**Coleções**

Tipos de dados especiais

**Tipo Objeto**

Podemos criar tipos de dados adicionais e depois fazer referência a eles dentro de outros objetos. Os tipos criados são gravados no esquema armazenado no banco de dados e as outras declarações que acessam o banco de dados podem fazer uso das definições desses tipos.

**1 – Criação do tipo t\_pessoa**

CREATE TYPE T\_PESSOA AS OBJECT (

CODIGO\_PESSOA NUMBER(5),

NOME\_PESSOA VARCHAR2(50),

ENDERECO VARCHAR2(50)) NOT FINAL;

**2 – Criação da tabela pessoa a partir do tipo t\_pessoa**

-- nao da erro, logo not final nao adianta de nada

CREATE TABLE PESSOA of T\_pessoa;

**3 – Fazendo consulta na tabela criada**

select \* from pessoa;

**4 – Incluindo dados na tabela criada**

insert into pessoa values (1,'leonardo','rio grande do sul');

**5 – Criação do tipo t\_fisica que herdou atributos de t\_pessoa**

CREATE TYPE T\_FISICA UNDER T\_PESSOA (

CPF CHAR(11),

SEXO CHAR(1));

**6 – Criação do tipo t\_juridica que herdou atributos de t\_pessoa**

CREATE TYPE T\_JURIDICA UNDER T\_PESSOA (

CNPJ CHAR(14),

INSC\_ESTADUAL VARCHAR(30));

**7 – Criação da tabela pessoa\_fisica a partir de t\_fisica e da tabela pessoa\_juridica a partir de t\_juridica**

CREATE TABLE PESSOA\_FISICA of T\_FISICA;

CREATE TABLE PESSOA\_JURIDICA of T\_JURIDICA;

**Tabelas aninhadas (nested tables)**

Suponha que precisemos armazenar os dados do funcionario como também de seus dependentes (CPF) em uma mesma tabela. Como o número de dependentes é variável uma possível solução seria com a utilização de tabelas aninhadas. Neste caso o campo dependente na tabela funcionario será uma tabela aninhada.

Para se criar uma tabela aninhada deve-se:

1 – Criar um tipo de dados do tipo tabela

CREATE OR REPLACE TYPE tabdep AS TABLE OF number(11);

/

2 – Criar tabela funcionario indicando a tabela aninhada

CREATE TABLE funcionario(mat NUMBER(5), nome varchar2(40), dependente tabdep)

NESTED TABLE dependente STORE AS dependente\_tab;

Para incluir dados na tabela aninhada.

INSERT INTO funcionario VALUES (1, 'leonardo', tabdep(11));

INSERT INTO funcionario VALUES (2, 'monica', tabdep(21, 22));

INSERT INTO funcionario VALUES (3, 'tais', tabdep(31, 32, 33));

Consultando os dados incluídos

Select \* from funcionario;

OU ENTÃO

SELECT mat, nome, COLUMN\_VALUE FROM funcionario t1, TABLE(t1.dependente) t2;

**O aninhamento de tabelas também pode ser feito com o uso do tipo de objetos. Ver exemplo a seguir:**

**1 – Criação do tipo de objeto t\_endereco**

CREATE OR REPLACE TYPE t\_endereco AS OBJECT (

rua VARCHAR2(50),

numero INTEGER,

cidade VARCHAR2(50),

uf CHAR(2),

cep CHAR(9));

**2 – Criação de lista\_enderecos do tipo tabela**

CREATE TYPE lista\_enderecos AS TABLE OF t\_endereco;

**3 – Criação da tabela cliente\_loja que tem o campo enderecos\_clientes sendo do tipo lista\_enderecos**

CREATE TABLE cliente\_loja (

codigo NUMBER(5),

nome VARCHAR2(50),

enderecos\_clientes lista\_enderecos)

NESTED TABLE enderecos\_clientes

STORE AS enderecos\_clientes\_tab;

**4 – Inclusão de linhas na tabela**

INSERT INTO cliente\_loja VALUES (111, 'ALICE',

lista\_enderecos(

t\_endereco('RUA FELICIDADE', 100, 'SAO PAULO','SP', '00000-000'),

t\_endereco('RUA ALEGRIA', 200, 'RIO DE JANEIRO','RJ', '11111-111')

)

);

INSERT INTO cliente\_loja VALUES (112, 'leonardo',

lista\_enderecos(

t\_endereco('RUA MONICA', 100, 'SAO CAETANO','SP', '00000-000'),

t\_endereco('RUA FERNANDA', 200, 'NITEROI','RJ', '11111-111')

)

);

**5 – Visualização dos dados**

select \* from cliente\_loja;

SELECT c.codigo, c.nome, e.rua

FROM cliente\_loja c, TABLE(c.enderecos\_clientes) e;

SELECT c.codigo, c.nome,e.\*

FROM cliente\_loja c, TABLE(c.enderecos\_clientes) e;

-- nao permite consultar dados oriundos de tabela aninhada

select \* from enderecos\_clientes\_tab;

1 – Crie o tipo de objeto esporte com os atributos codigo e nome. Crie os tipos de objeto abaixo, todos herdam do objeto esporte.

1.a – futebol com os atributos posicao e idade

1.b – boxe com os atributos peso e altura

1.c – corredor com os atributos modalidade e pais origem

Crie 3 tabelas a partir de cada um dos tipos de objeto criados em 1.a, 1.b e 1.c

Inclua dados na tabela, altere-os, apague alguns deles, visualize-os.

CREATE TYPE T\_esporte AS OBJECT (

CODIGO NUMBER(5),

NOME VARCHAR2(50)) NOT FINAL;

CREATE TYPE T\_futebol UNDER T\_esporte (

posicao varchar2(20),

idade number(3));

CREATE TYPE T\_boxe UNDER T\_esporte (

peso number(5,2),

altura number(3,2));

CREATE TYPE T\_corredor UNDER T\_esporte (

modalidade varchar2(20),

pais varchar2(20));

CREATE TABLE FUTEBOL of T\_futebol;

CREATE TABLE BOXE of T\_boxe;

CREATE TABLE CORREDOR of T\_corredor;

select \* from futebol;

select \* from boxe;

select \* from corredor;

2 - Crie uma tabela onde seja possível armazenar vários emails em tabela aninhada e outra coluna onde seja possível armazenar vários telefones em tabela aninhada.

CREATE OR REPLACE TYPE tabemail AS TABLE OF varchar2(20);

CREATE OR REPLACE TYPE tabtel AS TABLE OF number(9);

CREATE TABLE usuario(mat NUMBER(5), nome varchar2(10), email tabemail, tel tabtel)

NESTED TABLE email STORE AS tabemail\_tab,

NESTED TABLE tel STORE AS tabtel\_tab;

Gravar dados

INSERT INTO usuario VALUES (1, 'Ricardo', tabemail('ricardo@1'), tabtel(678901234) );

INSERT INTO usuario VALUES (2, 'Maria', tabemail('maria@1','maria@2'), tabtel(23112345));

INSERT INTO usuario VALUES (3, 'Monica', tabemail('monica@1'), tabtel(99790056,981837766));

INSERT INTO usuario VALUES (4, 'Leonardo', tabemail('leonardo@1','leonardo@2','leonardo@3'), tabtel(983506655,23112345));

SELECT mat, nome, t2.COLUMN\_VALUE FROM usuario t1, TABLE(t1.email) t2;

SELECT mat, nome, t2.COLUMN\_VALUE FROM usuario t1, TABLE(t1.tel) t2;

SELECT mat, nome, t2.COLUMN\_VALUE,t3.COLUMN\_VALUE FROM usuario t1, TABLE(t1.email) t2, TABLE(t1.tel) t3;

3 – Crie a tabela Olimpiadas com os campos ano, pais\_sede e as tabelas aninhadas do tipo futebol, boxe e corredor conforme tipos de objeto criados no exercício 1.

3.a – Faça consulta para retornar somente os nomes dos jogadores de futebol e do ano e pais\_sede.

3.b– Faça consulta para retornar somente os nomes dos boxistas e do ano e pais\_sede.

3.c – Faça consulta para retornar somente os nomes dos corredores e do ano e pais\_sede.

3.d – Faça consulta para retornar todos os dados, de todos os esportistas.

CREATE TYPE tb\_futebol AS TABLE OF t\_futebol;

CREATE TYPE tb\_boxe AS TABLE OF t\_boxe;

CREATE TYPE tb\_corredor AS TABLE OF t\_corredor;

CREATE TABLE olimpiadas(

ano number(4),

pais\_sede varchar2(20),

futebol tb\_futebol,

boxe tb\_boxe,

corredor tb\_corredor)

NESTED TABLE futebol STORE AS futebol\_tab,

NESTED TABLE boxe STORE AS boxe\_tab,

NESTED TABLE corredor STORE AS corredor\_tab;

insert into olimpiadas values (2024,'JAPAO',tb\_futebol(

t\_futebol (1,'leonardo','atacante',43),

t\_futebol (2,'Danrley','goleiro',53)

),

tb\_boxe(

t\_boxe(3,'Popo',75,1.80),

t\_boxe(4,'Maguila',120,1.90)

),

tb\_corredor(

t\_corredor(5,'Piquet','F1','Brasil'),

t\_corredor(6,'Nakajima','F2','Japao')

)

)

select o.ano,o.pais\_sede, f.\*

from olimpiadas o, TABLE(o.futebol) f

select o.ano,o.pais\_sede, b.\*

from olimpiadas o, TABLE(o.boxe) b

select o.ano,o.pais\_sede, c.\*

from olimpiadas o, TABLE(o.corredor) c

select o.ano,o.pais\_sede, f.\*, b.\*, c.\*

from olimpiadas o, TABLE(o.futebol) f,

TABLE(o.boxe) b, TABLE(o.corredor) c

4. Crie as tabelas do modo tradicional (4 tabelas) e retorne todos os dados como em 3.d com utilização do comando JOIN.

drop table futebol;

drop table boxe;

drop table olimpiadas;

drop table corredor;

CREATE table futebol (

CODIGO NUMBER(5),

NOME VARCHAR2(50),

posicao varchar2(20),

idade number(3),

ano number(4)

);

CREATE table boxe (

CODIGO NUMBER(5),

NOME VARCHAR2(50),

peso number(5,2),

altura number(3,2),

ano number(4)

);

CREATE table corredor (

CODIGO NUMBER(5),

NOME VARCHAR2(50),

modalidade varchar2(20),

pais varchar2(20),

ano number(4)

);

create table olimpiadas (

ano number(4),

pais\_sede varchar2(20)

);

insert into olimpiadas values (2024,'JAPAO');

insert into futebol values (1,'leonardo','atacante',43,2024);

insert into futebol values (2,'Danrley','goleiro',53,2024);

insert into boxe values (3,'Popo',75,1.80,2024);

insert into boxe values (4,'Maguila',120,1.90,2024);

insert into corredor values (5,'Piquet','F1','Brasil',2024);

insert into corredor values (6,'Nakajima','F2','Japao',2024);

SELECT o.\*, f.codigo, f.nome,f.posicao,f.idade,

b.codigo, b.nome,b.altura, b.peso,

c.codigo, c.nome,c.ano, c.pais

from olimpiadas o

left join futebol f on o.ano = f.ano

left join boxe b on o.ano = b.ano

left join corredor c on o.ano = c.ano