



Banco de Dados

Exercícios Mecânica

Prof. André Ulisses
andre.ulisses@edu.sc.senai.br

SELECT

Seleção dos dados

***, Função
Campos**

FROM

Origem dos dados

Tabela(s)

+ de uma tabela tem que ter Junção, sempre ligando pela chave estrangeira.

WHERE

Filtro do resultado

operação lógica

GROUP BY

Agrupar resultado

campos

Obrigatório sempre que tiver função de agregação (**SUM, MAX, MIN, AVG E COUNT**)
Os campos fora da função devem estar aqui

HAVING

Filtro usando função de agregação

função de agregação

ORDER BY

Ordenar resultado

campos

Consultas Joins

Nos Bancos de Dados relacionais a relação entre tabelas é o ponto fundamental da sua existência. Por isso muitas informações são acessadas através de junções entre tabelas.

Os relacionamentos são organizados através das chaves estrangeiras (FK).

Atenção: Tem-se sempre a preferencia ao juntar duas tabelas à partir dos campos utilizados na foreign key existente entre as tabelas.

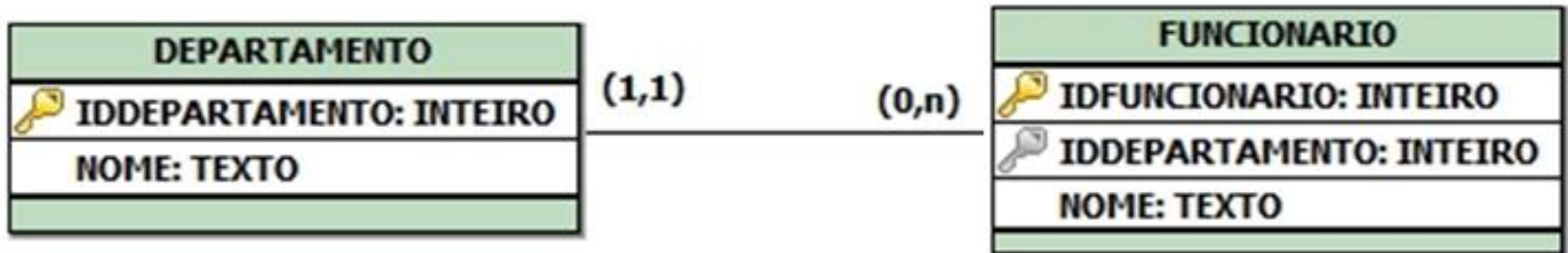
Consultas Joins

Quando precisamos acessar as informações entre duas tabelas, o SQL nos fornece um recurso para acesso simultâneo a tabelas que encontram-se relacionadas. Esse recurso é identificado como junção (Join).

Uma junção de tabelas cria uma pseudo-tabela derivada de duas ou mais tabelas de acordo com as regras especificadas, e que são parecidas com as regras da teoria dos conjuntos

Atenção: Tem-se preferência ao juntar duas tabelas à partir dos campos utilizados na **foreign key** existente entre as tabelas.

SQL - CONSULTA DE DADOS



```
CREATE TABLE DEPARTAMENTO (  
    IDDEPARTAMENTO INT PRIMARY KEY auto_increment,  
    NOME VARCHAR(100));  
  
CREATE TABLE FUNCIONARIO (  
    IDFUNCIONARIO INT PRIMARY KEY auto_increment,  
    IDDEPARTAMENTO INT,  
    NOME VARCHAR(100),  
    FOREIGN KEY (IDDEPARTAMENTO) REFERENCES DEPARTAMENTO (IDDEPARTAMENTO)  
);
```

Consultas Joins

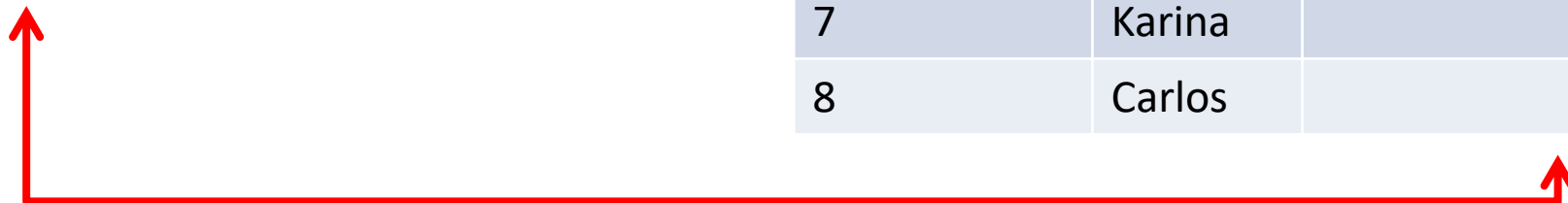
Exemplo:

Tabela **Departamento**

IDDEPARTAMENTO	NOME
1	Administração
2	Marketing
3	Financeiro
4	Cobrança
5	Logística
6	manutenção

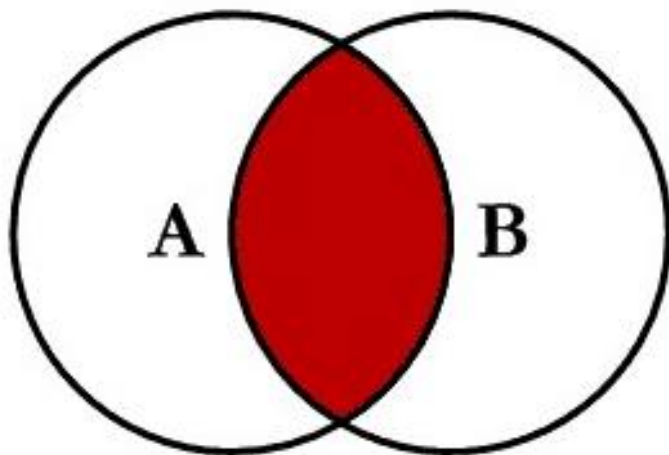
Tabela **Funcionário**

IDFUNCIONARIO	NOME	IDDEPARTAMENTO
1	João	1
2	Maria	2
3	Marcia	2
4	Ana	3
5	Lucas	3
6	Antonio	5
7	Karina	
8	Carlos	



Inner Join

As junções INNER JOIN se caracterizam por uma seleção que retorna apenas os dados que atendem a **condição de junção**



```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```

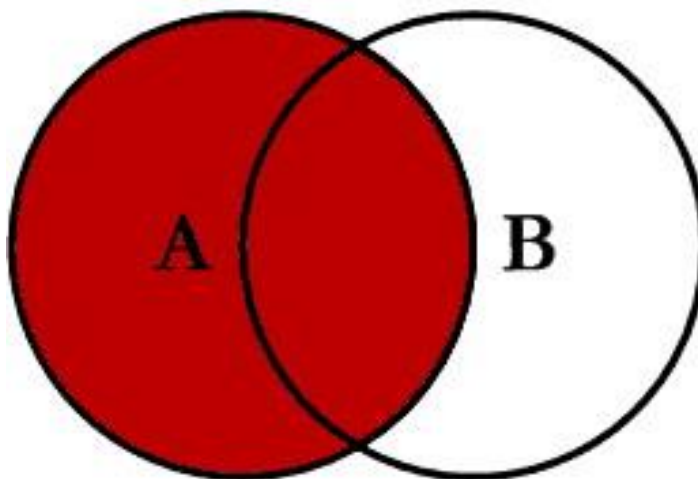

Consultas Joins – Inner Join

```
SELECT D.IDDEPARTAMENTO, D.NOME, F.IDFUNCIONARIO, F.NOME  
FROM DEPARTAMENTO AS D  
      INNER JOIN FUNCIONARIO AS F ON  
      D.IDDEPARTAMENTO = F.IDDEPARTAMENTO;
```

Iddepartamento	Nome	Idfuncionario	nome
1	Administração	1	João
2	Marketing	2	Maria
2	Marketing	3	Marcia
3	Financeiro	4	Ana
3	Financeiro	5	Lucas
5	Logística	6	Antonio

Left Join

As junções externas LEFT JOIN se caracterizam por uma seleção que retorna todos os dados da tabela da ESQUERDA (left) e apenas os dados que atendem a **condição de junção** da tabela da DIREITA.



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

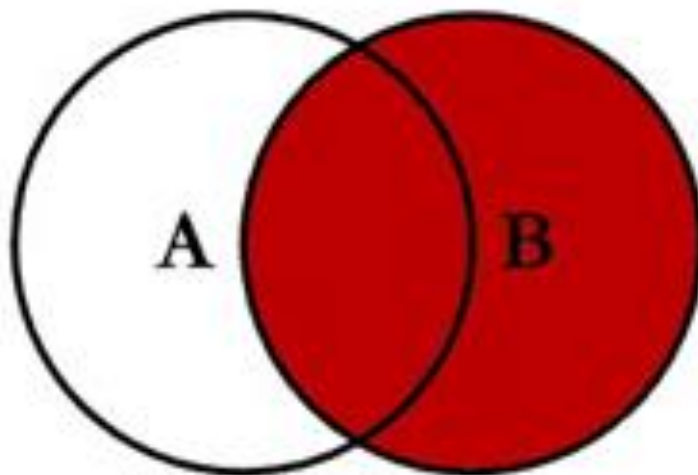
Consultas Joins – Left Join

```
SELECT D.IDDEPARTAMENTO, D.NOME, F.IDFUNCIONARIO, F.NOME
FROM DEPARTAMENTO AS D
LEFT JOIN FUNCIONARIO AS F ON
D.IDDEPARTAMENTO = F.IDDEPARTAMENTO;
```

Iddepartamento	Nome	Idfuncionario	nome
1	Administração	1	João
2	Marketing	2	Maria
2	Marketing	3	Marcia
3	Financeiro	4	Ana
3	Financeiro	5	Lucas
4	Cobrança		
5	Logística	6	Antonio
6	Manutenção		

Right Join

As junções externas RIGHT OUTER JOIN se caracterizam por uma seleção que retorna todos os dados da tabela da DIREITA (right) e apenas os dados que atendem a **condição de junção** da tabela da ESQUERDA.



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```

Consultas Joins – Right Join

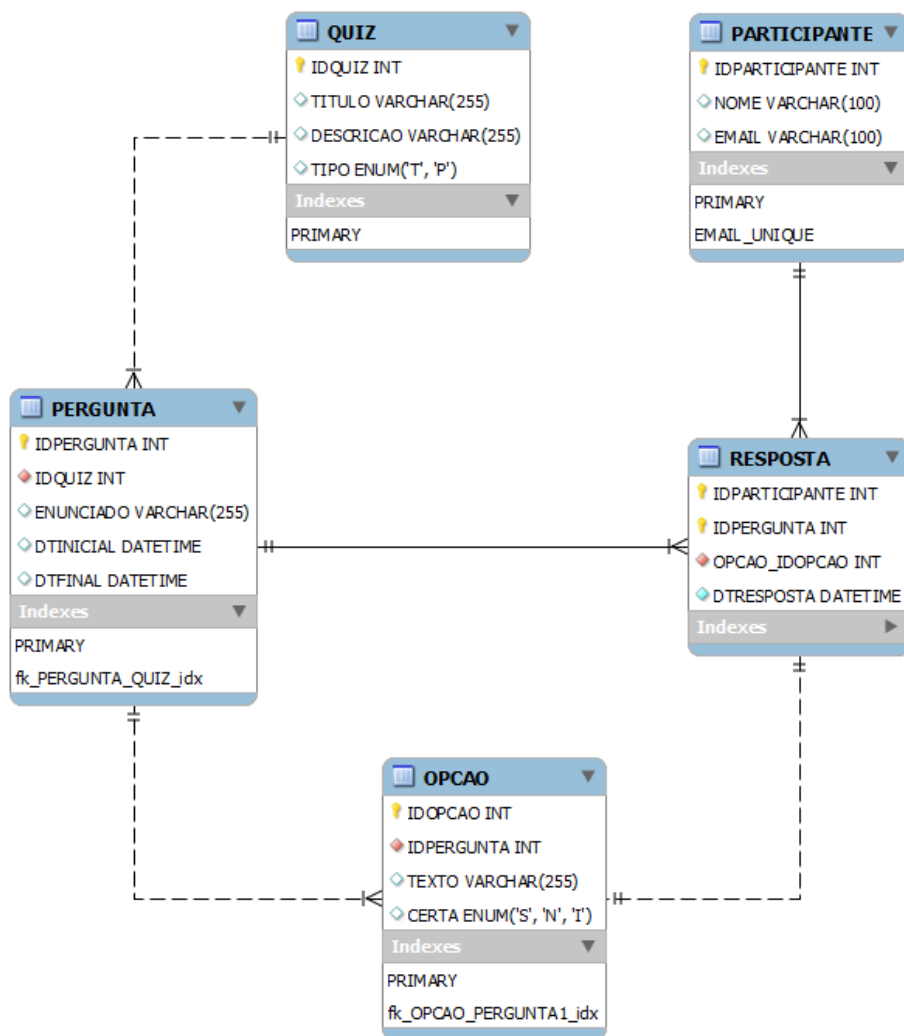
```
SELECT D.IDDEPARTAMENTO, D.NOME, F.IDFUNCIONARIO, F.NOME
FROM DEPARTAMENTO AS D
RIGHT JOIN FUNCIONARIO AS F ON
D.IDDEPARTAMENTO = F.IDDEPARTAMENTO;
```

Iddepartamento	Nome	Idfuncionario	nome
1	Administração	1	João
2	Marketing	2	Maria
2	Marketing	3	Marcia
3	Financeiro	4	Ana
3	Financeiro	5	Lucas
4	Logística	6	Antonio
		7	Karina
		8	Carlos

Consultando dados – Group By e suas Funções

O cláusula **GROUP BY** serve para agrupar os resultado em uma consulta SQL,

COUNT	Retorna a quantidade de registros com valores não-NULL diferentes
AVG	Retorna o valor médio
MIN	Retorna o menor valor
MAX	Retorna o maior valor
SUM	Retorna a soma dos valores



Exercício 2

Escreva os comandos SQL para listar o QUIZ com IDQUIZ = 2, as perguntas que pertencem esse QUIZ e as opções de resposta.

Exercício 3

Escreva os comandos SQL para listar todas as participações, coloque o nome e o e-mail do participante, a resposta escolhida, o enunciado da pergunta e o título e a descrição do QUIZ

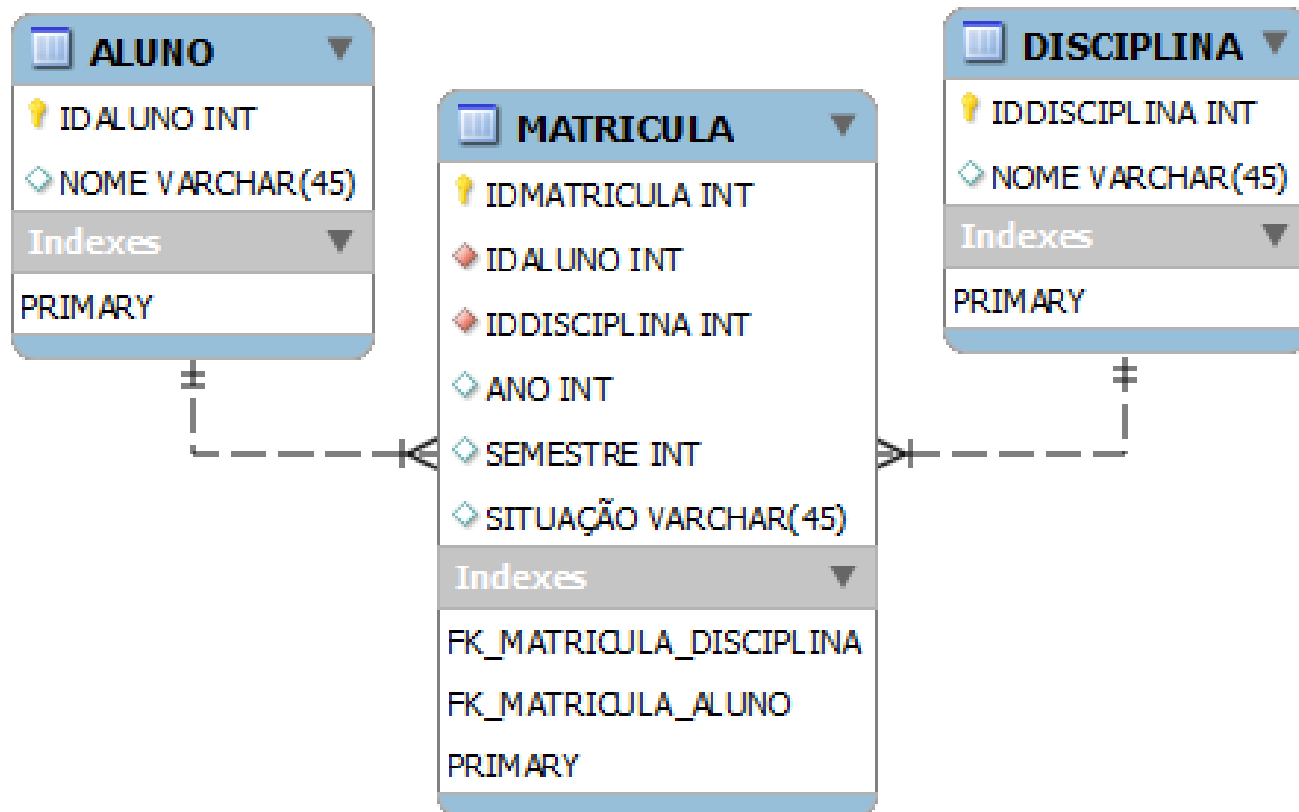
Exercício 4

Escreva os comandos SQL para listar somente as respostas certas, desde que o QUIZ seja do tipo “T”. Liste o código da pergunta, o enunciado, e todos os dados da opção.

Exercício 5

Para verificar qual pergunta teve mais participação foi utilizada uma consulta SQL que gerou o quadro abaixo, nesta tabela é listado o código da pergunta e a quantidade de participantes que respondeu essa questão.

IDPERGUNTA	COUNT(IDPARTICIPANTE)
1	36
2	73
3	22
4	128



Dados de Exemplo

ALUNO	
IDALUNO	NOME
1	ARIANO VILAR SUASSUNA
2	CECÍLIA BENEVIDES DE CARVALHO MEIRELES
3	MANUEL CARNEIRO DE SOUZA BANDEIRA FILHO

DISCIPLINA	
IDDISCIPLINA	NOME
1	PRODUÇÃO DA LEITURA E DA ESCRITA
2	INTRODUÇÃO AOS ESTUDOS LITERÁRIOS
3	ANTROPOLOGIA TEOLÓGICA

MATRICULA					
IDMATRICULA	IDALUNO	IDDISCIPLINA	ANO	SEMESTRE	SITUAÇÃO
1	1	1	2015	1	CONCLUÍDO
2	2	1	2015	1	CONCLUÍDO
3	1	2	2015	1	CONCLUÍDO
4	1	3	2015	2	MATRICULADO
5	2	1	2015	2	MATRICULADO
6	3	1	2015	1	CONCLUÍDO
7	3	2	2015	2	MATRICULADO
8	3	3	2015	1	CANCELADO
9	3	3	2015	2	MATRICULADO

Exercício 6

Crie uma consulta SQL que liste o código identificador da matrícula, o código identificador da disciplina, o nome da disciplina, o código identificador do aluno, o nome do aluno, o ano da matrícula, o semestre da matrícula e a situação da matrícula. Liste apenas as matrículas com ano igual a 2015 e semestre igual a 1.

Exercício 7

Crie uma consulta SQL que liste o código identificador do aluno, o nome do aluno, o código identificador da disciplina, o nome da disciplina e a situação da matrícula. Liste apenas as matrículas com ano igual a 2015 e semestre igual a 2 e a disciplina “ANTROPOLOGIA TEOLÓGICA”, ordenado pelo nome do aluno.

Exercício 8

Crie uma consulta para listar a quantidade de matrícula, quantidade de disciplinas, quantidade de alunos, agrupado por ano, semestre e situação, conforme imagem abaixo:

CONSULTA					
ANO	SEMESTRE	SITUACAO	COUNT(IDMATRICULA)	COUNT(IDALUNO)	COUNT(DISCIPLINA)
2015	1	CONCLUÍDO	4	3	2
2015	1	CANCELADO	1	1	1
2015	2	MATRICULADO	4	3	3