



Instituto de Tecnologia - ITEC  
Faculdade de Engenharia da Computação

# BANCO DE DADOS

## Unidade IV – SQL PARTE 2

Jan/2013

Fabíola Araújo

fpoliveira@ufpa.br

## Roteiro

- **Introdução ao SQL**
  - ▣ Histórico
- **Comandos DDL (Data Definition Language)**
  - ▣ Create, alter e drop
- **Comandos DML (Data Manipulation Language)**
  - ▣ Insert, delete, update e select

## SQL (Structured Query Language)

- **Histórico**
  - SEQUEL (Structured English **QUE**ry Language)
  - System R da IBM
  - Padrão SGBDs relacionais comerciais
  - ANSI (American National Standards Institute) e ISO (International Standards Organization)
- ▣ **Versões**
  - SQL-86 ou SQL-87
  - SQL-89
  - SQL-92 ou SQL 2
  - SQL-99 ou SQL 3
  - SQL-2003
  - SQL-2006
  - SQL-2008
  - SQL-2011

## SQL - Características

- Definição, consulta e atualização.
  - ▣ **DDL (Data Definition Language)** e **DML (Data Manipulation Language)**
- Visões
- Segurança e autorizações
- Restrição de integridade
- Controle de transações
- Aspectos da orientação a objetos

## Esquemas (Schemas)

- Tabelas e componentes pertencem a mesma aplicação.

- Criar esquema

```
CREATE SCHEMA Empresa AUTHORIZATION dbafabiola;
```

- Eliminar esquema

```
DROP SCHEMA Empresa CASCADE;
```

- ✓ Opção **RESTRICT**: Esquema eliminado se não tiver elementos.

## Esquemas (Schemas)

- ☠ **No Oracle**, ao criar um usuário, um esquema correspondente também é criado!

- Um usuário só pode estar relacionado a um único ESQUEMA.

- Um esquema é uma coleção de objetos que foram definidos pelo usuário.

- Ex. *Tables, triggers, constraints, indexes, views, sequences, stored procedures, synonyms, user-defined data types, database links e etc.*

## Criação de Usuários (Oracle)

```
Create user alunodba  
identified by teste  
default tablespace DAT01  
quota 15m on DATA  
password expires;
```

- Atribuições de permissões

```
Grant create session to alunodba;  
Grante dba to alunodba;
```

## Criação de Usuários (Oracle)

- Remoção de permissões

```
Revoke create session from alunodba;  
Revoke dba from alunodba;
```

## Roles (*Papéis*)

### □ Características

- Facilitar o gerenciamento de privilégios;
- Podem conter privilégios de sistema ou sobre objetos;
- Nome tem que ser único (usuários e roles);
- Não possuem proprietários e nem pertencem a nenhum esquema.

```
Create role mantenedor ;
```

## Roles (*Papéis*)

### □ Atribuição de permissões

```
Grant create table to mantenedor;  
Grant drop user to mantenedor;
```

### □ Remoção de permissões

```
Revoke create table from mantenedor;  
Revoke drop user from mantenedor;
```

## Roles (*Papéis*)

### □ Privilégios de **Sistema**

Categoria	Exemplos
<b>Table</b>	Create table, alter any table, drop any table, select any table, update any table, delete any table.
<b>Session</b>	Create session, alter session, restricted session.
<b>Index</b>	Create any index, alter any index, drop any index.
<b>Sequence</b>	Create sequence, alter any sequence, drop any sequence, select any sequence.

## Roles (*Papéis*)

### □ Privilégios sobre os **objetos**

Categoria	Exemplos
<b>Table</b>	Alter, delete, index, insert, references, select update.
<b>View</b>	Alter, delete, index, insert, select update.
<b>Sequence</b>	Alter, select.
<b>Procedure</b>	Alter, execute.

## Profiles

- Criação de profiles
  - ▣ Gerenciamento de contas dos usuários
    - Limite de recursos e senhas
  - ▣ Criação de um profile **default**

```
Create profile analista limit  
Sessions_per_user 2  
CPU_per_session 10000  
idle_time 60  
connect_time 480;
```

## Profiles

- Algumas limitações...

Categoria	Descrição
Password	"Idade" da senha, expiração, histórico, verificação complexa.
Conta do Usuário	Bloqueio, tempo de CPU, permissão E/S, tempo de inatividade, tempo de conexão, espaço de memória, conexões simultâneas.

## Profiles

- Remoção de **profiles**

```
Drop profile analista;  
Drop profile mantenedor cascade;
```

- Atribuição de profiles

```
Create user alunodba identified by teste  
profile analista  
default tablespace DAT01  
quota 15m on DATA  
password expires;
```

```
Alter user alunodba profile analista;
```

## Tabelas (*Tables*)

- **Ao criar tabelas...**
  - ▣ Nomes, atributos e restrições (*referencial, entidade, not null e outros*)
- **Tipos de dados**
  - ▣ Numéricos: integer (int), float, decimal (i,j), real...
  - ▣ Caracteres: char(n), varchar(n)...
  - ▣ Data e Hora: date (dia, ano, mês, hora, minuto e segundo)

## Tabelas (Oracle)

### Nome de tabelas e colunas

- Devem começar com letras
- Tamanho de 1-30 caracteres
- Devem conter apenas:
  - A-Z
  - a-z
  - 0-9
  - \_, \$ e #
- Não podem ter nomes duplicados!

## Tabelas (Oracle)

### Tipos de Dados

Tipo de Dado	Descrição
Varchar2(n)	Dado do tipo caracter (tamanho variável), no máximo 4000 bytes.
Char(n)	Dado do tipo caracter (tamanho fixo), no máximo 2000 bytes.
Number(p,s)	Dado do tipo numérico (tamanho variável).
Date	Dado do tipo data ou hora
Long	Dado do tipo caracter (tamanho variável), até 2Gb.
Clob	Dados do tipo caracter até 4Gb.
Blob	Dado do tipo binário até 4Gb.
Bfile	Dado binário armazenado em um arquivo externo (até 4Gb).

## Tabelas - Exemplos

```
CREATE TABLE Empregado
(Nome VARCHAR(15) NOT NULL,
Sobrenome VARCHAR(15) NOT NULL,
Cpf CHAR(9) NOT NULL primary key,
DataNasc DATE NOT NULL,
Endereco VARCHAR(30),
Salario DECIMAL(10,2),
Depto INTEGER NOT NULL,
Email VARCHAR2(25) UNIQUE,
FOREIGN KEY (Depto) REFERENCES Departamento
(NumeroDep));
```

## Tabelas - Exemplos

```
CREATE TABLE Departamento
(NomeDep VARCHAR(15) NOT NULL unique,
NumeroDep INTEGER NOT NULL primary key,
CpfGer CHAR(9) NOT NULL,
FOREIGN KEY (CpfGer) REFERENCES Empregado
(Cpf));
```

## Tabelas

- As chaves **primária** e **estrangeira** podem ser definidas **posteriormente** através do comando **ALTER TABLE**.
- A restrição de **PRIMARY KEY** especifica **um** ou **mais atributos** que compõe a chave primária.

## Tabelas

- Remoção de Tabelas

```
DROP TABLE Empresa CASCADE;
```

- Opção **restrict**
  - Tabela eliminada se não for referenciada em qualquer restrição (chave estrangeira) ou visão.

### ⚠ No Oracle

- Eliminar tabelas com dados e restrições (chave estrangeira)

```
Drop table Empregado cascade constraint;
```

## Restrições (Constraints)

- **Definição de Restrições**
  - Nome da restrição deve ser **único** dentro do esquema.
  - É **opcional**, porém é **bom atribuir nomes** as restrições para facilitar a procura no dicionário de dados (metadados).

## Restrições (Oracle)

- Exemplo

```
CREATE TABLE Empregado
(.....,
  Cpf CHAR(9) NOT NULL,
  DataNasc DATE,
  Email VARCHAR2(25),
  Depto INTEGER NOT NULL,
  CONSTRAINT Pk_Emp PRIMARY KEY (Cpf),
  CONSTRAINT Uk_Emp_email UNIQUE (email),
  CONSTRAINT Nn_Emp_datanasc NOT NULL,
  CONSTRAINT Fk_Emp_Depto FOREIGN KEY (Depto)
  REFERENCES Departamento (NumeroDep));
```

## Alterações de Tabelas

### □ Comando **ALTER**

- ▣ Alterar estrutura (definição) de uma tabela
  - Redefinir chave primária, chave estrangeira, *constraints* do tipo *unique* e *not null*, entre outros.
- ▣ Alterar atributos
  - Adicionar
  - Redefinir o tipo ou valor default
  - Remover

## Alterações de Tabelas

### □ Adicionar novo atributo

```
ALTER TABLE Empregado ADD servico VARCHAR(12);
```

- ▣ Valor para o novo atributo
  - Especificar cláusula DEFAULT
  - Comando UPDATE
  - Nenhuma cláusula DEFAULT → Valor NULO

## Alterações de Tabelas

### □ Remover um atributo

```
ALTER TABLE Empregado DROP endereco CASCADE;
```

#### ▣ Opção **restrict**

- Atributo eliminado se nenhuma visão ou restrição referenciar a coluna.

#### ☠ **No Oracle**

- Restrict é default e não precisa especificar.
- Remover um atributo

```
Alter table Empregado drop column endereco;
```

## Alterações de Tabelas

### □ Alterar atributo

#### ▣ Adicionar novo valor default

```
ALTER TABLE Empregado ALTER Cpf SET  
DEFAULT '1122334455';
```

#### ☠ **No Oracle**

- Alterar o valor default de um atributo.

```
Alter table Empregado modify CPF default '1122334455';
```

- Modificar o tipo de um atributo.

```
Alter table Empregado modify EMAIL varchar2(30);
```

## Alterações de Tabelas

### □ Alterar atributo

- ▣ **Eliminando o valor default** do atributo CPF

```
ALTER TABLE Empregado ALTER Cpf DROP DEFAULT;
```

### ☠ **No Oracle**

- Eliminar o valor default de um atributo

```
Alter table Empregado modify CPF default NULL;
```

## Comandos DML

### □ **Inserir dados na tabela**

- ▣ Inserir uma tupla de cada vez na relação;
- ▣ Valores inseridos na mesma ordem em que os atributos foram criados no comando "CREATE TABLE";
- ▣ Especificar a ordem de inserção dos dados nos atributos;
- ▣ Inserir dados a partir de uma seleção em outra tabela com os mesmos atributos (tabela idêntica).

## Exemplos - *INSERT*

```
INSERT INTO Empregado  
VALUES ('Ricardo', 'Oliveira', '123456789',  
'20-02-80', 'Tv. 14 de Março no. 42', 2000, 2);
```

```
INSERT INTO Empregado (nome, sobrenome, cpf, depto)  
VALUES ('RICARDO', 'OLIVEIRA', '123456789', 2);
```

```
INSERT INTO Empregado (nome, sobrenome, cpf, depto)  
AS SELECT nome, sobrenome, cpf, depto FROM Empregado_Old;
```

## Comandos DML

### □ **Eliminar dados da tabela**

- ▣ Pode-se utilizar uma cláusula de condição (WHERE).
- ▣ Caso a cláusula de condição não seja utilizada, TODAS as linhas da tabela serão eliminadas.

```
DELETE FROM Empregado WHERE Sobrenome = 'SILVA';
```

```
DELETE FROM Empregado;
```



## Exemplo - *DELETE*

- **Delete** com **Select** aninhado.

```
DELETE FROM Empregado
WHERE Depto IN
(SELECT NumeroDep FROM Departamento
WHERE NomeD = 'PESQUISA');
```

## Comandos DML

- **Atualização** de dados da tabela
  - ▣ Modificar valores de atributos de uma ou mais linhas (tuplas) da tabela.

```
UPDATE Projeto SET LocalizacaoP = 'BELEM'
WHERE NumeroDep = 5;
```

```
UPDATE Projeto SET LocalizacaoP = 'BELEM' , NumeroDep = 5
WHERE PNumero = 10;
```

## Exemplo - *UPDATE*

- **Update** com **Select** aninhado.

```
UPDATE Empregado
SET Salario = Salario *1.1
WHERE NumDep IN (SELECT Dnumero
FROM Departamento
WHERE Dnome = 'Pesquisa');
```

## Comandos DML

- **Consultas**

```
SELECT <Lista de Atributos>
FROM <Lista de Tabela>
WHERE <Condição>
```

- ▣ Ex1.

```
SELECT DataNasc, Endereco
FROM Empregado
WHERE Nome='Eduardo' AND Sobrenome='Silva';
```

## Comando **SELECT** utilizando **Join**

Empregado

Nome	IniciaisdoMeio	Sobrenome	CPE	DtNasc	End	Sexo	NSSSuper	NUD
------	----------------	-----------	-----	--------	-----	------	----------	-----

Departamento

NomeD	NumeroDep	CPFGer	DataInicioGer
-------	-----------	--------	---------------

### ▪ Ex1.

```
SELECT Nome, NomeD
FROM Empregado, Departamento
WHERE NUD=NumeroDep;
```

### ▪ Ex2.

```
SELECT Nome, NomeD
FROM Empregado Join Departamento on NUD=NumeroDep;
```

## Exemplos - **SELECT**

### ▣ Ex2. Utilizando ALIAS

```
SELECT Nome, Sobrenome, Endereco
FROM Empregado Emp, Departamento Dept
WHERE Dept.NomeDept='Informatica'
AND Emp.Depto = Dept.NumeroDept;
```

### ▣ Ex3. Sem Cláusula WHERE (Condição)

```
SELECT Cpf FROM Empregado;
```

## Exemplos - **SELECT**

### ▣ Ex4. Utilizando o asterisco (\*)

```
SELECT * FROM Empregado WHERE Depto = 5;
```

- Seleciona todos os atributos da tabela, sendo que apenas as tuplas de empregados que trabalham no Depto = 5.

## Comando **SELECT**

### ▣ Variação: **SELECT ALL** ou **SELECT DISTINCT**

- ▣ No SELECT DISTINCT elimina-se as tuplas duplicadas.
  - Operação onerosa
  - Usuário não deseja ver tuplas duplicadas no resultado
  - Função agregação
  - Opção ALL é **default**!
- ▣ Ex.

```
SELECT (ALL) Salario FROM Empregado;
```

## Comparação de *Substrings*

### □ Comando **LIKE**

- ▣ Compara somente partes de uma string de caracteres.
  - “%” → Substitui um número arbitrário de caracteres.
  - “\_” → Substitui um único caracter.

#### ▣ Ex1.

```
SELECT Nome, Sobrenome  
FROM Empregado  
WHERE Endereco LIKE '%NAZARE%';
```

## Exemplo - **LIKE**

#### ▣ Ex2.

```
SELECT Nome, Sobrenome  
FROM Empregado  
WHERE DataNasc LIKE '____1990';
```

## Concatenação de Dados

- A concatenação de dados só pode ser feita com dados do tipo **string** (varchar2, varchar, char e outros).

#### ▣ Ex.

```
SELECT Nome || Sobrenome AS NOME FROM Empregado;
```

## Operadores Aritméticos

### □ Operadores padrão da aritmética

- ▣ Soma (+)
- ▣ Subtração (-)
- ▣ Multiplicação (\*)
- ▣ Divisão (/)

- *Só podem ser aplicados em valores numéricos ou atributos com domínio numérico.*

## Exemplo

### Operadores Aritméticos

Empregado

Nome	IniciaisdoMeio	Sobrenome	CPF	DtNasc	End	Sexo	CPFuper	NUD
------	----------------	-----------	-----	--------	-----	------	---------	-----

Trabalha\_Em

CPFemp	NumP	Horas
--------	------	-------

Projeto

NomeP	NumeroP	LocalizacaoP	NumD
-------	---------	--------------	------

```
SELECT Nome, Sobrenome, Salário*1.1
FROM Empregado, Trabalha_Em, Projeto
WHERE CPF=CPFemp
AND NumP=NumeroP
AND NomeP= 'Projeto X';
```

## Operador de Comparação

### BETWEEN

- Compara se os valores de um atributo estão dentro de uma determinada **faixa**, fornecendo o **limite superior e inferior**.

```
SELECT *
FROM Empregado
WHERE (Salario BETWEEN 3000 AND 5000) AND NUD = 5;
```

```
SELECT *
FROM Empregado
WHERE Salario ≥ 3000 AND Salario ≤ 5000 AND NUD = 5;
```

## Ordenamento de Tuplas

### ORDER BY

- Ordenar o resultado pelo valor de **um** dos **atributos** ou de **vários atributos**.

Empregado

Nome	IniciaisdoMeio	Sobrenome	CPF	DtNasc	End	Sexo	CPFuper	NUD
------	----------------	-----------	-----	--------	-----	------	---------	-----

Trabalha\_Em

CPFemp	NumP	Horas
--------	------	-------

Projeto

NomeP	NumeroP	LocalizacaoP	NumD
-------	---------	--------------	------

Departamento

NomeD	NumeroDep	CPFGer	DataInicioGer
-------	-----------	--------	---------------

```
SELECT NomeD, Nome, NomeP
FROM Departamento, Empregado, Trabalha_Em, Projeto
WHERE NumeroDep = NUD AND CPF = CPFemp AND NumeroP = NumP
ORDER BY NomeD, Nome;
```

## Ordenamento de Tuplas

### ORDER BY

- Ordernamento **default** → Ascendente (**ASC**)
  - Ordem descendente (**DESC**) deve ser especificada.

```
SELECT NomeD, Nome, NomeP
FROM Departamento, Empregado, Trabalha_Em, Projeto
WHERE NumeroDep = NUD
AND CPF = CPFemp
AND NumeroP = NumP
ORDER BY NomeD (ASC), Nome DESC;
```

## Operador *IN*

- Comparação se o valor de um atributo está contido em um **conjunto de valores** especificados.

Empregado

Nome	IniciaisdoMeio	Sobrenome	CPF	DtNasc	End	Sexo	NSSSuper	NUD
------	----------------	-----------	-----	--------	-----	------	----------	-----

Departamento

NomeD	NumeroDep	CPFGer	DataInicioGer
-------	-----------	--------	---------------

Localizacoes\_Dep

NumeroDep	Localização_Dep
-----------	-----------------

Projeto

NomeP	NumeroP	LocalizacaoP	NumD
-------	---------	--------------	------

Trabalha\_Em

CPFEMP	NumP	Horas
--------	------	-------

## Operador *IN*

```
SELECT DISTINCT NumeroP
FROM Projeto
WHERE NumeroP IN
(SELECT NumeroP
FROM Projeto, Departamento, Empregado
WHERE NumD = NumeroDep
AND CPFGer = CPF AND Sobrenome = 'SILVA')
OR NumeroP IN
(SELECT NumP
FROM Trabalha_Em, Empregado
WHERE CPFEMP = CPF AND Sobrenome = 'SILVA');
```

## Conjuntos **Explícitos** e *Nulls*

- Utilizando operador **IN**

▣ Ex1.

```
SELECT DISTINCT Cpf
FROM Trabalha_Em
WHERE NumP IN (1,2,3);
```

▣ Ex2.

```
SELECT Nome, Sobrenome
FROM Empregado
WHERE CPFSuper IS NULL;
```

## “*Renomear*” Atributos

- Qualificador **AS**

▣ Ex.

```
SELECT E.Nome AS Nome_Empregado,
S.Nome AS Nome_Supervisor
FROM Empregado AS E, Empregado AS S
WHERE E.CPFSuper = S.CPF;
```

## Funções de Agregação

### □ Operador **COUNT**

- ▣ **Conta o número de linhas** (ou valores) baseado em uma cláusula de condição ou não.

#### ▣ **Ex1.**

```
SELECT COUNT(*)  
FROM Empregado, Departamento  
WHERE NUD = NumeroDep  
AND NomeD = 'PESQUISA';
```

## Exemplo - **COUNT**

#### ▣ **Ex2.**

```
SELECT COUNT(DISTINCT Salario) FROM Empregado;
```

#### ▣ **Ex3.**

```
SELECT Sobrenome, Nome  
FROM Empregado  
WHERE (SELECT COUNT(*)  
FROM Dependente  
WHERE CPF = CPFEmp) >= 2;
```

## Funções de Agregação

### □ Outras funções...

- ▣ Soma (**SUM**)
- ▣ Valor máximo (**MAX**)
- ▣ Valor mínimo (**MIN**)
- ▣ Média Aritmética (**AVG**)

#### ▣ **Ex1.**

```
SELECT SUM(Salario), MAX(Salario), Min(Salario), AVG(Salario)  
FROM Empregado;
```

## Exemplos Funções de Agregação

#### ▣ **Ex2.**

```
SELECT SUM(Salario), MAX(Salario), Min(Salario), AVG(Salario)  
FROM Empregado, Departamento  
WHERE NUD = NumeroDep AND NomeD = 'PESQUISA';
```

#### ▣ **Ex3.**

```
SELECT NUD, COUNT(*), AVG(Salario)  
FROM Empregado GROUP BY NUD;
```