

Banco de Dados

SQL – Parte 2

(musa@unifesp.br)

SQL

- SQL como Linguagem de Manipulação de Dados (DML):
 - Inclusão, exclusão e atualização de dados
 - Consulta de dados

Esquema exemplo

- Fornecedor (Fcod, Fnome, Status, #Cidcod, fone)
- Cidade (Ccod, Cnome, uf)
- Peça (Pcod, Pnome, Cor, Peso, #Ccod)
- Projeto (PRcod, Jnome, #Ccod)
- Fornecimento (#Fcod, #Pcod, #Prcod, Quantidade)

O Comando INSERT

**INSERT INTO <nome_tabela> [(<colunas>)]
VALUES (<lista de valores>)**

- insere linhas na tabela

Exemplos:

**INSERT INTO Peças
VALUES (9,'meia','azul', 0,3, 'Poa');**

**INSERT INTO Peças(Pcod, Pnome, Cor, Peso,Cidade)
VALUES (10,'calça','preta',2, 'Poa');**

O comando UPDATE

```
UPDATE <nome_tabela>  
SET <atribuição>  
WHERE <condição>
```

Alterar linhas da tabela

Exemplo:

```
UPDATE Peças  
SET Peso = 6  
WHERE Pnome = 'camiseta';
```

O comando DELETE

**DELETE FROM <nome_tabela>
[WHERE <condição>]**

Excluir linhas da tabela

Exemplo:

DELETE FROM Peças;

**DELETE FROM Peças
WHERE CIDADE='São Paulo';**

SQL - Consultas

1. Consultas Básicas
2. Consultas Complexas
3. Facilidades Adicionais

Consultas Básicas

```
SELECT <lista de atributos>  
FROM <lista de tabelas>  
[ WHERE <condição>; ]
```

O resultado de uma consulta SQL é uma relação.

- Exemplo: “Recuperar cores utilizadas para peças”

```
SELECT Cor  
FROM Peças
```


Consultas Básica

- “Recuperar cores utilizadas para peças”

Pcod	Pnome	Cor	Peso	Cidade
P1	calça	branco	1	C4
P2	blusa	branco	0,6	C5
P3	blusa	azul	0,6	C1
P4	camiseta	branco	0,8	C4
P5	saia	azul	0,3	C4
P6	saia	vermelho	0,3	C5

Cor
branco
branco
azul
branco
azul
vermelho

Consultas Básicas

```
SELECT DISTINCT Cor  
FROM Peças
```

Cor
branco
azul
vermelho

Consultas

- Uso da expressão WHERE
 - Operadores para comparações lógicas:
=, <>, <, <=, >, >
<coluna> BETWEEN 'valor' AND 'valor'
<coluna> LIKE '%banco de %'
<coluna> LIKE '19__'
 _ = um caracter
 % = sequência de caracteres

Consultas

Exemplo: “Recuperar nomes das cidades gaúchas”

```
SELECT Cnome  
FROM Cidade  
WHERE uf = 'RS'
```

Tabela do banco:

Ccod	Cnome	uf
C1	Santa Maria	RS
C2	Passo Fundo	RS
C3	Florianópolis	SC
C4	Porto Alegre	RS
C5	São Paulo	SP
C6	Curitiba	PR

Resultado:

Cnome
Santa Maria
Passo Fundo
Porto Alegre

Consultas envolvendo mais de uma tabela - Junção

- Usado quando a consulta SQL envolver duas tabelas.
- O usuário especifica os atributos de junção (condição de junção)
- A cláusula WHERE especifica os atributos usados para a junção das duas tabelas

Consultas envolvendo mais de uma tabela - Junção

- Exemplo: “Selecionar os nomes e status de fornecedores de Porto Alegre”

```
SELECT Fnome, status  
FROM fornecedor, cidade  
WHERE Cnome="Porto Alegre" AND Ccod=Cidcod
```

Consultas envolvendo mais de uma tabela - Junção

- SELECT Fnome, status
FROM fornecedor, cidade
WHERE Cnome="Porto Alegre" AND Ccod=Cidcod

Fcod	Fnome	Status	fone	Cidcod
F3	Ana	30	22222	C1
F4	Pedro	22	3333	C3
F1	João	10	333333	C4
F2	Maria	20	55555	C4
F5	Katia	40	299999	C6

Ccod	Cnome	uf
C1	Santa Maria	RS
C2	Passo Fundo	RS
C3	Florianópolis	SC
C4	Porto Alegre	RS
C5	São Paulo	SP
C6	Curitiba	PR

Fnome	Status
João	10
Maria	20

Resultado

Consultas envolvendo mais de uma tabela - Junção

- Exemplo: Listar os nomes de fornecedores de “Camiseta”.

```
SELECT Fnome
FROM Fornecedor, Peças, Fornecimento
WHERE Peça.Pcod=Fornecimento.Pcod AND
       Fornecedor.Fcod=Fornecimento.Fcod AND
       Pnome="Camiseta"
```

Atributos com o mesmo nome devem ser qualificados.

Uso de aliases

- Considerando a tabela:
Funcionario(Fcod, Fnome, FSupervisor)
- *“Selecionar nomes dos funcionários e nomes de seus chefes”*

```
SELECT F1.Fnome, F2.Fnome  
FROM Funcionario F1, Funcionario F2,  
WHERE F2.Fcod=F1.FSupervisor
```

Uso do *

- O asterisco (*) retorna todas as colunas da tabela

```
SELECT Fcod, Fnome, Status, Cidade  
FROM Fornecedor
```

é o mesmo que

```
SELECT *  
FROM Fornecedor
```

Consultas Aninhadas

Uso da cláusula IN

- Verifica se um valor está no resultado de outra consulta

```
SELECT PRnome  
FROM Projeto  
WHERE Icod
```

IN

```
(SELECT Icod  
FROM Instituição  
WHERE I.Inome="Empresa X")
```

Consultas Aninhadas

- Consulta equivalente sem ser aninhada:

```
SELECT PRnome  
FROM Projeto P, Instituição I  
WHERE P.Icod = I.Icod AND  
I.Inome="Empresa X")
```

Operador Existencial

Uso da cláusula EXISTS

```
SELECT Fnome  
FROM Fornecedor  
WHERE EXISTS  
(SELECT *  
FROM Fornecimento  
WHERE Fcod=Fornecedor.Fcod)
```

Recupera o nome do
fornecedor para o qual
há pelo menos um
fornecimento

Recupera as tuplas da
primeira consulta para
o qual exista pelo menos
uma correspondente
na segunda

Campos Vazios

Pode-se testar atributos nulos

- Uso de IS NULL (IS NOT NULL)

“Recuperar os nomes de fornecedores cujo status não foi informado”

```
SELECT Fnome  
FROM Fornecedor  
WHERE Status IS NOT NULL
```

Operadores Aritméticos

- Adição +
- Subtração -
- Multiplicação *
- Divisão /
- Aplicados diretamente a valores numéricos ou a atributos com domínio numérico
- Exemplo: *“Mostrar os pesos das peças se acrescidas de 10”*

```
SELECT Pnome, Peso+10  
FROM Peça
```

Funções Agregadas

- Média - AVG
- Mínimo - MIN
- Máximo - MAX
- Total - SUM
- Contagem - COUNT

As funções podem ser aplicadas sobre:

- Toda a tabela
- Grupos de Linhas (definidos através do GROUP BY - que será visto a seguir)

Funções Agregadas

- Exemplo:

“Obter o número de peças”

```
SELECT COUNT(*)  
FROM Peças
```

Modelo estendido de consultas

- Sintaxe

```
SELECT <Lista_de_colunas>  
FROM <Lista_de_tabelas>  
[WHERE <Condição>]  
[GROUP BY <Lista_de_Colunas>]  
[HAVING <Critério>]  
[ORDER BY <Lista_de_Colunas>]
```

Agrupamento

- Divide o resultado de uma consulta em “grupos” de tuplas cujos valores das colunas especificadas são iguais.
- Restrição: quando é usado o GROUP BY, no SELECT, pode ser utilizado:
 - uma constante
 - uma função de agregação calculada sobre o grupo
 - uma coluna que faz parte do GROUP BY
 - uma expressão envolvendo as outras alternativas.

Condição

- HAVING : especifica uma condição que cada “grupo” deve satisfazer.

“Obter o número de peças fornecidas para cada fornecedor, apenas para os que maior do que 600”

```
SELECT Fcod, COUNT(Pcod)
```

```
FROM Fornecimento
```

```
GROUP BY Fcod
```

```
HAVING COUNT(Pcod) > 600
```

Ordem

- ORDER BY - ordena as linhas da tabela resultante, por default, ordem crescente; ou uso de DESC

```
SELECT Pnome, Peso, cor  
FROM Peça  
WHERE cor='vermelha'  
ORDER BY Pnome DESC
```

Exercícios

Terminar Lista sobre base de dados Produção Cinematográfica