EJERCICIOS DE TRIGGERS EN PUBS

1. **Añade la columna EJEMPLARES\_VENDIDIOS a la tabla PUBLISHER. Rellénala con el resultado de sumar las unidades vendidas de todos sus libros (ytd\_sale).**

**ALTER TABLE PUBLISHER ADD EJEMPLARES\_VENDIDOS NUMBER(7);**

**UPDATE PUBLISHER**

**SET EJEMPLARES\_VENDIDOS = (SELECT SUM(YTD\_SALE)**

**FROM TITLE**

**WHERE PUB\_ID=PUBLISHER.PUB\_ID)**

**UPDATE PUBLISHER**

**SET EJEMPLARES\_VENDIDOS = 0 WHERE EJEMPLARES\_VENDIDOS IS NULL**

1. **Escribe un disparador que mantenga actualizada la columna del ejercicio anterior.**

**CREATE OR REPLACE TRIGGER T1 AFTER INSERT OR DELETE OR UPDATE ON TITLE**

**FOR EACH ROW**

**BEGIN**

**IF INSERTING THEN**

**UPDATE PUBLISHER**

**SET EJEMPLARES\_VENDIDOS = EJEMPLARES\_VENDIDOS + :NEW.YTD\_SALE WHERE :NEW.PUB\_ID=PUBLISHER.PUB\_ID;**

**DBMS\_OUTPUT.PUT\_LINE('TABLA PUBLISHER ACTUALIZADA.');**

**ELSE**

**IF UPDATING THEN**

**UPDATE PUBLISHER**

**SET EJEMPLARES\_VENDIDOS = EJEMPLARES\_VENDIDOS + :NEW.YTD\_SALE - :OLD.YTD\_SALE WHERE :NEW.PUB\_ID=PUBLISHER.PUB\_ID;**

**DBMS\_OUTPUT.PUT\_LINE('TABLA PUBLISHER ACTUALIZADA.');**

**ELSE**

**UPDATE PUBLISHER**

**SET EJEMPLARES\_VENDIDOS = EJEMPLARES\_VENDIDOS - :OLD.YTD\_SALE WHERE :OLD.PUB\_ID=PUBLISHER.PUB\_ID;**

**DBMS\_OUTPUT.PUT\_LINE('TABLA PUBLISHER ACTUALIZADA.');**

**END IF;**

**END IF;**

**END;**

**INSERT INTO TITLE VALUES ('PC4000', 'Prueba', 'popular\_comp', 1622, 20, 8000, 10, 10000, 'Prueba 2', SYSDATE);**

1. **Añade a la tabla TITLE la columna LAST\_UPDATE. Rellénala con la fecha de hoy para todos los títulos.**

**ALTER TABLE TITLE ADD LAST\_UPDATE DATE;**

**UPDATE TITLE**

**SET LAST\_UPDATE = SYSDATE;**

1. **Escribe un disparador que modifique el valor de LAST\_UPDATE cada vez que se produzca una modificación de la información de libros.**

**CREATE OR REPLACE TRIGGER T2 BEFORE UPDATE OR INSERT ON TITLE**

**FOR EACH ROW**

**BEGIN**

**:NEW.LAST\_UPDATE := SYSDATE;**

**END;**

**UPDATE TITLE**

**SET TITLE = 'HOLA' WHERE TITLE\_ID='PC4000';**

1. **La tabla ROYSCHED establece el porcentaje que deben ganar los autores en función de las ventas de títulos. Basándote en la información de esta tabla, se debe mantener actualizado el valor de la columna ROYALTY que corresponde a los autores.**

**CREATE OR REPLACE TRIGGER T3 BEFORE INSERT OR UPDATE ON TITLE**

**FOR EACH ROW**

**DECLARE**

**CONTADOR NUMBER(1);**

**BEGIN**

**SELECT COUNT(\*) INTO CONTADOR**

**FROM ROYSCHED**

**WHERE TITLE\_ID=:NEW.TITLE\_ID**

**AND :NEW.YTD\_SALE BETWEEN LORANGE AND HIRANGE;**

**IF CONTADOR=1 THEN**

**SELECT ROYALTY INTO :NEW.ROYALTY**

**FROM ROYSCHED**

**WHERE TITLE\_ID=:NEW.TITLE\_ID**

**AND :NEW.YTD\_SALE BETWEEN LORANGE AND HIRANGE;**

**END IF;**

**END;**

**UPDATE TITLE**

**SET YTD\_SALE=12000 WHERE TITLE\_ID='PS3333'**

1. **En la tabla EMPLOYEEPUB el valor de la columna JOB\_LVL debe estar comprendida siempre entre los valores min\_lvl y max\_lvl establecidos para el job\_id correspondiente. Emite un error en caso de que se pretenda establecer un valor incoherente.**

**create or replace TRIGGER T4 BEFORE UPDATE ON EMPLOYEEPUB**

**FOR EACH ROW**

**DECLARE**

**CONTADOR NUMBER(1);**

**BEGIN**

**SELECT COUNT(\*) INTO CONTADOR**

**FROM JOB**

**WHERE JOB\_ID=:NEW.JOB\_ID**

**AND :NEW.JOB\_LVL BETWEEN MIN\_LVL AND MAX\_LVL;**

**IF CONTADOR=0 THEN**

**RAISE\_APPLICATION\_ERROR(-20005, 'NUEVO VALOR DE JOB\_LVL NO PERMITIDO.');**

**END IF;**

**END;**

**UPDATE EMPLOYEEPUB**

**SET JOB\_LVL = 300 WHERE JOB\_ID=2**

1. **En la tabla TITLE, el título de un libro se puede repetir siempre y cuando se publique en una editorial distinta. Escribe un disparador que emita un error cuando un título de libro se pretenda duplicar de forma errónea.**

**CREATE OR REPLACE TRIGGER T5 BEFORE INSERT ON TITLE**

**FOR EACH ROW**

**DECLARE**

**CONTADOR NUMBER(1);**

**BEGIN**

**SELECT COUNT(\*) INTO CONTADOR**

**FROM TITLE**

**WHERE TITLE\_ID=:NEW.TITLE\_ID**

**AND PUB\_ID=:NEW.PUB\_ID;**

**IF CONTADOR=0 THEN**

**RAISE\_APPLICATION\_ERROR(-20006, 'TITULO DE LA MISMA EDITORIAL YA EXISTE.');**

**END IF;**

**END;**

1. **El valor de la columna ADVANCE de la tabla TITLE no se puede modificar (sin emitir error).**

**CREATE OR REPLACE TRIGGER T6 BEFORE UPDATE ON TITLE**

**FOR EACH ROW**

**WHEN (NEW.ADVANCE!=OLD.ADVANCE)**

**BEGIN**

**:NEW.ADVANCE:=:OLD.ADVANCE;**

**END;**

**UPDATE TITLE**

**SET ADVANCE= 10000 WHERE TITLE\_ID='PC1035';**

1. **Un titulo no puede cambiar de editorial (sin emitir error).**

**CREATE OR REPLACE TRIGGER T7 BEFORE UPDATE ON TITLE**

**FOR EACH ROW**

**WHEN (NEW.PUB\_ID!=OLD.PUB\_ID)**

**BEGIN**

**:NEW.PUB\_ID:=:OLD.PUB\_ID;**

**END;**

**UPDATE TITLE**

**SET PUB\_ID=0736 WHERE TITLE\_ID='PC1035';**