

Consultas – Parte 1

Banco de Dados I

Consultas

- Estrutura básica

select ***A1, A2, ..., An***
from ***r1, r2, ..., rm***
where ***P***

A1, A2, ..., An: atributos

r1, r2, ...rm: relações

P é o predicado da consulta

Consultas

- Estrutura básica

select ***A1, A2, ..., An***

from ***r1, r2, ..., rm***

where ***P***

EMPREGADO

ENOME	<u>SSN</u>	DATANASC	ENDERECO	DNUMERO	codcargo
-------	------------	----------	----------	---------	----------

chave primária (p.k.)

DEPARTAMENTO

DNOME	<u>DNUMERO</u>	DGERSSN
-------	----------------	---------

DEPT_LOCALIZACOES

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

DEPENDENTE

<u>codigo</u>	nome	essn
---------------	------	------

PROJETO

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

TRABALHA_EM

<u>SSN</u>	<u>PNUMERO</u>	HORAS
------------	----------------	-------

CARGO

<u>codigo</u>	nome	descricao
---------------	------	-----------

Consultas

□ Select

- Lista os atributos
- Encontre o primeiro nome de todos os empregados

select enome

from empregado

- O * depois do select retorna todos os atributos

select *

from empregado

Consultas

□ Select

■ SQL permite duplicatas

□ Select enome from empregado

Joao (o Joao com SSN = 001)

Joao (o Joao com SSN = 002)

Joao (o Joao com SSN = 003)

■ Distinct (remover duplicata)

□ Select distinct enome from empregado

■ ALL

□ Não remover duplicata

□ Padrão

Consultas

□ Cláusula Where

- Define uma expressão condicional que identifica as tuplas que deverão ser recuperadas pela consulta
- Diferente: <>
- Selecione o numero de todos os empregados que moram no endereço: Rua 16, Casa 13.

Select ssn

From empregado

Where endereco = 'Rua 16, Casa 13'

Consultas

□ Cláusula Where

- Selecione o numero dos empregados que tenham o primeiro nome Joao e último nome Silva

Select ssn

From empregado

Where enome = 'Joao' and datanasc > '15/05/1975'

Selecione o numero dos empregados que não tenham nome igual a "Jose Barbosa"

Select ssn

From empregado

Where enome <> 'Jose Barbosa'

Consultas

❑ Cláusula Where

- SQL inclui o operador between
empregado(ssn,enome,salario)

Selecione o ssn de todos os empregados que ganham mais de 1500 e menos de 10000

Select ssn

From empregado

Where salario between 1500 and 10000

Consultas

empregado(ssn,enome,salario)

□ Cláusula from

- Define as tabelas que serão consideradas na consulta

Select ssn

From empregado

Where salario between 1500 and 10000

Consultas

empregado(ssn,nome,salario)

- ❑ Operações com String
 - String são especificadas com apóstrofo
nome = 'Maria'
 - SQL permite o uso de caracteres especiais
 - ❑ % (corresponde a qualquer sequencia substring)
 - ❑ _ (corresponde a qualquer caracter)
-

Consultas

empregado(ssn, nome, salario)

- ❑ Operações com String

enome ('M%')
Maria
Maria
Melise
Marta Fonseca
Mauro Batista

ssn ('C_')
C1
C2

Consultas

- ❑ Operações com String
 - Operador like
 - caracteres especiais devem ser usados com o operador like

Select *

From R

Where coluna like

Consultas

empregado(ssn, nome, salario)

❑ Operações com String

Select enome
From empregado
Where enome like
 'M%'

Select ssn
From empregado
Where ssn like 'C_'

enome
Maria
Maria
Melise
Marta Fonseca
Mauro Batista

ssn
C1
C2

Consultas

- ❑ Operações com String

- ❑ scape

Comando usado para
permitir que um
caracter especial seja
especificado dentro da
string de busca

Select ssn

From empregado

Where ssn like 'C\%00%'
escape '\'

ssn
C%001
C%0024
C%0023344

Consultas

- ❑ Operações com String
 - ❑ SQL permite outras operações com string
 - ❑ concatenação (usando “||”)
 - ❑ conversão de maiúscula em minúscula (e vice-versa)
 - ❑ upper(atributo ou string)
 - ❑ lower(atributo ou string)
 - ❑ Retornar tamanho da string
 - ❑ Extração de substrings e outras
 - ❑ Alguns funções variam de um SGBD para outro
-

Consultas

□ Ordenação

- SQL permite a ordenação das tuplas
- Liste todos os empregado em ordem alfabética do primeiro nome

Select *

From empregado

Order by enome

- Operadores desc e asc
 - Ordem decrescente e crescente (padrão)
 - Acrescentados depois do nome do atributo
-

Consultas

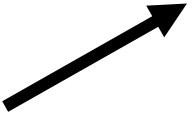
❑ Renomear

- SQL permite renomear atributos e relações


Select **dnome**
From departamento

Select **dnome** as **nome**
From departamento

OU (Select **dnome** **nome**
From departamento)



dnome
Pesquisa
RH
Graduação



nome
Pesquisa
RH
Graduação

Consultas

- Renomear
 - SQL permite renomear atributos e relações

Select d.dnome
From departamento d



**d é uma
variável
de tupla**

Exercícios – BD acadêmico

aluno(mat, cpf, identidade, nome, email)

professor(codigo, nome, cpf, email)

disciplina(codigo, nome, ementa)

turma(codigo, nome, disciplina, professor)

disciplina referencia disciplina(codigo)

professor referencia professor(codigo)

matricula(coda, codt, ano, semestre, nota, status)

coda referencia aluno (código)

codt referencia turma (codigo)

semestre = 1 ou semestre = 2

status = A (aprovado) ou R (reprovado) ou C (em curso)

Exercícios – BD acadêmico

- ❑ Retorne os dados de todos os alunos
 - ❑ Retorne o nome e o CPF dos alunos cujo email possui @unifei
 - ❑ Retorne o código das turmas que o professor com um determinado código foi alocado
 - ❑ Retorne o código das disciplinas cuja quantidade de créditos está em um determinado intervalo (os limites inferior e superior do intervalo você quem deve definir)
 - ❑ Retorne os dados das matrículas que estão em curso
-

Exercícios – BD empresa

- ❑ Retorne os dados de todos os funcionários
 - ❑ Retorne o nome e o CPF dos funcionários cujo email possui @unifei no endereço
 - ❑ Retorne o código dos projetos que o funcionário com um determinado código foi alocado
 - ❑ Retorne o código dos projetos e dos funcionários de todas as alocações que iniciaram entre as datas x e y (valor de x e y você é quem deve definir)
-

FEMPREGADO

ENOME	<u>SSN</u>	DATANASC	ENDERECO	DNUMERO	codcargo
-------	------------	----------	----------	---------	----------

chave primária (p.k.)



DEPARTAMENTO

DNOME	<u>DNUMERO</u>	DGERSSN
-------	----------------	---------

DEPT_LOCALIZACOES

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

DEPENDENTE

<u>codigo</u>	nome	essn
---------------	------	------

PROJETO

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

TRABALHA_EM

<u>SSN</u>	<u>PNUMERO</u>	HORAS
------------	----------------	-------

CARGO

<u>codigo</u>	nome	descricao
---------------	------	-----------



Exercícios – BD empresa 2

- ❑ Retorne todos os projetos cujo nome é Gerência de qualidade de dados
- ❑ Retorne todos os empregados que contém o nome da rua Teldomiro Santiago no endereço
- ❑ Retorne o nome completo de os empregados que nasceram antes de 10/01/2001
- ❑ Retorne o código dos projetos nos quais o ~~funcionario com SSN = E00120~~ trabalhou mais de 5 horas.

BD Admimissão

NrMatric	NmFunc	DtAdm	Sexo	CdCargo	CdDepto
0001	Maria	10/12/02	F	C1	D1
0002	João	11/05/07	M	C2	D2
0003	Paulo	12/08/05	M	C3	D1
0004	Fábio	04/06/06	M	C3	D1

CdCargo	NmCargo	Salário
C1	Gerente	10000
C2	Auxiliar de escritório	1500
C3	Analista de Sistemas	6000

CdDepto	NmDepto	Ram al
D1	Informática	1301
D2	Recursos Humanos	1302
D3	Contabilidade	1303

Produto Cartesiano

□ Retorne o nome e salário dos funcionários

- Nome é um atributo da relação FUNCIONÁRIO
 - Salário é um atributo da relação CARGO
 - Solução: concatenar atributos da relação FUNCIONÁRIO com relação CARGO
 - Produto cartesiano (FUNCIONARIO x CARGO)
 - FUNCIONARIO x CARGO
 - Possui tuplas que logicamente não fazem sentido
 - Dados da Maria que é uma gerente concatenados com os detalhes do cargo Analista de Sistemas
 - Solução: selecionar somente aquelas tuplas logicamente viáveis
-

Consulta – Produto Cartesiano

- Retornar a combinação de funcionário com cargo (todas as tuplas de funcionário combinadas a todas as tuplas de cargo)

```
Select *  
from funcionario f, cargo c
```

NrMatric	NmFunc	DtAdm	Sexo	F.CdCargo	C.CdCargo	NmCargo	Salário
0001	Maria	10/12/02	F	C1	C1	Gerente	10000
0001	Maria	10/12/02	F	C1	C2	Auxiliar de escritório	1500
0001	Maria	10/12/02	F	C1	C3	Analista de Sistemas	6000
0002	João	11/05/07	M	C2	C1	Gerente	10000
0002	João	11/05/07	M	C2	C2	Auxiliar de escritório	1500
0002	João	11/05/07	M	C2	C3	Analista de Sistemas	6000
0003	Paulo	12/08/05	M	C3	C1	Gerente	10000
0003	Paulo	12/08/05	M	C3	C2	Auxiliar de escritório	1500
0003	Paulo	12/08/05	M	C3	C3	Analista de Sistemas	6000
0004	Fábio	04/06/06	M	C3	C1	Gerente	10000
0004	Fábio	04/06/06	M	C3	C2	Auxiliar de escritório	1500
0004	Fábio	04/06/06	M	C3	C3	Analista de Sistemas	6000

NrMatric	NmFunc	DtAdm	Sexo	F.CdCargo	C.CdCargo	NmCargo	Salário
0001	Maria	10/12/02	F	C1	C1	Gerente	10000
0001	Maria	10/12/02	F	C1	C2	Auxiliar de escritório	1500
0001	Maria	10/12/02	F	C1	C3	Analista de Sistemas	6000
0002	João	11/05/07	M	C2	C1	Gerente	10000
0002	João	11/05/07	M	C2	C2	Auxiliar de escritório	1500
0002	João	11/05/07	M	C2	C3	Analista de Sistemas	6000
0003	Paulo	12/08/05	M	C3	C1	Gerente	10000
0003	Paulo	12/08/05	M	C3	C2	Auxiliar de escritório	1500
0003	Paulo	12/08/05	M	C3	C3	Analista de Sistemas	6000
0004	Fábio	04/06/06	M	C3	C1	Gerente	10000
0004	Fábio	04/06/06	M	C3	C2	Auxiliar de escritório	1500
0004	Fábio	04/06/06	M	C3	C3	Analista de Sistemas	6000

Consulta

□ Junção

- Operação de banco de dados que envolve:
 - Concatenação de tuplas diferentes (podem ou não ser de tabelas diferentes)
 - Seleção das tuplas com significado relevante
 - Para seleção, é necessário definir um critério (predicado da junção)
 - Critério pode ser definido
 - Cláusula Where
 - Cláusula JOIN
-

Consulta – Junção

Critério de junção definido na WHERE

- Selecionar somente aquelas tuplas logicamente viáveis
 - Tuplas com o mesmo código de cargo

```
Select funcionario.NmFunc, cargo.Salario  
from funcionario, cargo  
where funcionario.CdCargo = Cargo.CdCargo
```

NmFunc	Salário
Maria	10000
João	1500
Paulo	6000
Fábio	6000

Consultas

□ Cláusula Where - Junção

- Selecione o nome dos empregados que trabalham no departamento 'Pesquisa'

`select funcionario.NmFunc`

`from funcionario, departamento`

`where departamento.NmDept = 'Pesquisa'`

`and funcionario.CdDepto =`

`departamento.CdDepto`

cliente(codigo,nome,etc)

dependente(codigo,nome, codCliente, etc ...

- ❑ **Qual o nome do cliente referente ao dependente com codigo = D0001?**

Select nome

From cliente, dependente

Where codCliente = codigo

and codigo = "D0001"

Qual "nome" será retornado? Qual
"codigo" será considerado nas
cláusulas?

Consultas

- Renomear

Select c.nome

From cliente as c, dependente as d

where c.codigo = d.codCliente

and d.codigo = "D0001"

- c e d são variáveis de tuplas
-

Consultas

□ Renomear

***select c.nome as nome_cliente
from cliente c, dependente d
where d.codigo = "D0001"
and c.codigo = d.codCliente***

- c e d são variáveis de tuplas
 - no resultado, o atributo nome de cliente é exibido como nome_cliente
-

Exercícios – BD academico

- ❑ Retorne o nome das turmas com o nome dos seus respectivos professores
 - ❑ Retorne o nome das turmas do professor cujo CPF é igual a um determinado valor
 - ❑ Retorne o nome dos alunos e código das turmas somente para as matrículas com status reprovado
-

Exercícios – BD empresa

- ❑ Retorne o nome dos funcionários com os nome do seus repectivos departamentos
 - ❑ Retorne o nome do projeto e código dos funcionários alocados no projeto
 - ❑ Retorne o nome do funcionários com seus repectivos números de telephone
-

Exercícios – BD empresa2

- ❑ Retorne o nome dos empregados e o nome do departamento onde trabalham ordenados pelo nome do empregado (crescente) e nome do departamento (decrescente)
 - ❑ Retorne o nome do departamento e o nome do seu gerente
 - ❑ Retorne o nome do gerente do departamento RH
 - ❑ Retorne o nome dos projetos nos quais o funcionário com SSN = E00120 trabalhou mais de 5 horas.
-

cliente(nome_cliente, rua, cidade)

depositante(numero_conta, nome_cliente)

Consultas

- Junção com cláusula JOIN
 - Tabelas juntadas
 - Join (usado para especificar a junção na cláusula from)
 - Retorne a cidade do cliente da conta 'CT01'

Select c.cidade

From **cliente c join depositante d** on

c.nome_cliente = d.nome_cliente

where d.numero_conta = 'CT01'

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

Consultas

□ Junção

- Tabelas juntadas
- Join (usado para especificar a junção na cláusula from)
- Retorne o nome do cliente da conta 'CT01'
- Select nome_cliente

From **cliente c join depositante d** on
c.nome_cliente =
d.nome_cliente

where d.numero_conta = 'CT01'

Tabela juntada

Condição de junção

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

Consultas

□ Junção

Condição de junção no from com "on"

- Retorne o nome do cliente da conta 'CT01'

Select nome_cliente

**From cliente c join depositante d on
c.nome_cliente = d.nome_cliente**

where d.numero_conta = 'CT01'

Select nome_cliente

From cliente c, depositante d

**where d.numero_conta = 'CT01' and
c.nome_cliente = d.nome_cliente**

Condição de junção no where

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

Consultas

□ Tipos de Junção

- Inner join
- Retorna somente tuplas relacionadas
- Cláusula on especifica o critério da consulta
- Select cidade

From **cliente c inner join depositante d**
on c.nome_cliente = d.nome_cliente
where d.numero_conta = 'CT01'

cliente(nome_cliente, rua, cidade)

depositante(numero_conta, nome_cliente)

Consultas

□ Tipos de Junção

■ Natural join

- Não é necessário especificar o critério de junção. A junção é realizada comparando os atributos iguais das relações especificadas no join (os atributos devem ter o mesmo nome)

■ Select cidade

From **cliente c natural join depositante d**
where d.numero_conta = 'CT01'

Select *

From funcionario f natural join tempIntegral t

funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

tempIntegral

<i>nome_funcionário</i>	<i>Rua</i>	<i>cidade</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500

■ Junção externa

- ❑ Extensão da operação junção que evita a perda de informações
- ❑ Faz a junção natural e acrescenta ao resultado as tuplas de uma relação que não se relacionam com as tuplas da outra relação

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

■ Junção externa esquerda (primeira relação)

- Junção natural mais as tuplas da relação da esquerda que não estão relacionadas com a segunda relação
- funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

tempoIntegral

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

Select *

From funcionario left outer join tempIntegral

funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

tempIntegral

<i>nome_funcionário</i>	<i>Rua</i>	<i>cidade</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death Valley	<i>nulo</i>	<i>nulo</i>

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

Consultas

- ❑ Tipos de Junção
 - ❑ Left outer join
 - Retorna o nome de todos os cliente com número da conta, incluindo os cliente que não possuem conta
 - ❑ Select nome_cliente, numero_conta
From **cliente c left outer join**
depositante d
on c.nome_cliente = d.nome_cliente
-

Select *

From funcionario right outer join tempIntegral

■ Junção externa direita (segunda relação)

- Junção natural mais as tuplas da relação da direita que não estão relacionadas com a primeira relação funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

tempIntegral

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

Select *

From funcionario right outer join tempIntegral

funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

tempIntegral

<i>nome_funcionário</i>	<i>Rua</i>	<i>cidade</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Gates	<i>Nulo</i>	<i>nulo</i>	Redmond	5300

Select *

From funcionario full outer join tempIntegral

■ Junção externa completa

- Junção natural mais as tuplas das duas relações que não estão relacionadas

funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

tempIntegral

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

Select *

From funcionario full outer join tempIntegral

funcionario

<i>nome_funcionário</i>	<i>rua</i>	<i>Cidade</i>
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

<i>nome_funcionário</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500

tempIntegral

<i>nome_funcionário</i>	<i>Rua</i>	<i>cidade</i>	<i>nome_agência</i>	<i>salário</i>
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death Valley	<i>nulo</i>	<i>nulo</i>
Gates	<i>Nulo</i>	<i>nulo</i>	Redmond	5300

Consultas – BD acadêmico

- ❑ Retorne o nome do aluno e o código das turmas que está matriculado considerando incluindo os alunos que não estão matriculados em turmas.
 - ❑ Retorne o nome da disciplina e o código das turmas associadas incluindo as disciplinas que não possuem turma.
-

Consultas – BD empresa

- ❑ Retorne o nome do departamento e o nome dos seus respectivos funcionários incluindo os departamentos que não possuem funcionário
 - ❑ Retorne o nome do projeto e o código dos funcionários alocados incluindo os projetos que não possuem funcionários alocados
-

Consultas

depositante(nome_cliente, numero_conta)

tomador(nome_cliente, numero_emprestimo)

❑ Operações de conjunto

❑ União

- Encontre todos os clientes do banco que possuem um empréstimo, uma conta ou as duas coisas no banco
 - **(select nome_cliente from depositante)**
union
(select nome_cliente from tomador)
-

Consultas

depositante(nome_cliente, numero_conta)

tomador(nome_cliente, numero_emprestimo)

- ❑ Operações de conjunto

- ❑ Intersecção

- Encontre todos os clientes do banco que possuem um empréstimo e uma conta no banco

- **(select distinct nome_cliente from depositante)**
intersect

- (select distinct nome_cliente from tomador)**

Consultas

depositante(nome_cliente, numero_conta)

tomador(nome_cliente, numero_emprestimo)

❑ Operações de conjunto

❑ Subtração

- Encontre todos os clientes do banco que possuem uma conta no banco e não possuem empréstimo
 - **(select distinct nome_cliente from depositante) except (select nome_cliente from tomador)**
-

EMPREGADO

ENOME	<u>SSN</u>	DATANASC	ENDERECO	DNUMERO	codcargo
-------	------------	----------	----------	---------	----------

chave primária (p.k.)

DEPARTAMENTO

DNOME	<u>DNUMERO</u>	DGERSSN
-------	----------------	---------

DEPT_LOCALIZACOES

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

DEPENDENTE

<u>codigo</u>	nome	essn
---------------	------	------

PROJETO

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

TRABALHA_EM

<u>SSN</u>	<u>PNUMERO</u>	HORAS
------------	----------------	-------

CARGO

<u>codigo</u>	nome	descricao
---------------	------	-----------

Exercícios – BD empresa2

- ❑ Retorne todos os nomes de empregado e/ou dependente
 - ❑ Retorne todos os nomes de empregado que não são nomes de dependentes
 - ❑ Retorne todos os nomes de empregado que também são nomes de dependentes
-

Consultas

□ Valores nulos

- SQL permite usar parametro null nos predicados das consultas
 - Null (valor desconhecido ou que não existe)

```
select número_empréstimo  
  from empréstimo  
  where quantia is null
```
 - Is not Null é usado para testar se o valor não é nulo

```
select número_empréstimo  
  from empréstimo  
  where quantia is not null
```
-

Consultas

❑ Valores nulos

- Qualquer expressão aritmética envolvendo um valor nulo é nula

select quantia + 1000

from empréstimo

*Se quantia for nulo, então o resultado é nulo
(não retorna nada)*

Consultas

- Funções agregadas
 - avg: **valor médio**
 - min: **valor mínimo**
 - max: **valor máximo**
 - sum: **soma dos valores**
 - count: **número de valores**
-

Quantos Perryridge?

Média (saldo)

Soma (saldo)

Máximo ou Mínimo (saldo)

Consultas

❑ Função agregada

<i>nome_agência</i>	<i>número_conta</i>	<i>saldo</i>
Centro	A-102	400
Centro	A-201	900
Alvorada	A-217	750
Alvorada	A-215	750
Planalto	A-222	700

Consultas

conta(nome_agencia, numero_conta, saldo)

- ❑ Encontre a média dos saldos
 - **select avg** (*saldo*)
from *conta*
 - ❑ Encontre a média dos saldos das contas da agência Centro
 - **select avg** (*saldo*)
from *conta*
where *nome_agencia* = 'Centro'
-

Consultas

conta(nome_agencia, numero_conta, saldo)

- ❑ Encontre a soma dos saldos
 - **select sum** (*saldo*)
from *conta*
 - ❑ Encontre a soma dos saldos das contas da agência Centro
 - **select sum**(*saldo*)
from *conta*
where *nome_agencia* = `Centro`
-

Consultas

conta(nome_agencia, numero_conta, saldo)

- ❑ Encontre o valor máximo dos saldos
 - **select max** (*saldo*)
from *conta*

 - ❑ Encontre o valor máximo dos saldos das contas da agência Centro
 - **select max** (*saldo*)
from *conta*
where *nome_agencia* = `Centro`
-

Consultas

conta(nome_agencia, numero_conta, saldo)

- ❑ Retorne a quantidade de contas

select count(*) from conta

- ❑ *Retorne a quantidade de contas da agencia Centro*

select count(*) from conta

where nome_agencia = 'Centro'

Consultas

cliente(nome_cliente, rua, cidade)

- Encontre o número de tuplas da relação cliente
 - **select count(*)**
from cliente
 - *SQL permite usar count com o distinct*
 - *cliente(codigo, nome, rua, cidade)*
 - *Quantos nomes diferentes existem na relação cliente?*
 - **select count distinct (nome)**
from cliente
-

Consultas

depositante(nome_cliente, numero_conta)

- ❑ Retorne a quantidade de depósitos realizados
 - **Select count(*) from depositante**
 - ❑ Retorne o a quantidade de clientes que já depositaram em alguma conta
 - **select count distinct (nome_cliente) from depositante**
-

Consultas

- Funções agregadas com grupamento
 - Group by permite agrupar tuplas de uma relação considerando determinado (s) atributo (s) e usar funções agregadas considerando o grupamento

<i>nome_agência</i>	<i>número_conta</i>	<i>saldo</i>	
Centro	A-102	400	→ Grupo 1
Centro	A-201	900	
Alvorada	A-217	750	→ Grupo 2
Alvorada	A-215	750	
Planalto	A-222	700	→ Grupo 3

Consultas

- ❑ Funções agregadas com agrupamento
 - `select nome_agencia,max(saldo)`
`from conta`
`group by(nome_agencia)`

<i>nome_agência</i>	<i>número_conta</i>	<i>saldo</i>	
Centro	A-102	400	→ Grupo 1
Centro	A-201	900	
Alvorada	A-217	750	→ Grupo 2
Alvorada	A-215	750	
Planalto	A-222	700	→ Grupo 3

Consultas

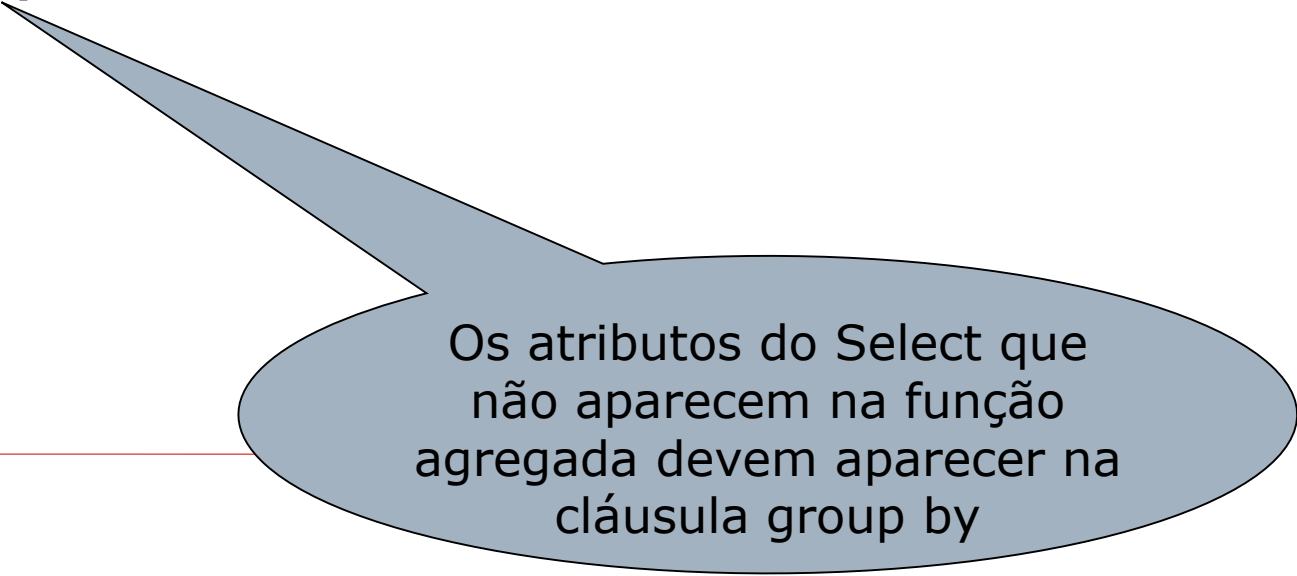
- ❑ Funções agregadas com agrupamento
 - `select nome_agencia,soma(saldo)`
`from conta`
`group by(nome_agencia)`

<i>nome_agência</i>	<i>número_conta</i>	<i>saldo</i>	
Centro	A-102	400	→ Grupo 1
Centro	A-201	900	
Alvorada	A-217	750	→ Grupo 2
Alvorada	A-215	750	
Planalto	A-222	700	→ Grupo 3

Consultas

cliente(codigo,nome,rua,numero,bairro)

- ❑ Funções agregadas com grupamento
 - Retorne a quantidade de clientes por rua
- ```
select rua, count(codigo)
from cliente
group by (rua)
```



Os atributos do Select que não aparecem na função agregada devem aparecer na cláusula group by

depositante(nome\_cliente, numero\_conta)  
conta(numero\_conta, nome\_agencia, saldo)

---

- ❑ Funções agregadas com agrupamento
  - Encontre o número de depositantes por agência

**Select** nome\_agencia, **count(distinct**  
nome\_cliente)

**From** conta, depositante

**Where** conta.numero\_conta =  
depositante.numero\_conta

**Group by** (nome\_agencia)

---

# Consultas

conta(numero\_conta,nome\_agencia,saldo)

---

## ❑ Funções agregadas

- SQL permite especificar um critério para os grupos de tuplas resultantes da consulta
  - Clausula having
  - Retorne a média de saldos das agencias considerando somente as agências com médias maiores que 1200
  - **select** *nome\_agencia*, **avg** (*saldo*)  
    **from** *conta*  
    **group by** *nome\_agência*  
    **having** **avg** (*saldo*) > 1200
-

# Consultas

conta(numero\_conta,nome\_agencia,saldo)

---

## □ Funções agregadas

- Clausula having
  - Where X having
    - O predicado da cláusula where é considerado primeiro
  - **select** *nome\_agencia*, **avg** (*saldo*)  
    **from** *conta*  
  
    **where** numero\_conta like `C01%`  
  
    **group by** *nome\_agência*  
        **having avg** (*saldo*) > 1200
-

# Exercícios – BD empresa2

## EMPREGADO

|       |            |          |          |         |          |
|-------|------------|----------|----------|---------|----------|
| ENOME | <u>SSN</u> | DATANASC | ENDERECO | DNUMERO | codcargo |
|-------|------------|----------|----------|---------|----------|

chave primária (p.k.)

---

## DEPARTAMENTO

|       |                |         |
|-------|----------------|---------|
| DNOME | <u>DNUMERO</u> | DGERSSN |
|-------|----------------|---------|

## DEPT\_LOCALIZACOES

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

## DEPENDENTE

|               |      |      |
|---------------|------|------|
| <u>codigo</u> | nome | essn |
|---------------|------|------|

## PROJETO

|       |                |              |      |
|-------|----------------|--------------|------|
| PNOME | <u>PNUMERO</u> | PLOCALIZACAO | DNUM |
|-------|----------------|--------------|------|

## TRABALHA\_EM

|            |                |       |
|------------|----------------|-------|
| <u>SSN</u> | <u>PNUMERO</u> | HORAS |
|------------|----------------|-------|

## CARGO

|               |      |           |
|---------------|------|-----------|
| <u>codigo</u> | nome | descricao |
|---------------|------|-----------|

---

# Exercícios – BD empresa2

---

- ❑ Retorne a soma de todas as horas trabalhadas.
  - ❑ Retorne a soma de todas as horas trabalhadas por projeto (número do projeto).
  - ❑ Retorne a soma de todas as horas trabalhadas por empregado (ssn do empregado).
  - ❑ Retorne a soma das horas trabalhadas por empregado (ssn) considerando somente aqueles que trabalharam mais de 60 horas.
  - ❑ Retorne o valor da quantidade máxima de horas trabalhadas.
  - ❑ Retorne o valor da quantidade máxima de horas trabalhadas por empregado.
  - ❑ Retorne o valor da quantidade máxima de horas trabalhadas no projeto de código P001.
  - ❑ Retorne o código do departamento e a quantidade de funcionários de cada um.
  - ❑ Retorne o nome do departamento e a quantidade de funcionários de cada um.
  - ❑ Retorne o nome do departamento e a quantidade de funcionários de cada um considerando somente os departamentos que possuem a string PRP no nome
  - ❑ Retorne o nome do departamento e a quantidade de funcionários de cada um considerando somente os departamentos que possuem a string PRP no nome e os departamentos que possuem mais de 5 empregados
-