

/*PRIMER PROBLEMA*/

//1- Elimine las tablas y créelas nuevamente

drop table inscriptos;

drop table cursos;

drop table socios;

drop table profesores;

```
create table socios(  
documento char(8) not null,  
nombre varchar2(40),  
domicilio varchar2(30),  
primary key (documento)  
);
```

```
create table profesores(  
documento char(8) not null,  
nombre varchar2(40),  
domicilio varchar2(30),  
primary key (documento)  
);
```

```
create table cursos(  
numero number(2),  
deporte varchar2(20),  
dia varchar2(15),  
documentoprofesor char(8),
```

```
constraint CK_inscriptos_dia
check (dia in('lunes','martes','miercoles','jueves','viernes','sabado')),
constraint FK_documentoprofesor
foreign key (documentoprofesor)
references profesores(documento),
primary key (numero)
);
```

```
create table inscriptos(
documentosocio char(8) not null,
numero number(2) not null,
matricula char(1),
constraint CK_inscriptos_matricula check (matricula in('s','n')),
constraint FK_documentosocio
foreign key (documentosocio)
references socios(documento),
constraint FK_numerocurso
foreign key (numero)
references cursos(numero),
primary key (documentosocio,numero)
);
```

//2- Ingrese algunos registros para todas las tablas

```
insert into socios values('30000000','Fabian Fuentes','Caseros 987');
insert into socios values('31111111','Gaston Garcia','Guemes 65');
insert into socios values('32222222','Hector Huerta','Sucre 534');
insert into socios values('33333333','Ines Irala','Bulnes 345');
```

```
insert into profesores values('22222222','Ana Acosta','Avellaneda 231');
insert into profesores values('23333333','Carlos Caseres','Colon 245');
insert into profesores values('24444444','Daniel Duarte','Sarmiento 987');
insert into profesores values('25555555','Esteban Lopez','Sucre 1204');
```

```
insert into cursos values(1,'tenis','lunes','22222222');
insert into cursos values(2,'tenis','martes','22222222');
insert into cursos values(3,'natacion','miercoles','22222222');
insert into cursos values(4,'natacion','jueves','23333333');
insert into cursos values(5,'natacion','viernes','23333333');
insert into cursos values(6,'futbol','sabado','24444444');
insert into cursos values(7,'futbol','lunes','24444444');
insert into cursos values(8,'basquet','martes','24444444');
```

```
insert into inscriptos values('30000000',1,'s');
insert into inscriptos values('30000000',3,'n');
insert into inscriptos values('30000000',6,null);
insert into inscriptos values('31111111',1,'s');
insert into inscriptos values('31111111',4,'s');
insert into inscriptos values('32222222',8,'s');
```

//3- Elimine la vista "vista_club"

```
drop view vista_club;
```

//4- Cree una vista en la que aparezca el nombre del socio, el deporte, el día, el nombre del profesor y el estado de la matrícula (deben incluirse los socios que no están inscriptos en ningún deporte, los cursos para los cuales no hay inscriptos y los profesores que no tienen designado deporte también)

```
CREATE VIEW vista_club AS
```

SELECT

s.nombre AS socio_nombre,
c.deporte AS deporte,
c.dia AS dia,
p.nombre AS profesor_nombre,
i.matricula AS matricula

FROM

socios s

LEFT JOIN inscriptos i ON s.documento = i.documento

FULL JOIN cursos c ON i.numero = c.numero

FULL JOIN profesores p ON c.documento = p.documento;

//5- Muestre la información contenida en la vista (11 registros)

SELECT * FROM vista_club;

//6- Realice una consulta a la vista donde muestre la cantidad de socios inscriptos en cada deporte (agrupe por deporte y día) ordenados por cantidad

SELECT deporte, dia, COUNT(socio_nombre) AS cantidad

FROM vista_club

WHERE socio_nombre IS NOT NULL

GROUP BY deporte, dia

ORDER BY cantidad;

//7- Muestre (consultando la vista) los cursos (deporte y día) para los cuales no hay inscriptos (3 registros)

SELECT deporte, dia

FROM vista_club

WHERE socio_nombre IS NULL AND deporte IS NOT NULL;

//8- Muestre los nombres de los socios que no se han inscripto en ningún curso (consultando la vista) (1 registro)

```
SELECT socio_nombre
```

```
FROM vista_club
```

```
WHERE deporte IS NULL AND socio_nombre IS NOT NULL;
```

//9- Muestre (consultando la vista) los profesores que no tienen asignado ningún deporte aún (1 registro)

```
SELECT profesor_nombre
```

```
FROM vista_club
```

```
WHERE deporte IS NULL AND profesor_nombre IS NOT NULL;
```

//10- Muestre (consultando la vista) el nombre de los socios que deben matrículas (1 registro)

```
SELECT DISTINCT socio_nombre
```

```
FROM vista_club
```

```
WHERE matricula = 'n' OR matricula IS NULL;
```

//11- Consulte la vista y muestre los nombres de los profesores y los días en que asisten al club para dictar sus clases (9 registros)

```
SELECT profesor_nombre, dia
```

```
FROM vista_club
```

```
WHERE profesor_nombre IS NOT NULL AND dia IS NOT NULL;
```

//12- Muestre la misma información anterior pero ordenada por día

```
SELECT profesor_nombre, dia
```

```
FROM vista_club
```

```
WHERE profesor_nombre IS NOT NULL AND dia IS NOT NULL
```

```
ORDER BY
```

CASE dia

WHEN 'lunes' THEN 1

WHEN 'martes' THEN 2

WHEN 'miercoles' THEN 3

WHEN 'jueves' THEN 4

WHEN 'viernes' THEN 5

WHEN 'sabado' THEN 6

WHEN 'domingo' THEN 7

END;

//13- Muestre todos los socios que son compañeros en tenis los lunes (2 registros)

SELECT socio_nombre

FROM vista_club

WHERE deporte = 'tenis' AND dia = 'lunes' AND socio_nombre IS NOT NULL;

//14- Intente crear una vista denominada "vista_inscriptos" que muestre la cantidad de inscriptos por curso, incluyendo el número del curso, el nombre del deporte y el día

CREATE VIEW vista_inscriptos AS

SELECT c.numero, c.deporte, c.dia, COUNT(i.documentosocio) AS cantidad

FROM cursos c

LEFT JOIN inscriptos i ON c.numero = i.numero

GROUP BY c.numero, c.deporte, c.dia;

//15- Elimine la vista "vista_inscriptos" y créela para que muestre la cantidad de inscriptos por curso, incluyendo el número del curso, el nombre del deporte y el día

DROP VIEW vista_inscriptos;

```
CREATE VIEW vista_inscriptos AS  
SELECT c.numero, c.deporte, c.dia, COUNT(i.documentosocio) AS cantidad  
FROM cursos c  
LEFT JOIN inscriptos i ON c.numero = i.numero  
GROUP BY c.numero, c.deporte, c.dia;
```

//16- Consulte la vista (9 registros)

```
SELECT * FROM vista_inscriptos;
```

/*SEGUNDO PROBLEMA*/

//1- Elimine las tabla "inscriptos", "socios" y "cursos"

```
drop table inscriptos;
```

```
drop table socios;
```

```
drop table cursos;
```

//2- Cree las tablas

```
create table socios(
```

```
documento char(8) not null,
```

```
nombre varchar2(40),
```

```
domicilio varchar2(30),
```

```
constraint PK_socios_documento
```

```
primary key (documento)
```

```
);
```

```
create table cursos(
```

```
numero number(2),
```

```
deporte varchar2(20),
```

```
dia varchar2(15),
```

```
constraint CK_inscriptos_dia check (dia  
in('lunes','martes','miercoles','jueves','viernes','sabado')),
```

```
profesor varchar2(20),
```

```
constraint PK_cursos_numero
```

```
primary key (numero)
```

```
);
```

```
create table inscriptos(
```



```

documentosocio char(8) not null,
numero number(2) not null,
matricula char(1),
constraint PK_inscriptos_documento_numero
primary key (documentosocio,numero),
constraint FK_inscriptos_documento
foreign key (documentosocio)
references socios(documento),
constraint FK_inscriptos_numero
foreign key (numero)
references cursos(numero)
);

```

//3- Ingrese algunos registros para todas las tablas

```
insert into socios values('30000000','Fabian Fuentes','Caseros 987');
```

```
insert into socios values('31111111','Gaston Garcia','Guemes 65');
```

```
insert into socios values('32222222','Hector Huerta','Sucre 534');
```

```
insert into socios values('33333333','Ines Irala','Bulnes 345');
```

```
insert into cursos values(1,'tenis','lunes','Ana Acosta');
```

```
insert into cursos values(2,'tenis','martes','Ana Acosta');
```

```
insert into cursos values(3,'natacion','miercoles','Ana Acosta');
```

```
insert into cursos values(4,'natacion','jueves','Carlos Caseres');
```

```
insert into cursos values(5,'futbol','sabado','Pedro Perez');
```

```
insert into cursos values(6,'futbol','lunes','Pedro Perez');
```

```
insert into cursos values(7,'basquet','viernes','Pedro Perez');
```

```
insert into inscriptos values('30000000',1,'s');
```

```
insert into inscriptos values('30000000',3,'n');
insert into inscriptos values('30000000',6,null);
insert into inscriptos values('3111111',1,'s');
insert into inscriptos values('3111111',4,'s');
insert into inscriptos values('32222222',1,'s');
insert into inscriptos values('32222222',7,'s');
```

//4- Realice un join para mostrar todos los datos de todas las tablas, sin repetirlos
(7 registros)

```
SELECT s.*, c.*, i.*
FROM socios s
INNER JOIN inscriptos i ON s.documento = i.documentosocio
INNER JOIN cursos c ON i.numero = c.numero;
```

//5- Elimine la vista "vista_cursos"

```
DROP VIEW vista_cursos;
```

//6- Cree la vista "vista_cursos" que muestre el número, deporte y día de todos los cursos.

```
CREATE VIEW vista_cursos AS
SELECT numero, deporte, dia
FROM cursos;
```

//7- Consulte la vista ordenada por deporte (7 registros)

```
SELECT * FROM vista_cursos ORDER BY deporte;
```

//8- Ingrese un registro mediante la vista "vista_cursos" y vea si afectó a "cursos"

```
INSERT INTO vista_cursos (numero, deporte, dia) VALUES (8, 'golf', 'lunes');
```

// Verificar tabla cursos:

```
SELECT * FROM cursos;
```

//9- Actualice un registro sobre la vista y vea si afectó a la tabla "cursos"

```
UPDATE vista_cursos SET dia = 'martes' WHERE numero = 1;
```

// Verificar cambios en cursos:

```
SELECT * FROM cursos;
```

//10- Elimine un registro de la vista para el cual no haya inscriptos y vea si afectó a "cursos"

```
UPDATE vista_cursos SET dia = 'martes' WHERE numero = 1;
```

// Verificar cambios en cursos:

```
SELECT * FROM cursos;
```

//11- Intente eliminar un registro de la vista para el cual haya inscriptos

```
DELETE FROM vista_cursos WHERE numero = 1; // Error por FK
```

//12- Elimine la vista "vista_inscriptos" y créela para que muestre el documento y nombre del socio, el numero de curso, el deporte y día de los cursos en los cuales está inscripto

```
DROP VIEW vista_inscriptos;
```

```
CREATE VIEW vista_inscriptos AS
```

```
SELECT s.documento, s.nombre, i.numero, c.deporte, c.dia
```

```
FROM socios s
```

```
JOIN inscriptos i ON s.documento = i.documentosocio
```

```
JOIN cursos c ON i.numero = c.numero;
```

//13- Intente ingresar un registro en la vista:

```
insert into vista_inscriptos values('32222222','Hector Huerta',6,'futbol','lunes');
```

//No lo permite porque la modificación afecta a más de una tabla base.

//14- Intente actualizar el documento de un socio (no lo permite)

```
UPDATE vista_inscriptos SET documento = '99999999' WHERE documento =  
'32222222';
```

//15- Elimine un registro mediante la vista

```
DELETE FROM vista_inscriptos  
WHERE documento = '32222222' AND numero = 7;
```

//16- Verifique que el registro se ha eliminado de "inscriptos"

```
SELECT * FROM inscriptos;
```

/*TERCER PROBLEMA*/

//1- Eliminamos la tabla y la creamos

drop table empleados;

```
create table empleados(  
    documento char(8),  
    nombre varchar2(20),  
    apellido varchar2(20),  
    sueldo number(6,2),  
    cantidadhijos number(2,0),  
    fechaingreso date,  
    primary key(documento)  
);
```

//2- Ingrese algunos registros

```
insert into empleados values('22222222','Juan','Perez',200,2,'10/10/1980');  
insert into empleados values('22333333','Luis','Lopez',250,0,'01/02/1990');  
insert into empleados values('22444444','Marta','Perez',350,1,'02/05/1995');  
insert into empleados values('22555555','Susana','Garcia',400,2,'15/12/2018');  
insert into empleados values('22666666','Jose Maria','Morales',500,3,'25/08/2015');
```

//3- Cree (o reemplace) el procedimiento almacenado llamado
"pa_aumentarsueldo" que aumente los sueldos inferiores al promedio en un 20%

```
CREATE OR REPLACE PROCEDURE pa_aumentarsueldo AS v_promedio  
NUMBER;
```

```
BEGIN
```

```
-- Calcular promedio de sueldos
```

```
SELECT AVG(sueldo) INTO v_promedio FROM empleados;
```

```
-- Aumentar sueldos inferiores al promedio en 20%
```

```
UPDATE empleados
```

```
SET sueldo = sueldo * 1.20
```

```
WHERE sueldo < v_promedio;
```

```
COMMIT;
```

```
END pa_aumentarsueldo;
```

```
/*4- Ejecute el procedimiento creado anteriormente*/
```

```
EXECUTE pa_aumentarsueldo;
```

```
//5- Verifique que los sueldos han aumentado
```

```
SELECT * FROM empleados;
```

```
//6- Ejecute el procedimiento nuevamente
```

```
EXECUTE pa_aumentarsueldo;
```

```
//7- Verifique que los sueldos han aumentado
```

```
SELECT * FROM empleados;
```

```
//8- Elimine la tabla "empleados_antiguos"
```

```
DROP TABLE empleados_antiguos PURGE;
```

```
//9- Cree la tabla "empleados_antiguos"
```

```
CREATE TABLE empleados_antiguos(
```

```
documento CHAR(8),
```

```
nombre VARCHAR2(40)
);
```

//10- Cree (o reemplace) un procedimiento almacenado que ingrese en la tabla "empleados_antiguos" el documento, nombre y apellido (concatenados) de todos los empleados de la tabla "empleados" que ingresaron a la empresa hace más de 10 años

```
CREATE OR REPLACE PROCEDURE pa_cargar_antiguos AS
BEGIN
    INSERT INTO empleados_antiguos (documento, nombre)
    SELECT documento, nombre || ' ' || apellido
    FROM empleados
    WHERE fechaingreso < ADD_MONTHS(SYSDATE, -120); -- 10 años = 120 meses

    COMMIT;
END pa_cargar_antiguos;
```

//11- Ejecute el procedimiento creado anteriormente

```
EXECUTE pa_cargar_antiguos;
```

//12- Verifique que la tabla "empleados_antiguos" ahora tiene registros (3 registros)

```
SELECT * FROM empleados_antiguos;
```

/*CUARTO PROBLEMA*/

//1- Elimine la tabla y créela con la siguiente estructura

drop table pacientes;

create table pacientes(

documento char(8),

nombre varchar2(30),

edad number(2),

sexo char(1)

);

//2- Ingrese los siguientes registros

insert into pacientes values('1111111','Acosta Ana',40,'f');

insert into pacientes values('22222222','Bustos Betina',35,'f');

insert into pacientes values('33333333','Caseres Carlos',18,'m');

insert into pacientes values('44444444','Dominguez Diego',6,'m');

insert into pacientes values('15555555','Fuentes Fabiana',55,'f');

insert into pacientes values('26666666','Gomez Gaston',38,'m');

insert into pacientes values('37777777','Irala Ines',16,'f');

insert into pacientes values('38888888','Juarez Julieta',17,'f');

insert into pacientes values('40000000','Lopez Lucas',3,'m');

//3- Realice una función que reciba la edad del paciente y retorne la cadena "menor" o "mayor" según sea menor a 18 años o no

CREATE OR REPLACE FUNCTION f_edad_tipo(p_edad IN NUMBER)

RETURN VARCHAR2

IS


```
BEGIN  
  
IF p_edad < 18 THEN  
    RETURN 'menor';  
  
ELSE  
    RETURN 'mayor';  
  
END IF;  
  
END f_edad_tipo;
```

//4- Realice una función que reciba el caracter correspondiente al sexo del paciente y retorne la cadena "femenino" o "masculino"

```
CREATE OR REPLACE FUNCTION f_sexo_tipo(p_sexo IN CHAR)  
RETURN VARCHAR2  
IS  
BEGIN  
    RETURN CASE p_sexo  
        WHEN 'f' THEN 'femenino'  
        WHEN 'm' THEN 'masculino'  
    END;  
  
END f_sexo_tipo;
```

//5- Realice un "select" mostrando el nombre del paciente y empleando las funciones de los puntos 3 y 4, dos columnas que indiquen si es mayor o menor de edad y el sexo.

```
SELECT  
  
    nombre AS paciente,  
  
    f_edad_tipo(edad) AS "Tipo Edad",  
  
    f_sexo_tipo(sexo) AS Sexo  
  
FROM pacientes;
```


/*QUINTO PROBLEMA*/

//1- Elimine la tabla y créela con la siguiente estructura

DROP TABLE empleados;

CREATE TABLE empleados(

 nombre VARCHAR2(40),

 sueldo NUMBER(6,2)

);

//2- Ingrese algunos registros

INSERT INTO empleados VALUES('Acosta Ana',550);

INSERT INTO empleados VALUES('Bustos Bernardo',850);

INSERT INTO empleados VALUES('Caseros Carolina',900);

INSERT INTO empleados VALUES('Dominguez Daniel',490);

INSERT INTO empleados VALUES('Fuentes Fabiola',820);

INSERT INTO empleados VALUES('Gomez Gaston',740);

INSERT INTO empleados VALUES('Huerta Hernan',1050);

//3- Muestre la suma total de todos los sueldos realizando un "select" (5400)

SELECT SUM(sueldo) FROM empleados;

//4- Se necesita incrementar los sueldos en forma proporcional, en un 10% cada vez y controlar que la suma total de sueldos no sea menor a \$7000, si lo es, el bucle debe continuar y volver a incrementar los sueldos, en caso de superarlo, se saldrá del ciclo repetitivo; es decir, este bucle continuará el incremento de sueldos hasta que la suma de los mismos llegue o supere los 7000.

DECLARE

 v_total NUMBER;

```
BEGIN  
  
LOOP  
  
    SELECT SUM(sueldo) INTO v_total FROM empleados;  
  
    EXIT WHEN v_total >= 7000;  
  
    UPDATE empleados SET sueldo = sueldo * 1.10;  
  
END LOOP;  
  
COMMIT;  
  
END;
```

//5- Verifique que los sueldos han sido incrementados y la suma de todos los sueldos es superior a 7000

```
SELECT SUM(sueldo) FROM empleados;
```

//6- Muestre el sueldo máximo realizando un "select"

```
SELECT MAX(sueldo) FROM empleados;
```

//7- Se necesita incrementar los sueldos en forma proporcional, en un 5% cada vez y controlar que el sueldo máximo alcance o supere los \$1600, al llegar o superarlo, el bucle debe finalizar. Incluya una variable contador que cuente cuántas veces se repite el bucle

```
DECLARE  
  
    v_max NUMBER;  
  
    v_counter NUMBER := 0;  
  
BEGIN  
  
    LOOP  
  
        SELECT MAX(sueldo) INTO v_max FROM empleados;  
  
        EXIT WHEN v_max >= 1600;  
  
        UPDATE empleados SET sueldo = sueldo * 1.05;  
  
        v_counter := v_counter + 1;
```

```
END LOOP;  
  
COMMIT;  
  
END;
```

//8- Verifique que los sueldos han sido incrementados y el sueldo máximo es igual o superior a 1600

```
SELECT MAX(sueldo) FROM empleados;
```

//9- Muestre el sueldo mínimo realizando un "select"

```
SELECT MIN(sueldo) FROM empleados;
```

//10- Se necesita incrementar los sueldos en forma proporcional, en un 10% cada vez y controlar que el sueldo mínimo no supere los \$900. Emplee la sintaxis "if CONDICION then exit"

```
DECLARE
```

```
    v_min NUMBER;
```

```
BEGIN
```

```
    LOOP
```

```
        UPDATE empleados SET sueldo = sueldo * 1.10;
```

```
        SELECT MIN(sueldo) INTO v_min FROM empleados;
```

```
        IF v_min > 900 THEN
```

```
            EXIT;
```

```
        END IF;
```

```
    END LOOP;
```

```
    COMMIT;
```

```
END;
```

//11- Muestre el sueldo mínimo realizando un "select"

```
SELECT MIN(sueldo) FROM empleados;
```