

REVISÃO DE SQL

PARTE 1

Sérgio Mergen

Esquema de Exemplo

Proj (idP, nome, duracao, custo, idD)

idD referencia depto

Func (idF, nome, salario, idD, idChefe)

idD referencia depto

idChefe referencia Func

Depto (idD, nome, predio, idDiretor)

idDiretor referencia Func

Aloc (idP, idF, funcao)

idP referencia proj

idF referencia func

Cláusula Select

Exemplo: encontre os nomes e a duração de todos projetos:

Proj				
idP	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

select *nome, duracao*
from *proj*

Resposta	
nome	duracao
ABC	3
Lucrei	2

Cláusula Select

Retornar o nome dos funcionários e o seu salário anual

Func			
idF	nome	salario	idD
1	Marcos	3.000	1
2	Ana	4.000	2
3	João	2.500	1

select *nome, salario * 12*
from *func*

Func	
nome	salario * 12
Marcos	36.000
Ana	48.000
João	30.000

Cláusula From

Encontre o produto cartesiano dos projetos e departamentos

Proj				
idP	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

Depto		
idD	nome	predio
1	TI	3
2	Marketing	2

```
select *  
from depto, proj
```

Resposta							
idP	nome	duracao	custo	idD	idD	nome	predio
1	ABC	3	12.000	1	1	TI	3
1	ABC	3	12.000	1	2	Marketing	2
2	Lucrei	2	30.000	1	1	TI	3
2	Lucrei	2	30.000	1	2	Marketing	2

Cláusula From

Encontre o nome de cada departamento e o nome de todos os respectivos projetos desses departamentos

Proj				
idProj	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

Depto		
idD	nome	predio
1	TI	3
2	Marketing	2

```
select  depto.nome, proj.nome
from    depto, proj
where   depto.idD = proj.idD
```

Resposta	
Depto.nome	Projeto.nome
TI	ABC
TI	Lucrei

Cláusula From

Encontre o nome de cada departamento e o nome de todos os respectivos projetos desses departamentos **com duração estimada de 3 anos**.

Proj				
id	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1

Depto		
idD	nome	predio
1	TI	3
2	Marketing	2

```
select  depto.nome, proj.nome
from    depto, proj
where   depto.idD = proj.idD
and     duracao = 3
```

Resposta	
Depto.nome	Projeto.nome
ABC	TI

A operação de Renomeação

Encontre o nome de cada departamento e de seus respectivos diretores, renomeando as colunas para DEPTO e DIRETOR

Func			
idF	nome	salario	idD
1	Marcos	3.000	1
2	Ana	4.000	2
3	João	2.500	1

Depto			
idD	nome	predio	idDiretor
1	TI	3 - centro	1
2	RH	2 - norte	2

```
select depto.nome as DEPTO, func.nome as DIRETOR  
from depto, func  
where depto.idDiretor = func.idF
```

Resposta	
DEPTO	DIRETOR
TI	Marcos
RH	Ana

Variáveis de tupla

Para cada projeto, apresente seu nome, nome dos funcionários alocados e a função desses funcionários no projeto.

Func	
idF	nome
1	Marcos
2	Ana
3	João

Proj	
idP	nome
1	Lucrei
2	Caos

Aloc		
idP	idF	função
1	1	Coordenador
1	3	Analista

```
select  p.nome, f.nome, funcao
from    proj as p, func as f, alocacao a
where   a.idP = p.idP
and     a.idF = f.idF
```

Resposta		
p.nome	f.nome	Funcao
Lucrei	Marcos	Coordenador
Lucrei	João	Analista

Operações com Strings (LIKE)

Ex. Listar todos os nomes de departamentos cuja descrição do prédio inclua a subcadeia “centro”.

Deppto		
idD	nome	predio
1	TI	3 – centro
2	RH	5 - centro
3	Marketing	3 - norte

```
select *  
from depto  
where predio like '%centro%'
```

Resposta		
idD	nome	predio
1	TI	3 – centro
2	RH	5 - centro

Ordenando as tuplas

Ex. Listar em ordem de duração os nomes de todos os projetos do departamento de TI

Proj				
idP	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	1
3	Show	1	10.000	2

Depto		
idD	nome	predio
1	TI	3
2	Marketing	2

```
select      *  
from        proj p, depto d  
where       p.idD = d.idD and d.nome = 'TI'  
order by    duracao
```

Resposta							
idP	nome	duracao	custo	idD	idD	nome	predio
2	Lucrei	2	30.000	1	1	TI	3
1	ABC	3	12.000	1	1	TI	3

AGREGAÇÃO

Funções de Agregação

- Estas funções aplicam-se a conjuntos de valores de uma coluna de uma relação, devolvendo um valor

avg: valor médio

min: valor mínimo

max: valor máximo

sum: soma dos valores

count: número de valores

Funções de Agregação (cont.)

Determinar o salário médio dos funcionários.

Func		
idF	nome	salario
1	Marcos	10000
2	Ana	6000
3	João	5000

```
select avg (salario)  
from func
```

Resposta
Avg(salario)
7000

Funções de Agregação (cont.)

Calcular o número de departamentos.

Deppto		
idD	nome	idDiretor
1	TI	1
2	RH	5
3	Marketing	3

```
select count (*)  
from depto
```

Resposta
Count(*)
3

Funções de Agregação (cont.)

Encontrar o número de funcionários alocados em projetos.

Aloc		
idP	idF	função
1	1	Coordenador
1	3	Analista
2	1	Coordenador

```
select count (distinct idF)  
from aloc
```

Resposta
Count(distinct idF)
2

Funções de agregação – Group By

Ex. Listar o salário médio por departamento.

Func			
idF	nome	Salario	idD
1	Marcos	10000	1
2	Ana	6000	2
3	João	5000	1

```
select    idD, avg (salario)
from      funci
group by  idD
```

Resposta	
idD	avg (salario)
1	7500
2	6000

Funções de Agregação – Cláusula Having

Ex. Listar os nomes de todas os departamentos cuja média de salário seja superior a R\$5.000.

Func			
idF	nome	salario	idD
1	Marcos	10000	1
2	Ana	6000	2
3	João	5000	1

Depto			
idD	nome	predio	idDiretor
1	TI	3 - centro	1
2	RH	2 - norte	2

```
select      d.nome, avg (salario)
from        func f, depto d
where       f.idD = d.idD
group by    d.nome
having      avg (salario) > 7000
```

Resposta	
nome	Avg(salario)
TI	7500

JUNÇÕES

Tipos de Junção

- Junção interna: os registros das tabelas precisam casar entre si.
 - (Inner) join
 - Natural Join
- Junção cruzada: cada registro de um lado da junção casa com todos registros do outro lado
 - Cross join
- Junção externa: um registro de uma tabela não necessariamente precisa ter uma correspondência
 - Veremos na próxima aula

Operações de Junção

- Condição de junção
 - define quais as tuplas que são combinadas nas duas tabelas
 - Cross join e natal join não usam condições.

assim	Junções com condição	Condições de Junção
	(inner) join	on <predicate> using (A_1, A_2, \dots, A_n)
ou assim	Junções sem condição	
	Cross join Natural join	

Exemplo

Proj				
idP	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	
4	Caos	10	100.000	1

Depto		
idD	nome	predio
1	TI	3
2	Marketing	2
3	RH	2

*SELECT * FROM
proj p INNER JOIN depto d ON p.idD = d.idD*

Resposta							
idP	nome	duracao	custo	idD	idD	nome	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3

Exemplo

Proj				
idP	nome	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	
4	Caos	10	100.000	1

Depto		
idD	nome	predio
1	TI	3
2	Marketing	2
3	RH	2

*SELECT * FROM
proj p INNER JOIN depto d USING (id)*

Resposta							
idP	nome	duracao	custo	idD	idD	nome	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3

Exemplo

Proj				
idP	nomeP	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2
3	Genesis	2	15.000	
4	Caos	10	100.000	1

Depto		
idD	nomeD	predio
1	TI	3
2	Marketing	2
3	RH	2

*SELECT * FROM
proj p NATURAL JOIN depto d*

Resposta							
idP	nomeP	duracao	custo	idD	idD	nomeD	predio
1	ABC	3	12.000	1	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2
4	Caos	10	100.000	1	1	TI	3

Exemplo

Proj				
idP	nomeP	duracao	custo	idD
1	ABC	3	12.000	1
2	Lucrei	2	30.000	2

Depto		
idD	nomeD	predio
1	TI	3
2	Marketing	2

*SELECT * FROM
proj p CROSS JOIN depto d*

Resposta							
idP	nomeP	duracao	custo	idD	idD	nomeD	predio
1	ABC	3	12.000	1	1	TI	3
1	ABC	3	12.000	1	2	Marketing	2
2	Lucrei	2	30.000	2	1	TI	3
2	Lucrei	2	30.000	2	2	Marketing	2

Junção versus Produto Cartesiano

- Listar os dados de projeto e do seu respectivo departamento para projetos com custo estimado superior a R\$15.000.

```
select  p.*, d.nome  
from    proj p, depto d  
where   p.idD = d.idD  
and     custo > 15000
```

versus

```
select  p.*, d.nome  
from    proj p natural inner join depto d  
where   custo > 15000
```

- A última separa claramente onde ir buscar os dados de onde se colocam condições “de filtragem” (seleção)

Atividade Individual

- Os exercícios a seguir são baseados em dados representados pelo esquema abaixo

movie (movie_id, title)

person (person_id, person_name)

Movie_cast (movie_id, person_id, character_name, cast_order)

movie_id referencia movie

person_id referencia person

- O banco de dados está disponível no moodle

Atividade Individual

Com base no modelo relacional do slide anterior, escreva consultas em SQL para responder o que se pede abaixo:

1. Exibir a quantidade de filmes lançados em cada ano. Ordenar o resultado por ano
2. Exibir o nome dos filmes que tenham mais do que 20 personagens no elenco. Mostrar a quantidade de personagens
3. Para cada ator, mostrar seu nome, e os anos do seu primeiro e do seu último filme lançados

REVISÃO DE SQL

PARTE 1

Sérgio Mergen