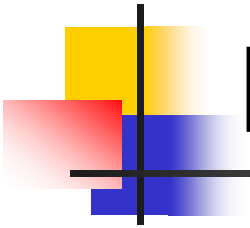
SQL (Structured Query Language) – Parte II

Universidade Federal do Maranhão - UFMA
Departamento de Informática
Banco de Dados I
Profª.MSc Simara Rocha
simararocha@gmail.com/simara@deinf.ufma.br

Referências: Elmasri, R. and Navathe, S.B. Sistemas de Bancos de Dados.Editora Addison-Wesley, 4ª edição, 2005.
Date, C.J. Introdução a Sistemas de Bancos de Dados. Editora Campus, 8ª edição, 2004.
Korth, H.F. e Silberschatz, A. Sistemas de Bancos de Dados. Makron Books, 5ª edição, 2006.

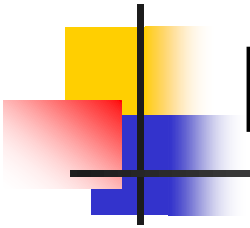


DML

- As operações de manipulação para a DML são: SELECT, INSERT, UPDATE, DELETE
- A forma básica do comando Select é:

  SELECT <lista atributos>

  FROM <lista tabelas>
  WHERE <condição>
- O resultado de qualquer comando SELECT é uma tabela (relação)
- O campo FROM realiza produto cartesiano das tabelas relacionadas
- WHERE deve trazer as condições de junção



DML - Select

- **SELECT**
  - Lista os atributos desejados como resultados de uma consulta;
  - Corresponde a operação de projeção da álgebra relacional;
  - Lista de atributos pode ser substituído por “*” (todos);
- **From**
  - Lista de relação (tabelas) a serem usados na execução da expressão;
- **WHERE**
  - São definidos critérios de pesquisa envolvendo atributos das relações (tabelas) definidas na cláusula from;
  - Corresponde ao predicado da operação de seleção da álgebra relacional;
  - Comando opcional;

DML - Select

- **Esquema relacional baseado no Oracle:**

EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)

EMPNO = Código do Empregado
MGR = Código do Gerente
ENAME = Nome do Empregado
SAL = Salário
JOB = Função do Empregado
COMM = Comissão
HIREDATE= Data de Admissão\Nasc.
DEPTNO = Código Departamento

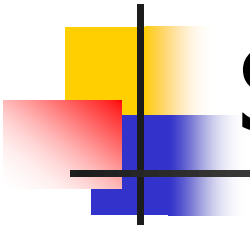
DEPT (DEPTNO, DNAME, LOC)

DEPTNO = Código Departamento
DNAME = Nome do Departamento
LOC = Localização

SALGRADE (GRADE, LOSAL, HISAL)

GRADE = Nível do Salário
LOSAL = Menor Salário no Nível
HISAL = Maior Salário no Nível

Select - Exemplo



- Selecionar todos os atributos de cada empregado:

  SELECT * FROM emp;

- Selecionar todos os empregados com emprego de Gerente (Manager):

  SELECT *

  FROM emp

  WHERE job = 'MANAGER'

- Comparação de valores (WHERE):
  - = , > , >=, <, <= , <> , Or

Select - Exemplo

- Selecionar os nomes de todos os empregados Gerentes com salários maior que R$ 500,00

  SELECT ename

  FROM emp

  Where job **=** 'MANAGER' **and** sal **>** 500;

Select

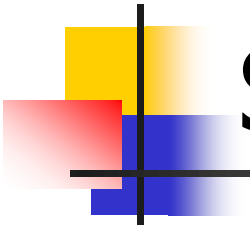
- **Order By**
  - Especifica a seqüência de ordenação da tabela criada pela consulta;
  - Comando opcional;
  - Qualificador opcional : asc / desc.

  Select A1, A2,...,An

  From r1,r2,...rm

  Where P

  Order by A1, A2 **desc**
- Onde A são os campos a serem selecionados, R são as tabelas, e P é uma condição.



Select - Exemplo

- Selecionar para cada empregado o seu nome e a sua função ordenado primeiramente por função e depois por empregado (em ordem decrescente):

  SELECT ename, job

  FROM emp

  ORDER BY job, ename **desc**

Select

- **Between**
  Faz uma pesquisa entre uma faixa de valores (**inclusive**) para um campo da tabela.
- **Not Between**
  Faz uma pesquisa descartando uma faixa de valores.
- Exemplo
  - Selecionar todos empregados que tenham comissão entre 0 e 1000

```
Select *
from emp
where comm between 0 and 1000;
```

Select

- **IN**
  - Consulta a presença de um campo em um conjunto de valores
- **NOT IN**
  - Consulta a NÃO presença de um campo em um conjunto de valores
- Exemplo:
  - Selecionar todos os empregados do departamento 5 e 6

```
select *
from emp
where deptno IN (5, 6)
```

Select

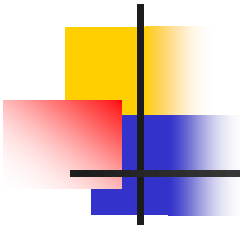
- **Like**
  - Compara a existência de uma caractere em uma determinada posição em um string.
  - ‘%’ representa qualquer seqüência de n caracteres.
- **Not Like**
  - Compara a não existência de um caractere em uma determinada posição em um string.
- Exemplo:
  - Selecionar o nome e código dos empregados que o nome inicia com a letra 'C'.

```
SELECT EMPNO,ENAME
FROM EMP
WHERE ENAME LIKE 'C%';
```

Select

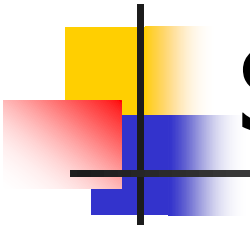
- **Is Null**
  - Verifica se o valor do campo comparado é vazio;
- **Is Not Null**
  - Verifica se o valor do campo comparado não é vazio;
- Exemplo:
  - Selecionar o nome, código e salário dos empregados que não possuem comissão.

```
SELECT EMPNO,ENAME,SAL
FROM EMP
WHERE COMM IS NULL;
```



Select

- **EXISTS ou NOT EXISTS**
  - É usada para verificar se o resultado de uma consulta aninhada é vazio ou não. É sempre usado em conjunto com uma query aninhada
    - SELECT e.ename
      FROM emp e
      WHERE **EXISTS** (SELECT *
      FROM dependente d
      WHERE e.empto = d.empto)



Select

- Operadores Aritméticos
  - + adição
  - - Subtração
  - * Multiplicação
  - / Divisão
- Exemplo:
  - Selecionar o nome dos empregados, seu salário e o salário mais 300:

    SELECT ENAME, SAL, **SAL + 300**

    FROM EMP;

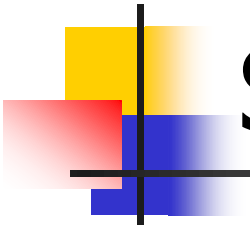
Select

- Renomeando uma coluna (Rename – Álgebra Relacional) ou tabela:
- Tabela:

  SELECT * FROM EMP **AS** Empregado
- Coluna:

  SELECT ENAME **AS** NOME, SAL **AS** SALÁRIO FROM EMP;
  - Saída:

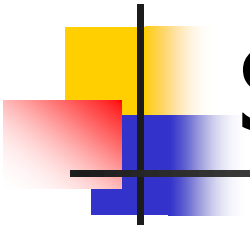
nome	salário
João da Silva	1000
Pedro Paulo	500



Select

- **Distinct**
  - Para Eliminar as linhas duplicadas utiliza-se a palavra DISTINCT na cláusula SELECT.
- Exemplo:
  - SELECT **DISTINCT** DEPTNO

    FROM EMP



Select

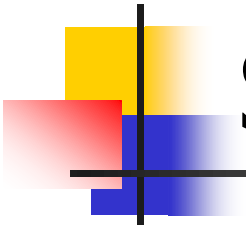
- Funções de agrupamento
  - AVG ([DISTINCT | ALL] n) - valor médio de n, ignorando valores nulos.
  - COUNT ([DISTINCT | ALL] expr *) - número de vezes que a expressão de número EXPR avalia algo diferente de NULO. '*' faz com que COUNT conte todas as linhas selecionadas, incluindo duplicadas e linhas com nulos.
  - MAX ([DISTINCT | ALL] expr) - valor máximo de expr
  - MIN ([DISTINCT | ALL] expr) - valor mínimo de min
  - SUM ([DISTINCT | ALL] n) - soma os valores de n ignorando valores nulos

Select

- Funções de agrupamento
  - Selecionar a média de salário.
    - SELECT AVG(SAL) FROM EMP;
  - Selecionar total de salário pago pela empresa.
    - SELECT SUM(SAL) FROM EMP;
  - Selecionar a quantidade de empregados.
    - SELECT COUNT(ENAME) FROM EMP;
  - Selecionar o maior salário entre os gerentes.
    - SELECT MAX(SAL)

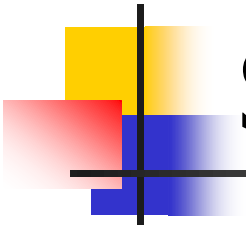
      FROM EMP

      WHERE JOB='MANAGER’



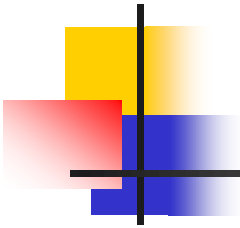
Select

- **GROUP BY**
  - Forma grupos com as tuplas da tabela especificada na cláusula from, que possuem o mesmo valor no atributo especificado na cláusula grupo by;
  - Para ter resultado em ordem, deve ser especificado também a cláusula order by (após a cláusula group by).
  - Ex.: Recuperar o número de empregados por departamento
    - SELECT DEPTNO, COUNT(EMPNO)
      FROM EMP
      GROUP BY DEPTNO



Select

- **HAVING**
  - Having: usada em conjunto com GROUP BY para permitir a inclusão de condições nos grupos.
  - Ex.: Selecionar a média de salário dos departamentos com média superior a 700.
    - SELECT AVG(SAL), DEPTNO
      FROM EMP
      GROUP BY DEPTNO
      HAVING AVG(SAL)>700;

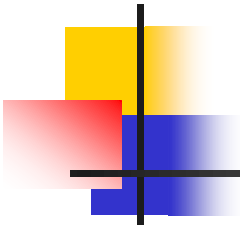


Select

- **Subconsultas ou Consultas Aninhadas**: São consultas que possuem consultas completas dentro de sua cláusula WHERE
  - Motivação: Algumas queries requerem que valores do BD sejam buscados e então usados numa condição
  - Ex1: Empregados que possuem salário maior do que a média:
    - SELECT ENAME, SAL
      FROM EMP
      WHERE SAL > **(SELECT AVG(SAL) FROM EMP)**;
  - Ex2.: Empregados que possuem salário menor do que a média dos MANAGERS com seu respectivo cargo:
    - SELECT ENAME, JOB, SAL
      FROM EMP
      WHERE SAL < **(SELECT AVG(SAL) FROM EMP WHERE JOB = 'MANAGER')**;

Select

- **EQUI-JOIN**
  - O relacionamento entre a tabela EMP e a tabela DEPT é um EQUIJOIN, ou seja, o operador de comparação ' = ' é utilizado.
  - A condição de Join é especificada na cláusula WHERE.
  - Para juntar as duas tabelas EMP e DEPT:
    - SELECT EMP.ENAME, EMP.JOB, DEPT.DNAME
      FROM EMP, DEPT
      WHERE EMP.DEPTNO = DEPT.DEPTNO;



Select

- **Natural Join**
  - Junção ocorreria através da igualdades de colunas com o mesmo nome
    - SELECT * FROM EMP **NATURAL JOIN** DEPT
- **Condition Join**
  - Deixa claro quais colunas farão parte da junção
    - SELECT * FROM EMP JOIN DEPT ON emp.deptno = dept.deptno (explícito)
  - Mesma coisa que:
    - SELECT * FROM EMP, DEPT
      WHERE emp.deptno = dept.deptno (implícito)

Select

- **Outer Join (Junção Externa)**
  - Tratamento para colunas que poderão ter valor NULL (mesmo conceito da Álgebra Relacional).
- **Left Outer Join (Junção Externa à Esquerda)**
  - SELECT * FROM T1 **LEFT OUTER JOIN** T2 **ON** T1.C1 = T2.C3

T1

C1	C2
10	15
20	25

T2

C3	C4
10	BB
15	DD

Junção left outer de T1 com T2

C1	C2	C3	C4
10	15	10	BB
20	25	Null	Null

Select

- **Right Outer Join (Junção Externa à Direita)**
  - SELECT * FROM T1 **RIGHT OUTER JOIN** T2 **ON** T1.C1 = T2.C3

T1

C1	C2
10	15
20	25

T2

C3	C4
10	BB
15	DD

Junção right outer de T1 com T2

C1	C2	C3	C4
10	15	10	BB
Null	Null	15	DD

Select

- **Full Outer Join (Junção Externa Total)**
  - SELECT * FROM T1 **FULL OUTER JOIN** T2 **ON** T1.C1 = T2.C3

T1

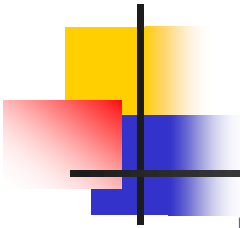
C1	C2
10	15
20	25

T2

C3	C4
10	BB
15	DD

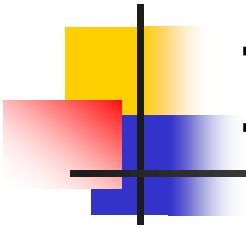
Junção full outer de T1 com T2

C1	C2	C3	C4
10	15	10	BB
20	25	Null	Null
Null	Null	15	DD



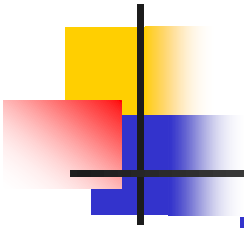
Select

- SQL implementa a operação UNIÃO da álgebra relacional. É requerido que as relações sejam compatíveis de união
  - Exemplo: Recupere o nome dos empregados do departamento 5 ou os gerentes do depto 5
    - select ename from emp where deptno = 5
      union
      select ename from emp where deptno = 5 and job = “Manager”
- Além da operação UNIÃO, o SQL implementa as operações INTERSEÇÃO e DIFERENÇA (∩ e -)
  - INTERSECT
  - MINUS ou EXCEPT
- A sintaxe para os dois comandos é a mesma da UNIÃO



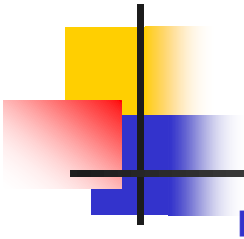
Insert

- Usado para adicionar uma tupla a uma relação
  - Sintaxe: INSERT INTO tabela (lista colunas)
    VALUES (lista de valores atômicos)
    - Ex. INSERT INTO Emp (EMPNO, ENAME, JOB, DEPTNO) VALUES (1,'Maria Silva', 'Administrator', 5)
- A inserção será rejeitada se tentarmos omitir um atributo que não permite valores nulos (NOT NULL)



Delete

- Remove tuplas de uma relação
- Sintaxe:
  - DELETE FROM tabela
    [WHERE condição]
- Obs.: Se omitirmos a cláusula WHERE, então o DELETE deve ser aplicado a todas as tuplas da relação. Porém, a relação permanece no BD como uma relação vazia.



Update

- Modifica o valor de atributos de uma ou mais tuplas.
- Sintaxe:
  - UPDATE tabela
    SET lista_atributos com atribuições de valores
    [WHERE condição]
- Obs.: omitir a cláusula WHERE implica que o UPDATE deve ser aplicado a todas as tuplas da relação