

DESARROLLO BBDD – GRUPO 7

PRÁCTICA 1-2



INDICE

1. EJERCICIOS PRÁCTICA 1 – PARTE 2	1
1.1 EJERCICIO 1	1
1.2 EJERCICIO 2	3
1.3 EJERCICIO 3	4
1.4 EJERCICIO 4	12
1.5 EJERCICIO 5	14

COMPONENTES DEL GRUPO	TRABAJO
Juan Perea Campos	25%
Pablo Moraga Navas	25%
José Javier Sevilla Toboso	25%
Rodrigo Díaz Hellín-Valera	25%

1. EJERCICIOS PRÁCTICA 1 – PARTE 2

1.1 EJERCICIO 1

TABLAS

```

create table cuenta
(
    nro_cuenta char(10),
    balance int,
    primary key (nro_cuenta)
);

create table transaccion
(
    nro_cuenta char(10),
    hora_modificacion date NOT NULL,
    id_cliente char(10),
    ant_balance int,
    act_balance int,
    primary key (nro_cuenta, hora_modificacion)
);

insert into cuenta(nro_cuenta, balance) values ('001', '15000');
insert into cuenta(nro_cuenta, balance) values ('002', '10000');
insert into cuenta(nro_cuenta, balance) values ('003', '20000');
insert into transaccion(nro_cuenta, hora_modificacion, id_cliente, ant_balance, act_balance) values
('001', '15/05/1997', 'c1', '100', '200');
insert into transaccion(nro_cuenta, hora_modificacion, id_cliente, ant_balance, act_balance) values
('001', '28/02/2017', 'c2', '300', '500');
insert into transaccion(nro_cuenta, hora_modificacion, id_cliente, ant_balance, act_balance) values
('001', '1/02/1995', 'c3', '200', '300');

```

	NRO_CUENTA	BALANCE
1	001	50000
2	002	80000
3	003	20000

	NRO_CUENTA	HORA_MODIFICACION	ID_CLIENTE	ANT_BALANCE	ACT_BALANCE
1	001	15/05/97	c1	100	200
2	001	28/02/17	c2	300	500
3	001	01/02/95	c3	200	300

TRIGGER

```

create or replace trigger ejercicio_1
after insert or update of balance on cuenta
for each row
begin
insert into transaccion (nro_cuenta, hora_modificacion, id_cliente, ant_balance, act_balance)
values (:new.nro_cuenta, sysdate, 555, :old.balance, :new.balance);
end;

```

UPDATE

```

update cuenta set balance = '50000' where nro_cuenta = '001';
update cuenta set balance = '80000' where nro_cuenta = '002';

```

RESULTADO

	NRO_CU...	HORA_MODIFICACION	ID_CLIENTE	ANT_BALANCE	ACT_BALANCE
1	001	15/05/97	c1	100	200
2	001	28/02/17	c2	300	500
3	001	01/02/95	c3	200	300
4	001	04/04/18	555	50000	50000
5	002	04/04/18	555	80000	80000

1.2 EJERCICIO 2

TABLAS

```
CREATE TABLE distance (
ruta CHAR(10) CONSTRAINT PKDDistancia PRIMARY KEY,
distancia_k NUMBER,
distancia_m NUMBER
)
```

TRIGGERS

```
CREATE OR REPLACE TRIGGER insert_ejercicio2
BEFORE INSERT ON distance
FOR EACH ROW
BEGIN
IF: NEW.distancia_k IS NOT NULL AND :NEW.distancia_k > 0 THEN
:NEW.distancia_m := :NEW.distancia_k*0.621371;
ELSIF :NEW.distancia_m IS NOT NULL AND :NEW.distancia_m > 0 THEN
:NEW.distancia_k := :NEW.distancia_m*1.609344;
END IF;
END;
```

```
CREATE OR REPLACE TRIGGER update_ejercicio2
BEFORE INSERT ON distance
FOR EACH ROW
BEGIN
IF: NEW.distancia_k <> :NEW.distancia_k THEN
:NEW.distancia_m := :NEW.distancia_k*0.621371;
ELSIF :OLD.distancia_m <> :NEW.distancia_m THEN
:NEW.distancia_k := :NEW.distancia_m*1.609344;
END IF;
END;
```

INSERT

```
INSERT INTO distance VALUES ('Ruta1', 2000,0);
INSERT INTO distance VALUES ('Ruta2', 0,5000);
```

UPDATE

```
UPDATE distance SET distancia_m=1500 WHERE ruta='Ruta1';
UPDATE distance SET distancia_k=3000 WHERE ruta='Ruta2';
SELECT* FROM distance;
```

1.3 EJERCICIO 3

TABLAS

```
DROP TABLE DEPARTMENT;
DROP TABLE EMPLOYEE;
DROP TABLE SITE_DEPTS;
DROP TABLE PROJECT;
DROP TABLE WORK_IN;
```

```
CREATE TABLE DEPARTMENT(
  NAMED VARCHAR(15) NOT NULL,
  NUMBERD NUMBER NOT NULL,
  DATEINICGTE DATE,
  PRIMARY KEY (NUMBERD));
```

```
CREATE TABLE EMPLOYEE(
  NAMEP VARCHAR(15) NOT NULL,
  INIC CHAR,
  SURNAME VARCHAR(15) NOT NULL,
  NSS CHAR(9) NOT NULL,
  DATEN DATE,
  ADRESS VARCHAR(30),
  SEX CHAR,
  SALARY DECIMAL(10,2),
  ND NUMBER NOT NULL,
  PRIMARY KEY(NSS));
```

```
CREATE TABLE SITE_DEPTS(
  NUMBERD NUMBER NOT NULL,
  SITED VARCHAR(15) NOT NULL,
  PRIMARY KEY (NUMBERD, SITED));
```

```
CREATE TABLE PROJECT(
  NAMEPR VARCHAR(15) NOT NULL,
  NUMBERPR INT NOT NULL,
  SITEPR VARCHAR(15),
  NUMD INT NOT NULL,
  PRIMARY KEY (NUMBERPR));
```

```
CREATE TABLE WORK_IN(
  NSSE CHAR(9) NOT NULL,
  NUMP INT NOT NULL,
  HOURS DECIMAL(2,1) NOT NULL,
  PRIMARY KEY(NSSE, NUMP));
```

TRIGGER

```

CREATE OR REPLACE TRIGGER insert_empleado
BEFORE INSERT ON EMPLOYEE

FOR EACH ROW

DECLARE
    numD DEPARTMENT.NUMBERD%TYPE;
BEGIN
    SELECT numberd into numD FROM DEPARTMENT WHERE numberd=:new.nd;

    IF numD<>:new.nd THEN
        DBMS_STANDARD.raise_application_error(20000, 'Error al insertar:
        El departamento no existe.');
```

INSERT

```

INSERT INTO DEPARTMENT VALUES('Informática',1,'');
INSERT INTO EMPLOYEE VALUES ('Rodrigo','R','Díaz-Hellin','nss1','',
|'Criptana','H',2500.00,1);
SELECT * FROM EMPLOYEE;
SELECT * FROM DEPARTMENT;
```

RESULTADO

NAMEP	I SURNAME	NSS	DATEN	ADRESS	S	SALARY	ND
Rodrigo	R Díaz-Hellin	nss1		Criptana	H	2500	1

NAMED	NUMBERD	DATEINICGTE
Informática	1	

TRIGGER

```

CREATE OR REPLACE TRIGGER insert_SITE_DEPTS
BEFORE INSERT ON SITE_DEPTS

FOR EACH ROW

DECLARE
    numD1 SITE_DEPTS.NUMBERD%TYPE;
BEGIN
    SELECT numberd into numD1 FROM DEPARTMENT WHERE numberd=:new.numberd;

    IF numD1<>:new.numberd THEN
        DBMS_STANDARD.raise_application_error(20000, 'Error al insertar: El
        departamento no existe.');
```

INSERT

```

INSERT INTO DEPARTMENT VALUES('Matematicas',2,'');
INSERT INTO SITE_DEPTS VALUES(2,'Site2');
SELECT * FROM SITE_DEPTS;
SELECT * FROM DEPARTMENT;

```

RESULTADO

NUMBERD	SITED
2	Site2

NAMED	NUMBERD	DATEINICGTE
Informática	1	
Matematicas	2	

TRIGGER

```

CREATE OR REPLACE TRIGGER insert_PROJECT
BEFORE INSERT ON PROJECT

FOR EACH ROW

DECLARE
    numD2 DEPARTMENT.NUMBERD%TYPE;
BEGIN
    SELECT numberd into numD2 FROM DEPARTMENT WHERE numberd=:new.NUMD;

    IF numD2<>:new.NUMD THEN
        DBMS_STANDARD.raise_application_error(20000, 'Error al insertar:
        El departamento no existe.');
```

INSERT

```

INSERT INTO DEPARTMENT VALUES('Fisica',3,'');
INSERT INTO PROJECT VALUES('Proyecto Fisica',1,'SitePR1',3);
SELECT * FROM PROJECT;
SELECT * FROM DEPARTMENT;

```


TRIGGER

```

CREATE OR REPLACE TRIGGER insert_WORK_IN
BEFORE INSERT ON WORK_IN

FOR EACH ROW

DECLARE
    numD3 EMPLOYEE.NSS%TYPE;
    numD33 PROJECT.NUMBERPR%TYPE;

BEGIN
    SELECT nss into numD3 FROM EMPLOYEE WHERE nss=:new.nsse;
    SELECT numberpr into numD3 FROM PROJECT WHERE numberpr=:new.nump;

    IF numD3<>:new.NSSE AND numD33<>:new.NUMP THEN
        DBMS_STANDARD.raise_application_error(20000,'Error al insertar:
        El empleado y el proyecto no existe.');
```

```

END;
```

INSERT

```

INSERT INTO WORK_IN VALUES ('nss1',1,1);
SELECT * FROM WORK_IN;
SELECT * FROM DEPARTMENT;
```

RESULTADO

NSSE	NUMP	HOURS
nss1	1	1

NAMED	NUMBERD	DATEINICGTE
Informática	1	
Matematicas	2	
Fisica	3	

TRIGGER

```

CREATE OR REPLACE TRIGGER update_empleado
BEFORE UPDATE ON EMPLOYEE

FOR EACH ROW

DECLARE
    numD DEPARTMENT.NUMBERD%TYPE;
BEGIN
    IF numD<>:new.nd THEN
        SELECT numberd into numD FROM DEPARTMENT WHERE numberd=:new.nd;
        DBMS_STANDARD.raise_application_error(20000,'Error al insertar:
        El departamento no existe.');
```

```

    END IF;

END;
```

RESULTADO

NAMEP	I SURNAME	NSS	DATEN	ADRESS	S	SALARY	ND
Rodrigo	R Díaz-Hellin	nss1		Criptana	H	2500	2

NAMED	NUMBERD	DATEINICGTE
Informática	1	
Matemáticas	2	
Física	3	

TRIGGER

```

CREATE OR REPLACE TRIGGER update_SITE_DEPTS
BEFORE UPDATE ON SITE_DEPTS

FOR EACH ROW

DECLARE
    numD1 SITE_DEPTS.NUMBERD%TYPE;
BEGIN
    SELECT numberd into numD1 FROM DEPARTMENT WHERE numberd=:new.numberd;

    IF numD1<>:new.numberd THEN
        DBMS_STANDARD.raise_application_error(20000, 'Error al insertar:
        El departamento no existe.');
```

```

    END IF;

END;
```

UPDATE

```
UPDATE SITE_DEPTS SET numberd=2 WHERE numberd=1;
SELECT * FROM SITE_DEPTS;
SELECT * FROM DEPARTMENT
```

RESULTADO

NUMBERD	SITED
2	Site2

NAMED	NUMBERD	DATEINICGTE
Informática	1	
Matematicas	2	
Fisica	3	

TRIGGER

```
CREATE OR REPLACE TRIGGER update_PROJECT
BEFORE INSERT ON PROJECT

FOR EACH ROW

DECLARE
    numD2 DEPARTMENT.NUMBERD%TYPE;

BEGIN
    SELECT numberd INTO numD2 FROM DEPARTMENT WHERE numberd=:new.NUMD;

    IF numD2<>:new.NUMD THEN
        DBMS_STANDARD.raise_application_error(20000, 'Error al insertar:
        El departamento no existe.');
```

UPDATE

```
UPDATE PROJECT SET NUMD=2 WHERE NUMD=1;
SELECT * FROM PROJECT;
SELECT * FROM DEPARTMENT;
```

RESULTADO

NAMEPR	NUMBERPR	SITEPR	NUMD
Proyecto Fisica	1	SitePR1	3

NAMED	NUMBERD	DATEINICGTE
Informática	1	
Matematicas	2	
Fisica	3	

TRIGGER

```
CREATE OR REPLACE TRIGGER update_WORK_IN
BEFORE UPDATE ON WORK_IN
```

```
FOR EACH ROW
```

```
DECLARE
```

```
numD3 EMPLOYEE.NSS%TYPE;
```

```
numD33 PROJECT.NUMBERPR%TYPE;
```

```
BEGIN
```

```
SELECT nss into numD3 FROM EMPLOYEE WHERE nss=:new.nsse;
```

```
SELECT numberpr into numD33 FROM PROJECT WHERE numberpr=:new.nump;
```

```
IF numD3<>:new.NSSE AND numD33<>:new.NUMP THEN
```

```
DBMS_STANDARD.raise_application_error(20000, 'Error al insertar:
```

```
El empleado y el proyecto no existen.');
```

```
END IF;
```

```
END;
```

RESULTADO

NSSE	NUMP	HOURS
nss2	2	1

INSERT

```
INSERT INTO EMPLOYEE VALUES('Joseja','J','Sevilla','nss4','','LaRoda','H',6000.00,3);
```

```
INSERT INTO PROJECT VALUES('Quimica',5,'SitePR1',3);
```

```
UPDATE WORK_IN SET nsse='nss4',nump=2 WHERE nsse='nss1' AND nump=1;
```

```
SELECT * FROM WORK_IN;
```

```
SELECT * FROM DEPARTMENT;
```

TRIGGERS

```
CREATE OR REPLACE TRIGGER delete_departamento
AFTER DELETE ON department
FOR EACH ROW

BEGIN
    DELETE FROM employee WHERE nd = :OLD.numberd;
    DELETE FROM SITE_DEPTS WHERE numberd = :OLD.numberd;
    DELETE FROM PROJECT WHERE numd = :OLD.numberd;
    DELETE FROM WORK_IN WHERE nump = :OLD.numberd;

END;

CREATE OR REPLACE TRIGGER delete_empleado
AFTER DELETE ON EMPLOYEE
FOR EACH ROW
BEGIN
    DELETE FROM WORK_IN WHERE nsse = :OLD.nss;

END;

CREATE OR REPLACE TRIGGER delete_PROJECT
AFTER DELETE ON PROJECT
FOR EACH ROW
BEGIN
    DELETE FROM WORK_IN WHERE NUMP = :OLD.numberpr;

END;
```

1.4 EJERCICIO 4

TABLAS

```
CREATE TABLE empleado(
nss number,
nombre char(10),
salario number,
num_dep char(10),
primary key(nss));

CREATE TABLE departamento(
num_dep char(10),
nombre char(10),
presupuesto number,
primary key(num_dep)
);
```

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	NSS		NUMBER		No	(null)		1		(null)
2	NOMBRE		CHAR(10 BYTE)		Yes	(null)		2		(null)
3	SALARIO		NUMBER		Yes	(null)		3		(null)
4	NUM_DEP		CHAR(10 BYTE)		Yes	(null)		4		(null)

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	NUM_DEP		CHAR(10 BYTE)		No	(null)		1		(null)
2	NOMBRE		CHAR(10 BYTE)		Yes	(null)		2		(null)
3	PRESUPUESTO		NUMBER		Yes	(null)		3		(null)

```
INSERT INTO departamento VALUES ('012','Gest Admin','13000');
```

```
INSERT INTO departamento VALUES ('013','Informat','20000');
```

```
INSERT INTO empleado VALUES ('1','Joseja JJ','3000','013');
```

```
INSERT INTO empleado VALUES ('2','Jupecan','100','012');
```

❖	NSS	❖	NOMBRE	❖	SALARIO	❖	NUM_DEP
1	1		Joseja JJ		3000		013
2	2		Jupecan		100		012

TRIGGER

```
CREATE OR REPLACE TRIGGER triggerempleado4
AFTER INSERT ON EMPLEADO
FOR EACH ROW
DECLARE
presu number;
presu2 number;
BEGIN
pres := :new.salario;
SELECT presupuesto INTO presu2 FROM departamento
WHERE departamento.num_dep= : new.num_dep;
UPDATE departamento SET presupuesto = presu2+presu
WHERE departamento.num_dep=:new.num_dep;
END;
```

INSERT

```
INSERT INTO empleado VALUES ('3','Rodri','3000','013');

INSERT INTO empleado VALUES ('4','Pablo','50000','012');
```

RESULTADO

	NUM_DEP	NOMBRE	PRESUPUESTO
1	012	GestAdmi	63000
2	013	Inform	23000

1.5 EJERCICIO 5

TABLAS

```
drop table discografica;
drop table genero;

create table discografica(
id int,
nombre varchar(60),
numero_generos int,
numero_operaciones int,
totales_generos int,
numero_operaciones_tot int,
primary key (id)
);

create table genero(
id int,
siglas varchar(5),
nombre varchar(60),
bpm int,
discografica int,
primary key (id)
);

insert into discografica values (1, 'Sony Music', 0,0,0,0);
insert into discografica values (2, 'Warner Bros Records', 0,0,0,0);
insert into discografica values (3, 'BMG Music', 0,0,0,0);
insert into discografica values (4, 'EMI', 0,0,0,0);
insert into discografica values (5, 'Universal Music Group', 0,0,0,0);
```


TRIGGERS

```

create or replace trigger e5_1
after insert on genero
for each row
begin
update discografica set numero_generos = numero_generos+1 where discografica.id=:new.discografica;
dbms_output.put_line('after row');
end;

create or replace trigger e5_2
after insert on genero
for each row
begin
update discografica set numero_operaciones = numero_operaciones+1 where discografica.id=:new.discografica;
dbms_output.put_line('before row');
end;

create or replace trigger e5_3
after insert on genero
begin
update discografica set totales_generos = totales_generos+1;
dbms_output.put_line('after statement');
end;

create or replace trigger e5_4
after insert on genero
begin
update discografica set numero_operaciones_tot = numero_operaciones_tot+1;
dbms_output.put_line('before statement');
end;

```

INSERT

```

select * from discografica;
insert into genero values (1, 'TEC', 'Techno', 100,1);
insert into genero values (2, 'BLU', 'Blue', 50,5);
insert into genero values (3, 'JAZ', 'Jazz', 80,3);
insert into genero values (4, 'CLA', 'Clásica', 70,4);
insert into genero values (5, 'ELE', 'Electro', 110,4);
insert into genero values (6, 'RRR', 'Rock and Roll', 100,1);
select * from discografica;

```

RESULTADO

ID	NOMBRE	NUMERO_GENEROS	NUMERO_OPERACIONES	TOTALES_GENEROS	NUMERO_OPERACIONES_TOT
1	1 Sony Music	0	0	0	0
2	2 Warner Bros Records	0	0	0	0
3	3 BMG Music	0	0	0	0
4	4 EMI	0	0	0	0
5	5 Universal Music Group	0	0	0	0

ID	NOMBRE	NUMERO_GENEROS	NUMERO_OPERACIONES	TOTALES_GENEROS	NUMERO_OPERACIONES_TOT
1	1 Sony Music	2	2	6	6
2	2 Warner Bros Records	0	0	6	6
3	3 BMG Music	1	1	6	6
4	4 EMI	2	2	6	6
5	5 Universal Music Group	1	1	6	6