TALLER DE BASE DE DATOS

(PROCEDURE, FUNCTION Y TRIGGER)

ALUMNO:

JORGE IVAN URUETA RAMOS

MG: LUIS GARCIA CUIDA

UNIVERSIDAD DE CORDOBA
MONTERÍA-CÓRDOBA
BASE DE DATOS

14 DE NOV. DE 1

Ejercicio 1

Crear procedimiento que pasado los parámetros necesarios (JOB_ID,

JOB_TITLE, MIN_SALARY, MAX_SALARY). Inserte un nuevo trabajo en la tabla JOBS Se debe validar que el trabajo no exista en la tabla JOBS.

Solución del punto 1

Se crea un procedure con nombre ejercicio1 y se piden por parámetro los

datos pedidos en el ejercicio Y se declaró un dato llamado valor de tipo integer , que me servirá para comparar de la siguiente forma.

```
CREATE OR REPLACE PROCEDURE   EJERCICIO1(JOB_ID2  JOBS.JOB_ID%TYPE,
JOB_TITLE2 JOBS.JOB_TITLE%TYPE,MIN_SALARY2 JOBS.MIN_SALARY %TYPE,MAX_SALARY2 JOBS.MAX_SALARY%TYPE)
IS
VALOR INTEGER;
```

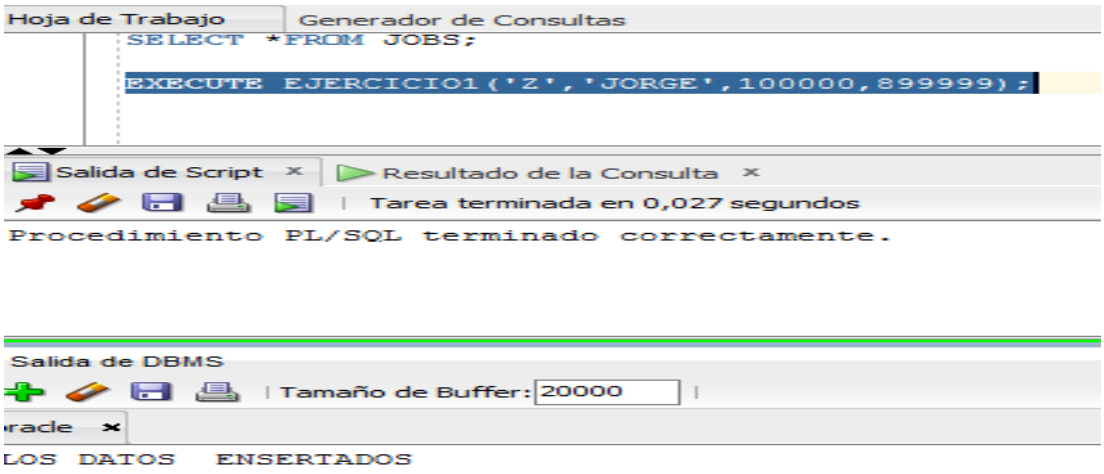
En el bloque de la función se hace una consulta que me contara las veces que se repite el JOB_ID2 que es el dato que no se puede repetir y de almacenara la consulta que me cuenta en un cursor implícito llamado valor luego se valida y si cumple las condiciones se almacenara el empleado y si no mandara por pantalla de comandos un mensaje de error.

```
BEGIN
SELECT COUNT(*) INTO VALOR FROM JOBS WHERE JOB_ID2=JOBS.JOB_ID;
IF(VALOR <1)THEN
INSERT INTO JOBS(JOB_ID,JOB_TITLE,MIN_SALARY,MAX_SALARY)

VALUES(JOB_ID2,JOB_TITLE2,MIN_SALARY2,MAX_SALARY2);
DBMS_OUTPUT.PUT_LINE('LOS DATOS  ENSERTADOS');
ELSIF (VALOR<>0)THEN
DBMS_OUTPUT.PUT_LINE('LOS DATOS  YA  EXISTE O  UNO DE LOS  DATOS INSERTADOS NO ESTA EN  SU DOMINIO ');
END IF;
END  EJERCICIO1;
```

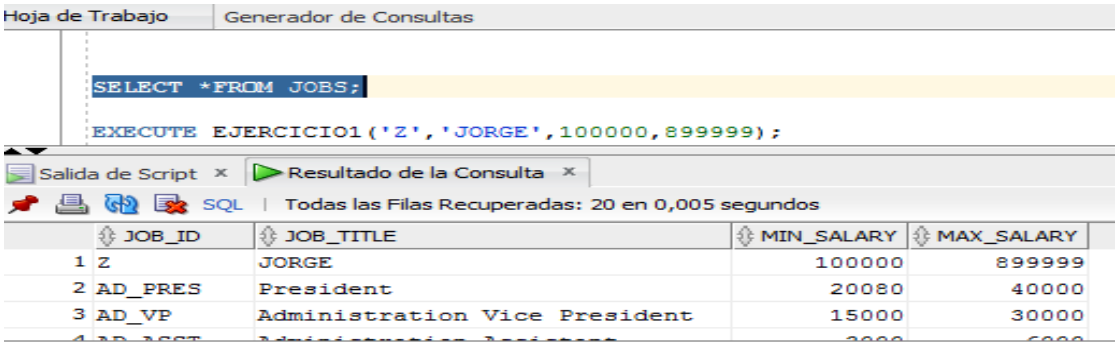
Se hace una consulta para ver los datos que están .

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	AD_PRES	President	20080	40000
2	AD_VP	Administration Vice President	15000	30000
3	AD_ASST	Administration Assistant	3000	6000
4	FI_MGR	Finance Manager	8200	16000
5	FI_ACCOUNT	Accountant	4200	9000
6	AC_MGR	Accounting Manager	8200	16000
7	AC_ACCOUNT	Public Accountant	4200	9000
8	SA_MAN	Sales Manager	10000	20080
9	SA_REP	Sales Representative	6000	12008
10	PU_MAN	Purchasing Manager	8000	15000
11	PU_CLERK	Purchasing Clerk	2500	5500
12	ST_MAN	Stock Manager	5500	8500
13	ST_CLERK	Stock Clerk	2008	5000
14	SH_CLERK	Shipping Clerk	2500	5500
15	IT_PROG	Programmer	4000	10000
16	MK_MAN	Marketing Manager	9000	15000
17	MK_REP	Marketing Representative	4000	9000
18	HR_REP	Human Resources Representative	4000	9000
19	PR_REP	Public Relations Representative	4500	10500

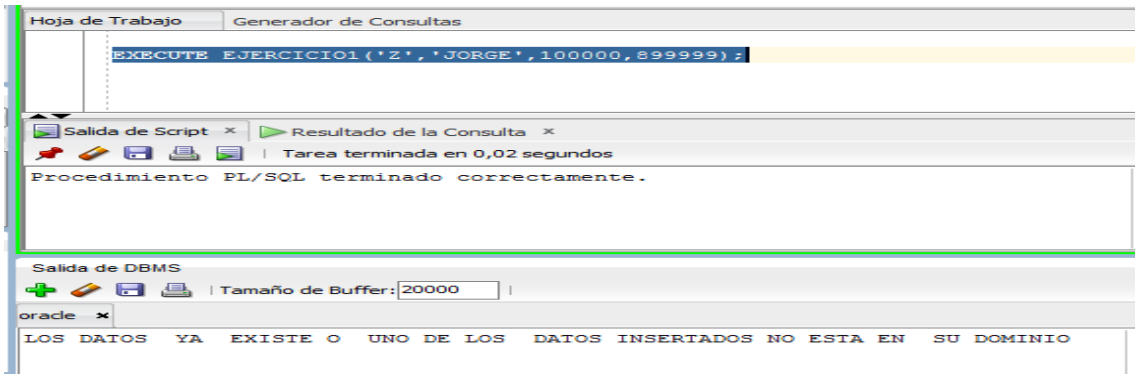
Ejecutamos el procedure con el comando execute y se menten dentro de este los datos a insertar de la siguiente manera y luego me manda un mensaje que dice que el dato fue insertado .



Y hacemos la selección nuevamente y vemos que se inserta correctamente.



Si se intenta insertar nuevamente el mismo dato mandara un mensaje de error .



El ejercicio quedaría de la siguiente forma

Hoja de Trabajo | Generador de Consultas

```
CREATE OR REPLACE PROCEDURE   EJERCICIO1(JOB_ID2  JOBS.JOB_ID%TYPE,
JOB_TITLE2 JOBS.JOB_TITLE%TYPE,MIN_SALARY2 JOBS.MIN_SALARY %TYPE,MAX_SALARY2 JOBS.MAX_SALARY%TYPE)
IS
VALOR INTEGER;
BEGIN
SELECT COUNT(*) INTO VALOR FROM JOBS WHERE JOB_ID2=JOBS.JOB_ID;
IF(VALOR <1)THEN
INSERT INTO JOBS(JOB_ID,JOB_TITLE,MIN_SALARY,MAX_SALARY)

VALUES(JOB_ID2,JOB_TITLE2,MIN_SALARY2,MAX_SALARY2);
DBMS_OUTPUT.PUT_LINE('LOS DATOS  ENSERTADOS');
ELSIF (VALOR<>0)THEN
DBMS_OUTPUT.PUT_LINE('LOS DATOS  YA  EXISTE O  UNO DE LOS  DATOS INSERTADOS NO ESTA EN  SU DOMINIO ');
END IF;
END  EJERCICIO1;


EXECUTE EJERCICIO1('Z','JORGE',100000,899999);

SELECT *FROM JOBS;
```

Ejercicio 2

Crear un procedimiento que pasado el código del empleado, nombre, código del departamento al cual va estar adscrito, código del puesto (JOB_id), (campo employee_id) (tabla EMPLOYEES) y demás campos necesarios para su creación; agregue el trabajador a la tabla EMPLOYYES. Se debe validar que el codigo asociado al trabajador a crear no exista en la tabla porque si no genera error de duplicidad en la llave primaria.

Solución de punto 2

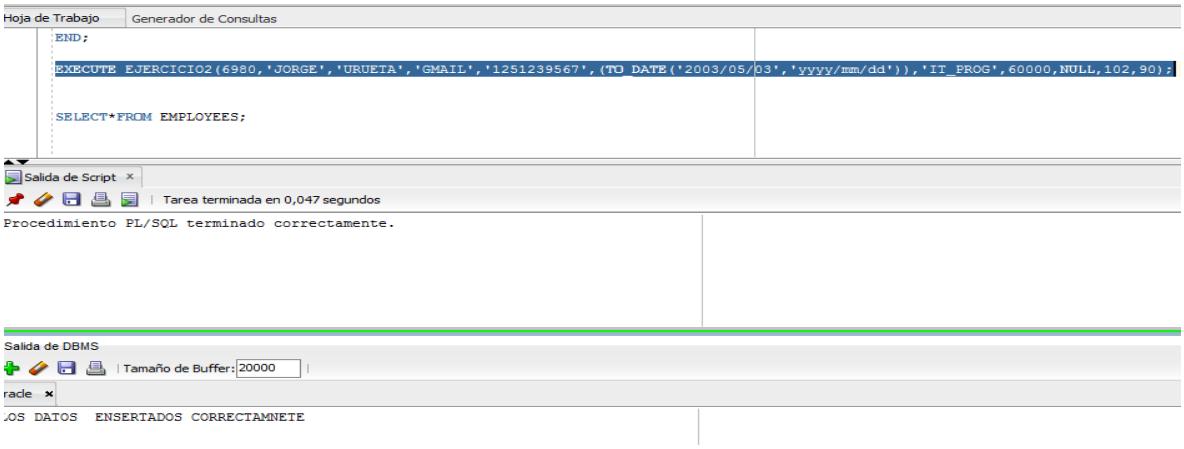
Se crea un procedure donde se piden por parámetros los datos que pide el ejercicio se declaran 4 variable que se utilizaran para condicionar y se puedan meter los datos en la tabla enployees sin irrumpir con la llaves foráneas y las restricciones de la tabla .

```
CREATE OR  REPLACE PROCEDURE EJERCICIO2(EMPOYEE_ID2 EMPLOYEES.EMPLOYEE_ID%TYPE,
FIRST_NAME2 EMPLOYEES.FIRST_NAME%TYPE,
LAST_NAME2 EMPLOYEES.LAST_NAME%TYPE,
EMAIL2 EMPLOYEES.EMAIL%TYPE,
PHONE_NUMBER2 EMPLOYEES.PHONE_NUMBER%TYPE,
HIRE_DATE2 EMPLOYEES.HIRE_DATE%TYPE,
JOB_ID2 EMPLOYEES.JOB_ID%TYPE,
SALARY2 EMPLOYEES.SALARY%TYPE,
COMMISSION_PCT2 EMPLOYEES.COMMISSION_PCT%TYPE,
MANAGER_ID2 EMPLOYEES.MANAGER_ID%TYPE,
DEPARTMENT_ID2 EMPLOYEES.DEPARTMENT_ID%TYPE)
IS
EMPLO_ID INTEGER;
JO_ID INTEGER;
MANA_ID2 INTEGER;
DEPART_ID INTEGER;
```

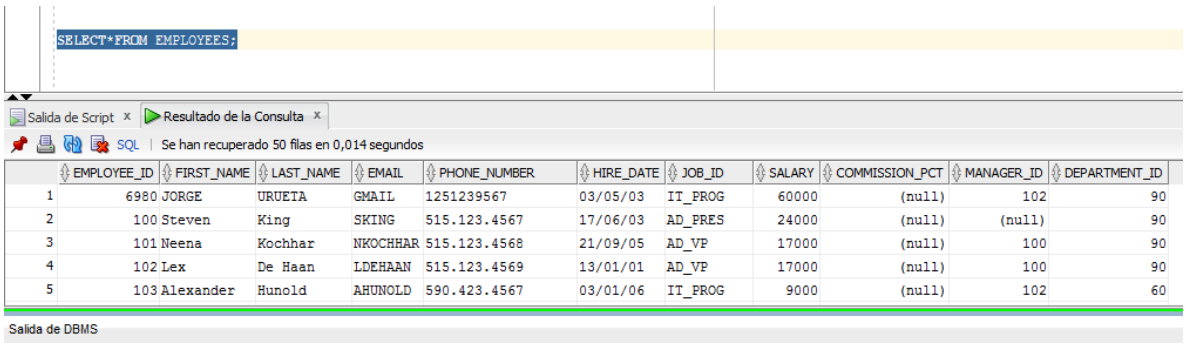
En el bloque se hacen las respectivas selecciones para contar si se repite algún dato que se va insertar cada selección tiene un into que mete el dato de que cantidad de veces se repite el dato . con ese dato en el into ya se puede comparar o condicionar la forma en que se va ingresar el dato .

```
BEGIN
MANA_ID2:=NULL;
select count(*) INTO  EMPLO_ID FROM EMPLOYEES WHERE EMPLOYEES.EMPLOYEE_ID=EMPOYEE_ID2;
SELECT COUNT(*)INTO JO_ID FROM JOBS WHERE JOBS.JOB_ID=JOB_ID2;
SELECT COUNT(*) INTO MANA_ID2  FROM EMPLOYEES WHERE EMPLOYEES.EMPLOYEE_ID=MANAGER_ID2;
SELECT COUNT(*) INTO DEPART_ID  FROM DEPARTMENTS WHERE DEPARTMENTS.DEPARTMENT_ID=DEPARTMENT_ID2;
IF (EMPLO_ID <1 AND JO_ID>0 AND(( MANA_ID2 >0) OR (MANA_ID2 IS NULL ))AND DEPART_ID >0)THEN

INSERT INTO EMPLOYEES(EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMMISSION_PCT,MANAGER_ID,DEPARTMENT_ID)
VALUES(EMPOYEE_ID2,FIRST_NAME2,LAST_NAME2,EMAIL2,PHONE_NUMBER2,HIRE_DATE2,JOB_ID2,SALARY2,COMMISSION_PCT2,MANAGER_ID2,DEPARTMENT_ID2);
DBMS_OUTPUT.PUT_LINE('LOS DATOS  ENSERTADOS CORRECTAMNETE');

ELSE
DBMS_OUTPUT.PUT_LINE('ERROR EL DATO YA EXISTE  O  UNO DE LOS DATOS  NO CORESPONDE A SU DOMINIO');
END IF;
END;
```

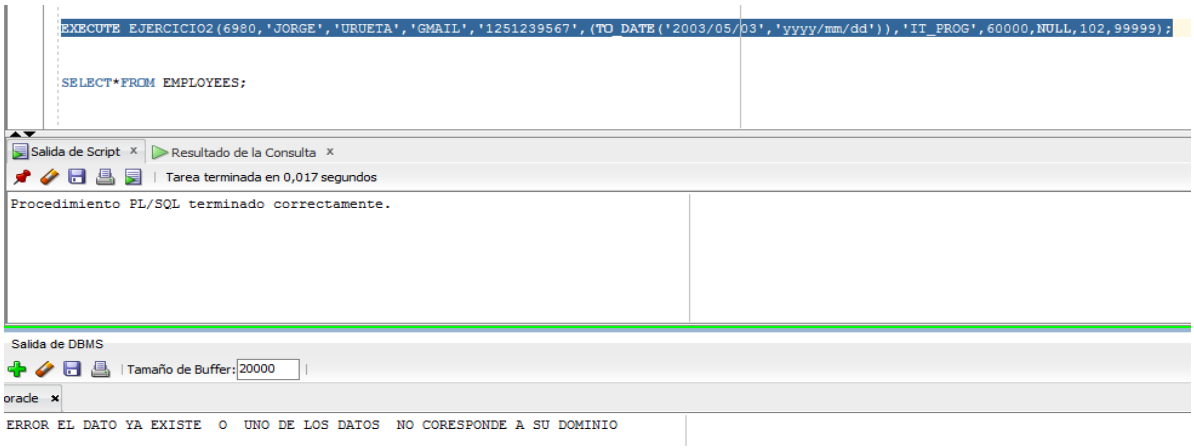
Ejecutamos el procedure con un execute y pasamos los parámetros que se quieren insertar .



Hacemos la consulta y vemos que el dato fue insertado.



Si volvemos a meter el mismo dato o irrumpe alguna llave foránea mandara un mensaje de error en este caso colocaremos un deparment_id que no exista .



El ejercicio completo quedaría de la siguiente forma.

```
CREATE OR  REPLACE PROCEDURE EJERCICIO2(EMPOYEE_ID2 EMPLOYEES.EMPLOYEE_ID%TYPE,
FIRST_NAME2 EMPLOYEES.FIRST_NAME%TYPE,
LAST_NAME2 EMPLOYEES.LAST_NAME%TYPE,
EMAIL2 EMPLOYEES.EMAIL%TYPE,
PHONE_NUMBER2 EMPLOYEES.PHONE_NUMBER%TYPE,
HIRE_DATE2 EMPLOYEES.HIRE_DATE%TYPE,
JOB_ID2 EMPLOYEES.JOB_ID%TYPE,
SALARY2 EMPLOYEES.SALARY%TYPE,
COMMISSION_PCT2 EMPLOYEES.COMMISSION_PCT%TYPE,
MANAGER_ID2 EMPLOYEES.MANAGER_ID%TYPE,
DEPARTMENT_ID2 EMPLOYEES.DEPARTMENT_ID%TYPE)
IS
EMPLO_ID INTEGER;
JO_ID INTEGER;
MANA_ID2 INTEGER;
DEPART_ID INTEGER;
BEGIN
MANA_ID2:=NULL;
select count(*) INTO  EMPLO_ID FROM EMPLOYEES WHERE EMPLOYEES.EMPLOYEE_ID=EMPOYEE_ID2;
SELECT COUNT(*)INTO JO_ID FROM JOBS WHERE JOBS.JOB_ID=JOB_ID2;
SELECT COUNT(*) INTO MANA_ID2  FROM EMPLOYEES WHERE EMPLOYEES.EMPLOYEE_ID=MANAGER_ID2;
SELECT COUNT(*) INTO DEPART_ID  FROM DEPARTMENTS WHERE DEPARTMENTS.DEPARTMENT_ID=DEPARTMENT_ID2;
IF (EMPLO_ID <1 AND JO_ID>0 AND(( MANA_ID2 >0) OR (MANA_ID2 IS NULL ))AND DEPART_ID >0)THEN

INSERT INTO EMPLOYEES(EMPLOYEE_ID,FIRST_NAME,LAST_NAME,EMAIL,PHONE_NUMBER,HIRE_DATE,JOB_ID,SALARY,COMMISSION_PCT,MANAGER_ID,DEPARTMENT_ID)
VALUES(EMPOYEE_ID2,FIRST_NAME2,LAST_NAME2,EMAIL2,PHONE_NUMBER2,HIRE_DATE2,JOB_ID2,SALARY2,COMMISSION_PCT2,MANAGER_ID2,DEPARTMENT_ID2);
DBMS_OUTPUT.PUT_LINE('LOS DATOS  ENSERTADOS CORRECTAMNETE');

ELSE
DBMS_OUTPUT.PUT_LINE('ERROR EL DATO YA EXISTE  O  UNO DE LOS DATOS  NO CORESPONDE A SU DOMINIO');
END IF;
END;
```

Ejercicio 3

Cree una función que liste cuantos trabajadores hay por Departamento, la lista debe estar enumerada y ordenada por nombre del departamento.

Solución de punto 3

Creamos un objeto que nos almacenara la consulta en este caso son el nombre del departamento y el total de empleado por departamento luego de crear el objeto crearemos un objeto derivado de este para manipularlo en el la función seguidamente .

```
CREATE OR REPLACE TYPE EJER3 AS OBJECT(
DEPARTMENT_NAME VARCHAR(30),
TOTAL_EMPLEADOS INTEGER
);
CREATE OR REPLACE TYPE DATOSEJER3 AS TABLE OF EJER3;
```

Luego procedemos con la función llamamos al objeto derivado como una variable global luego en el bloque se hace una consulta la cual se quiere , tipo cast multiset que permite meter los datos de la consulta dentro del tipo objeto y luego retornarlas dentro de la función utilizamos una tabla temporal llamada dual para hacer el select .

```
create or replace FUNCTION  EJERCICIO3
RETURN DATOSEJER3 IS
CANT DATOSEJER3;
BEGIN
select
CAST( MULTISET( SELECT D.DEPARTMENT_NAME,COUNT(E.DEPARTMENT_ID )"CANTIDAD"
  FROM EMPLOYEES E,DEPARTMENTS D
  WHERE  D. DEPARTMENT_ID=E.DEPARTMENT_ID
  GROUP BY  D.DEPARTMENT_NAME ORDER BY  D.DEPARTMENT_NAME ASC
    ) as DATOSEJER3)INTO CANT
    FROM DUAL;
    RETURN  CANT;
END EJERCICIO3;
```

Se procede a ejecutar la función para mostrar la consulta que se requiere los departamentos .

```
select* from table (ejercicio3);
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 11 en 0,0

	DEPARTMENT_NAME	TOTAL_EMPLEADOS
1	Accounting	2
2	Administration	1
3	Executive	4
4	Finance	6
5	Human Resources	1
6	IT	5
7	Marketing	2
8	Public Relations	1
9	Purchasing	6
10	Sales	34
11	Shipping	45

Ejercicio 4

Realice un trigger que cada vez que se inserte un dato en la tabla DEPARMENT.

Registre esos mismos datos, junto con la fecha en que hizo la inserción y la operación, en una tabla nueva llamada aud_department. Esta tabla deben crearla con los mismos campos de la tabla departments, añadiendo otros campos llamados operación y fecha_registro.

Se crea la tabla que con los datos de la tabal departamento más la de la fecha de registro

```
CREATE   TABLE aud_department (
DEPARTMENT_ID  NUMBER(4,0),
DEPARTMENT_NAME VARCHAR2(30),
MANAGER_ID  NUMBER(6,0),
LOCATION_ID NUMBER(4,0),
FECHA_REGISTRO DATE,
CONSTRAINT DEPART PRIMARY  KEY(DEPARTMENT_ID)
);
```

Se realiza el trigger que realizara la acción de copiar los mismo datos que se vallan insertando en la tabla departamento y a estos se le agregar la fecha con el comando sysdate que me trae la fecha actual

```
CREATE OR REPLACE  TRIGGER INSERTAR AFTER INSERT OR  UPDATE ON DEPARTMENTS FOR EACH ROW
BEGIN
Insert into aud_department(DEPARTMENT_ID,DEPARTMENT_NAME,LOCATION_ID,MANAGER_ID,FECHA_REGISTRO)
values (:NEW.DEPARTMENT_ID,:NEW.DEPARTMENT_NAME,:NEW.LOCATION_ID,:NEW.MANAGER_ID,SYSDATE);
END;
```

Se procede a insertar un dato en la tabla departamento

```
INSERT INTO DEPARTMENTS(DEPARTMENT_ID,DEPARTMENT_NAME,MANAGER_ID,LOCATION_ID)VALUES(290,'ADMIN',102,1700);
```

Y se hace un select en la tabla aud_department para ver si el trigger migro los datos de la tabla departamento a esta

```
INSERT INTO DEPARTMENTS(DEPARTMENT_ID,DEPARTMENT_NAME,MANAGER_ID,LOCATION_ID)VALUES(290,'ADMIN',102,1700);

SELECT ROWNUM"NUMERO_TABLA",D.* FROM aud_department D;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0,003 segundos

	NUMERO_TABLA	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID	FECHA_REGISTRO
1	1	290	ADMIN	102	1700	14/11/15

El ejercicio completo quedaría de la siguiente forma

```
CREATE   TABLE aud_department (
DEPARTMENT_ID  NUMBER(4,0),
DEPARTMENT_NAME VARCHAR2(30),
MANAGER_ID  NUMBER(6,0),
LOCATION_ID NUMBER(4,0),
FECHA_REGISTRO DATE,
CONSTRAINT DEPART PRIMARY  KEY(DEPARTMENT_ID)
);


CREATE OR REPLACE  TRIGGER INSERTAR AFTER INSERT OR  UPDATE ON DEPARTMENTS FOR EACH ROW
BEGIN
Insert into aud_department(DEPARTMENT_ID,DEPARTMENT_NAME,LOCATION_ID,MANAGER_ID,FECHA_REGISTRO)
values (:NEW.DEPARTMENT_ID,:NEW.DEPARTMENT_NAME,:NEW.LOCATION_ID,:NEW.MANAGER_ID,SYSDATE);
END;

INSERT INTO DEPARTMENTS(DEPARTMENT_ID,DEPARTMENT_NAME,MANAGER_ID,LOCATION_ID)VALUES(290,'ADMIN',102,1700);

SELECT ROWNUM"NUMERO_TABLA",D.* FROM aud_department D;
```

Punto extra

Mostrar que función realiza el trigger ejecutado, si es un insert un update o un delete .

Solución

Para ello se creó una tabla con los mismos datos de la tabla aud_department con un dato más que va hacer el mensaje con la operación que se hizo en el trigger se lo que se hizo fue solo una validación que para ello se utilizaron palabras reservadas como deleting inserting y updating que me permitieron identificar qué operación se realizan y así migrar los datos de la tabla departamento con la fecha y la operación que se realizó en este caso se hizo un insert .

```
CREATE   TABLE aud_department2 (
DEPARTMENT_ID  NUMBER(4,0),
DEPARTMENT_NAME VARCHAR2(30),
MANAGER_ID  NUMBER(6,0),
LOCATION_ID NUMBER(4,0),
FECHA_REGISTRO DATE,
METODO VARCHAR(20),
CONSTRAINT DEP PRIMARY  KEY(DEPARTMENT_ID)
);
CREATE OR REPLACE  TRIGGER INSERTAR AFTER INSERT OR  UPDATE ON DEPARTMENTS
FOR EACH ROW
BEGIN
IF(INSERTING)THEN
Insert into aud_department2(DEPARTMENT_ID,DEPARTMENT_NAME,LOCATION_ID,MANAGER_ID,FECHA_REGISTRO,METODO)
values (:NEW.DEPARTMENT_ID,:NEW.DEPARTMENT_NAME,:NEW.LOCATION_ID,:NEW.MANAGER_ID,SYSDATE,'DATO  INSERTADO');
ELSIF(UPDATING)THEN
Insert into aud_department2(DEPARTMENT_ID,DEPARTMENT_NAME,LOCATION_ID,MANAGER_ID,FECHA_REGISTRO,METODO)
values (:NEW.DEPARTMENT_ID,:NEW.DEPARTMENT_NAME,:NEW.LOCATION_ID,:NEW.MANAGER_ID,SYSDATE,'DATO ACTUALIZADO');
ELSIF(DELETING)THEN
Insert into aud_department2(DEPARTMENT_ID,DEPARTMENT_NAME,LOCATION_ID,MANAGER_ID,FECHA_REGISTRO,METODO)
values (:NEW.DEPARTMENT_ID,:NEW.DEPARTMENT_NAME,:NEW.LOCATION_ID,:NEW.MANAGER_ID,SYSDATE,'DATO ELIMINADO');
dbms_output.put_line(' ELIMINADO');
END IF;
END;
INSERT INTO DEPARTMENTS(DEPARTMENT_ID,DEPARTMENT_NAME,MANAGER_ID,LOCATION_ID)VALUES(280,'ADMIN',102,1700);
```

Para mostrar que operación realizo se hace una consulta a la tabla que en este caso se llamó aud_department2

```
SELECT ROWNUM"NUMERO_TABLA",D.* FROM aud_department2 D;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0,001 segundos

	NUMERO_TABLA	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID	FECHA_REGISTRO	METODO
1	1	280	ADMIN	102	1700	14/11/15	DATO INSERTADO