

PROGRAMAÇÃO DE BANCO DE DADOS

PROGRAMAÇÃO DE BANCO DE DADOS

PROGRAMAÇÃO DE BANCO DE DADOS

- 80 horas/aula
- 20 semanas

EMENTA: Componentes da Linguagem SQL. Comandos de Definição de Dados. Comandos de Controle de Dados. Encadeamento de Tabelas. Visões, Índices e Consultas

OBJETIVOS: Utilizar a linguagem SQL tendo como suporte um banco de dados relacional, criando tabelas e manipulando dados através de comandos DDL e DML.

PROGRAMAÇÃO DE BANCO DE DADOS

- Bibliografia Básica
 - MORAIS, A. L.. Desvendando a Programação com o SQL Server 2014 - Guia de referência prático para aprender a programar
 - utilizando o SQL Server 2014. Ciencia Moderna, 2017.
 - GILMORE, W. Jason. Dominando PHP e MYSQL do iniciante ao profissional. Starlin Alta Consult, 2008.
 - DAMAS, L. M. D. SQL – Structured Query Language. LTC, 2007. ISBN: 8521615582.
- Bibliografia Complementar
 - SOUZA, M A. SQL, PL/SQL, SQL PLUS: Manual de referência completo e objetivo. Rio de Janeiro: Ciência Moderna, 2004.
 - PRICE, J. Oracle database 11g SQL. Porto Alegre: Bookman, 2009.
 - MILANI, A. PostgreSQL: Guia do Programador. São Paulo: Novatec, 2008.
 - MANZANO, J. A. N. G. PostgreSQL 8.3.0. interativo: guia de orientação e desenvolvimento. São Paulo: Érica, 2008.

PROGRAMAÇÃO DE BANCO DE DADOS

- Critérios de Avaliação Fórmula : $((P1+P2)/2)*0.6)+(T1*0.4)$ Legendas :
- Primeira Avaliação - Avaliar o conteúdo do primeiro bimestre. – P1
- Segunda Avaliação - Avaliar o conteúdo do segundo bimestre. – P2
- Trabalho final da disciplina - Trabalho final para avaliar o conteúdo da disciplina. – T1.

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

• Ferramentas de Apoio:

- Oracle Database 11g Express Edition



- Sql Developer



- Sql Developer Data Modeler



- Toad for Oracle



PROGRAMAÇÃO DE BANCO DE DADOS



Downloads de Software

Download gratuito, aprendizado gratuito, avaliação ilimitada

Para downloads maiores ou para funções como a retomada de download, é altamente recomendável o uso de um gerenciador de download.

Desenvolvedores:

Todos os downloads são gratuitos, e cada um vem com uma [Licença de Desenvolvimento](#) (inglês) que lhe permite utilizar versões completas dos produtos somente enquanto estiver desenvolvendo e elaborando o protótipo dos seus aplicativos (ou estritamente para sua própria educação). Em alguns casos, determinados downloads (como as versões Beta) têm licenças com termos ligeiramente diferentes. Você pode comprar produtos com licenças para uso total a qualquer momento na [Loja](#) (inglês) online ou com seu representante de vendas.

Clientes:

Se você já tiver uma licença comercial você deve baixar o software do nosso [Oracle Software Delivery Cloud](#), que é especificamente desenhado para a satisfação do cliente. Para patches, consulte [My Oracle Support](#).

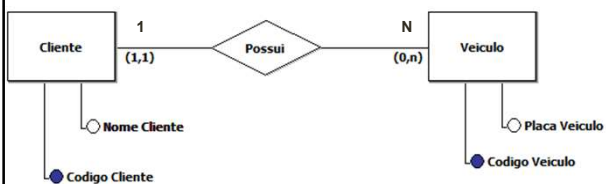
PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

- Revisão Modelagem de Dados
- Conversão do Modelo Conceitual para o Modelo Relacional/Lógico.

Modelo Conceitual (DER)



PROGRAMAÇÃO DE BANCO DE DADOS

• Revisão Modelagem de Dados

Modelo Conceitual

- Entidade
- Atributo
- Atributo Identificador
- Ocorrências de Entidade
- Relacionamento

Modelo Lógico

- Tabela
- Campo
- Chave Primaria (PK)
- Registros
- { Tabela
Chave Estrangeira (FK)

PROGRAMAÇÃO DE BANCO DE DADOS

• Revisão Modelagem de Dados



- Chave Estrangeira ficará na tabela do lado N



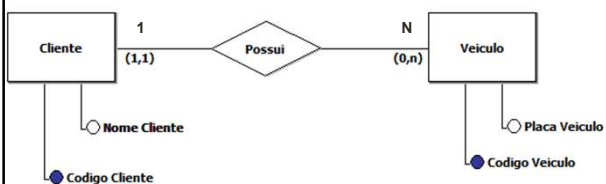
- O Relacionamento será transformado em tabela.
- Essa nova tabela hospedar as duas Chaves Estrangeiras



- Verificamos a necessidade de construir duas tabelas
- A Chave estrangeira ficará na tabela com menor probabilidade de hospedar registros

PROGRAMAÇÃO DE BANCO DE DADOS

Modelo Conceitual (DER)



PROGRAMAÇÃO DE BANCO DE DADOS

Modelo Lógico

TB_CLI (cd_cli, nm_cli)

TB_VEI (cd_vei, pl_vei, cd_cli)
cd_cli referencia TB_CLI

PROGRAMAÇÃO DE BANCO DE DADOS

Modelo Lógico

Notação Crows Foot "Pé de Galinha" – James Martin



PROGRAMAÇÃO DE BANCO DE DADOS

?

PROGRAMAÇÃO DE BANCO DE DADOS

• Revisão Desenvolvimento de Banco de Dados

• Dado

- Valor isolado

376,89	6	62,82
--------	---	-------

• Informação

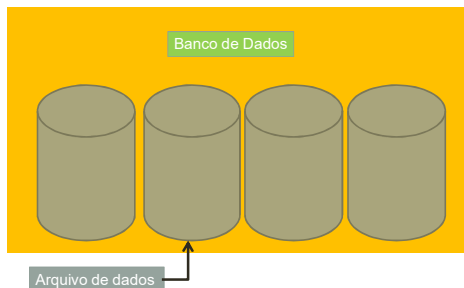
- Dado contextualizado

VI_conta	Qtd_pes	VI_ind
376,89	6	62,82

PROGRAMAÇÃO DE BANCO DE DADOS

Banco de Dados

- Arquivo(s) no(s) qual(is) os dados são armazenados para que possam ser manipulados e, dessa forma, produzir informação



PROGRAMAÇÃO DE BANCO DE DADOS

Banco de Dados

- Um banco de dados é uma coleção de dados tratados como uma unidade.
- A finalidade de um banco de dados é armazenar e recuperar informações relacionadas.

Oracle Database 11g: Administration Workshop I

PROGRAMAÇÃO DE BANCO DE DADOS

Sistema Gerenciador de Banco de Dados

- Um Sistema Gerenciador de Banco de Dados (SGBD – DBMS Relational DataBase Management System) gere uma grande quantidade de dados em um ambiente multiusuário, afim de que todos possam acessar simultaneamente os mesmos dados. Isto é realizado enquanto proporciona um alto desempenho, ao mesmo tempo que evita o acesso não autorizado e fornece soluções eficientes para recuperação de falha.

Oracle Database 11g: Administration Workshop

PROGRAMAÇÃO DE BANCO DE DADOS

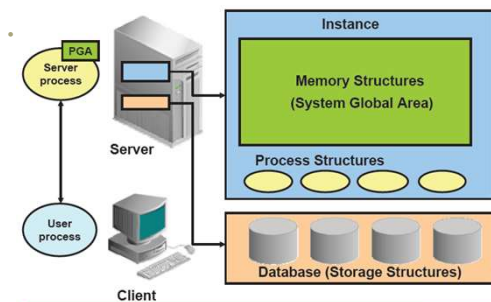
Sistema Gerenciador de Banco de Dados

- Autocontenção.
- Independência dos Dados.
- Abstração dos Dados.
- Visões.
- Transações.
- Controle de Acesso.

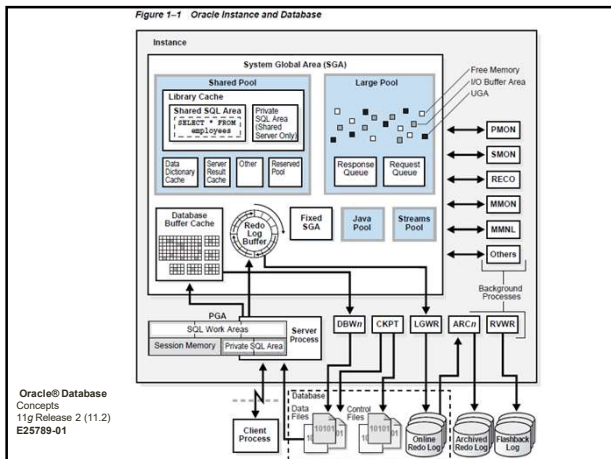
Marcio Novelli

PROGRAMAÇÃO DE BANCO DE DADOS

Oracle Database Server Architecture: Overview



Copyright © 2009, Oracle. All rights reserved.



PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

- Linguagem de pesquisa para Banco de Dados Relacional.
- Oferece funções de recuperação e atualização de Banco de Dados.
- Características inspiradas na álgebra relacional.
- Pode ser usada de forma *on-line* ou partir de um programa.
- Especifica qual informação deseja e não como vai obtê-la.
- Processa conjunto de registros em vez de um simples registro por vez.

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

- Desenvolvida originalmente no início dos anos 70 nos laboratórios da IBM.
- Parte integrante do projeto System R, que tinha por objetivo demonstrar a viabilidade da implementação do *Modelo Relacional* (Edgar Frank "Ted" Codd (August 23, 1923 – April 18, 2003)).
- Originalmente denominada SEQUEL - "Structured English Query Language" (Linguagem de Consulta Estruturada, em Inglês)
- Devido a simplicidade de seu uso, foi rapidamente absorvida pelo mercado. Consequentemente varias "versões" surgiram, desenvolvidas por outros fabricantes.
- 1986 foi padronizada pelo America National Standards Institute (ANSI).
- 1987 revisada pelo International Organization for Standardization (ISO)

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

- 1992 Revisão da linguagem -SQL92.
- 1999 Revisão da linguagem -SQL1999 (SQL3) - Foram incluídas: expressões regulares de emparelhamento \ *queries* recursivas \ gatilhos (*triggers*) \ características de orientação a objeto.
- 2003 Revisão da linguagem -SQL2003 – Introduzido características relacionadas ao XML \ sequências padronizadas
- A mais recente padrão SQL foi adotada em Julho de 2008 e é chamado de SQL: 2008. Os nomes formais da presente norma são:
 - ANSI/ISO/IEC 9075:2008, "Database Language SQL"
 - ANSI/ISO/IEC 9075-14:2008, "Database Language SQL"
 - ISO/IEC 9075:2008, "Database Language SQL"
 - ISO/IEC 9075-14:2008, "Database Language SQL"

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

Sub conjuntos de comandos da linguagem SQL

DDL – Data Definition Language - Concentra comandos que viabilizam a manutenção de objetos do banco de dados:

CREATE
ALTER
DROP
TRUNCATE

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

```
CREATE TABLE plants (  
  plant_id NUMBER PRIMARY KEY,  
  common_name VARCHAR2(15) );
```

```
ALTER TABLE plants ADD  
( latin_name VARCHAR2(40) );
```

```
DROP TABLE plants;
```

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

Sub conjuntos de comandos da linguagem SQL

**DML – Data Manipulation Language -
Concentra comandos que viabilizam a
manutenção de DADOS do banco de dados:**

**INSERT
UPDATE
DELETE**

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

```
INSERT INTO employees  
(employee_id, last_name, email, job_id, hire_date, salary)  
VALUES  
(1234, 'Mascis', 'JMASCIS', 'IT_PROG', '14-FEB-2008',  
9000);
```

```
UPDATE employees SET salary=9100  
WHERE employee_id=1234;
```

```
DELETE FROM employees WHERE employee_id=1234;
```

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language
Sub conjuntos de comandos da linguagem SQL

DQL – Data Query Language - Concentra comandos que viabilizam a recuperação (consulta) de DADOS do banco de dados:

SELECT

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

SELECT * FROM employees;

SELECT cli_nome, cli_cpf FROM tb_clientes;

SELECT tb_funcionarios.fnc_nome,
tb_dependentes.dep_nome

FROM tb_funcionarios

INNER JOIN tb_dependentes

USING (fnc_codigo);

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language
Sub conjuntos de comandos da linguagem SQL

TCL– Transaction Control – Registram Permanentemente as alterações feitas em linhas ou desfazem essas alterações:

COMMIT

ROLLBACK

SAVEPOINT

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

SAVEPOINT before_salary_update;

UPDATE employees SET salary=9100
WHERE employee_id=1234;

ROLLBACK TO SAVEPOINT before_salary_update;

UPDATE employees SET salary=9200 WHERE
employee_id=1234;

COMMIT;

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

Sub conjuntos de comandos da linguagem SQL

**DCL– Data Control Language – Alteram as
permissões nas estruturas (objetos) de
banco de dados:**

**GRANT
REVOKE**

PROGRAMAÇÃO DE BANCO DE DADOS

SQL – Structured Query Language

CREATE TABLE plantas (
cd_planta NUMBER PRIMARY KEY,
nm_planta VARCHAR2(15));

GRANT SELECT ON plantas TO scott;

REVOKE SELECT ON plantas FROM scott;

DROP TABLE plantas;

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS

- Interface iterativa guiada por comandos do SGBD Oracle para viabilizar consultas, manutenção dos dados e geração de relatórios.



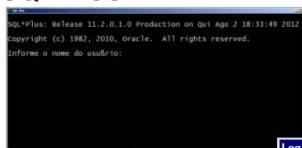
Executar Linha de Comandos SQL



SQL Plus

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS



Log On

User Name:

Password:

Host String:

OK Cancel

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS

```
SQL*Plus: Release 11.2.0.1.0 Production on Ter Ago 14 17:57:10 2012
Copyright (c) 1982, 2010, Oracle. All rights reserved.
Informe o nome do usuário: scott
Informe a senha:
Conectado a:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL>
```

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS

```
SQL> select * from vw_cliente;

CD_CLI  NM_CLI
-----
3476 Antonio
2790 PAULO CEZAR
5476 Otávio Pe SOUZA
4134 Paula
2345 Marcelo Souza
8734 Cezar
9087 Vilma
2546 MARCELO SOUZA
9734 tste

9 linhas selecionadas.
SQL> _
```

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS – Comando / Configurações

cl scr - limpa a área de trabalho

sql> cl scr

disconnect - desconecta o usuário da instancia

sql> disc

connect - conecta o usuário a uma instancia

sql> conn scott/tiger

exit - encerra o aplicativo sqlplus

sql> exit

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS – Comando / Configurações

edit - edita instrução disponível no Buffer
sql> ed

run - exibe e executa instrução disponível no Buffer
sql> run

/ - executa instrução disponível no Buffer
sql> /

@ - permite a execução de um arquivo que contenha um bloco de comando sql
sql> @ c:\temp\script.sql

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS – Comando / Configurações

spool - permite a gravação dos comandos e seus resultados em um arquivo
sql> spool c:\temp\result.txt
sql> select table_name from user_tables;
sql> spool off

describe - permite a visualização da estrutura de uma tabela
sql> desc user_tables

PROGRAMAÇÃO DE BANCO DE DADOS

SQLPLUS – Comando / Configurações

set - altera configurações do ambiente da ferramenta SQLPLUS

sql> SET PAGESIZE 999	Quantidade de linhas da página
sql> SET LINESIZE 999	Quantidade de caracteres na linha
sql> SET HEADING OFF / ON	Suprime/Exibe o cabeçalho das colunas
sql> SET SERVEROUTPUT OFF / ON	Suprime/Exibe mensagens do DBMS_Output
sql> SET PAUSE OFF / ON	Pausa entre paginas

show - exibe valores configurados pelo comando SET
sql> show all

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

Tipos de dados

Tipos Escalares

- Especifica o formato de armazenamento e alcance dos valores que uma variável pode assumir.

Tipos Compostos

- Tipos compostos são aqueles que contêm uma ou mais variáveis escalares (que também são chamadas neste contexto de atributos).

Tipos de Referência

- Assemelham-se aos ponteiros da linguagem C. O tipo de referência mais conhecido é o cursor. Um cursor é um handle ou um ponteiro para uma área de memória conhecida como área de contexto.

Tipo de Objeto

- Objeto da PL/SQL que contém tantos atributos, isto é, variáveis de outros tipos, como métodos. A partir do Oracle 9i o usuário pode criar seus próprios tipos de objeto e trabalhar inclusive com subtipos e com herança.

Tipos escalares

Família numérica:	Família de caracteres:	Família de data/intervalo
BINARY_INTEGER DEC	CHAR	DATE
DECIMAL	CHARACTER	INTERVAL DAY TO SECOND ⁶
DOUBLE PRECISION	LONG	INTERVAL YEAR TO MONTH ³
FLOAT	NCHAR ⁷	TIMESTAMP ³
INT INTEGER	NVARCHAR2 ⁷	TIMESTAMP WITH TIME ZONE ⁸
NATURAL	STRING	TIMESTAMP WITH LOCAL TIME ZONE ⁸
NATURALN	VARCHAR	
	VARCHAR2	
		Família booleana:
		BOOLEAN
NUMBER	Família de Rowid:	
NUMERIC	ROWID	
PLS_INTEGER	UROWID ³	
POSITIVE	Família de confiança:	Família de Raw:
POSITIVEN	MLSLABEL	RAW LONG RAW
REAL		
SIGNTYPE		
SMALLINT		

TIPOS COMPOSTOS

RECORD NESTED
TABLE INDEX-BY
TABLE VARRAY⁹

TIPOS LOB

BFILE¹
LOB¹
CLOB¹
NLOB¹

TIPOS DE REFERENCIA

REF CURSOR REF
object type⁴

TIPOS DE OBJETO

object type⁴
SYS_ANYTYPE⁵
SYS_ANYDATA⁵
SYS_ANYDATASET⁵

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

DDL – DATA DEFINITION LANGUAGE

- Linguagem para definição de objetos.
- Permite a criação e manutenção de objetos de um banco de dados.

PROGRAMAÇÃO DE BANCO DE DADOS

Criação de Tabelas Persistentes

```
CREATE TABLE [Owner.]TableName  
(  
    {{ColumnDefinition} [...]}  
    [PRIMARY KEY (ColumnName [...]) ]  
    [[CONSTRAINT ForeignKeyName]  
        FOREIGN KEY ([ColumnName] [...])  
        REFERENCES RefTableName  
        [(ColumnName [...])] [ON DELETE CASCADE]] [...]  
    ]  
)  
[AS SelectQuery]
```

PROGRAMAÇÃO DE BANCO DE DADOS

Criação de Tabelas Temporárias

```
CREATE GLOBAL TEMPORARY TABLE [Owner.]TableName
(
    {{ColumnDefinition} [...]}
    [PRIMARY KEY (ColumnName [...])] |
    [[CONSTRAINT ForeignKeyName]
     FOREIGN KEY ([ColumnName] [...])
     REFERENCES RefTableName
     [(ColumnName [...])] [ON DELETE CASCADE]] [...]]
    }
)
```

PROGRAMAÇÃO DE BANCO DE DADOS

Exemplos

CREATE TABLE autor

```
( nm_autor    VARCHAR(15),
  sn_autor    VARCHAR(15),
  sx_autor    CHAR(1) check (sx_autor in ('M','F')),
  ds_autor    VARCHAR(2000),
  PRIMARY KEY (nm_autor, sn_autor));
```

CREATE TABLE livro

```
( cd_livro    NUMBER (5),
  tt_livro    VARCHAR(100) UNIQUE,
  nm_autor    VARCHAR(15),
  sn_autor    VARCHAR(15),
  pr_livro    DECIMAL(5,2) NOT NULL,
  PRIMARY KEY (cd_livro),
  FOREIGN KEY (nm_autor, sn_autor)
  REFERENCES autor(nm_autor, sn_autor));
```

PROGRAMAÇÃO DE BANCO DE DADOS

Alteração de Tabelas

```
ALTER TABLE [Owner.]TableName
ADD [COLUMN] ColumnName ColumnDataType [DEFAULT DefaultVal]
[[NOT] INLINE] [UNIQUE] [NULL]]
```

```
ALTER TABLE [Owner.]TableName
DROP {[COLUMN] ColumnName | (ColumnName [... ] )}
```

```
ALTER TABLE [Owner.]TableName
ADD CONSTRAINT ConstraintName PRIMARY KEY (ColumnName [... ])
```

```
ALTER TABLE [Owner.]TableName
MODIFY (ColumnName DEFAULT DefaultVal)
```

PROGRAMAÇÃO DE BANCO DE DADOS

Exemplos

```
alter table tb_notas add cod_nota number(1);

alter table tb_notas drop constraint tb_notas_pk;

alter table tb_notas add constraint tb_notas_pk primary key
(cod_nota_aluno, cod_nota_disc, cod_nota);

alter table tb_notas add constraint cod_nota_ck
check (cod_nota in (1,2,3,4));
```

PROGRAMAÇÃO DE BANCO DE DADOS

Exemplos

```
alter table tb_notas modify nota_aluno_1 number(3,1);

alter table tb_alunos rename column sexo_aluno
to sex_aluno;

alter table tb_notas add constraint tb_notas_alunos_fk
foreign key (cod_nota_aluno)
references tb_alunos (cod_aluno);

alter table tb_notas rename to tbl_notas;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Exclusão de Tabelas

```
DROP TABLE [Owner.]TableName
```

Exclusão de Tabelas

```
DROP TABLE tb_alunos;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Exclusão de Todas as Linhas de uma Tabela

```
TRUNCATE TABLE [Owner.]TableName
```

Exclusão de Tabelas

```
TRUNCATE TABLE tb_notas;
```

PROGRAMAÇÃO DE BANCO DE DADOS

DML – DATA MANIPULATION LANGUAGE

- Linguagem para manipulação de dados.
- Permite a manutenção dos dados de um banco de dados.

- **INSERT**
- **UPDATE**
- **DELETE**

PROGRAMAÇÃO DE BANCO DE DADOS

Inserindo Registros em uma tabela

```
INSERT INTO [Owner.]TableName (ColumnName [...])  
VALUES (Valor [...]);
```

**Obs.: Valores alfanuméricos devem estar entre
aspas simples - 'exemplo'**

PROGRAMAÇÃO DE BANCO DE DADOS

Exemplos

```
INSERT INTO authors  
  (firstname, lastname, sexo, description)  
VALUES  
  ('Kleber', 'Silva', 'M', 'Especialista em Banco de Dados');
```

PROGRAMAÇÃO DE BANCO DE DADOS

Alterando Registros em uma tabela

```
UPDATE [Owner.]TableName  
SET ColumnName = Valor [,...]  
WHERE <Condição>;
```

**Obs.: Valores alfanuméricos devem estar entre
aspas simples - 'exemplo'**

PROGRAMAÇÃO DE BANCO DE DADOS

Exemplos

```
UPDATE authors SET  
  firstname = 'Cleber'  
WHERE lastname = 'Silva';  
-----  
UPDATE employees SET  
  job_id='SA_MAN',  
  salary=salary+1000,  
  department_id=140  
WHERE last_name='Jones' ;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Excluindo Registros em uma tabela

```
DELETE <FROM> [Owner.]TableName  
WHERE <Condição>;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Exemplos

```
DELETE FROM authors  
WHERE lastname = 'Silva';
```

```
DELETE employees  
WHERE id = 2347;
```

PROGRAMAÇÃO DE BANCO DE DADOS

DQL – DATA QUERY LANGUAGE

• **SELECT**

PROGRAMAÇÃO DE BANCO DE DADOS

Recupera (consulta) dados armazenados em uma ou mais tabelas.

Select * from *tabela*;

Select *coluna1*, *coluna2*, (...) from *tabela*;

PROGRAMAÇÃO DE BANCO DE DADOS

Podemos inserir em uma instrução SELECT:

- expressões aritméticas
- apelidos (alias)
- colunas concatenadas
- literais

PROGRAMAÇÃO DE BANCO DE DADOS

expressões aritméticas:

- + soma
- subtração
- * multiplicação
- / divisão

• prioridade: (entre), * e /, + e - depois

• esquerda para a direita, em caso de operadores de prioridade igual.

PROGRAMAÇÃO DE BANCO DE DADOS

Recupera (consulta) dados armazenados em uma ou mais tabelas.

```
SELECT sal* 0.10 + sal  
FROM emp;
```

```
SELECT 1897 * 0.10 + 1897  
FROM dual;
```

```
SELECT sysdate  
FROM dual;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Apelidos:

- Alias (apelidos) para colunas ou tabelas;
- Apenas em tempo de execução;

```
SELECT sal * 0.10 + sal as Salario  
FROM emp;
```

```
SELECT e.sal * 0.10 + e.sal as "Salario Corrigido"  
FROM emp e;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Colunas concatenadas:

Podemos agrupar o valor de duas ou mais colunas (ou literais), provenientes de uma consulta, com ||;

```
SELECT 'nome do funcionario ' || ename  
FROM emp;
```

```
SELECT 'nome do funcionario ' || ename as "Relacao de Funcionarios"  
FROM emp;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula WHERE:

Possibilita a definição de condição para limitar a ação do comando. Dessa forma podemos restringir a quantidade de linhas exibidas em uma consulta.

```
SELECT    cd_funcionario, nm_funcionario
FROM      tb_funcionario
WHERE     cd_cargo = 764;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula WHERE Operadores:

Operador	Operação
=	igual a
>	maior que
>=	maior que ou igual a
<	menor que
<=	menor que ou igual a
<>	diferente

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula WHERE Operadores AND / OR:

AND – OR

Para criarmos condições complexas, podemos concatenar condições com os operadores AND e OR.

```
SELECT    ename, sal
FROM      emp
WHERE     job = 'SALESMAN' and comm = 300;
```

```
SELECT    ename, sal
FROM      emp
WHERE     job = 'SALESMAN' or comm = 300;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula WHERE Operadores:

Operador	Operação
BETWEEN ... AND ...	entre dois valores (inclusive)
IN (<i>list</i>)	combina com qualquer um da lista de valores
LIKE (_ %)	combina com uma padrão
IS NULL	é um valor nulo

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula WHERE Operadores:

Operador	Operação
NOT BETWEEN	não entre 2 valores dados
NOT IN	não na lista de valores dados
NOT LIKE	não semelhante a uma <i>string</i> dada
IS NOT NULL	não é valor nulo

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula ORDER BY:

Possibilita a ordenação do resultado de uma consulta pelo(s) campo(s) indicado(s).

```
SELECT  cd_funcionario, nm_funcionario
FROM    tb_funcionario
WHERE   cd_cargo = 764 and cd_locacao = 32
ORDER BY cd_funcionario;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Clausula ORDER BY opção DESC:

A opção DESC possibilita a inversão da ordenação da consulta.

```
SELECT    cd_funcionario, nm_funcionario
FROM      tb_funcionario
WHERE     cd_cargo = 764 and cd_locacao = 32
ORDER BY  cd_funcionario DESC;
```

PROGRAMAÇÃO DE BANCO DE DADOS

Formato geral:

```
SELECT { *, coluna [alias], ...}
FROM tabela
WHERE condições
ORDER BY {coluna, expressão,...} [ASC|DESC]
```

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES

- Executar cálculos em dados
- Modificar itens de dados individuais
- Manipular a saída de um grupo de linhas
- Alterar formatos de datas para mostrá-los
- Converter tipos de dados de colunas.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES

- CHARACTER
- NUMBER
- DATE
- CONVERSION
- GROUP

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CARACTER

- Aceitam caracteres como entrada
- Retornam caracteres ou números.

LOWER	INSTR
UPPER	LTRIM
INITCAP	RTRIM
LPAD	LENGTH
RPAD	SUBSTR
TRANSLATE	REPLACE

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CARACTER

LOWER (col | valor)

Transforma os caracteres alfabéticos que estão em maiúsculas para minúsculas.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CARACTER

UPPER (col | valor)

Transforma os caracteres alfabéticos que estão em minúsculas para maiúsculas..

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CARACTER

INITCAP (col | valor)

Transforma a primeira letra da *string* em maiúscula.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CARACTER

LPAD (col | valor , n, 'string')
RPAD (col | valor , n, 'string')

Preenche a coluna ou literal da esquerda (lpad) ou direita (rpad) até o tamanho total de *n* posições. O espaço restante é preenchido com a *string* informada. Se a *string* for omitida, o valor é preenchido com espaços em branco.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CHARACTER

SUBSTR (col | valor , pos, n)

Retorna uma *substring* de *n* extraída da coluna ou literal, começando na posição *pos*.
Se *n* é omitido, a *string* é extraída da posição *pos* até o final.
Numeração de caracteres em uma *string*: esquerda para direita, começando com 1.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CHARACTER

INSTR (col | valor , 'string', pos, n)

Localiza a posição do caracter da *enésima* ocorrência de 'string' na coluna ou literal começando na posição *pos*.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CHARACTER

RTRIM (col | valor , 'chars')
LTRIM (col | valor , 'chars')

Remove da direita ou da esquerda da coluna ou literal as ocorrências de *chars*.
Se *char* não é especificada, serão removidos os espaços em branco.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CHARACTER

LENGTH (col | valor)

Retorna o número de caracteres (ou dígitos) da coluna ou literal.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CHARACTER

TRANSLATE (col | valor, de, para)

Transforma o caracter *de* para o caracter *para*.
Mais de um caracter podem ser combinados.
Todas as ocorrências do caracter *de* serão traduzidas pelo correspondente caracter *para*.
Se o correspondente caracter *para* não é fornecido, o caracter *de* é removido

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CHARACTER

REPLACE (col | valor, 'string1', 'string2')

Transforma a coluna ou literal com cada ocorrência de *string1* substituída por *string2*.
Se *string2* for omitida, todas as ocorrências de *string1* serão removidas.

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CARACTER

- Aceitam caracteres como entrada
- Retornam caracteres ou números.

LOWER	INSTR
UPPER	LTRIM
INITCAP	RTRIM
LPAD	LENGTH
RPAD	SUBSTR
TRANSLATE	REPLACE

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

- Aceitam números como entrada e retornam valores numéricos

ROUND	TRUNC
CEIL	FLOOR
POWER	SQRT
SIGN	ABS
MOD	

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

ROUND (col | valor, n)

- Arredonda a coluna expressão ou valor para n casas decimais.
- Se n é omitido, o arredondamento elimina as casas decimais.
- Se n é negativo, os números à esquerda do ponto decimal são arredondados para zero.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

TRUNC (col | valor, n)

- Trunca a coluna expressão ou valor para n casas decimais.
- Se n é omitido, o truncamento elimina as casas decimais.
- Se n é negativo, os números à esquerda do ponto decimal são truncados para zero.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

CEIL (col | valor)

- Encontra o maior inteiro menor ou igual ao valor.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

FLOOR (col | valor)

- Encontra o menor inteiro maior ou igual ao valor

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

POWER (col | valor, n)

- Eleva o valor à enésima potência.
- n pode ser negativo..

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

SQRT (col | valor)

- Calcula a raiz do valor.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

SIGN (col | valor)

- Retorna -1, 0, 1 se o valor é um número negativo, zero ou positivo respectivamente.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

ABS (col | valor)

Retorna o valor absoluto.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

MOD (valor1 | valor2)

Encontra o resto da divisão dos valores :*valor1* por *valor2*.

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES NUMÉRICAS

• Aceitam números como entrada e retornam valores numéricos

ROUND	TRUNC
CEIL	FLOOR
POWER	SQRT
SIGN	ABS
MOD	

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

O Oracle armazena as datas em um formato numérico interno, composto por 7 bytes. Pode variar de **1-jan-4712 AC** até **31-dez-4712 DC**.

Cada byte armazena um dos componentes abaixo:

- Século
- Ano
- Mês
- Dia
- Horas
- Minutos
- Segundos

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

Mudando a Visualização das datas da sessão.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYYMMDD';
```

```
ALTER SESSION SET NLS_DATE_FORMAT = 'HH24:MI:SS';
```

Ou

```
SELECT TO_CHAR(SYSDATE, 'HH24:MI:SS') FROM DUAL;
```

```
SELECT TO_CHAR(SYSDATE, 'YYYY MM DD') FROM DUAL;
```

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

SYSDATE

SYSDATE é uma pseudo-coluna que retorna o valor corrente da data e hora.
É comum selecionar SYSDATE a partir da tabela interna e vazia chamada DUAL.

SELECT SYSDATE FROM DUAL;

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

É possível realizar cálculos com datas, utilizando operadores aritméticos.

Operações possíveis:

- **date + number** — adiciona um número de dias à data, retornando uma data
- **date - number** — subtrai um número de dias da data, retornando uma data
- **date - date** — subtrai uma data de outra, retornando um número de dias.
- **date + number/24** — adiciona um número de horas à data, retornando uma data.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

MONTHS_BETWEEN (data1, data2)

Retorna o número de meses dentre *data1* e *data2*, positivo ou negativo.

ADD_MONTHS (data, n)

Acrescenta **n** meses a data.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

NEXT_DAY (data1, char)

Retorna a data do próximo dia da semana especificado, seguinte à *data1*.
Char pode ser um número representando um dia, ou uma *string*..

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

ROUND (data1)

• Retorna *data1* com a hora definida como 12:00AM (meia-noite).

ROUND (data1,'MONTH')

• Retorna o primeiro dia do mês contido em *data1* se ela estiver na primeira metade do mês, caso contrário, o primeiro dia do próximo mês.

ROUND (data1,'YEAR')

• Retorna o primeiro dia do ano contido em *data1* se ela estiver na primeira metade do ano, caso contrário, o primeiro dia do próximo ano.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DATA

TRUNC (data,'MONTH')

• Retorna o primeiro dia do mês contido em *data*.

TRUNC (data,'YEAR')

• Retorna o primeiro dia do ano contido em *data*.

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

TO_CHAR (num | data, ['fmt'])

Converte o *número* ou *data* para o formato *fmt*.
Se *fmt* for omitido será usado o formato *default* DD-MON-YY para data ou um *char* para números.
Os formatos somente afetam a forma como o dado é mostrado, não alterando seu valor interno.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

TO_NUMBER (char)

Converte *char* que contém números para *number*.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

TO_DATE (char, 'fmt')

Converte o valor de *char* representando uma data, em uma data de acordo com o formato *fmt* especificado.

Se *fmt* for omitido, será usado DD-MON-YY..

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

Formatos de datas :

SCC ou CC –	Século; 'S' coloca o prefixo '-' para BC
YYYY ou SYYYY –	Ano; 'S' coloca o prefixo '-' para BC
YYY ou YY ou Y –	Últimos 3,2,1 dígitos do ano
Y, YYY –	Ano com vírgula na posição indicada
SYEAR ou YEAR –	Ano; 'S' coloca o prefixo '-' para BC
BC ou AD –	Indicador BC /AD
B.C. ou A.D. –	Indicador BC /AD
Q –	Trimestre
MM –	Mês
MONTH –	Nome do mês, com 9 caracteres
MON –	Nome do mês, com 3 caracteres
WW ou W –	Semana do ano ou mês
DDD ou DD ou D –	Dia do ano, mês ou semana
DAY –	Nome do dia, com 9 caracteres

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

Formatos de datas :

DY –	Nome do dia, abreviado com 3 caracteres
J –	Data juliana; número de dias desde 31-dez-4713 AC
AM ou PM –	Indicador AM/PM
A.M. ou P.M. –	Indicador AM/PM
HH ou HH12 –	Hora do dia (1-12)
HH24 –	Hora do dia (1-24)
MI –	Minutos
SS –	Segundo
SSSSS –	Segundos depois da meia-noite (0-86399)
/, –	Pontuação reproduzida no resultado
"..." –	String reproduzida no resultado
fm –	Prefixo de MONTH ou DAY para eliminar espaços
TH –	Sufixo para número ordinal
SP –	Sufixo para <i>spelled-out</i> (soletrar) o número (ex:DDSP)
SPTH –	Sufixo para <i>spelled-out</i> (soletrar) o número(ex:DDSPTH)

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

Formatos de números:

Formato	Significado	Exemplo	
9	Posição numérica	999999	1234
0	Mostra zeros à esquerda	099999	001234
\$	Cifrão flutuante	\$999999	\$1234
.	Ponto decimal na posição indicada	999999.99	1234.00
,	Vírgula na posição indicada	999,999	1,234
MI	Sinal MI à direita de negativos	999999MI	1234-
PR	Parênteses em negativos	999999PR	(1234)
EEEE	Notação científica	99.99EEEE	1.234E+03
V	Multiplica por 10n (n=9s ap"os V)	9999V99	123400
B	Mostra zeros como brancos	B9999.99	1234.00

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

DECODE (col, busca1, resultado1, [busca2, resultado2, ...] default)

- Realiza busca condicionais, funcionando como um *case* ou *if*.
- *Col* e comparado com cada argumento busca e retorna o correspondente resultado quando a combinação for verdadeira.
- Se o valor *default* for omitido, será retornado um NULL..

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

NVL (col , val)

Converte um valor NULL de col para val.

PROGRAMAÇÃO DE BANCO DE DADOS



PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE GRUPO

Função	Valor Retornado
AVG (coluna)	Média, ignorando valores Nulos
COUNT (coluna)	Número de vezes que ocorre valor
MAX (coluna)	Máximo valor
MIN (coluna)	Mínimo valor
STDDEV (coluna)	Desvio padrão, ignorando nulls.
SUM (coluna)	Soma valores, ignorando nulls.
VARIANCE (coluna)	Variância, ignorando nulls.
GREATEST (coluna)	Obtém o maior valor da lista.
LEAST (coluna)	Obtém o menor valor da lista.

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

Cláusula GROUP BY

Utilizada para agrupar o resultado de uma função de grupo em grupos menores. Os agrupamentos menores serão baseados no valor do(s) campo(s) indicado(s)

```
SELECT JOB, AVG(SAL)
FROM EMP
GROUP BY JOB;

JOB          AVG(SAL)
-----
ANALYST      3000
CLERK        1037.5
MANAGER      2758.3333
PRESIDENT    5000
SALESMAN     1400
```

PROGRAMAÇÃO DE BANCO DE DADOS

FUNÇÕES DE CONVERSÃO

Cláusula HAVING

Utilizada na construção de restrições com funções de grupo

```
SELECT JOB,  
        MAX(SAL),  
        COUNT(*)  
FROM EMP  
GROUP BY JOB  
HAVING COUNT(*) >2;
```

JOB	MAX(SAL)	COUNT(*)
-----	-----	-----
CLERK	1300	4
SALESMAN	1600	4
MANAGER	2975	3