

Banco de Dados – Ciência da Computação

SQL (*Structured Query Language*)

Prof. Mariane Moreira

mariane.souza@unifal-mg.edu.br

Universidade Federal de Alfenas

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Consultas

- **Junção**

- Operador **"join"** (usado para especificar a junção na cláusula from).

cliente(nome_cliente, rua, cidade)

depositante(numero_conta,nome_cliente)

- Ex: retorne o nome do cliente da conta 'CT01'.

- Select nome_cliente

from **cliente c join depositante d** condição de junção

c.nome_cliente = d.nome_cliente

where **d.numero_conta = 'CT01'** condição de seleção

Consultas

- Tipos de Junção

- Inner join

- Retorna somente tuplas relacionadas
 - Mesmo resultado de Join.

cliente(nome_cliente, rua, cidade)
depositante(numero_conta,nome_cliente)

- Select nome_cliente
from **cliente c inner join depositante d** on
c.nome_cliente = d.nome_cliente
where d.numero_conta = 'CT01'

Consultas

- Tipos de Junção

- **Natural join**

- Não é necessário especificar o critério de junção. A junção é realizada comparando os atributos com mesmo nome das relações especificadas no join.

cliente(nome_cliente, rua, cidade)

depositante(numero_conta, nome_cliente)

- Select nome_cliente

from **cliente c natural join depositante d**

where d.numero_conta = 'CT01'

Consultas

- Tipos de Junção
 - **Left Outer join**
 - Realiza a operação join normalmente
 - Retorna todos os registros da tabela da **esquerda**, independente de relacionamento com algum registro da tabela da direita
- `Select nome_cliente`
`From cliente c left outter join depositante d`
`on c.nome_cliente = d.nome_cliente`
`where d.numero_conta = 'CT01'`

Consultas

- Tipos de Junção
 - **Right Outer join**
 - Realiza a operação join normalmente
 - Retorna todos os registros da tabela da **direita**, independente de relacionamento com algum registro da tabela da esquerda
- `Select nome_cliente`
`From cliente c right outer join depositante d`
`on c.nome_cliente = d.nome_cliente`
`where d.numero_conta = 'CT01'`

Consultas

- Tipos de Junção

- Full Outer join

- Realiza a operação join normalmente
 - Retorna todos os registros da tabela da **direita e esquerda**, independente de relacionamento em cada uma.
- cliente(nome_cliente, rua, cidade)
depositante(numero_conta, nome_cliente)

- Select nome_cliente

From **cliente c full outer join depositante d**

on c.nome_cliente = d.nome_cliente

where d.numero_conta = 'CT01'

Obs: O "Natural" pode ser adicionado em todas as variações

Consultas

- Consultas aninhadas (subconsulta, subqueries, *nested* queries).
 - Uso de uma consulta do tipo **select-from-where** na cláusula “**where**” de outra consulta.
 - Objetivo: consultar uma **informação no banco** para ser usada na cláusula **where** da consulta externa.
 - Uma subconsulta pode retornar:
 - Valor (pode ser usado o operador = na comparação).
 - Conjunto ou multiconjunto (relação)

Consultas

- Consultas aninhadas que retornam valor
 - Pode ser usado o operador =
 - Neste caso, a subconsulta deve retornar apenas **um valor**.

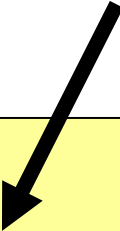
conta(numero_conta, nome_agencia, saldo)

Selecione as contas que tenham o saldo igual ao da conta C11.

SUBCONSULTA

Select *

from conta **where** saldo = **(select saldo from conta where numero_conta = 'C11')**



Consultas

- Consultas aninhadas que retornam conjunto ou multiconjunto
- Neste caso, devem ser utilizados **operadores de conjunto na comparação**.
 - Operadores IN e NOT IN

conta(numero_conta, nome_agencia, saldo)

emprestimo(numero_emprestimo, nome_agencia, quantia)


Selecione as contas que possuem empréstimo
na quantia superior a 20.000

SUBCONSULTA

Select *

from conta

where nome_agencia **in** (select nome_agencia from
emprestimo where quantia >20000)



Consultas

- Consultas aninhadas - **exemplos**
 - depositante(nome_cliente, numero_conta)
tomador(nome_cliente, numero_emprestimo)
 - Encontre todos os clientes do banco que possuem uma conta e um empréstimo.
 - ***select nome_cliente from depositante
where nome_cliente in
(select nome_cliente from tomador)***

Consultas

- Consultas aninhadas - **exemplos**
 - Operador **NOT IN**
 - depositante(nome_cliente, numero_conta)
tomador(nome_cliente, numero_emprestimo)
 - Encontre todos os clientes do banco que possuem conta e que não possuem empréstimo
 - ***select nome_cliente from depositante
where nome_cliente not in
(select nome_cliente from tomador)***

Consultas

- Consultas aninhadas - operador **Some/Any**
 - Operador usado para comparar o valor de '**a**' (normalmente um atributo) a um conjunto ou multiconjunto **A**.
 - **a** = some **A** (**a** tem que ser = a algum valor em **A**).
 - Além do =, os demais operadores de comparação podem ser usados (>, <, <=, >=, <>)

Consultas

- Consultas aninhadas - operador **Some/Any**

conta(numero_conta, nome_agencia, saldo)

- Ex: Retorne as contas que possuem saldo maior que o de **alguma conta** da agência Halfed.
- **Select * from** conta
where saldo > some (**select saldo from conta where nome_agencia = 'Halfed'**)

Consultas

- Consultas aninhadas - operador **ALL**
 - Operador usado para comparar o valor de '**a**' (normalmente um atributo) a um conjunto ou multiconjunto **A**.
 - **a** > all **A** (a tem que ser maior que **todos** os valores em A).
 - Também podem ser utilizados outros operadores...

Consultas

- Consultas aninhadas - operador **ALL**

conta(numero_conta, nome_agencia, saldo)

- Ex: Retorne as contas que possuem saldo maior que **todas** as contas da agência Halfed.

- **Select * from** conta

where saldo > ALL (**select saldo from conta
where nome_agencia = 'Halfed'**)

Consultas

- Subconsultas – variáveis locais e globais
 - As variáveis de tuplas definidas nas consultas internas (subconsultas) tem escopo local
 - Só existem na consulta aninhada e não são reconhecidas na consulta principal.
 - Por outro lado, variáveis declaradas na consulta principal tem escopo global e são reconhecidas nas subconsultas.
 - Aplicação: mesmo nome de atributos → necessidade de diferenciar através de renomeação das tabelas.

Consultas

- Subconsultas – variáveis locais e globais

dependente (dcodigo,pnome,sexo,essn)

- Ex: Retorne o primeiro nome dos empregados que possuem dependentes com o mesmo primeiro nome e sexo do empregado.

- Select pnome from empregado e
where ss in

**(select essn from dependente d
where e.pnome = d.pnome
and e.sexo = d.sexo)**

——— Consulta
mais interna

Consultas

Formas equivalentes de se realizar a mesma consulta:

```
Select pnome from empregado e  
where ssn in
```

```
(select essn from dependente d  
where e.pnome = d.pnome  
and      e.sexo = d.sexo)
```

Com consulta aninhada...

```
Select pnome  
from empregado e, dependente d  
where e.ssn = d.essn and e.pnome = d.pnome and  
      e.sexo = d.sexo
```

Sem consulta aninhada...

Consultas

- Consultas aninhadas - operador Exists

- Verifica se o resultado da subconsulta é vazio.
- Ex: Retorne o primeiro nome dos empregados **caso exista** dependentes com o mesmo primeiro nome e sexo do empregado.
- Select pnome from empregado e
where exists

**(select * from dependente d
where e.ssn = d.essn
and e.pnome = d.pnome
and e.sexo = d.sexo)**

A busca é feita em todo o banco,
A cada vez que existir, a cláusula
Exists retorna true e este registro
é retornado no resultado.

Consultas

- Consultas aninhadas - operador Exists
 - Ex: Encontre os empregados que possuem conta e empréstimo no banco.
 - Select nome_cliente from depositante d
where exists
**(select nome_cliente from tomador t where
t.nome_cliente = d.nome_cliente)**

Consultas

- Consultas aninhadas – operador **Not Exists**
- Ex: Encontre os empregados que possuem uma conta e não possuem empréstimo no banco.
- Select nome_cliente from depositante d
where not exists
**(select nome_cliente from tomador t where
t.nome_cliente = d.nome_cliente)**

Consultas

- **Consultas aninhadas - operador Unique:** testa se o resultado da subconsulta possui valores **duplicados** ou não.

Retorne os nomes dos clientes que possuem **apenas uma** conta na agência Halfeld,

depositante(nome_cliente, número_conta)

conta(número_conta, nome_agencia, saldo)

Select nome_cliente from depositante d
where unique

**(select número_conta from conta c
where c.número_conta = d.número_conta
and nome_agencia = 'Halfeld')**

Caso o resultado da consulta aninhada seja de uma única tupla, UNIQUE retorna true e o nome do cliente é retornado no resultado.

Consultas

- Divisão

- Não existe um operador de divisão em SQL
- A operação pode ser simulada utilizando os operadores NOT EXISTS e EXCEPT.
- Ex: Encontre os empregados que trabalham em todos os projetos.
- `Select e.pnome from empregado e where NOT EXISTS ((Select p.pnumero from projeto p) EXCEPT (Select t.pnumero from trabalhaem t where t.ssn = e.ssn))`
 - Dados todos os números de projetos, tirando os que o empregado trabalhar, o resultado deve ser vazio (NOT EXISTS = TRUE) ou seja, o conjunto de projetos em que o empregado trabalha deve ser o conjunto total de projetos que existe.

Exercícios

EMPREGADO

PNOME	MINICIAL	UNOME	<u>SSN</u>	DATANASC	ENDERECO	SEXO	SALARIO	SUPERSSN	DNO
-------	----------	-------	------------	----------	----------	------	---------	----------	-----

DEPARTAMENTO

DNOME	<u>DNUMERO</u>	GERSSN	GERDATAINICIO
-------	----------------	--------	---------------

DEPTO_LOCALIZACOES

<u>DNUMERO</u>	<u>DLOCALIZACAO</u>
----------------	---------------------

PROJETO

PJNOME	<u>PNUMERO</u>	PLOCALIZACAO	DNUM
--------	----------------	--------------	------

TRABALHA_EM

<u>ESSN</u>	<u>PNO</u>	HORAS
-------------	------------	-------

Exercícios

- Usando join, retorne o nome (em ordem crescente) de todos os departamentos localizados no pavilhão A
- Usando join, retorne o primeiro nome (em ordem decrescente) de todos os gerentes de departamentos localizados no pavilhão A
- Usando subconsultas, retorne o primeiro nome de todos os empregados que trabalham no projeto "ALFA"
- Retorne todos os nomes de projeto que possuem empregados que trabalham mais de 10 horas

Exercícios

- Retorne a média de horas trabalhadas por cada empregado considerando somente as médias maiores que 12.
- Retorne a média de horas trabalhadas por cada empregado considerando somente as médias maiores que 14 e empregados do departamento D01.
- Retorne todos os nomes de empregados com seus respectivos nomes de departamentos, inclusive os empregados que não possuem departamento (suponha que não seja obrigatório especificar o departamento do empregado)
- Retorne os nomes de todos os empregados que trabalham mais que algum empregado no projeto 'P01'

Exercícios

- Encontre todos os empregados (ssn) que trabalham com a mesma combinação (projeto, horas) do empregado com código E001.
- Retorne os nomes dos empregados que ganham mais que todos os empregados do departamento D05.
- Encontre os projetos controlados por todos os departamentos.
- Retorne os nomes dos empregados que não possuem dependentes.

Próximas aulas...

- SQL (Continuação)