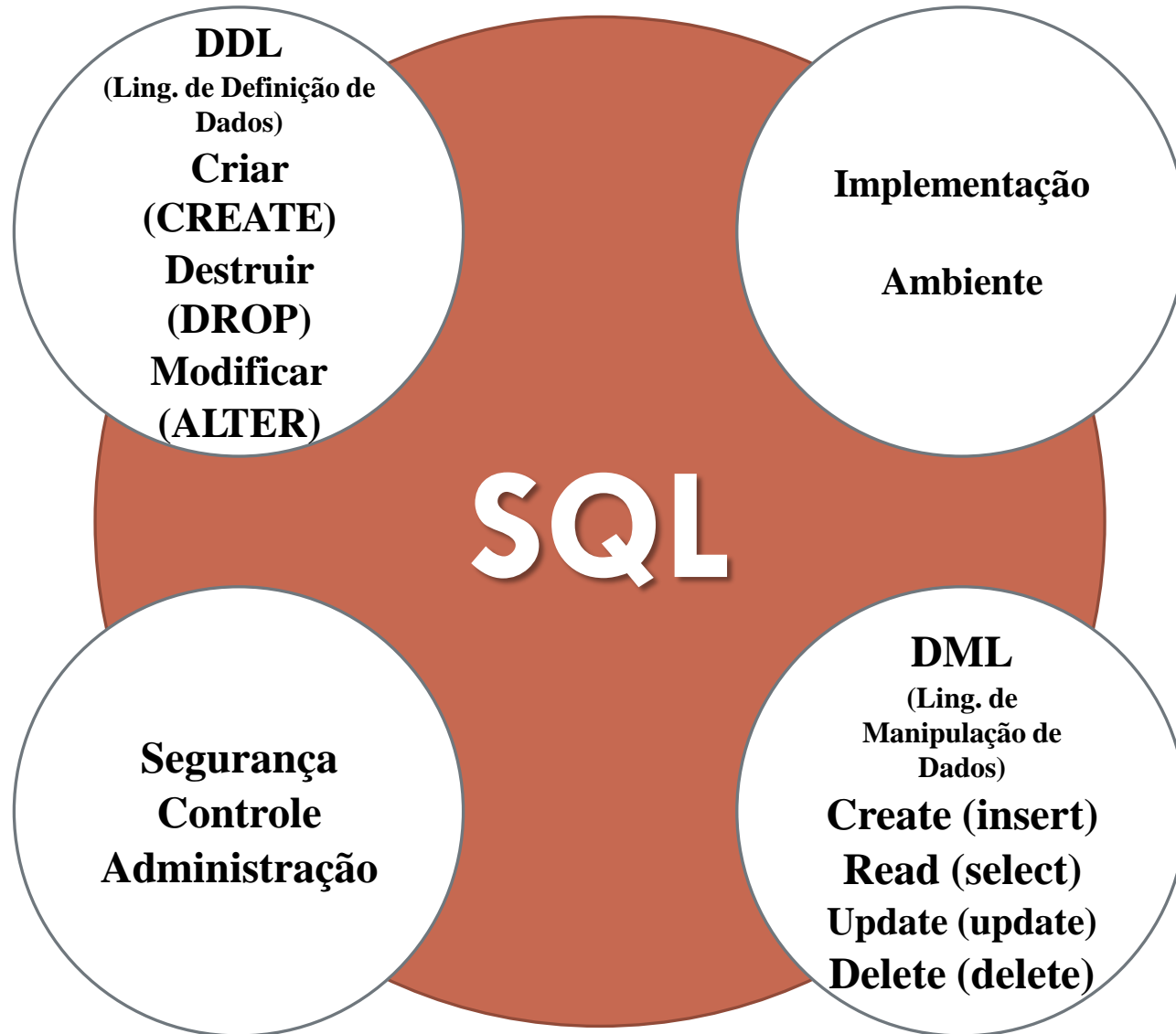




BANCO DE DADOS ADS — 4 — FATEC

Valéria Maria Volpe

USO DO SQL



COMANDOS SQL (PADRÃO ANSI)

- ❑ Criação e destruição de tabelas
- ❑ Extração de dados de uma tabela
- ❑ Inserção, modificação e remoção de dados
- ❑ Definição de visões
- ❑ Execução transacional
- ❑ Definição de privilégios de acesso

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Comando SELECT

■ Consultando atributos

- **SELECT** <lista de atributos> **FROM** Tabela

■ Exemplo: listar nome e no. de matrícula da tabela Alunos

- **SELECT** nome, cod_RM **FROM** Alunos

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Comando SELECT

■ Consultando todos os atributos

- `SELECT * FROM Tabela`

■ Exemplo: listar todos os atributos da tabela Alunos

- `SELECT * FROM Alunos`

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Comando **SELECT**

■ Consultando tuplas de tabelas

- **SELECT** <lista de atributos> **FROM** Tabela **WHERE** <comparação>

- Operadores Relacionais: >, >=, <, <=, =, <>

- Operadores Lógicos: and, or, not

■ Exemplo: listar nome e cod_RM do aluno 1213453

- **SELECT** nome, cod_RM **FROM** Alunos **WHERE** cod_RM = 1213453

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Operadores SQL

- **BETWEEN** e **NOT BETWEEN** substituem o uso dos operadores \leq e \geq
 - **SELECT** <lista de atributos> **FROM** Tabela **WHERE** <atributo> **BETWEEN** <valor1> **AND** <valor2>
- Exemplo: listar todos os empregados com salário entre R\$1.000, e R\$ 2.000,00
 - **SELECT** * **FROM** Empregados **WHERE** salario **BETWEEN** 1000.00 **AND** 2000.00

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Operadores SQL

- **LIKE** e **NOT LIKE** só operam com atributos do tipo char (varchar). Operam como = ou <>. Utiliza o % para substituir uma palavra ou parte dela.

- **SELECT** <lista de atributos> **FROM** Tabela **WHERE** <atributo> **LIKE** <valor1>

- Exemplo: listar todos os empregados que tem como primeiro nome João

- **SELECT** * **FROM** Empregados **WHERE** nome **LIKE** 'João%'

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Operadores SQL

- **IN** e **NOT IN** seleciona dados que estão ou não contidos em um conjunto valores.
 - **SELECT** <lista de atributos> **FROM** Tabela **WHERE** <atributo> **IN** <valores>
- Exemplo: listar todos os empregados com parentesco M-mãe, P-pai, F-filho
 - **SELECT** * **FROM** Empregados **WHERE** parentesco **IN** ('M', 'P', 'F')

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Operadores SQL

- **IS NULL** e **IS NOT NULL** seleciona atributos com valor NULL ou NOT NULL.

- **SELECT** <lista de atributos> **FROM** Tabela **WHERE** <atributo> **IS NULL**

- Exemplo: listar todos os empregados que não tenham telefone definido

- **SELECT** * **FROM** Empregados **WHERE** telefone **IS NULL**

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Cláusula **ORDER BY**

- **ORDER BY** é usada para ordenar por um atributo. A ordenação pode ser ascendente (ASC) ou descendente (DESC).

- **SELECT** <lista de atributos> **FROM** Tabela <WHERE> <comparação> **ORDER BY**
 <atributo> {**ASC** | **DESC**}

- Exemplo: listar todos os empregados ordenados por nome

- **SELECT * FROM Empregados ORDER BY nome ASC**

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

□ Cláusula **DISTINCT**

- **DISTINCT** é usada para eliminar tuplas repetidas do resultado da consulta.

- **SELECT DISTINCT** <atributos> **FROM** Tabela

- Exemplo: listar os diferentes salários dos empregados

- **SELECT DISTINCT** salario **FROM** Empregados

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

□ Cláusula **GROUP BY**

- **GROUP BY** organiza os dados selecionados em grupos. Os atributos usados no **SELECT** **devem aparecer** no **GROUP BY**.

- **SELECT** <lista de atributos> **FROM** Tabela **GROUP BY** <lista de atributos>
- Exemplo: listar os nomes dos empregados organizados por idade
- **SELECT** nome, idade **FROM** Empregados **GROUP BY** idade

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

□ Cláusula **HAVING**

- **HAVING** usada para agrupar dados de forma condicional. A cláusula **HAVING** é diferente do **WHERE**. O **WHERE** restringe o resultado obtido **sempre** após o uso do **FROM**, já a **HAVING** filtra o retorno do **agrupamento**.
 - **SELECT** <lista de atributos> **FROM** Tabela **GROUP BY** <lista de atributos> **HAVING** <condição>
 - Exemplo: agrupar os produtos com base nos tipos e fabricantes disponíveis, mas apenas os registros cuja quantidade supera as **200** unidades em estoque
 - **SELECT** tipo, fabricante, **SUM**(quantidade) **AS** quantidade estoque **FROM** Produtos **ORDER BY** nome, fabricante **HAVING SUM**(quantidade) **>= 200**

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Realizando cálculos com informações selecionadas (consultadas)

- O comando **SELECT** permite criar atributos que não pertencem a tabela a partir de cálculos sobre atributos que pertencem a tabelas.
- Exemplo: listar o nome e o salário novo dos empregados com aumento de 5% para aqueles que tem 5 ou mais anos de tempo de serviço
 - **SELECT** nome, salario, (salario * 1.05) as *salario_novo*, tempo_serv **FROM**
Empregados **WHERE** tempo_serv **>=** 5

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Funções agregadas disparadas a partir do SELECT

- Funções agregadas operam em um conjunto de linhas para fornecer um resultado por grupo, sendo esse conjunto de linhas uma tabela inteira ou a uma divisão em grupos.
 - A função **MAX** seleciona o **maior valor** de uma coluna entre o conjunto de dados armazenados na tabela.
 - A função **MIN** seleciona o **menor valor** de uma coluna entre o conjunto de dados armazenados na tabela.
 - A função **SUM** faz a **soma dos valores** de uma coluna entre o conjunto de dados armazenados na tabela.
 - A função **AVG** traz a **média de valores** de uma determinada coluna informada na consulta.
 - A função **COUNT** responsável por trazer o **total de ocorrências (contar)** da coluna informada.

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Funções agregadas MAX

- Exemplo: Suponhamos que seja necessário saber exatamente qual o **maior preço** entre todos os produtos cadastrados na tabela produto.
- **SELECT MAX(PRECO) AS maior_preço FROM Produtos**

❑ Funções agregadas MIN

- Exemplo: Suponhamos que seja necessário saber exatamente qual o **menor preço** entre todos os produtos cadastrados na tabela produto.
- **SELECT MIN(PRECO) AS menor_preço FROM Produtos**

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

☐ Funções agregadas SUM

- Exemplo: listar os tipos e fabricantes dos produtos disponíveis, mas apenas os registros cuja quantidade supera as **200** unidades em estoque
- **SELECT** tipo, fabricante, **SUM**(quantidade) **AS** quantidade_estoque **FROM** Produtos **WHERE SUM**(quantidade) **>= 200**

☐ Funções agregadas AVG

- Exemplo: listar o salário médios dos empregados ativos na empresa
- **SELECT AVG**(salario) **AS** media_salarial **FROM** empregados **WHERE** status = 1

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

☐ Funções agregadas COUNT

- Exemplo: deseja-se saber exatamente qual a quantidade de produtos existentes no banco de dados.
- **SELECT COUNT**(produto_id) **AS** Total **FROM** Produtos

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Junção de tabelas (JOIN)

- A junção de tabelas é feito para combinar duas ou mais tabelas por meio de alguma chave ou valor comum entre elas.
- Citar as tabelas envolvidas na junção após o **FROM**.
- Usar o nome da tabela ou alias (apelidos) para evitar ambiguidade e facilitar a escrita do comando **SELECT**.

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Junção de tabelas (JOIN)

- Exemplo: listar o nome do empregados e o departamento em que trabalha
- **SELECT E.nome, D.nome**
- **FROM** empregados **E**, departamentos **D**
- **WHERE E.idEmp = D.idEmp**

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Junção de tabelas (JOIN)

- Para juntar tabelas pode-se utilizar as cláusulas **(NOT) LIKE**, **(NOT) IN**, **IS (NOT) NULL** misturadas aos operadores **AND**, **OR** e **NOT** nas equações de junção (cláusula **WHERE**)

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Junção de tabelas (JOIN)

- Exemplo: Listar os departamentos que têm projetos com número superior a 9 e localizados em RJ ou SP ordenados por nome de departamento

```
SELECT D.nome, P.nome FROM departamentos D, projetos P  
WHERE P.local IN ('SP', 'RJ')  
AND P.numero > 9 AND P.num_depto = D.num_depto  
ORDER BY D.nome
```

EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Junção de tabelas (JOIN)

- É possível **agrupar tabelas** usando mais de um atributo
- Exemplo: Encontre o total de projetos de cada funcionário

```
SELECT E.num_depto, E.nome, COUNT(*) AS Total  
FROM trabalhaEm T, empregados E  
WHERE E.idEmp = T.idEmp  
GROUP BY E.num_depto, E.nome  
ORDER BY E.num_depto
```


EXTRAÇÃO DE DADOS DE TABELAS - CONSULTAS

❑ Junção de tabelas (JOIN)

- É possível juntar mais de **duas tabelas** usando atributo em comum
- Exemplo: Listar o nome dos empregados, com seu respectivo departamento que trabalhem mais de 20 horas em algum projeto

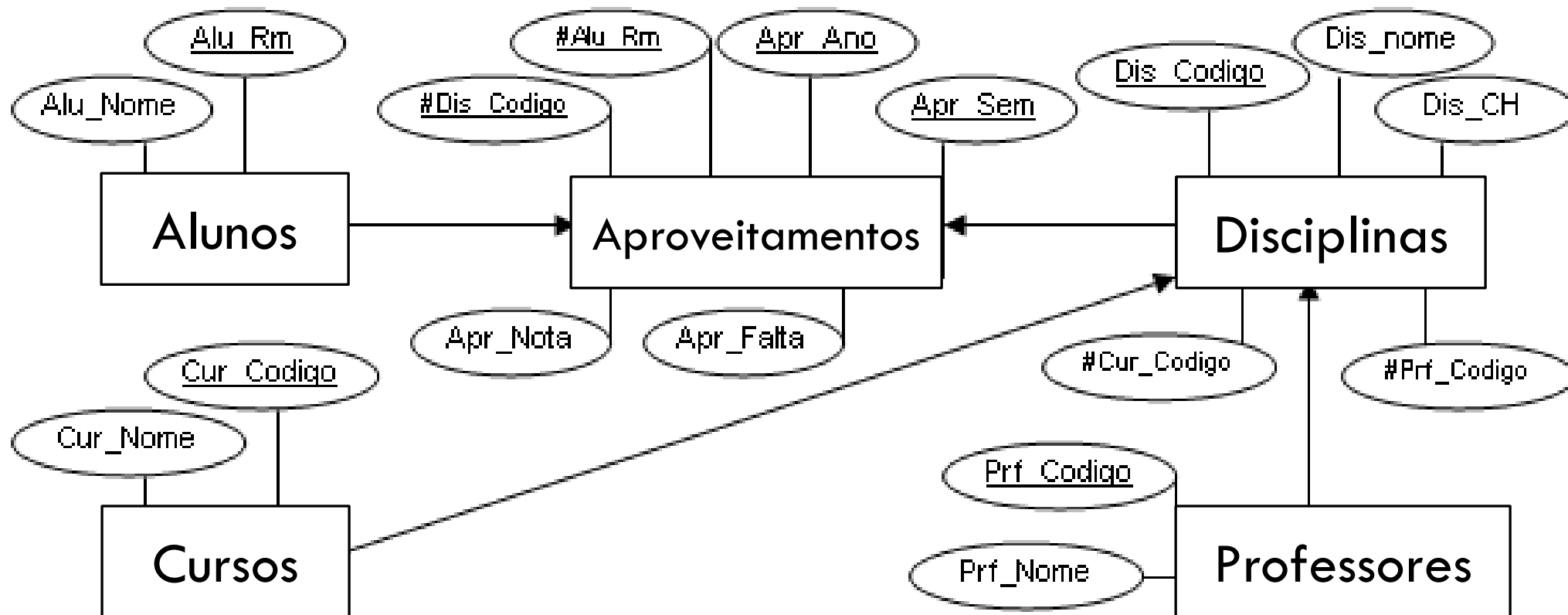
SELECT **E**.nome, **D**.nome,

FROM empregados **E** , departamento **D**, trabalhaEm **T**

WHERE **T**.horas > 20 **AND** **E**.idEmp = **T**.idEmp **AND** **E**.num_depto = **D**.num_depto

JUNÇÃO DE TABELAS - PRÁTICA

Considere o Diagrama de Estrutura de Dados - DED



JUNÇÃO DE TABELAS - PRÁTICA

- ❑ Mapeamento para o Modelo Lógico (Modelo Relacional)
 - alunos (alu_RM, nome)
 - professores (prf_codigo, prf_nome)
 - cursos (cur_codigo, cur_nome)
 - disciplinas(dis_codigo, dis_nome, dis_CH, #cur_código, #prf_codigo)
 - Aproveitamentos(#alu_RM, #dis_codigo, apr_ano, apr_sem, apr_nota, apr_falta)

BIBLIOGRAFIA

Básica:

DATE, C.J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2000.

ELMASRI, R. e NAVATHE, S. **Sistemas de Banco de Dados**. 3. ed. Rio de Janeiro: LTC, 2002.

SILBERSCHATZ, A. e KORTH, H.F. **Sistema de bancos de dados**. 3 ed. São Paulo: Makron Books, 1999.

Complementar:

BORGES JÚNIOR, S.R. **Bancos de dados simples e prático**. Pará de Minas: Virtual Books, 2011

DATE, C.J. **Guia para o padrão SQL**. Rio de Janeiro: Campus, 1989.

KERN, V.M. **Bancos de dados relacionais: teoria e prática de projeto**. São Paulo: Érica, 1994.

KOCH, G. e K. LONEY. **ORACLE: the complete reference**. 3. ed. Berkeley: Oracle Press, 1995.

MACHADO, F.; M. ABREU. **Projeto de Banco de Dados: uma visão prática**. 3. ed. São Paulo: Érica, 1998.