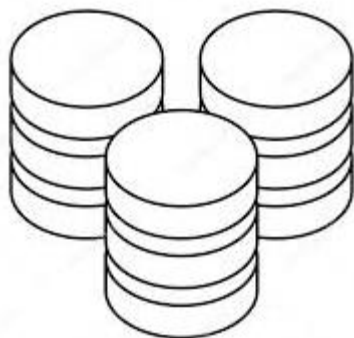


Banco de Dados II

Subqueries
Create table as
Reescrita de consultas



Profa. Damires Souza
damires@ifpb.edu.br



Select – From – Where

Select <nome(s) da(s) coluna(s)>
From <tabela ou lista de tabelas>
Where <restrições>;

$\pi_{\text{<colunas>}} (\sigma_{\text{<restrições>}} (\text{<tabela>}))$



Projeção



Seleção



Um pipeline!!

```
select * from produto;
```

```
Select descricao
```

```
From produto
```

```
Where unidade IN ('KG', 'L', 'M');
```

descricao
character varying (20)

Queijo

Leite

Linho

Feijao

Açucar

```
Select round(avg(valor),0)
```

```
From produto;
```

round
numeric



14

Posso usar o resultado de cada consulta para outra?

Subqueries

- **Subquery** (subconsulta) é um comando **SELECT** que **foi incluído** em outro comando **SELECT**, **UPDATE**, **INSERT**, **DELETE**...
 - ou dentro de outra subquery...
- **Por quê??**
 - Algumas consultas requerem
 - **valores** obtidos a partir de outra consulta
 - um **conjunto de linhas** obtido a partir de outra consulta

Exemplo

- Problema: Que produtos participam em qualquer pedido cuja quantidade de produtos, em itens, seja igual a 10?

Select p.descricao

From produto p

Where p.codprod in (select i.codprod

From itenspedido i

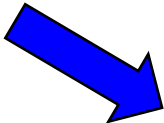
Where i.quantidade = 10);

Consulta Interna
ou subconsulta
ou consulta aninhada

Consulta
externa



codprod	
integer	
	1
	4
	5



descricao
character varying (20)
Queijo
Linho
Feijao

Outro Exemplo

- Problema: Quais vendedores ganham **salário fixo** abaixo da média?

```
select v.nome  
      From vendedor v  
      Where v.salariofixo  
            < (Select round(AVG(v.salariofixo),1)  
              From vendedor v);
```

Qual o valor da Média de salário?

Qual o resultado da query?

Operadores

Operador	Definição
Exists R	Verdadeira se e somente se R não é vazia
s IN R	Verdadeira se s é igual a um dos valores de R
s > ALL R	Verdadeira se s é maior que todos os valores da relação R
s > ANY R	Verdadeira se s é maior que no mínimo um valor na relação R

s <> ALL R \Leftrightarrow s NOT IN R

s = ANY R \Leftrightarrow s IN R

****** Os operadores Exists, any, in e all podem ser negados (**NOT**); o operador **>** pode ser substituído por qualquer outro (<, =, <>)

Exemplos

- > ALL (1, 2, 3) ??
=> maior do que 3
- > ANY (1, 2, 3) ??
=> maior do que 1

```
Select p.descricao
From produto p
Where p.valor >
ANY (Select max(pr.valor)
      From produto pr
      group by pr.unidade) ;
```

E com ALL??

```
Select p.unidade,
max(p.valor)
From produto p
group by p.unidade
```



unidade character (2)	max numeric
KG	22.00
L	10.00
M	24.00
G	6.12

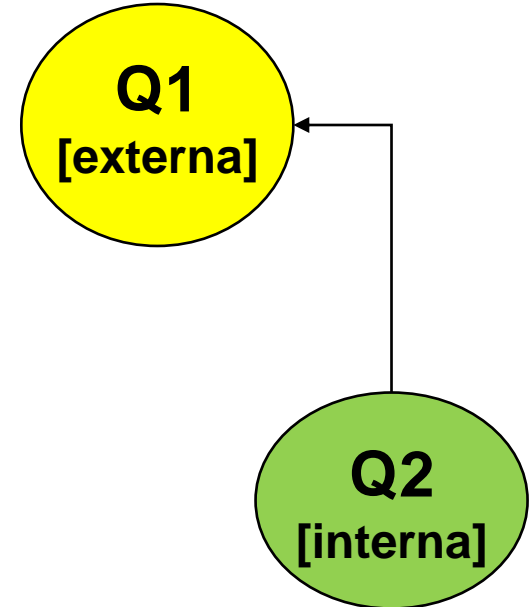


descricao character varying (20)
Queijo
Leite
Linho
Feijao
Açucar

Consultas correlacionadas

** Ou consultas correlatas

- Se a condição na cláusula WHERE da consulta aninhada/subconsulta utiliza **um atributo** de tabela declarada na clausula FROM da **consulta externa** então elas são chamadas de **correlacionadas**
 - Acontece quando **uma subquery referencia colunas da consulta externa**



Exemplo

**** Faça: insert into produto values (default, 'XXX', 1.2, 'KG')**

- Problema: Quais os produtos que **não** estão presentes em nenhum pedido?
- Solução:

select codprod, descricao

From produto **P**

Where not exists (select *

From itenspedido i

Where **i.codprod = P.codprod**);



Teste de
Existência

Como??

1. Carrega a tabela do FROM da **externa** na memória
2. Para cada linha desta tabela, verifica se o resultado da consulta **interna** existe
3. Se o resultado interno existe, retorna a projeção do SELECT externo
4. Para cada linha da tabela Itenspedido
 - Se existe um item cujo **i.codprod** seja igual ao seu **P.codprod**
 - Então retorna a linha do itempedido

codprod [PK] integer	descricao character varying (20)
1	Queijo
4	Linho
5	Feijao
2	Chocolate
3	Leite
6	Açucar
7	XXX

numped [PK] integer	codprod [PK] integer	quantidade integer
2	2	35
2	3	30
1	1	10
4	5	35
5	4	10
3	4	35
1	5	10



Subconsulta pode estar em..

- **Select:**

- Pode ser colocada nas cláusulas **select**,
from, **where**

- **DML:**

- Insert
 - Update
 - Delete

- **Create table as**



Exemplo com SELECT

Select distinct

(Select COUNT(*) from cliente where cidade like 'João Pessoa') AS "Quantidade de Pessoenses",

(Select COUNT(*) from cliente where cidade like 'Recife') AS "Quantidade de Recifenses"

From cliente;

Qual o resultado da query?

Exemplo com INSERT

- Problema: Cadastrar como cliente os vendedores que possuem **faixa de comissão 'A'**. Usar para código do cliente o mesmo código de vendedor acrescido do valor 10.

insert into cliente (**codcli, nome**)

(select **codvend + 10, nome**
from vendedor

where faixacomissao like 'A');

codcli [PK] integer	nome character varying (30)	endereço character varying (30)	telefone character varying (12)	insce character varying (10)	cnpj character varying (10)	cidade character varying (10)
5	Ana Moura	Nego, 321	32465432	87325	780978	Joao Pessoa
6	Cassandra Doura	[null]	[null]	786525	79876	Recife
7	Cicero Novaes	[null]	[null]	123525	432534	Natal
8	Marcos Araruna	Sergipe,267	43265432	900325	800534	Joao Pessoa
9	Marcia Araruna2	Sergipe 267	43265432	900325	800534	Joao Pessoa
13	Carla Gomes	[null]	[null]	[null]	[null]	[null]

Exemplo com UPDATE

- Problema: Acrescentar 2,5% ao preço unitário dos produtos que estejam **abaixo da média** dos preços para aqueles comprados a quilo.

Update produto

Set valor = valor*1.025

Where valor < (select avg(valor)

From produto

Where unidade = 'KG');

codprod [PK] integer	descricao character varying (20)	valor numeric (10,2)	unidade character (2)
2	Chocolate	6.27	G
5	Feijao	12.00	KG
1	Queijo	22.00	KG
6	Açucar	9.23	KG
7	XXX	1.23	KG
3	Leite	10.25	L
4	Linho	24.00	M

Exemplo com DELETE

Obs: antes da próxima operação, execute:

```
insert into pedido values(100,10,'12/10/2020',4,null);
```

Problema: Apagar todos os registros de pedidos realizados por vendedores fantasmas (**codvend = null**).

```
delete from pedido P
where not exists (select nome
                  from vendedor v
                  where v.codvend = P.codvend);
```

Qual o resultado da query?

É uma consulta correlacionada?

Qual o resultado da query?

```
Select p.data
from pedido p
where p.numped in
    (select i.numped
     from itenspedido i
     where i.codprod in
         (select pr.codprod
          from produto pr
          where descricao like 'Chocolate'))
```

Faça...

- Como reescrever essa consulta usando **JOIN**?



Create Table as

- Pode-se criar uma tabela a partir de outra(s)

CREATE TABLE AS

- Cria e carrega uma tabela com dados produzidos pelo comando **SELECT** (como uma **subconsulta**).
- As **colunas da tabela** possuem os nomes e tipos de dados associados às **colunas da saída do SELECT** (projeção)
 - Pode-se substituir os nomes das colunas fornecendo-se uma lista explícita de novos nomes de colunas

Create Table AS

```
CREATE [ [ LOCAL ] { TEMPORARY | TEMP } ]  
TABLE nome_da_tabela [ (nome_da_coluna [,  
...]) ] AS consulta
```

- A tabela pode, opcionalmente, ser criada como uma **tabela temporária**.
 - É automaticamente eliminada no final da sessão.

Exemplo

```
CREATE TABLE pedidoVendedor AS
```

```
select p.numped, v.nome
```

```
from pedido p join vendedor v on p.codvend =  
v.codvend
```

```
where data < '12/12/2020';
```

➤ **Select * from pedidoVendedor;**

➤ **insert into pedidoVendedor values(201, 'Bruno Assis');**

Como ficou a **integridade** de chave?

numped integer	nome character varying (30)
1	Juan Gomes
2	Juan Gomes
3	Josefa Cirino
4	Joao Peregrino
5	Carla Gomes
6	Joao Peregrino

numped integer	nome character varying (30)
1	Juan Gomes
2	Juan Gomes
3	Josefa Cirino
4	Joao Peregrino
5	Carla Gomes
6	Joao Peregrino
201	Bruno Assis

O que acontece aqui??

```
create table vendedor1 as  
  select * from vendedor where 1=2;  
  
select * from vendedor1;
```

Vamos testar?



O que acontece aqui??

➤ **Antijoin** com subquery

```
SELECT c.nome, c.codcli  
FROM cliente c  
WHERE c.codcli NOT IN  
      (SELECT p.codcli  
       FROM pedido p)  
ORDER BY c.nome;
```

Anti join é um tipo de junção o qual retorna linhas de uma tabela que **não combinam** com um conjunto específico de outra tabela.

O que acontece aqui??

➤ **Antijoin** com join

```
Select c.nome, c.codcli  
from cliente c left join pedido p on c.codcli =  
    p.codcli  
where p.codcli is null  
Order by c.nome;
```

Veja:

```
Select c.nome, c.codcli as cliente, p.codcli as  
    ClienteemPedido, numped  
from cliente c left join pedido p on c.codcli = p.codcli  
Order by c.nome;
```