

Objetivo:

Crear un esquema genérico de auditoría para auditar los inserts updates y deletes sobre tablas.

1) Crear la siguiente tabla de auditoría:

Nom_tabla: AUDITORIA

Campos:

NRO_AUDIT	ENTERO	PK	(Crear sequence o SERIAL, IDENTITY o AUTOINCREMENT)
NOM_TABLA	VARCHAR(30)	NOT NULL	
OPERACION	CHAR		(VALIDAR valores (I: Inserted, O: Old Updated, N:New Updated,D: Deleted))
ROW_DATA	VARCHAR(255)	NOT NULL	
USUARIO	VARCHAR(30)	DEFAULT	USER (ver según motor)
FECHA	DATE	DEFAULT	CURRENT (ver según motor)

```
CREATE TABLE AUDITORIA(  
  NRO_AUDIT NUMBER PRIMARY KEY,  
  NOM_TABLA VARCHAR(30) NOT NULL,  
  OPERACION CHAR(1) CHECK (OPERACION IN ('I','O','N','D')),  
  ROW_DATA VARCHAR(255) NOT NULL,  
  USUARIO VARCHAR(30) DEFAULT USER,  
  FECHA DATE DEFAULT SYSDATE)  
/  
CREATE SEQUENCE "SQ_PK_AUDITORIA" MINVALUE 1 MAXVALUE 9999 INCREMENT  
BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE  
/
```

- 2) Crear un procedure `INS_DATA` que inserte en la tabla auditoría una fila con los parámetros de entrada enviados.

Parametros de entrada: `NOM_TABLA` , `OPERACION`, `ROW_DATA`

Especificación del cuerpo del procedure

Debe insertar en la tabla auditoría una fila con los datos recibidos.

```
CREATE OR REPLACE PROCEDURE INS_DATA (N_TABLA IN  
AUDITORIA.NOM_TABLA%TYPE, D_OPER IN AUDITORIA.OPERACION%TYPE, D_DATA IN  
AUDITORIA.ROW_DATA%TYPE)  
IS  
BEGIN  
INSERT INTO AUDITORIA (NRO_AUDIT,NOM_TABLA,OPERACION,ROW_DATA) VALUES  
(SQ_PK_AUDITORIA.NEXTVAL,N_TABLA,D_OPER,D_DATA);  
END;
```

Que sucede con el campo `USUARIO` y `FECHA` de la tabla auditoría si no los incluyen en el insert?

- 3) Crear un trigger de insert sobre la tabla MANUFACT con las siguientes características:
FOREACH ROW 'INS_MANUFACT'
Ejecuta el procedure INS_DATA pasándole los parámetros necesarios a partir de los datos INSERTADOS en la tabla.

El valor NOM_TABLA deberá ser "MANUFACT"

El valor OPERACIÓN deberá ser "I"(i mayúscula)

El campo ROW_DATA lo debe componer concatenando los valores INSERTADOS en la tabla origen y separándolo por "|" cada valor.

Ej: "ANZ"|"Adfasdfslfids"|"12"

CREATE OR REPLACE TRIGGER INS_MANUFACT

AFTER

insert on MANUFACT

FOR EACH ROW

DECLARE

D_DATA VARCHAR(255);

begin

D_DATA := :NEW.MANU_CODE || '|' || :NEW.MANU_NAME || '|' || :NEW.LEAD_TIME;

INS_DATA ('MANUFACT', 'I', D_DATA);

end;

INSERT INTO MANUFACT VALUES('ZZZ','ZAFFA STORE',23)

- 4) Crear un trigger de DELETE sobre la tabla MANUFACT con las siguientes características:
FOREACH ROW
Ejecuta el procedure INS_DATA pasándole los parámetros necesarios a partir de los datos de la fila/s BORRADAS en la tabla.

El valor NOM_TABLA deberá ser "MANUFACT"

El valor OPERACIÓN deberá ser "D"

El campo ROW_DATA lo debe componer concatenando los valores de la fila BORRADA de la tabla origen y separándolo por "|" cada valor.

Ej: "ANZ"|"Adfasdfslfsls"|"12"

CREATE OR REPLACE TRIGGER DEL_MANUFACT

AFTER

DELETE on MANUFACT

FOR EACH ROW

DECLARE

D_DATA VARCHAR(255);

begin

D_DATA := :OLD.MANU_CODE||' '||:OLD.MANU_NAME||' '||:OLD.LEAD_TIME;

INS_DATA ('MANUFACT', 'D',D_DATA);

end;

DELETE FROM MANUFACT WHERE MANU_CODE='ZZZ'

SELECT * FROM AUDITORIA ORDER BY NRO_AUDIT

- 5) Crear un trigger de UPDATE sobre la tabla MANUFACT con las siguientes características:
FOREACH ROW
Ejecutar dos veces el procedure INS_DATA pasándole los parámetros necesarios a partir de los datos INSERTADOS en la tabla.

El valor NOM_TABLA deberá ser "MANUFACT"

El valor OPERACIÓN deberá ser "O" (Old Fila vieja antes de la modificación)

El campo ROW_DATA lo debe componer concatenando los valores de la fila ORIGINAL anterior a la MODIFICACION de la tabla origen y separándolo por "|" cada valor.

Ej: "ANZ"|"Adfasdfslfsls"|"12"

El valor NOM_TABLA deberá ser "MANUFACT"

El valor OPERACIÓN deberá ser "N" (NEW Fila nueva producto de la modificación)

El campo ROW_DATA lo debe componer concatenando los valores de la fila NUEVA posterior a la MODIFICACION de la tabla origen y separándolo por "|" cada valor.

Ej: "ANZ"|"Adfasdfslfsls"|"12"

CREATE OR REPLACE TRIGGER "UPD_MANUFACT"

AFTER

UPDATE on MANUFACT

FOR EACH ROW

DECLARE

D_DATA VARCHAR(255);

begin

D_DATA := :OLD.MANU_CODE||'|'||:OLD.MANU_NAME||'|'||:OLD.LEAD_TIME;

INS_DATA ('MANUFACT', 'O',D_DATA);

D_DATA := :NEW.MANU_CODE||'|'||:NEW.MANU_NAME||'|'||:NEW.LEAD_TIME;

INS_DATA ('MANUFACT', 'N',D_DATA);

end;

6) Realizar las siguientes pruebas

- a. INSERTAR el siguiente registro "XXX","Xtra large","23"
- b. UPDATE el reg donde MANU_CODE="XXX" cambiando la descripción por "Extra Large"
Que valores contiene la tabla AUDITORIA?
- c. INSERTAR NUEVAMENTE el siguiente registro "XXX","Xtra large","23"
Que sucedió?
- d. INSERTAR el siguiente registro "ZZZ","Zampini SA","11"
- e. BORRAR el registro cuyo MANU_CODE="ZZZ"
Que nuevos registros contiene la tabla AUDITORIA?

INSERT INTO MANUFACT VALUES ('XXX', 'Xtra large',23)

UPDATE MANUFACT SET MANU_NAME='Extra Large' where MANU_CODE='XXX'

INSERT INTO MANUFACT VALUES ('XXX', 'Xtra large',23)

Da un error por clave primaria duplicada

INSERT INTO MANUFACT VALUES ('ZZZ', 'Zampini SA',11)

DELETE FROM MANUFACT WHERE MANU_CODE='ZZZ'

7) Crear la siguiente tabla de log de errores:

Nom_tabla: AUDIT_ERROR

Campos:

NRO_ERROR	ENTERO	PK	(Crear sequence o SERIAL, IDENTITY o AUTOINCREMENT)
SQL_ERR	INTEGER		
ISAM_ERR	INTEGER		
ERROR_INFO	CHAR(70)		
NOM_TABLA	VARCHAR(30)	NOT NULL	
OPERACION	CHAR		
ROW_DATA	VARCHAR(255)	NOT NULL	
USUARIO	VARCHAR(30)	DEFAULT USER	
FECHA	DATETIME YEAR TO FRACTION	DEFAULT CURRENT	
ERRSTATUS	CHAR	(Validar P:Pendiente, F: Finalizado)	DEFAULT "P"

CREATE TABLE AUDIT_ERROR

(NRO_ERROR NUMBER PRIMARY KEY,

SQL_ERR NUMBER,

ISAM_ERR NUMBER,

ERROR_INFO CHAR(70),

NOM_TABLA VARCHAR(30) NOT NULL,

OPERACION CHAR(1),

ROW_DATA VARCHAR(255) NOT NULL,

USUARIO VARCHAR(30) DEFAULT USER,

FECHA DATE DEFAULT SYSDATE,

ERRSTATUS CHAR(1) DEFAULT 'P' CHECK (ERRSTATUS IN ('P','F')))

**CREATE SEQUENCE "SQ_PK_AUDIT_ERROR" MINVALUE 1 MAXVALUE 9999 INCREMENT
BY 1 START WITH 1 NOCACHE NOORDER NOCYCLE**

/

PRUEBA DE INSERT

INSERT INTO AUDIT_ERROR

(NRO_ERROR,SQL_ERR,ISAM_ERR,ERROR_INFO,NOM_TABLA,OPERACION,ROW_DATA)

VALUES (sq_pk_audit_error.nextval,-206,1200,'sfdsfdsfdfsdfa','MANUFACT',

'I','DFAFSDSDFDSF')

- 8) Modificar el procedure INS_DATA agregando un bloque de manejo de excepciones que ante cualquier error que ocurra en la operación de auditoría, grabe una fila en la tabla ERROR_AUDIT ingresando los datos asociados al error y los datos de la fila fallida en la inserción de la tabla AUDITORÍA.

El campo ERRSTATUS no se incluye en el INSERT y se actualizará por DEFAULT en "P".

```
create or replace PROCEDURE INS_DATA (N_TABLA IN AUDITORIA.NOM_TABLA%TYPE,  
D_OPER IN AUDITORIA.OPERACION%TYPE, D_DATA IN AUDITORIA.ROW_DATA%TYPE)  
IS  
err_num NUMERIC;  
err_msg CHAR(70);  
BEGIN  
    INSERT INTO AUDITORIA (NRO_AUDIT,NOM_TABLA,OPERACION,ROW_DATA) VALUES  
(SQ_PK_AUDITORIA.NEXTVAL,N_TABLA,D_OPER,D_DATA);  
EXCEPTION  
WHEN OTHERS THEN  
    err_num := SQLCODE;  
    err_msg := SUBSTR(SQLERRM,1,70);  
    INSERT INTO audit_error  
(NRO_ERROR,SQL_ERR,ISAM_ERR,ERROR_INFO,NOM_TABLA,OPERACION,ROW_DATA)  
VALUES  
(sq_pk_audit_error.NEXTVAL,err_num,err_num,err_msg,N_TABLA,D_OPER,D_DATA);  
END;
```


9) Realizar las siguientes pruebas

- a. Borre y recree la secuencia sq_pk_auditoria comenzando con el anteúltimo número existente en la tabla auditoría campo nro_audit

DROP SEQUENCE SQ_PK_AUDITORIA

CREATE SEQUENCE "SQ_PK_AUDITORIA" MINVALUE 1 MAXVALUE 9999 INCREMENT BY 1 START WITH 45 NOCACHE NOORDER NOCYCLE

- b. Realizar los siguientes INSERTs del siguiente
'WWW','Wahington WWW',12
'TTT','Tamina SA',12

Falló la transacción original?

Se grabó la información en la tabla auditoría?

Que sucedió?

- c. Ejecutar un DELETE completo de la tabla MANUFACT sin cláusula WHERE.
Cuál fue el resultado del mismo?
Que información se grabó en la tabla AUDITORÍA?
Se grabaron errores en la tabla ERROR_AUDIT? Por qué?

DELETE FROM MANUFACT

10) Realizar un procedimiento REPROCESA_DATA

Objetivo: Actualizar la tabla AUDITORIA con todas las filas de auditoría que no pudieron ser insertadas, que se grabaron en la tabla ERROR_AUDIT y que su estado sea Pendiente (ERRSTATUS=P).

Por cada fila de la tabla ERROR_AUDIT cuyo campo ERRSTATUS sea igual a P se deberán realizar las siguientes operaciones:

INSERTAR en la tabla AUDITORIA los datos de la fila original.

MODIFICAR el campo ERRSTATUS de la fila procesada con el valor "F"

Tener en cuenta:

- Que no se pueda modificar en forma concurrente por otro usuario o proceso la fila que se está tratando en este PROCEDURE.
- El proceso completo de reproceso debe tomarse como un bloque, o se procesan todas las filas o no se procesa ninguna.

```

CREATE OR REPLACE PROCEDURE REPROCESA_DATA
IS
CURSOR Cur1 IS SELECT
nro_error,nom_tabla,operacion,row_data,user,fecha FROM
audit_error WHERE errstatus='P' FOR UPDATE;
d_audit auditoria%ROWTYPE;

BEGIN
OPEN Cur1;
LOOP
    FETCH Cur1 INTO d_audit;
    IF Cur1%FOUND THEN
        INSERT INTO auditoria VALUES
(sq_pk_auditoria.nextval,d_audit.nom_tabla,d_audit.operacion,d_a
udit.row_data,d_audit.usuario,d_audit.fecha);
        UPDATE audit_error SET errstatus = 'F' WHERE CURRENT OF
Cur1;
    ELSE
        EXIT;
    END IF;
END LOOP;
CLOSE Cur1;
COMMIT;

EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
END REPROCESA_DATA;

```

11) Realizar las siguientes pruebas

- a. Consultar de la tabla ERROR_AUDIT todas las filas cuyo estado sea Pendiente de reprocesar.
Cuántas filas son?

```
SELECT * FROM audit_error WHERE errstatus='P'
```

- b. Ejecutar el procedimiento REPROCESA_DATA
Evaluar las tablas AUDITORIA y ERROR_AUDIT e indique el resultado obtenido.

```

begin
reprocesa_data;
END;

```

- 12) Crear los triggers de INSERT para las tablas ORDERS e ITEMS, tomando como referencia los triggers de la tabla MANUFACT.

CREATE OR REPLACE TRIGGER INS_ORDERS

AFTER

insert on ORDERS

FOR EACH ROW

DECLARE

D_DATA VARCHAR(255);

begin

D_DATA :=

:NEW.ORDER_NUM||' '||:NEW.ORDER_DATE||' '||:NEW.CUSTOMER_NUM||' '||:NEW.SHIP_INSTRUCT||' '||:NEW.BACKLOG||' '||:NEW.PO_NUM||' '||:NEW.SHIP_DATE||' '||:NEW.SHIP_WEIGHT||' '||:NEW.SHIP_CHARGE||' '||:NEW.PAID_DATE;

INS_DATA ('ORDERS', 'I',D_DATA);

end;

CREATE OR REPLACE TRIGGER INS_ITEMS

AFTER

insert on ITEMS

FOR EACH ROW

DECLARE

D_DATA VARCHAR(255);

begin

D_DATA :=

:NEW.ITEM_NUM||' '||:NEW.ORDER_NUM||' '||:NEW.STOCK_NUM||' '||:NEW.MANU_CODE||' '||:NEW.QUANTITY ||' '||:NEW.TOTAL_PRICE;

INS_DATA ('ITEMS', 'I',D_DATA);

end;

13) Realice una prueba de los mismos con la siguiente info.

- a. Inserte en la tabla ORDERS una nueva orden basándose en el SELECT de la tabla ORDERS para los order_num IN (1001,1002)., tener en cuenta que el campo order_num debe respetar una secuencia.
 - i. Crear una sequence sq_pk_orders que comience con el nro 2001
 - ii. La primer Orden deberá ser igual a la 1001 y la siguiente a la 1002

```
CREATE SEQUENCE "SQ_PK_ORDERS" MINVALUE 1 MAXVALUE 999999 INCREMENT BY  
1 START WITH 2001 NOCACHE NOORDER NOCYCLE
```

INSERT INTO ORDERS

```
SELECT SQ_PK_ORDERS.NEXTVAL,order_date,customer_num,ship_instruct,backlog,  
po_num, ship_date,ship_weight,ship_charge,paid_date  
FROM orders  
WHERE order_num in (1001,1002)
```

- b. Verifique los números de orden generados

```
select * from orders where order_num >2000
```

```
ORDER_NUM ORDER_DATE CUSTOMER_NUM SHIP_INSTRUCT BACKLOG PO_NUM SHIP_DATE SHIP_WEIGHT SHIP_CHARGE PAID_DATE  
2001 20/05/98 104 express n B77836 01/06/98 20,4 10 22/07/98  
2002 21/05/98 101 PO on box; deliver to back door only n 9270 26/05/98 50,6 15,3 03/06/98
```

- c. Inserte en la tabla ITEMS los ítems para la 1ra orden generada, basándose en los ítems de la Orden 1001.

INSERT INTO ITEMS

```
SELECT ITEM_NUM,2001,STOCK_NUM,MANU_CODE,QUANTITY,TOTAL_PRICE  
FROM ITEMS  
WHERE order_num =1001
```

- d. Inserte en la tabla ITEMS los ítems para la 2da orden generada, basándose en los ítems de la Orden 1002.

INSERT INTO ITEMS

```
SELECT ITEM_NUM,2002,STOCK_NUM,MANU_CODE,QUANTITY,TOTAL_PRICE  
FROM ITEMS  
WHERE order_num =1002
```

Que nuevas filas observa en la tabla AUDITORIA.

SELECT * FROM AUDITORIA ORDER BY nro_audit DESC

NRO_AUDIT	NOM_TABLA	OPERACION	ROW_DATA	USUARIO	FECHA
57	ITEMS I	2	2002 3 HSK 1 240	ANONYMOUS	26/06/12
56	ITEMS I	1	2002 4 HSK 1 960	ANONYMOUS	26/06/12
55	ITEMS I	1	2001 1 HRO 1 250	ANONYMOUS	26/06/12
53	ORDERS I	2002 21/05/98 101	PO on box; deliver to back door only n 9270 26/05/98 50,6 15,3 03/06/98	ANONYMOUS	26/06/12
52	ORDERS I	2001 20/05/98 104	express n B77836 01/06/98 20,4 10 22/07/98	ANONYMOUS	26/06/12