

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 SERIAL 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 DATETIME YEAR TO SECOND DEFAULT CURRENT);
```

- 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 PROCEDURE ins_data (d_tabla LIKE auditoria.nom_tabla, d_oper LIKE  
auditoria.operacion, d_data LIKE auditoria.row_data)  
INSERT INTO auditoria (nro_audit, nom_tabla,operacion,row_data)  
VALUES (0,d_tabla,d_oper,d_data);  
END PROCEDURE;
```

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"|"Adfasdfslflds"|"12"

CREATE TRIGGER INS_MANUFACT insert on MANUFACT

REFERENCING NEW AS NUE

**FOR EACH ROW (EXECUTE PROCEDURE INS_DATA ('MANUFACT','I',
'NUE.MANU_CODE'||'|'||NUE.MANU_NAME'||'|'||NUE.LEAD_TIME')
);**

INSERT INTO MANUFACT VALUES('ZZZ','FABRICANTE ZZZ', '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"|"Adfasdfslfids"|"12"

```
CREATE TRIGGER del_manufact DELETE ON manufact  
REFERENCING OLD AS VIE  
FOR EACH ROW (EXECUTE PROCEDURE INS_DATA ('MANUFACT','D',  
'VIE.MANU_CODE||'|'|VIE.MANU_NAME||'|'|VIE.LEAD_TIME')  
);
```

```
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"|"Adfasdfslfids"|"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"|"Adfasdfslfids"|"12"

```

CREATE TRIGGER upd_manufact UPDATE ON manufact
REFERENCING NEW as NUE OLD AS VIE
FOR EACH ROW (EXECUTE PROCEDURE INS_DATA ('MANUFACT','O',
'VIE.MANU_CODE||'|'||VIE.MANU_NAME||'|'||VIE.LEAD_TIME'),
EXECUTE PROCEDURE INS_DATA ('MANUFACT','N',
'NUE.MANU_CODE||'|'||NUE.MANU_NAME||'|'||NUE.LEAD_TIME')
);

```

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 SERIAL PRIMARY KEY,
sql_err INTEGER,
isam_err INTEGER,
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 DATETIME YEAR TO SECOND DEFAULT CURRENT,
errstatus CHAR(1) DEFAULT 'P' CHECK (errstatus IN ('P','F')) )

```

PRUEBA DE INSERT

```

INSERT INTO audit_error
(nro_error, sql_err, isam_err, error_info, nom_tabla, operaci3n, row_data)
VALUES (0,-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 operaci3n 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 inserci3n de la tabla AUDITORÍA.
El campo ERRSTATUS no se incluye en el INSERT y se actualizará por DEFAULT en "P".

```

CREATE PROCEDURE INS_DATA (d_tabla LIKE auditoria.nom_tabla, d_oper LIKE
auditoria.operacion, d_data LIKE auditoria.row_data)

```

```

DEFINE sql_err int;
DEFINE isam_err int;
DEFINE error_info char(70);
ON EXCEPTION SET sql_err, isam_err, error_info

```

```

INSERT INTO audit_error

```

```
(nro_error, sql_err, isam_err,error_info,nom_tabla, operacion,row_data)
VALUES (0,sql_err,isam_err,error_info,d_tabla,d_oper,d_data);
```

```
END EXCEPTION;
```

```
INSERT INTO auditoria (nro_audit, nom_tabla,operacion,row_data)
VALUES (0,d_tabla,d_oper,d_data);
```

```
END PROCEDURE;
```

9) Realizar las siguientes pruebas

- a. Abra una nueva sesión y desde ella lockee la tabla auditoria en modo exclusivo dentro de una transacción.

```
BEGIN WORK;
```

```
LOCK TABLE auditoria IN EXCLUSIVE MODE;
```

- 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 PROCEDURE reprocesa_data ()
  DEFINE n_error LIKE audit_error.nro_error;
  DEFINE d_tabla LIKE audit_error.nom_tabla;
  DEFINE d_oper LIKE audit_error.operacion;
  DEFINE d_data LIKE audit_error.row_data;
  DEFINE d_user LIKE audit_error.usuario;
  DEFINE f_fecha LIKE audit_error.fecha;
  DEFINE sql_err int;
  DEFINE isam_err int;
  DEFINE error_info char(70);

  ON EXCEPTION SET sql_err, isam_err, error_info
    ROLLBACK WORK;
  END EXCEPTION

  BEGIN WORK;

  FOREACH upd_cursor FOR
    SELECT nro_error,nom_tabla,operacion,row_data,usuario,fecha
    INTO n_error,d_tabla,d_oper,d_data,d_user,f_fecha
    FROM audit_error
    WHERE errstatus='P'

    INSERT INTO auditoria
      (nro_audit,nom_tabla,operacion,row_data,usuario,fecha)
      VALUES (0,d_tabla,d_oper,d_data,d_user,f_fecha);

    UPDATE audit_error SET errstatus = 'F' WHERE CURRENT OF upd_cursor;

  END FOREACH;

  COMMIT WORK;

END PROCEDURE;
```

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;
```

Opcional.

- 12) Crear los triggers de INSERT,DELETE y UPDATE para las tabla **STATE**, tomando como referencia los triggers de la tabla MANUFACT.

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

- a. Inserte en la tabla STATE una fila de prueba.
- b. Modifique algún atributo de la fila ingresada
- c. Borre la fila ingresada originalmente.

Que nuevas filas observa en la tabla AUDITORIA.

```
SELECT * FROM AUDITORIA ORDER BY nro_audit DESC
```