

### Consultas SQL

TIAGO G MORAES

### Roteiro



#### □Revisão DML

oInsert, delete update e select

#### ■Consultas SQL

- o Eliminando resultados duplicados
- o Renomeando atributos e relações
- OUtilizando WHERE
- Ordenando resultados
- o Agrupando dados

BANCO DE DADO:

2

### Revisão DML



#### □DML: Data Manipulation Language

o Manipulação dados: inserir, alterar, deletar e consultar dados

#### □Inserindo dados

o INSERT:

INSERT INTO <nome\_tabela> [ (<col1>, <col2>, ...)]
VALUES (<val1>, <val2>, ...)

#### VALUES

■Variantes:

o Mais linhas
 INSERT INTO pessoa VALUES (1, 'João'), (2, 'Maria')
o A partir de uma tabela

INSERT INTO aluno (cod, nome) (SELECT \* FROM pessoa)

→[] = opcionalidade → ⇒ = preencher com algum valor →{} = escolha um dos itens

BANCO DE DADOS

### Revisão DML



### □DML: Data Manipulation Language

o Manipulação dados: inserir, alterar, deletar e consultar dados

#### ■Excluindo dados

ODELETE

DELETE FROM <tabela> [WHERE <condição>]

· Cuidado para não deletar todas linhas de uma tabela. Para isso use o where

→[] = opcionalidade → ⇒ = preencher com algum valor → {} = escolha um dos iten:

BANCO DE DADOS

## Revisão DML



### ■Atualizando dados

oUpdate:

UPDATE <nome\_tabela> SET <coll>=<vall>,...
<coln>=<valn> [WHERE <CONDIÇÃO>]

### ■Buscando dados

oSelect:

SELECT <colunas> FROM <tabela> [WHERE <condição>]

→[] = opcionalidade → ⇒ = preencher com algum valor →{} = escolha um dos itens

o Porém esse formato de consulta é limitado....

NCO DE DADOS

### Resultados duplicados



# □A relação resposta a uma *SELECT* pode conter linhas iguais (duplicadas)

- Eliminar duplicadas → "distinct"
- Exemplo:

SELECT distinct cargo

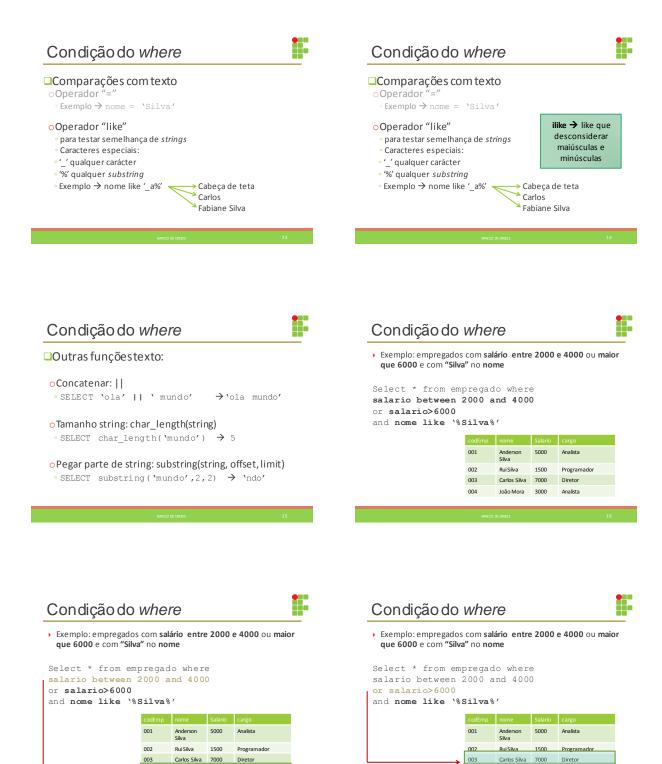
FROM empregado

cargo
Analista
Programador
Diretor

codEmp	nome	Salario	cargo
001	Anderson	5000	Analista
002	Rui	1500	Programador
003	Carlos	7000	Diretor
004	João	3000	Analista

ANCO DE DADOS





004

João Mora 3000

Analista

004

João Mora

3000

Analista

### Condição do where

 Exemplo: empregados com salário entre 2000 e 4000 ou maior que 6000 e com "Silva" no nome

Select \* from empregado where salario between 2000 and 4000 or salario>6000

and nome like \%Silva%'



TO DE DADOS 19

### Condição do where



 Exemplo: empregados com salário entre 2000 e 4000 ou maior que 6000 e com "Silva" no nome

Select \* from empregado where salario between 2000 and 4000 or salario>6000 and nome like '%Silva%'

Cuidado com valores nulos null>1000? • Is null • Is not null

TO THE DATIONS

### Orden ando resultados



- □Order by → serve para ordenar por um ou mais campos
- Utilizado ao final da select
- o A ordenação é feita pelo primeiro campo passado
- $^{\circ}$  Em caso de empate é ordenado pelo segundo campo e assim sucessivamente

### Orden ando resultados



- □Order by → serve para ordenar por um ou mais campos
- Utilizado ao final da select
- o A ordenação é feita pelo primeiro campo passado
- · Em caso de empate é ordenado pelo segundo campo e assim sucessivamente
- oCrescente → "asc" select \* from empregado

	nome	Salario	cargo	
004	João	3000	Analista	empate
001	Anderson	5000	Analista	
003	Carlos	7000	Diretor	
002	Rui	1500	Programador	r

BANCO DE DATOS 22

### Limitando dados



□Limit → limita o número de linhas na resposta oElimina os últimos

□Offset → pula linhas superiores da resposta

select \* from empregado
limit 2 offset 1

codEmp	nome	Salario	cargo
004	João	3000	Analista
001	Anderson	5000	Analista
003	Carlos	7000	Diretor
002	Dui	1500	Drogramador

ANCO DE DADOS 23

## Operações com grupos





■União → AUB



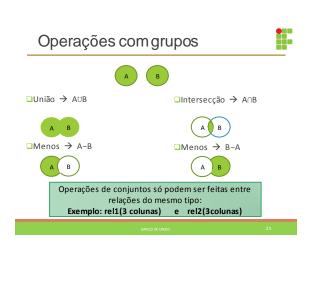
□Menos → A-B

□Intersecção → A∩B



A B

BANCO DE DADOS 24



### Operações com grupos



Union distinct (padrão)

#### □Union

SELECT nome FROM empregado UNION

SELECT nome FROM cliente;

□Intersect

SELECT nome FROM empregado INTERSECT

SELECT nome FROM cliente;

□Except (Minus)

SELECT nome FROM empregado

EXCEPT

SELECT nome FROM cliente;

## Agrupando dados



#### □Funções agregadas

- o Tomam uma coleção de valores e retornam um único valor
- Média (average) → avg ()
- ∘ Mínimo → min()
- · Máximo → max ()
- Somatório → sum()
- ∘ Contagem (Número de linhas) → count ()

## Agrupando dados



#### □Funções agregadas

- o Tomam uma coleção de valores e retornam um único valor
- Média (average) → avg()
- Mínimo → min()
- Máximo → max ()
- · Somatório → sum ()
- Contagem (Número de linhas) → count ()

### •Exemplos:

SELECT count(\*) FROM empregado SELECT avg(salario) AS media FROM

## Agrupando dados



### □Cláusula GROUP BY

- OUtilizado para dividir uma relação em subgrupos
- Os subgrupos são divididos com base em atributos (colunas)

## Agrupando dados



### □Cláusula GROUP BY

- OUtilizado para dividir uma relação em subgrupos
- Os subgrupos são divididos com base em atributos (colunas)
- OAs funções de agregação atuam separadamente em cada subgrupo
- o Exemplo: qual a média dos salários de cada cargo SELECT cargo, avg(salario) AS media FROM empregado GROUP BY cargo

## Exemplo GROUP BY



▶ Exemplo: qual a média dos salários de cada cargo?

SELECT cargo, avg(salario) AS media FROM empregado GROUP BY cargo

codEmp	Nome	salario	cargo
001	Tiago	5000	Analista
002	João	3000	Analista
003	Rui	1500	Programador
004	Julio	2000	Programador
005	Ana	2500	Programador
006	Maria	2000	Programador
007	Carlos	7000	Diretor

TO DE DADOS 31

## Exemplo GROUP BY



Exemplo: qual a média dos salários de cada cargo?

SELECT cargo, avg(salario) AS media FROM empregado GROUP BY cargo

codEmp	Nome	salario	cargo			
001	Tiago	5000	Analista	I)		
002	João	3000	Analista		Cargo	Media
003	Rui	1500	Programador	i î	Analista	4000
004	Julio	2000	Programador		Programador	2000
005	Ana	2500	Programador	1 /	Diretor	7000
006	Maria	2000	Programador			
007	Carlos	7000	Diretor			

## Agrupando dados



### □Cláusula HAVING

- Define uma condição para a seleção de cada subgrupo
- Deve ser usado com o GROUP BY
- o Funcionamento semelhante ao WHERE
- WHERE analisa a condição por linha
- · HAVING analisa a condição por subgrupo

DE DADOS 3

### Exemplo GROUP BY c/HAVING



Exemplo: qual a média dos salários de cada cargo com mais de um empregado?

SELECT cargo, avg(salario) AS media FROM empregado GROUP BY cargo HAVING count(\*)>1

codEmp	Nome	salario	cargo
001	Tiago	5000	Analista
002	João	3000	Analista
003	Rui	1500	Programador
004	Julio	2000	Programador
005	Ana	2500	Programador
006	Maria	2000	Programador
007	Carlos	7000	Diretor

## Exemplo GROUP BY c/HAVING



Exemplo: qual a média dos salários de cada cargo com mais de um empregado?

SELECT cargo, avg(salario) AS media FROM empregado GROUP BY cargo HAVING count(\*)>1

codEmp	Nome	salario	cargo			
001	Tiago	5000	Analista	1		
002	João	3000	Analista		Cargo	Media
003	Rui	1500	Programador	i í	Analista	4000
004	Julio	2000	Programador		Programador	2000
005	Ana	2500	Programador	1		
006	Maria	2000	Programador	J	Não satisfaz a condição	
007	Carlos	7000	Diretor	<b>−</b> ‰→	do HAVING	

### Selects aninhadas



□O resultado de uma *select* é uma relação

o Pode ser usada como tal

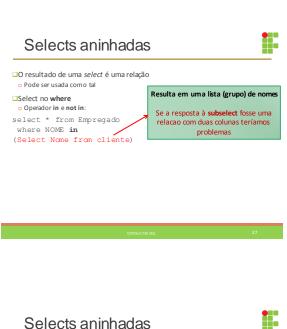
□Select no where

o Operador in e not in:

select \* from Empregado
 where NOME in

(Select Nome from cliente)

CONSULTAS SOL







□O resultado de uma select é uma relação

o Pode ser usada como tal

■Select no where o Operador in e not in:

select \* from Empregado

where NOME in

(Select Nome from cliente)

#### □Select no from

select S.CARGO, S.OTDE, SM from (select CARGO, count(CARGO) as QTDE, avg(SALARIO) as SM from empregado group by CARGO) S

where avg(SALARIO) > 2000



□O resultado de uma select é uma relação

o Pode ser usada como tal

#### ■Select no where

o Operador in e not in:

select \* from Empregado where NOME in

(Select Nome from cliente)

#### □Select no from

select S.CARGO, S.QTDE, SM from

(select CARGO, count(CARGO) as QTDE, avg(SALARIO) as SM from empregado group by CARGO) S

where avg(SALARIO) > 2000

Trata-se a resposta da

subselect como uma

tabela(relação) de nome S

### Selects aninhadas



#### ■O comando With

- o Guarda a resposta de uma consulta (tabela) em uma variável
- o É como se fosse criada uma View temporária
- · fica em memória apenas durante a execução da consulta
- o Ajuda na clareza de consultas complexas (dividir para conquistar!)

#### ■Exemplo

### with consulta as

(SELECT cargo, count(cargo) as qtde, avg(salario) as sm FROM empregado GROUP BY cargo)

SELECT \* FROM consulta WHERE sm>2000

## Junções



□É possível executar junções entre tabelas. (ou o produto cartesiano)

relacao1 JOIN relacao2

□ As tabelas devem ter um campo para serem vinculadas.

□Útil para vincular tabelas que se relacionam por chave estrangeira

■Tipos de junções

- oInterna
- Externa
- o Externa à esquerda
- o Externa à direita
- Natural

## Junções - Produto Cartesiano



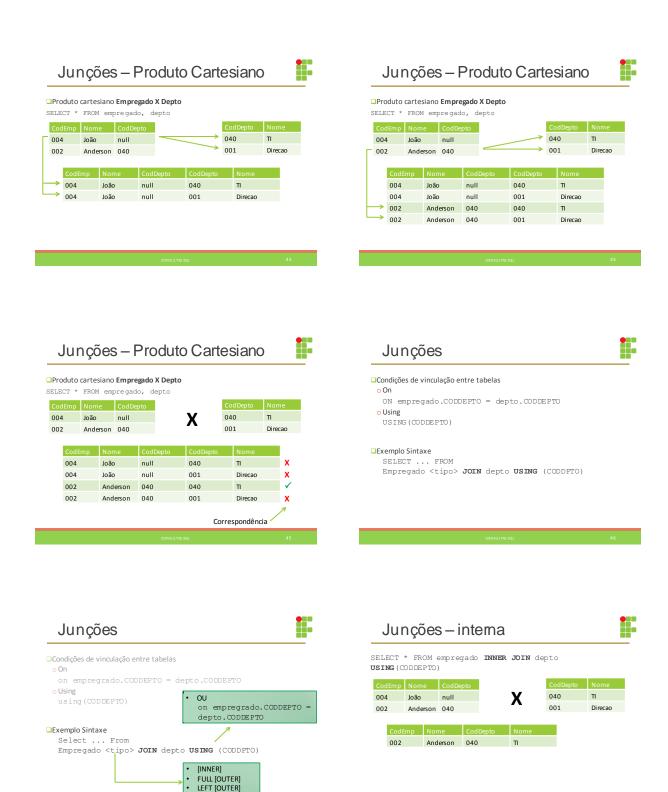
□Produto cartesiano Empregado X Depto

-	DELECT	FROM EM	niegado,	ue
	CodEmp	Nome	CodDepto	
	004	João	null	
	002	Anderson	040	

	CodDepto	Nome
Χ	040	TI
^	001	Direcao

CodEmp	Nome	CodDepto	CodDepto	Nome
004	João	null	040	TI
004	João	null	001	Direcao
002	Anderson	040	040	TI
002	Anderson	040	001	Direcao

7



RIGHT [OUTER] NATURAL





CONSULTAS SQL 53