ERP – gestão empresarial...

SQL: SQL é uma linguagem para bancos relacionais, relacional funciona com tabelas e, é muito organizado, é uma coleção de relações entre tabelas de duas dimensões todas as tabelas tem ID,

No SQL:

Criando o usuário HR

--

-- Curso - Oracle PL/SQL Especialista: do Básico ao Avançado

--

-- Autor - Emílio Scudero

--

-- Seção 03 - PL/SQL Fundamentos - Instalação e Configuração das conexões no Oracle SQL Developer

--

-- Aula 3 - Criando o Usuário HR

-- Criando o Usuário HR

/\*

No Oracle SQL Developer conecte-se como usuário SYS utilizando a sua conexão para o usuário SYS

Abra este script no Oracle SQL Developer e execute (F5)

/\*

/\*

Configura a sessão para o pluggable database XEPDB1

\*/

ALTER SESSION SET CONTAINER = XEPDB1;

/\*

Remove a conta do usuário HR e remove todos os seus objetos em cascata

\*/

DROP USER HR CASCADE;

/\*

Cria a conta do usuário HR com a senha hr e concede os privilégios

\*/

CREATE USER HR

IDENTIFIED BY hr

DEFAULT TABLESPACE users

TEMPORARY TABLESPACE TEMP

QUOTA unlimited on users

QUOTA 0 on system;

GRANT CONNECT, RESOURCE TO HR;

GRANT CREATE SESSION, CREATE VIEW, CREATE TABLE, ALTER SESSION, CREATE SEQUENCE, CREATE PROCEDURE, CREATE TRIGGER TO HR;

GRANT CREATE SYNONYM, CREATE DATABASE LINK, UNLIMITED TABLESPACE TO HR;

Criando uma conexão para o usuário HR no SQL Developer

--

-- Curso - Oracle PL/SQL Especialista: do Básico ao Avançado

--

-- Autor - Emílio Scudero

--

-- Seção 03 - PL/SQL Fundamentos - Instalação e Configuração das conexões no Oracle SQL Developer

--

-- Aula 4 - Criando uma conexão para o usuário HR no SQL Developer

-- Criando uma conexão para o usuário HR no SQL Developer

1. Criando uma conexão para o usuário HR no SQL Developer

1.1 No SQL Developer

Com o botão direito, selecione Oracle Conexões => Selecione Nova Conexão

Name: hr\_XEPDB1

Nome do usuário: hr

Senha: hr

Atribuição: padrão

Tipo de conexão: Básico

Nome do host: localhost

Porta: 1521

Nome do Serviço: XEPDB1

Clique em Testar

Clique em Salvar

Clique em Conectar, informe a senha do hr (hr)

Clique OK

2. Executando um comando na área de Trabalho

2.1 Execute o comando:

SELECT SYSDATE

FROM dual;

2.2 Desconecte

Preferencias – editor de códigos – fluxo de linhas- mostrar numero de linha – ajuda para ver quando dar algum erro

Criando os objetos do schema do usuário HR

Curso - Oracle PL/SQL Especialista: do Básico ao Avançado

--

-- Autor - Emílio Scudero

--

-- Seção 03 - PL/SQL Fundamentos - Instalação e Configuração das conexões no Oracle SQL Developer

--

-- Aula 5 - Criando os objetos do schema do usuário HR

-- Criando os objetos do schema do usuário HR

/\*

No Oracle SQL Developer conecte-se como usuário HR utilizando a sua conexão para o usuário HR

Abra este script no Oracle SQL Developer e execute (F5)

/\*

/\*

Remove as Tabelas caso já existam

\*/

DROP TABLE job\_history CASCADE CONSTRAINTS;

DROP TABLE departments CASCADE CONSTRAINTS;

DROP TABLE employees CASCADE CONSTRAINTS;

DROP TABLE jobs CASCADE CONSTRAINTS;

DROP TABLE locations CASCADE CONSTRAINTS;

DROP TABLE countries CASCADE CONSTRAINTS;

DROP TABLE regions CASCADE CONSTRAINTS;

DROP SEQUENCE locations\_seq;

DROP SEQUENCE departments\_seq;

DROP SEQUENCE employees\_seq;

ALTER SESSION SET NLS\_LANGUAGE=American;

ALTER SESSION SET NLS\_TERRITORY=America;

/\*

Criação das Tabelas

\*/

/\* Regions \*/

CREATE TABLE regions (

region\_id NUMBER CONSTRAINT regions\_id\_nn NOT NULL,

region\_name VARCHAR2(25)

);

CREATE UNIQUE INDEX reg\_id\_pk

ON regions (region\_id);

ALTER TABLE regions

ADD CONSTRAINT reg\_id\_pk

PRIMARY KEY (region\_id);

/\* Countries \*/

CREATE TABLE countries (

country\_id CHAR(2) CONSTRAINT country\_id\_nn NOT NULL,

country\_name VARCHAR2(40),

region\_id NUMBER,

CONSTRAINT country\_c\_id\_pk PRIMARY KEY (country\_id)

)

ORGANIZATION INDEX;

ALTER TABLE countries

ADD CONSTRAINT countr\_reg\_fk

FOREIGN KEY (region\_id)

REFERENCES regions(region\_id);

/\* Locations \*/

CREATE TABLE locations (

location\_id NUMBER(4),

street\_address VARCHAR2(40),

postal\_code VARCHAR2(12),

city VARCHAR2(30) CONSTRAINT loc\_city\_nn NOT NULL,

state\_province VARCHAR2(25),

country\_id CHAR(2)

);

Populando os objetos do schema do usuário HR

Criando Índices, inserindo comentários e outros no schema do usuário HR

IMPORTANTE

Cada Linha de dados em uma tabela é unicamente identificado por uma primary key (PK)

Para estabelecer uma relação entre duas tabelas definimos como foreing Keys (FK)

Null = ausência de valor

SQL tem poucos comandos, mas esses poucos são muitos poderosos

A linguagem SQL é definida por um conjunto de comandos, e esse conjunto tem alguns sub conjuntos, conjunto de comandos:

DML = Linguagem de Manipulação de Dados. São os comandos que interagem com os dados dentro das tabelas.

EXEMPLOS:

SELECT: para recuperar linhas,

INSERT: para inserir linha a uma tabela,

UPDATE: para atualizar linhas de uma tabela,

DELETE: para deletar uma linha de uma tabela,

MERGE: para comparar duas tabelas a origem e a destino, e atualizar a destino.

DDL = Linguagem de Definição de Dados. São os comandos que interagem com os objetos do banco

CREAETE: para criar tabelas, índices,

ALTER: para alterar estruturas de tabelas,

DROP: para remover tabelas,

RENAME: para renomear um objeto,

TRUNCATE: para deixar uma tabela vazia,

COMMENT: para adicionar um comentário para uma tabela, ou coluna

Data control language – COMANDOS DE CONTROLE DE SEGURANÇA

DCL: GRANT: para passar privilégios

REVOKE: para revogar privilégios

Transaction Control - Controle de transações

COMMIT: para efetivar uma transação, não pode mais ser desfeita

Rollback: desfaz a transação

SAVEPOINT: Criar um ponto de controle na transação.

COMANDOS

DESCRIBE: Me mostra a estrutura da tabela que estou pesquisando

DESCRIBE employees

DESC employess

DESC departments

DESC Jobs

DESC departments

\* : exibe todas colunas da tabela

FROM: é para indicar qual é o nome da tabela

SELECT \*

FROM employees;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Se eu quero exibir somente algumas colunas da tabela faço desta forma:

SELECT employee\_id, first\_name, last\_name, salary

FROM employees;

SELECT department\_id, department\_name, manager\_id

FROM departments;

Utilizando operadores aritméticos

SELECT first\_name, last\_name, salary, salary \* 1.15

FROM employees;

Esse comando ira me mostrar o salario multiplicado por 1.15 da tabela employees

Regras de Precedência de Operadores

SELECT first\_name, last\_name, salary, salary + 100 \* 1.15

FROM employees;

Dessa forma acima o calculo que sera feito primeiro é o de multiplicação, mas como eu quero que some primeiro faço da seguinte forma abaixo

-- Utilizando parênteses para alterar a precedência

SELECT first\_name, last\_name, salary, (salary + 100) \* 1.15

FROM employees;

Utilizando Alias de Coluna

SELECT first\_name AS nome, last\_name AS sobrenome, salary AS salário

FROM employees;

Desta forma acima eu altero o nome de uma coluna com AS, na parte abaixo altero sem

SELECT first\_name nome, last\_name sobrenome, salary salário

FROM employees;

SELECT first\_name "Nome", last\_name "Sobrenome", salary "Salário ($)", commission\_pct "Percentual de comissão"

FROM employees;

Utilizando Operador de concatenação

SELECT first\_name || ' ' || last\_name || ', data de admissão: ' || hire\_date "Funcionário"

FROM employees;

Desta forma irei juntar essas informações

Utilizando Operador de concatenação e Strings de caracteres

SELECT first\_name || ' ' || last\_name || ', data de admissão: ' || hire\_date "Funcionário"

FROM employees;

­­

Utilizando Operador alternativo para aspas

SELECT department\_name || q'[ Department's Manager Id: ]'

|| manager\_id "Departamento e Gerente"

FROM departments;

Dessa forma posso usar aspas durante meu SELECT sem acabar com a STRING

Utilizando DISTINCT para eliminar linhas duplicadas

SELECT DISTINCT department\_id

FROM employees;

SELECT DISTINCT last\_name

FROM employees;

SELECT DISTINCT last\_name, first\_name

FROM employees;

Utilizando DISTINCT faço que a informação que eu quero apareça apenas uma vez

Utilizando a cláusula WHERE

SELECT employee\_id, last\_name, job\_id, department\_id

FROM employees

WHERE department\_id = 60;

Where é uma condição, então ira exibir apenas o nome, job id, do departament que é do 60, e assim por diante

SELECT employee\_id, last\_name, job\_id, department\_id

FROM employees

WHERE job\_id = 'IT\_PROG';

Utilizando Strings de caractere na cláusula WHERE

SELECT first\_name, last\_name, job\_id, department\_id, hire\_date

FROM employees

WHERE last\_name = 'King';

Desta forma ira buscar na tabela as informações pedidas, contanto que tenha a informação King dentro da tabela

Utilizando Strings de caractere com datas na cláusula WHERE

SELECT first\_name, last\_name, job\_id, department\_id, hire\_date

FROM employees

WHERE hire\_date = '30/01/04';

Desta forma ira trazer as informações pedidas contanto que o a informação dentro da coluna do “hire date” seja igual 30/01/04