SQL
III - Consulta

Disciplina: Laboratório de Banco de Dados

Select

- SELECT ... FROM ... WHERE ...
  - Lista atributos de uma ou mais tabelas de acordo com alguma condição.

Select

SELECT <lista de atributos e funções>
FROM <lista de tabelas>
[WHERE predicado]
[GROUP BY <atributos de agrupamento>]
[HAVING <condição para agrupamento>]
[ORDER BY <lista de atributos>] ;

- Cláusula SELECT
  - Lista os atributos e/ou as funções a serem exibidos no resultado da consulta.
- Cláusula FROM
  - Especifica as relações a serem examinadas na avaliação da consulta.
- Cláusula WHERE
  - Especifica as condições para a seleção das tuplas no resultado da consulta;
    - As condições devem ser definidas sobre os atributos das relações que aparecem na cláusula FROM.
  - Pode ser omitida.

  Para mais informações acessem:
  http://www.postgresql.org/docs/9.3/static/sql-select.html

Select

```
SELECT * FROM “EMPREGADO”;
SELECT * FROM PUBLIC.“EMPREGADO”;
SELECT * FROM “empregado”;
SELECT * FROM “public”.EMPREGADO;
SELECT * FROM public.empregado;
SELECT * FROM empregado;
```

Select

```
SELECT datanasc, endereco
FROM empregado
WHERE Pnome = 'Alicia' AND Unome= 'Zelaya';

SELECT Pjnome, Plocalizacao
FROM projeto
WHERE Pnumero = ‘10' AND Dnum = 4;
```

Select

- Ordem de apresentação dos atributos:
  - SELECT
- Duas ou mais tuplas podem possuir valores idênticos de atributos;
  - Eliminação de tuplas duplicadas;
    - SELECT DISTINCT
- Cláusula ORDER BY
  - define a ordem de apresentação dos dados;
  - cada campo pode conter ordem ascendente (ASC) ou descendente (DESC);

Order By

Exemplo:

```
SELECT nome_dependente, sexo, datanasc, parentesco
FROM dependente
ORDER BY sexo DESC, nome_dependente ASC
```

```
SELECT superssn, pnome, sexo, salario
FROM empregado
ORDER BY superssn DESC, pnome
```

Select

- Operadores:
  - Conjunção de condições: AND
  - Disjunção de condições: OR
  - Negação de condições: NOT
  - =, <>, >, <, >=, <=
  - Entre dois valores: BETWEEN... AND
  - Compara cadeias de caracteres: LIKE, ILIKE ou NOT LIKE
    - *% (porcentagem): substitui qualquer string*
    - *_ (underscore): substitui qualquer caractere*
      - WHERE Pnome LIKE ‘ Jo%’
        - » qualquer string que se inicie com ‘ Jo ’
      - WHERE Pnome LIKE ‘ Joh_’
        - » qualquer string de 4 caracteres que se inicie com ‘Joh’

Select

```
SELECT Pnome, SSN
FROM empregado
WHERE Pnome LIKE ‘Jo%’ ;

SELECT Pnome, SSN
FROM empregado
WHERE Pnome LIKE ‘Joh_’ ;
```

Select

```
SELECT Pnome, SSN
FROM empregado
WHERE Pnome ILIKE ‘Jo%’ AND
        Sexo = ‘F’ AND
        Dno = 5;
```

Operações sobre conjuntos

- Operações sobre conjuntos:
  - União: UNION
  - Intersecção: INTERSECT
  - Diferença: EXCEPT

Mais informações acessem:

http://www.postgresql.org/docs/9.4/static/queries-union.html

Union

- Exemplo:
  - Listar o nome, sexo e data de nascimento de todos os empregados e dependentes que nasceram após 1945

```
SELECT Pnome
FROM empregado
WHERE datanasc > ’01/01/1945’
UNION ALL
SELECT nome_dependente
FROM dependente
WHERE datanasc > ’01/01/1945’
```

Union

- Exemplo:
  - Listar o nome e o tipo ‘P’ projeto e ‘D’ departamento de todos os departamentos com identificação 5 ou 4 e projetos que estejam alocados nestes departamentos

```
SELECT Dnome as nome, ‘D’ as tipo
FROM departamento
WHERE DNUMERO = 5 OR DNUMERO = 4
UNION
SELECT Pjnome as nome, ‘P’ as tipo
FROM projeto
WHERE DNUM = 5 OR DNUM = 4
```

Intersect

– INSERT INTO dependente values
(333445555, ‘Alicia’ , ‘F’ , ‘2013 -05- 27’ , ‘Neta’);

- Exemplo:
  – Liste os nomes dos dependentes que tem nome igual ao de algum empregado.

```
SELECT Nome_dependente
FROM dependente
INTERSECT
SELECT Pnome
FROM empregado
```

Except

- Exemplo:
  - Liste os nomes dos empregados que não têm dependentes.

```
SELECT Pnome
FROM empregado
EXCEPT
SELECT Pnome
FROM empregado, dependente
WHERE empregado.Ssn = dependente.Essn;
```

Exercício

- Liste os nomes dos empregados que não têm filhos (filho e/ou filha).

Exercício - Resposta

- Liste os nomes dos empregados que não têm filhos (filho e/ou filha).

```
SELECT Pnome
FROM empregado
EXCEPT
SELECT Pnome
FROM empregado, dependente
WHERE empregado.Ssn =
dependente.Essn AND
dependente.parentesco ILIKE 'Filh_';
```

Operações sobre conjuntos

- UNION (R U S)
  - Une todas as linhas selecionadas por duas consultas, eliminando as linhas duplicadas;
  - Gera uma relação que contém todas as tuplas pertencentes a R, a S, ou a ambas R e S.
- UNION ALL
  - Une todas as linhas selecionadas por duas consultas, inclusive as linhas duplicadas.
- INTERSECT (R n S)
  - Gera uma relação que contém todas as tuplas pertencentes tanto a R quanto a S.
- EXCEPT (R - S)
  - Gera uma relação que contém todas as tuplas pertencentes a R que não pertencem a S.

Subconsultas aninhadas

- Subconsulta
  - Expressão SELECT ... FROM ... WHERE ... aninhada dentro de outra consulta.
- Aplicações mais comuns:
  - Testes para membros de conjuntos;
  - Comparações de conjuntos;
  - Cardinalidade de conjuntos.
- IN
  - Testa se um atributo ou uma lista de atributos é membro do conjunto.
- NOT IN
  - Verifica a ausência de um membro em um conjunto.
- Conjunto:
  - Coleção de valores produzidos por uma cláusula SELECT ... FROM ... WHERE ...

Subconsultas aninhadas

- Exemplo:
  - Liste os nomes dos dependentes que tem nome igual a de algum empregado.

```
SELECT Nome_dependente
FROM dependente
WHERE Nome_dependente IN
        (SELECT Pnome FROM empregado)
```

Comparação de conjuntos

- SOME
  - ... WHERE salario > SOME (lista)
    - A condição é verdadeira quando salario for maior que algum dos resultados presentes na lista (resultado de uma consulta).

Comparação de conjuntos

- Exemplo:
  - Liste os nomes do empregados que têm salário superior a algum empregado do departamento 4.

```
SELECT Pnome
FROM empregado
WHERE salario > SOME
 (SELECT salario
 FROM empregado
 WHERE Dno = 4);
```

Comparação de conjuntos

- ALL
  - ... WHERE salario > ALL (lista)
    - A condição é verdadeira quando salario for maior que todos os resultados presentes na lista (resultado de uma consulta).

Comparação de conjuntos

- Exemplo:
  - Liste os nomes do empregados que têm salário superior ao salário de todos os empregados do departamento 4.

```
SELECT Pnome
FROM  empregado
WHERE salario > ALL
 (SELECT salario
 FROM empregado
 WHERE Dno = 4);
```

Cardinalidade de conjuntos

- EXISTS
  - ... WHERE EXISTS (lista)
    - A condição é verdadeira quando a lista (resultado de uma consulta) não for vazia.
- NOT EXISTS
  - ... WHERE NOT EXISTS (lista)
    - A condição é verdadeira quando a lista for vazia.

Cardinalidade de conjuntos

- Exemplo:
  – Liste o nome e número dos departamentos que não possuem projetos associados.

```
SELECT Dnome, dnumero
FROM departamento
WHERE NOT EXISTS
        (SELECT Dnum
        FROM projeto
        WHERE projeto.Dnum = departamento.Dnumero)
```

Junção

- Idéia:
  - Concatenar tuplas relacionadas de duas relações em tuplas únicas.
- Passos:
  - Formar um produto cartesiano das relações;
  - Fazer uma seleção forçando igualdade sobre os atributos que aparecem nas relações.

Junção

- Usar SELECT e WHERE
    - Atributos com mesmo nome são especificados usando nomes de tabelas e atributos (nome_tabela.nome_atributo).
- Cláusula FROM
  - Possui mais do que uma tabela.
- Cláusula WHERE
  - Inclui as condições de junção.

Junção

1. Nome do empregado juntamente com nome do seu departamento:

```
SELECT Pnome, Dnome
FROM empregado, departamento
WHERE empregado.Dno =
      departamento.Dnumero;
```

2. Nome do empregado, nome do seu departamento e a localização do departamento:

```
SELECT Pnome, Dnome, Dlocalizacao
FROM  empregado, departamento,  depto_localizacoes
WHERE empregado.Dno =
      departamento.Dnumero AND
      depto_localizacoes. Dnumero =
      empregado.Dno;
```

Junção

3. Nome do empregado, nome do seu departamento e seus dependentes:

```
SELECT Pnome, Dnome, Nome_dependente
FROM empregado,
     departamento,
     dependente
WHERE empregado.Dno =
     departamento.Dnumero AND
     dependente.Essn=
     empregado.Ssn;
```

Join

1. Nome do empregado juntamente com o nome do seu departamento:

```
SELECT Pnome, Dnome
FROM empregado, departamento
WHERE empregado.Dno =
      departamento.Dnumero;
```

```
SELECT Pnome, Dnome
FROM empregado
 JOIN  departamento  ON empregado.Dno = departamento.Dnumero;
```

```
SELECT e.Pnome, d.Dnome
FROM empregado e
 JOIN  departamento d   ON e.Dno = d.Dnumero;
```

Join

3. Nome do empregado, seu departamento e seus dependentes:

```
SELECT Pnome, Dnome, Nome_dependente
FROM empregado
    JOIN departamento
        ON empregado.Dno =
        departamento.Dnumero
    JOIN dependente
        ON dependente.Essn =
        empregado.Ssn;
```

As

- Renomear:
  - Atributos
    - Deve aparecer na cláusula SELECT;
    - Útil para a visualização das respostas na tela.
  - Relações
    - Deve aparecer na cláusula FROM;
    - Útil quando a mesma relação é utilizada mais do que uma vez na mesma consulta.
- Sintaxe
  - nome_antigo AS nome_novo

As

- Nome do empregado, nome do seu departamento e seus dependentes:

```
SELECT Pnome AS nome_empregado,
        Dnome AS nome_departamento,
        Nome_dependente AS dependente
FROM empregado AS emp,
        departamento AS depto,
        dependente AS dep
WHERE emp.Dno = depto.Dnumero AND dep.Essn=
        emp.Ssn;
```

Order by

- Ordena as tuplas resultantes de uma consulta:
  - ASC: ordem ascendente (padrão);
  - DESC: ordem descendente.
- Ordenação pode ser especificada em vários atributos:
  - Ordenação referente ao primeiro atributo é prioritária;
  - Se houver valores repetidos, então é utilizada a ordenação referente ao segundo atributo, e assim por diante.

Exercício

- Liste os dados dos empregados e seus departamentos. Ordene o resultado pelo nome do departamento.

Exercício - Resposta

- Liste os dados dos empregados e seus departamentos. Ordene o resultado pelo nome do departamento.

```
SELECT *
FROM empregado,
departamento
WHERE empregado.Dno =
departamento.Dnumero
ORDER BY Dnome;
```

Order by

- Se houver valores repetidos, então é utilizada a ordenação referente ao segundo atributo, e assim por diante.

```
SELECT Pnome, Minicial, Unome, Dnome
FROM empregado, departamento
WHERE empregado.Dno =
departamento.Dnumero
ORDER BY Dnome;
```

```
SELECT Pnome, Minicial, Unome, Dnome
FROM empregado, departamento
WHERE empregado.Dno =
departamento.Dnumero
ORDER BY Dnome, Pnome;
```

Funções de agregação

- Recebem uma coleção de valores como entrada;
- Retornam um único valor como saída.
  - Média: AVG( )
  - Mínimo: MIN( )
  - Máximo: MAX( )
  - Total: SUM( )
  - Contagem: COUNT( )
  -
- Obs.:
  - DISTINCT: não considera valores duplicados
  - ALL: inclui valores duplicados

Exercício

- Qual a média dos salários dos empregados?
- Qual a soma dos salários dos empregados?
- Qual é o salário mais baixo dos salários dos empregados?
- Qual é o salário mais alto dos salários dos empregados?

Exercício - Resposta

- Qual a média dos salários dos empregados?

  SELECT AVG(salario)
  FROM empregado

- Qual a soma dos salários dos empregados?

- Qual é o salário mais baixo dos salários dos empregados?

- Qual é o salário mais alto dos salários dos empregados?

Exercício - Resposta

- Qual a média dos salários dos empregados?
- Qual a soma dos salários dos empregados?

```
SELECT SUM(salario)
FROM empregado
```

- Qual é o salário mais baixo dos salários dos empregados?
- Qual é o salário mais alto dos salários dos empregados?

Exercício - Resposta

- Qual a média dos salários dos empregados?
- Qual a soma dos salários dos empregados?
- Qual é o salário mais baixo dos salários dos empregados?

```
SELECT MIN(salario)
FROM empregado
```

- Qual é o salário mais alto dos salários dos empregados?

Exercício - Resposta

- Qual a média dos salários dos empregados?
- Qual a soma dos salários dos empregados?
- Qual é o salário mais baixo dos salários dos empregados?
- Qual é o salário mais alto dos salários dos empregados?

```
SELECT MAX(salario)
FROM empregado
```

Exercício

- Quantos supervisores existem na relação empregado?

Exercício - Resposta

- Quantos supervisores existem na relação empregado?

```
SELECT COUNT(SUPERSSN)
FROM empregado
```

Exercício

- Quantos supervisores existem na relação empregado?

```
SELECT COUNT(SUPERSSN)
FROM empregado
```

- Quantos supervisores diferentes existem na relação empregado?

Exercício - Resposta

- Quantos supervisores existem na relação empregado?

```
SELECT COUNT(SUPERSSN)
FROM empregado
```

- Quantos supervisores existem na relação empregado?

```
SELECT COUNT(DISTINCT SUPERSSN)
FROM empregado
```

Group by

- Permite aplicar uma função de agregação não somente a um conjunto de tuplas, mas a um grupo de um conjunto de tuplas;

- Grupo de um conjunto de tuplas:
  - Conjunto de tuplas que possuem o mesmo valor para os atributos de agrupamento.

Group by

- Qual o maior salário, o menor salário e a média de salários na relação empregado por supervisor?

```
SELECT MIN(salario), MAX(salario), AVG(salario)
FROM empregado
GROUP BY SUPERSSN;

SELECT SUPERSSN, MIN(salario), MAX(salario),
AVG(salario)
FROM empregado
GROUP BY SUPERSSN;
```

Having

- Especifica uma condição de seleção para grupos;

- Recupera os valores para as funções somente para aqueles grupos que satisfazem à condição imposta na cláusula HAVING.

Having

- Qual o maior salário, o menor salário e a média de salários na relação empregado por supervisor, para médias salariais superiores a 30000?

```
SELECT MIN(salario), MAX(salario), AVG(salario)
FROM empregado
GROUP BY SUPERSSN
HAVING AVG(salario) > 30000

SELECT SUPERSSN, MIN(salario), MAX(salario), AVG(salario)
FROM empregado
GROUP BY SUPERSSN
HAVING AVG(salario) > 30000
```

Tarefa 06 – Lista de Exercícios

1. Liste os nomes dos empregados e os projetos para os quais cada empregado trabalha. Ordene o resultado pelo nome do projeto, em ordem ascendente. Renomeie as colunas exibidas para Nome do Empregado e Nome do Projeto.

2. Resolva o exercício anterior usando a cláusula JOIN.

3. Liste os nomes dos departamentos e seus respectivos projetos. Ordene o resultado pelo nome do departamento, em ordem ascendente. Renomeie as colunas exibidas para Nome do Departamento e Nome do Projeto.

4. Resolva o exercício anterior usando a cláusula JOIN.

5. Liste os nomes dos departamentos que possuem projetos usando a cláusula IN.

Tarefa 06 – Lista de Exercícios

6. Selecione os nomes e salários dos empregados que ganham entre 40000 e 50000.

7. Resolva o exercício anterior usando a cláusula BETWEEN... AND.

8. Selecione os nomes do empregados que tem dependentes, usando a cláusula INTERSECT.

9. Selecione os nomes do empregados que tem dependentes, usando apenas a cláusula IN.

10. Selecione os nomes do empregados que tem dependentes, usando apenas a cláusula EXISTS.

Tarefa 06 – Lista de Exercícios

11. Liste a quantidade de empregados cadastrados.

12. Liste a quantidade de empregados por supervisor, bem como o ssn do supervisor.

13. Liste a quantidade de empregados por supervisor e o ssn do supervisor, de forma que apenas os supervisores com mais de dois empregados supervisionados sejam listados.

14. Liste os números dos departamentos, bem como seus gastos totais com salários de empregados, para os casos em que os gastos são superiores à média dos gastos (com salário de todos os empregados). Ordene o resultado de forma descendente pelo gasto.