de funcionários que trabalham em cada projeto



# SQL

- Para cada departamento, recuperar o número do departamento, o número de funcionário no departamento e o seu salário médio
- ```
SELECT DNR, COUNT(*), AVG(SALARIO)
FROM FUNCIONARIO
GROUP BY DNR;
```

SQL

- Resultado da consulta

Pnome	Minicial	Unome	Cpf	...	Salario	Cpf_supervisor	Dnr		Dnr	Count (*)	Avg (Salario)
João	B	Silva	12345678966		30.000	33344555587	5	}	5	4	33.250
Fernando	T	Wong	33344555587		40.000	88866555576	5		4	3	31.000
Ronaldo	K	Lima	66688444476		38.000	33344555587	5		1	1	55.000
Joice	A	Leite	45345345376	...	25.000	33344555587	5	}	Resultado de C24		
Alice	J	Zelaya	99988777767		25.000	98765432168	4				
Jennifer	S	Souza	98765432168		43.000	88866555576	4				
André	V	Pereira	98798798733		25.000	98765432168	4				
Jorge	E	Brito	88866555576		55.000	NULL	1	}			

Agrupamento de tuplas FUNCIONARIO pelo valor de Dnr

SQL

- Para cada projeto, recuperar o número e nome do projeto e o número de funcionários que trabalham nesse projeto
- ```
SELECT P.PROJNUMERO, P.PROJNOME, COUNT(*)
FROM (PROJETO P JOIN TRABALHA_EM T
ON (P.PROJNUMERO = T.PNR))
GROUP BY P.PROJNUMERO;
```
- O agrupamento e as funções são aplicadas após as junções das relações envolvidas na consulta

# SQL

- O operador HAVING oferece uma condição sobre o grupo de tuplas associado a cada valor dos atributos de agrupamento
- Somente os grupos que satisfazem a condição são recuperados no resultado da consulta
- Para cada projeto em que mais de dois funcionários trabalham, recupere o nome do projeto e o número de funcionários que trabalham no projeto, ordenado pelo nome do projeto

# SQL

- ```
SELECT PROJNOME, COUNT(*)  
FROM PROJETO, TRABALHA_EM  
WHERE PNR = PROJNUMERO  
GROUP BY PROJNOME  
HAVING COUNT(*) > 2  
ORDER BY PROJNOME;
```

SQL

Projnome	Projnumero	...	Ecpf	Pnr	Horas
ProdutoX	1		12345678966	1	32,5
ProdutoX	1		45345345376	1	20,0
ProdutoY	2		12345678966	2	7,5
ProdutoY	2		45345345376	2	20,0
ProdutoY	2		33344555587	2	10,0
ProdutoZ	3		66688444476	3	40,0
ProdutoZ	3		33344555587	3	10,0
Informatização	10	...	33344555587	10	10,0
Informatização	10		99988777767	10	10,0
Informatização	10		98798798733	10	35,0
Reorganização	20		33344555587	20	10,0
Reorganização	20		98765432168	20	15,0
Reorganização	20		88866555576	20	NULL
Novos Benefícios	30		98798798733	30	5,0
Novos Benefícios	30		98765432168	30	20,0
Novos Benefícios	30		99988777767	30	30,0

Estes grupos não são seleccionados pela condição de HAVING de C26.

Após aplicar a cláusula WHERE, mas antes de aplicar HAVING

SQL

Projnome	Projnumero	...	Fcpf	Pnr	Horas		Projnome	Count (*)
ProdutoY	2		12345678966	2	7,5	→	ProdutoY	3
ProdutoY	2		45345345376	2	20,0		Informatização	3
ProdutoY	2		33344555587	2	10,0		Reorganização	3
Informatização	10		33344555587	10	10,0	→	Novos Benefícios	3
Informatização	10	...	99988777767	10	10,0		Resultado de C26 (Projnumero não mostrado)	
Informatização	10		98798798733	10	35,0			
Reorganização	20		33344555587	20	10,0			
Reorganização	20		98765432168	20	15,0	→		
Reorganização	20		88866555576	20	NULL			
Novos Benefícios	30		98798798733	30	5,0			
Novos Benefícios	30		98765432168	30	20,0	→		
Novos Benefícios	30		99988777767	30	30,0			

Após aplicar a condição da cláusula HAVING

SQL

- Para cada departamento que tem mais de 2 funcionários, recuperar o número do departamento e o número de seus funcionários que estão ganhando mais de 5000 mil

```
SELECT DNOME  
FROM DEPARTAMENTO, FUNCIONARIO  
WHERE DNUMERO = DNR  
AND SALARIO > 5000  
GROUP BY DNOME  
HAVING COUNT(*) > 2;
```


SQL

- Consulta errada
- Pois só retornará os departamentos que tenham mais do que 2 funcionários que ganham mais do que 5mil
- A pergunta é
- Os departamento que tenham mais do que 2 funcionário, retornar os que ganham mais de 5 mil

SQL

- ```
SELECT DNOME,PNOME
FROM DEPARTAMENTO, FUNCIONARIO
WHERE DNUMERO = DNR
AND SALARIO > 5000
AND DNUMERO IN (SELECT DNR FROM
FUNCIONARIO GROUP BY DNR
HAVING COUNT(*) > 2);
```

# Exercícios

- Para cada departamento, recupere o nome do departamento e a média do salario do departamento, onde a média seja maior que 3200 reais
- Para cada departamento, recupere o nome do departamento, o nome do funcionário e seu salário que seja maior que a media de todos os salários
- Para cada departamento, recupere o nome do departamento e a quantidade de funcionários onde o salário seja maior que a média do salario de todos os departamentos

# Exercícios

- Recuperar o numero do departamento e o CPF do funcionário do sexo feminino que ganha mais de 2000 reais para cada departamento
- Recuperar o nome de todos os funcionários que trabalham no departamento que tem o funcionário com o maior salário entre todos os funcionários
- Recupere o nome dos funcionários que ganham pelo menos 1000 reais a mais que o funcionário que recebe menos na empresa