

1. Cree la tabla TOUR copiando los datos de la tabla DOCENCIA.TOUR2017. Utilice:

```
CREATE TABLE TOUR AS SELECT * FROM DOCENCIA.TOUR2017
```

2. Divida la tabla en dos: una tabla llamada INFO_PERSONAL con los atributos personales del ciclista (ID, NOMBRE, NACIONALIDAD) y otra tabla llamada INFO_PROFESIONAL con los atributos de su equipo (ID, EQUIPO, PAIS, CATEGORIA). Use el atributo ID como enlace en ambas tablas, definiéndolo como clave primaria en las dos.
3. Cree una vista TOUR_VIEW que permita a las aplicaciones poder seguir viendo un solo objeto, como estaba la tabla original. Es decir, cree una vista para que la consulta `SELECT * FROM TOUR_VIEW` produzca el mismo resultado que producía antes `SELECT * FROM TOUR`. De este modo se garantiza la independencia de los datos.
4. Sin embargo, la independencia se debería poder asegurar también en inserción. Cree un disparador llamado TR_INSERTA_CICLISTA que recibe la petición de insertar una tupla nueva en TOUR_VIEW y produce dos inserciones, una por cada tabla actual. Observe que el disparador ha de ser de INSTEAD OF.
5. Cree una tabla llamada LOG_INSERTION que almacena las acciones de INSERT sobre la tabla TOUR. La idea es que las aplicaciones que aún hacían inserciones a través de la tabla antigua sean detectadas. La tabla tendrá dos campos, USUARIO y FECHA, guardando así el usuario ORACLE que pide la inserción y el instante en que se hace. Es decir, cuando una aplicación pida hacer un INSERT en TOUR, se almacena en LOG_INSERTION desde qué usuario ORACLE se hace la petición y su fecha completa, para que el administrador detecte estas aplicaciones y pueda avisar a los programadores de que cambien la tabla por la nueva vista.
6. Volvamos a la tabla TOUR. Cree una vista llamada TOUR_SPAIN con los ciclistas españoles Y una vista llamada TOUR_ITALY con los ciclistas italianos (Nacionalidad).
7. Pruebe a hacer una inserción de un nuevo ciclista español a través de la vista TOUR_SPAIN.
8. ¿Y si nuevo ciclista no es español, puede insertar a través de ella? Pruébalo ¿Puede verlo?

9. ¿Cómo la modificaría para que no se pueda insertar información que no puede luego verse? Revise en el manual la cláusula CHECK OPTION en la creación de vistas.
10. ¿Esta modificación es para cualquier operación de actualización o puede decirse que funcione en INSERT y no en UPDATE? ¿Hay alguna manera de hacer esta especie de filtro según la operación? No lo haga, pero indique cómo se haría.
11. Modifique la vista TOUR_SPAIN para que sea de sólo lectura y pruebe a insertar una fila a través de ella.
12. Cree una vista llamada V_TOUR_FRANCE con los ciclistas franceses. Ceda permiso de borrado sobre la vista V_TOUR_FRANCE a otro esquema (pídale el usuario a un compañero).
13. Borre un piloto francés desde ese esquema. Borre un piloto no francés.
14. Cree una vista CICLISTAS_POR_EQUIPOS donde para cada equipo dice el nombre del equipo (NOMBRE) y su número de ciclistas (NUM_CICLISTAS). ¿Puede insertar sobre ella? ¿Porqué?
15. Cree una tabla llamada TB_CICLISTAS_POR_EQUIPOS con la misma información que la vista anterior. Cree un disparador llamado TR_CICLISTAS_POR_EQUIPOS que cada vez que se inserta un nuevo ciclista en la tabla TOUR, suma en la tabla TB_CICLISTAS_POR_EQUIPOS una unidad al atributo NUM_CICLISTAS de la tupla correspondiente al equipo.
16. ¿Puede extender el disparador para que quite una unidad de NUM_CICLISTAS si se ha hecho un DELETE de TOUR?
17. ¿Y que hay que hacer para que el disparador cubra el caso de hacer un UPDATE sobre TOUR cambiando a un ciclista de equipo?

/*EJERCICIO 1*/

```
CREATE TABLE TOUR AS SELECT * FROM DOCENCIA.TOUR2017
```

/*EJERCICIO 2*/

```
create table INFO_PERSONAL(  
  ID NUMBER (3,0) PRIMARY KEY,  
  NOMBRE VARCHAR2 (50),  
  NACIONALIDAD VARCHAR2(3)  
);
```

```
CREATE TABLE INFO_PROFESIONAL (  
  ID NUMBER (3,0) PRIMARY KEY,  
  EQUIPO VARCHAR(50),  
  PAIS VARCHAR(30),  
  CATEGORIA VARCHAR(20)  
);
```

```
INSERT INTO INFO_PERSONAL  
SELECT ID, NAME, NATIONALITY FROM TOUR;
```

```
INSERT INTO INFO_PROFESIONAL  
SELECT ID, TEAM, COUNTRY, CATEGORY FROM TOUR;
```

/*EJERCICIO 3*/

```
CREATE VIEW TOUR_VIEW AS  
SELECT PER.ID, PRO.EQUIPO, PRO.PAIS, PER,NOMBRE, PER.NACIONALIDAD,  
PRO.CATEGORIA  
FROM INFO_PERSONAL PER JOIN INFO_PROFESIONAL PRO  
ON(PER.ID=PRO.ID);
```

```
SELECT * FROM TOUR;
```

```
SELECT * FROM TOUR_VIEW;
```

```
--PRUEBA, NO SE VE EN LA VISTA PORQUE LO INSERTAMO EN TOUR--  
--INSERT INTO TOUR VALUES (500, 'UMA INFORMÁTICA', 'ESPAÑA', 'ENRIQUE',  
'ESP','WORLDTOUR');--
```

/*EJERCICIO 4*/

```
CREATE OR REPLACE TRIGGER TR_INSERTA_CICLISTA
INSTEAD OF
INSERT
ON TOUR_VIEW
FOR EACH ROW
BEGIN
    INSERT INTO INFO_PERSONAL VALUES(
        :NEW.ID, :NEW.NOMBRE, :NEW.NACIONALIDAD);
    INSERT INTO INFO_PROFESIONAL VALUES(
        :NEW.ID, :NEW.EQUIPO, :NEW.PAIS, :NEW.CATEGORIA);
END TR_INSERTA_CICLISTA;
/
--PRUEBA--
INSERT INTO TOUR_VIEW VALUES(500, 'UMA INFORMÁTICA', 'ESPAÑA',
'ENRIQUE', 'ESP','WORLDTOUR');
```

/*EJERCICIO 5*/

```
--CREA NUEVA TABLA--
CREATE TABLE LOG_INSERTION(
    USUARIO VARCHAR2(12),
    FECHA DATE
);

--GUARDA QUIENES INTENTAN ALTERAR TOUR Y CUANDO--
CREATE OR REPLACE TRIGGER TR_LOG_INSERTION
AFTER
INSERT OR DELETE OR UPDATE
ON TOUR
BEGIN
    INSERT INTO LOG_INSERTION VALUES ( user, sysdate);
END TR_LOG_INSERTION;

--PRUEBA--
INSERT INTO TOUR VALUES(501, 'UMA INFORMÁTICA', 'ESPAÑA', 'ENRIQUE',
'ESP','WORLDTOUR');
```

/*EJERCICIO 6*/

```
CREATE VIEW TOUR_ITALY AS  
SELECT* FROM TOUR WHERE UPPER(NACIONALITY) = 'ITA';
```

```
CREATE VIEW TOUR_SPAIN AS  
SELECT* FROM TOUR WHERE UPPER(NACIONALITY) = 'ESP';
```

--PARA ASEGURAR QUE SE CUMPLA LA CONDICIÓN DE WHERE Y SINO
SALTE ERROR HAY QUE ESCRIBIR LO SIGUIENTE:--
--WITH CHECK OPTION--

/*EJERCICIO 7*/

```
INSERT INTO TOUR_SPAIN VALUES (1, 'TEAM CHERRY', 'Spain', 'MARCOS  
RAMIREZ',  
'ESP', 'WorldTour')
```

/*EJERCICIO 8*/

```
INSERT INTO TOUR_SPAIN VALUES (1, 'TEAM CHERRY', 'Spain', 'CARLOS  
JIMENEZ',  
'ITA', 'WorldTour')
```

--Se inserta pero no sale reflejado.--

/*EJERCICIO 9*/

```
create view TOUR_SPAIN as select*from TOUR  
where UPPER(NACIONALITY) = 'ESP' WITH CHECK OPTION;
```

```
create view TOUR_ITALY as select*from TOUR  
where UPPER(NACIONALITY) = 'ITA'; WITH CHECK OPTION;
```

/*EJERCICIO 11*/

```
CREATE OR REPLACE VIEW "UBD2285"."TOUR_SPAIN" ("ID", "TEAM",  
"COUNTRY", "NAME", "NACIONALITY", "CATEGORY") AS  
select"ID","TEAM","COUNTRY","NAME","NACIONALITY","CATEGORY"from TOUR  
where UPPER(NACIONALITY) = 'ESP';
```