Exercicios banco de dados

CREATE TABLE PRODS

(

codigo NUMERIC(3) NOT NULL,

nome VARCHAR(50) NOT NULL,

preco NUMERIC (5,2) NOT NULL,

tipo CHAR(1) NULL, -- [S]uprimento, [C]omponente, [P]eriférico

CONSTRAINT PK1 PRIMARY KEY (codigo)

);

SELECT \*

FROM prods;

INSERT INTO PRODS VALUES( 10, 'HD' ,200 ,'C');

INSERT INTO PRODS VALUES( 11, 'Memoria' ,250 ,'C');

INSERT INTO PRODS VALUES( 12, 'Impressora' ,680 ,'P');

INSERT INTO PRODS VALUES( 13, 'Processador' ,600 ,'C');

INSERT INTO PRODS VALUES( 14, 'DVD-RW' ,2 ,'S');

INSERT INTO PRODS VALUES( 15, 'Papel A4' ,19 ,'S');

INSERT INTO PRODS VALUES( 16, 'Scanner' ,199 ,'P');

-- Quantos produtos existem na tabela PRODS?

SELECT COUNT(\*)

FROM prods;

-- b) Quantos tipos de produtos existem na tabela PRODS?

SELECT COUNT(DISTINCT tipo)

FROM prods;

-- c) Quantos produtos existem de cada tipo?

SELECT tipo, COUNT(\*)

FROM prods

GROUP BY tipo;

DESC prods;

-- d) Qual a média de preço de todos os produtos?

SELECT AVG(preco)

FROM prods;

-- e) Qual a média de preço dos suprimentos (tipo ‘S’)?

SELECT AVG(preco)

FROM prods

Where tipo = 'S';

-- f) Qual a média de preço dos produtos de cada tipo?

SELECT tipo, AVG(preco) Media

FROM prods

GROUP BY tipo;

DESC prods;

SELECT \* FROM prods;

ALTER TABLE PRODS ADD (usuario NUMBER(1) NULL);

UPDATE PRODS

SET usuario = 1

WHERE codigo IN (10,12,13,14);

UPDATE PRODS

SET usuario = 2

WHERE usuario IS NULL;

SELECT tipo, usuario, AVG(preco)

FROM PRODS

GROUP BY tipo, usuario

ORDER BY tipo, usuario;

-- Deixando um produto sem usuario associado

UPDATE PRODS

SET usuario = NULL

WHERE codigo = 13;

-- Executar novamente as linhas do select acima

UPDATE PRODS

SET usuario = 2

WHERE codigo = 14;

-- Uso de HAVING grupos que farao parte

SELECT tipo, usuario, AVG(preco)

FROM PRODS

GROUP BY tipo, usuario

HAVING AVG(preco) > 300

ORDER BY tipo, usuario;

DESC prods;

select \* from prods;

INSERT INTO PRODS(codigo,nome,preco,tipo,usuario)

VALUES(9, 'HD2' ,400 ,'C',1);

SELECT codigo, preco

FROM prods

WHERE codigo = 9;

SELECT codigo,nome, preco

FROM prods

WHERE preco > 200;

SELECT codigo,nome, preco

FROM prods

WHERE preco >

(SELECT preco

FROM prods

WHERE codigo = 9);

SELECT codigo, preco,tipo

FROM prods

WHERE tipo = 'C';

SELECT codigo,nome, tipo, preco

FROM prods

WHERE preco > 200;

SELECT codigo,nome, tipo, preco

FROM prods

WHERE (preco > 200) and (tipo = 'C');

SELECT codigo,nome, tipo, preco, rowid

FROM prods;

-- Criar indices

-- CREATE INDEX nomeindex ON prods (CampoIndice);

CREATE SEQUENCE seq\_titulacoes START WITH 6 ;

SELECT seq\_titulacoes.currval FROM DUAL;

CREATE TABLE TITULACOES

(

cod\_titulacao NUMBER(4) DEFAULT seq\_titulacoes.nextval NOT NULL,

titulo VARCHAR (20) NOT NULL,

CONSTRAINT pk\_titulacoes PRIMARY KEY (cod\_titulacao)

) ;

Parte 2

CREATE TABLE PRODS

(

codigo NUMERIC(3) NOT NULL,

nome VARCHAR(50) NOT NULL,

preco NUMERIC (5,2) NOT NULL,

tipo CHAR(1) NULL, -- [S]uprimento, [C]omponente, [P]eriférico

CONSTRAINT PK1 PRIMARY KEY (codigo)

);

SELECT \*

FROM prods;

desc PRODS;

SELECT AVG(preco)

FROM prods

WHERE TIPO = 'C';

INSERT INTO PRODS VALUES( 10, 'HD' ,200 ,'C');

INSERT INTO PRODS VALUES( 11, 'Memoria' ,250 ,'C');

INSERT INTO PRODS VALUES( 12, 'Impressora' ,680 ,'P');

INSERT INTO PRODS VALUES( 13, 'Processador' ,600 ,'C');

INSERT INTO PRODS VALUES( 14, 'DVD-RW' ,2 ,'S');

INSERT INTO PRODS VALUES( 15, 'Papel A4' ,19 ,'S');

INSERT INTO PRODS VALUES( 16, 'Scanner' ,199 ,'P');

-- Quantos produtos existem na tabela PRODS?

SELECT COUNT(\*)

FROM prods;

-- b) Quantos tipos de produtos existem na tabela PRODS?

SELECT COUNT(DISTINCT tipo)

FROM prods;

-- c) Quantos produtos existem de cada tipo?

SELECT tipo, COUNT(\*)

FROM prods

GROUP BY tipo;

DESC prods;

-- d) Qual a média de preço de todos os produtos?

SELECT AVG(preco)

FROM prods;

-- e) Qual a média de preço dos suprimentos (tipo ‘S’)?

SELECT AVG(preco)

FROM prods

Where tipo = 'S';

-- f) Qual a média de preço dos produtos de cada tipo?

SELECT tipo, AVG(preco) Media

FROM prods

GROUP BY tipo;

DESC prods;

SELECT \* FROM prods;

ALTER TABLE PRODS ADD (usuario NUMBER(1) NULL);

UPDATE PRODS

SET usuario = 1

WHERE codigo IN (10,12,13,14);

UPDATE PRODS

SET usuario = 2

WHERE usuario IS NULL;

SELECT tipo, usuario, AVG(preco)

FROM PRODS

GROUP BY tipo, usuario

ORDER BY tipo, usuario;

-- Deixando um produto sem usuario associado

UPDATE PRODS

SET usuario = NULL

WHERE codigo = 13;

-- Executar novamente as linhas do select acima

UPDATE PRODS

SET usuario = 2

WHERE codigo = 14;

-- Uso de HAVING grupos que farao parte

SELECT tipo, usuario, AVG(preco)

FROM PRODS

GROUP BY tipo, usuario

HAVING AVG(preco) > 300

ORDER BY tipo, usuario;

DESC prods;

select \* from prods;

INSERT INTO PRODS(codigo,nome,preco,tipo,usuario)

VALUES(9, 'HD2' ,400 ,'C',1);

SELECT codigo, preco

FROM prods

WHERE codigo = 9;

SELECT codigo,nome, preco

FROM prods

WHERE preco > 200;

SELECT codigo,nome, preco

FROM prods

WHERE preco >

(SELECT preco

FROM prods

WHERE codigo = 9);

SELECT codigo, preco,tipo

FROM prods

WHERE tipo = 'C';

SELECT codigo,nome, tipo, preco

FROM prods

WHERE preco > 200;

SELECT codigo,nome, tipo, preco

FROM prods

WHERE (preco > 200) and (tipo = 'C');

SELECT codigo,nome, tipo, preco, rowid

FROM prods;

-- Criar indices

-- CREATE INDEX nomeindex ON prods (CampoIndice);

CREATE SEQUENCE seq\_titulacoes START WITH 6 ;

SELECT seq\_titulacoes.currval FROM DUAL;

CREATE TABLE TITULACOES

(

cod\_titulacao NUMBER(4) DEFAULT seq\_titulacoes.nextval NOT NULL,

titulo VARCHAR (20) NOT NULL,

CONSTRAINT pk\_titulacoes PRIMARY KEY (cod\_titulacao)

) ;

CREATE TABLE autores (

cod\_autor NUMBER (4) NOT NULL,

nome VARCHAR2 (100) NOT NULL,

descricao VARCHAR2 (1024),

CONSTRAINT PK\_AUTORES PRIMARY KEY (cod\_autor)

);

select count(\*) from autores;

select count(nome) from autores;

select count() from autores; -- erro

select count(\*) from table autores;

CREATE TABLE Estados (

uf CHAR ( 2 ),

nome VARCHAR2 ( 40 ) NOT NULL,

regiao CHAR ( 2 ) NULL,

area NUMBER(10,2),

CONSTRAINT pk\_estados primary key (uf)

);

CREATE TABLE Cidades (

cod\_cidade NUMBER ( 4 ) NOT NULL,

nome VARCHAR2 ( 60 ) NOT NULL,

uf CHAR ( 2 ),

CONSTRAINT pk\_cidades PRIMARY KEY (cod\_cidade),

CONSTRAINT fk\_est\_cid FOREIGN KEY (uf) REFERENCES estados (uf)

);

select e.nome, count(\*) from cidades c join estados e using(uf) group by e.nome having regiao = 'S' where count(\*) > 500;

select e.nome, count(\*) from cidades e join estados c using(uf) group by e.nome where regiao = 'S' having count(\*) > 500;

select e.nome, count(\*) from cidades c join estados e using(uf) group by e.nome having regiao = 'S' where count(\*) > 500;

select e.nome, count(\*) from cidades c join estados e using(uf) where regiao = 'S' group by e.nome having count(\*) > 500; -- correta

select \* from autores;

desc autores;

CREATE TABLE autores (

cod\_autor NUMBER (4) NOT NULL,

nome VARCHAR2 (100) NOT NULL,

descricao VARCHAR2 (1024),

CONSTRAINT PK\_AUTORES PRIMARY KEY (cod\_autor)

);

INSERT INTO autores VALUES( 2, 'Carlos' ,'Professos de Frances');

update autores set nome = 'Joao', descricao='Professor de Ingles' where cod\_autor=1;

update nome = 'Rodrigo Silva' on table autores where cod\_autor=1; -- erro

update autores set nome = 'Rodrigo Silva' where cod\_autor=1; -- correto

update autores set nome = 'Rodrigo Silva', descricao='Professor de BD' where cod\_autor=1; -- correto

update autores set nome = 'Rodrigo Silva', descricao='Professor de BD'; correto

update autores set nome = 'Rodrigo Silva' and descricao='Professor de BD' where cod\_autor=1; -- erro

CREATE TABLE Filmes (

titulo VARCHAR(250) NOT NULL,

ano NUMBER(4),

diretor VARCHAR(100),

genero VARCHAR(20) NOT NULL,

atoresprincipais VARCHAR(1000)

);

insert into filmes (titulo, ano, diretor) values ('Pulp Fiction', null, 'Quentin Tarantino'); -- ERRO

insert into filmes values ('Pulp Fiction', null, 'Quentin Tarantino', 'Crime,Thriller', 'John Travolta, Uma Thurman and Samuel L. Jackson');

insert into filmes (titulo, ano, diretor, genero, atoresprincipais)

values ('Pulp Fiction', 1994, null, 'Crime,Thriller', 'John Travolta, Uma Thurman and Samuel L. Jackson');

insert into filmes (titulo, ano, diretor, genero, atoresprincipais)

values ('Pulp Fiction', 1994, null, 'Crime,Thriller', 'John Travolta, Uma Thurman and Samuel L. Jackson');

select \* from estados;

DESC estados;

CREATE TABLE estados2 (

uf CHAR ( 2 ) NOT NULL,

nome VARCHAR2 ( 40 ) NOT NULL,

regiao CHAR ( 2 ) NOT NULL,

CONSTRAINT PK\_ESTADOS2 PRIMARY KEY (uf)

);

select \* from estados2;

INSERT INTO estados2 (uf,nome,regiao) VALUES( 'MT', 'Mato Grosso' ,'CO');

INSERT INTO estados2 (uf,nome,regiao) VALUES( 'GO', 'Goias' ,'CO');

INSERT INTO estados2 (uf,nome,regiao) VALUES( 'SP', 'São Paulo' ,'SD');

INSERT INTO estados2 (uf,nome,regiao) VALUES( 'MG', 'Minas Gerais' ,'SD');

select regiao, count(\*) from estados2 group by uf;

select uf, count(\*) from estados2 group by regiao;

select regiao, count(\*) from estados2 group by regiao;

select regiao, count(uf) from estados2;