

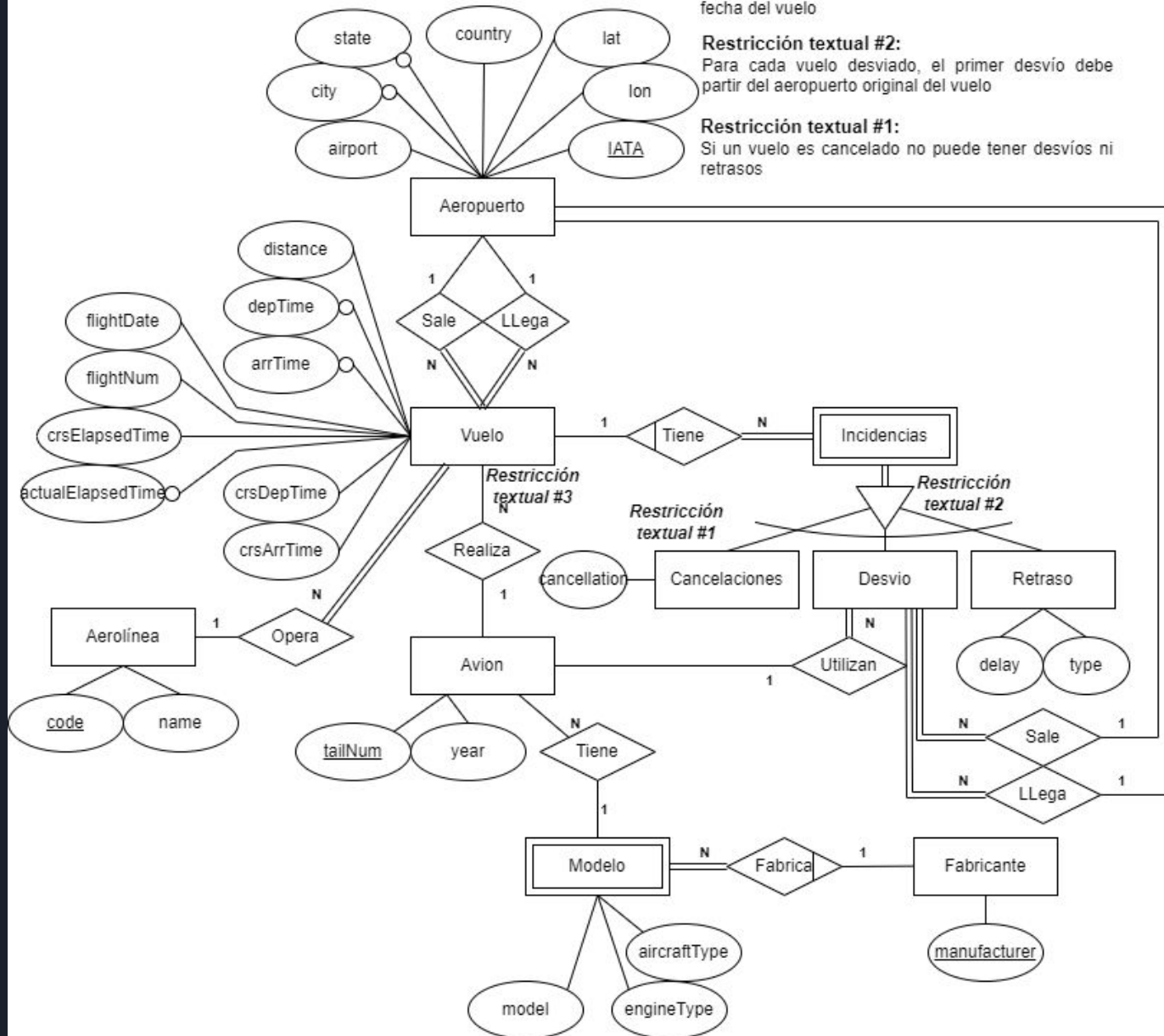


BBDD - Práctica 3

Héctor Toral
Francisco Javier Pizarro
Pablo López

29/05/2022

Modelo E/R





Alternativas a nuestro modelo:

- En vez de tener una tabla incidencias débil de vuelo habíamos pensado tener 2 relaciones débiles y un atributo cancelado finalmente se descarto ya que finalmente se tradujo la especialización con 3 tablas (cancelaciones, desvíos, retrasos) y el atributo de cancelado no era necesario para ninguna de las consultas por lo que no supondría ninguna mejora de rendimiento según los requisitos actuales.
- Otra alternativa era tener la entidad de aerolíneas con una relación 1:N con la tabla de aviones pero esto no nos sacaría todos los datos correctos a la hora de realizar consultas, además de que perjudica el rendimiento a la hora de tener que hacer un join de más al sacarlo de la relación con avión, pudiendo hacerlo directamente desde la tabla de vuelo

Modelo relacional

Aeropuerto = (iata: cadena(4); **airport**: cadena(45), no nulo; **country**: cadena(30), no nulo; **state**: cadena(2); **city**: cadena(35), no nulo; **lat**: racional, no nulo; **lon**: racional, no nulo);
único (lat, lon)

Fabricante= (manufacturer: cadena(30));

Modelo= (id: cadena(40); **model**: cadena(20), no nulo; **aircraftType**: cadena(25), no nulo; **engineType**: cadena(15), no nulo; manufacturer: cadena(30), no nulo);
único (model, manufacturer)

Aerolinea= (code: cadena(7); **name**: cadena(100), no nulo);


Avion= (tailNum: cadena(7); modelId: cadena(40), **year**: entero);
verificar que: year >= 1890

Vuelo= (id: cadena(40); **flightDate**: cadena(10), no nulo; **flightNum**: entero, no nulo; **distance**: entero, no nulo; origin: cadena(4), no nulo; **destination**: cadena(4), no nulo; **tailNum**: cadena(7);
carrier: cadena(7), no nulo; **depTime**: cadena(4), **arrTime**: cadena(4), **crsDepTime**: cadena(4), no nulo; **crsArrTime**: cadena(4), no nulo; **crsElapsedTime**: entero, no nulo; **actualElapsedTime**: entero);
único (origin, carrier, flightNum, flightDate)
verificar que: distance >= 0, origin <> destination

Cancelaciones= (id: cadena(40); idVuelo: cadena(40), no nulo; **cancellation**: cadena(100), no nulo);
único (idVuelo, cancellation)

Desvios= (id: cadena(40); idVuelo: cadena(40), no nulo; origin: cadena(40), no nulo; tailNum: cadena(7), no nulo; destination: cadena(4), no nulo);
único (idVuelo, origin, tailNum, destination)

Retrasos= (id: cadena(40); idVuelo: cadena(40), no nulo; **delay**: entero, no nulo; **delayType**: cadena(50), no nulo);
verificar que: **delay** >= 0
único (idVuelo, delay, delayType)



Número de compañías que operan en al menos cinco aeropuertos de Alaska

```
select count(*) as "Número de aerolíneas"
from (
  select carrier
  from (
    select distinct V.origin, carrier
    from aeropuerto A, vuelo V
    where A.iata = V.origin and A.state = 'AK'
  ) A
  group by carrier
  having count(*) >= 5
) N;
```

Aeropuerto en el que operan los aviones más modernos

```
select S.airport, S.iata, FLOOR(S.average)
from edadMediaAeropuertos S
where S.average = (
    select MIN(average) as minimo
    from edadMediaAeropuertos A
);

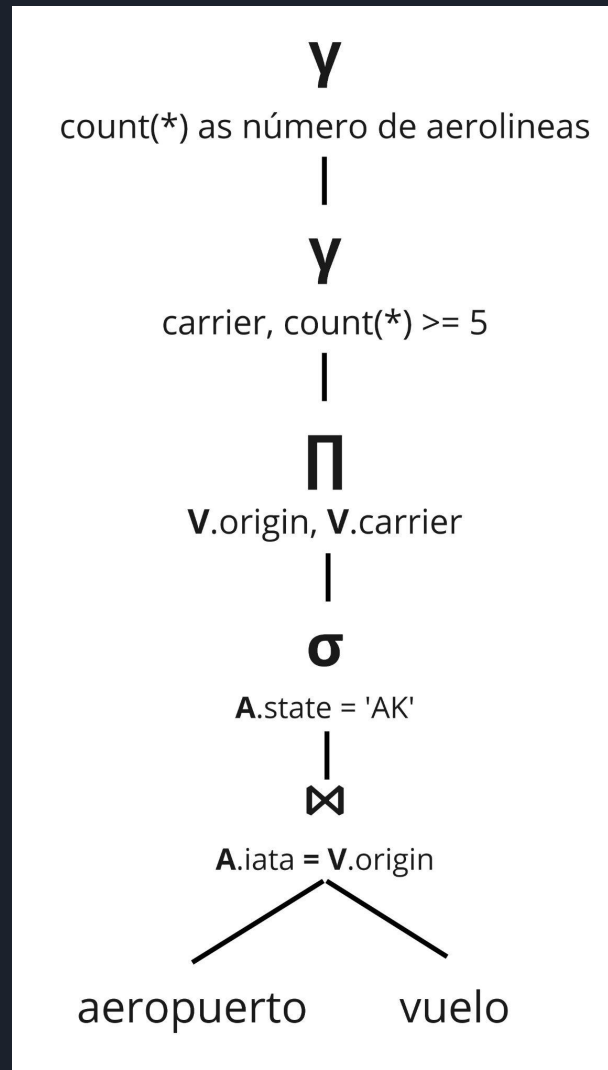
create view edadMediaAeropuertos as
select AE.airport, AE.iata, AVG(2022 - AV.year) as average
from aeropuerto AE, vuelo V, avion AV
where (AE.iata = V.origin or AE.iata = V.origin) and AV.tailNum = V.tailNum
and
    AV.year is not null
group by AE.airport, AE.iata;
```

Compañía con el mayor porcentaje de vuelos que despegan y aterrizan en un mismo estado.

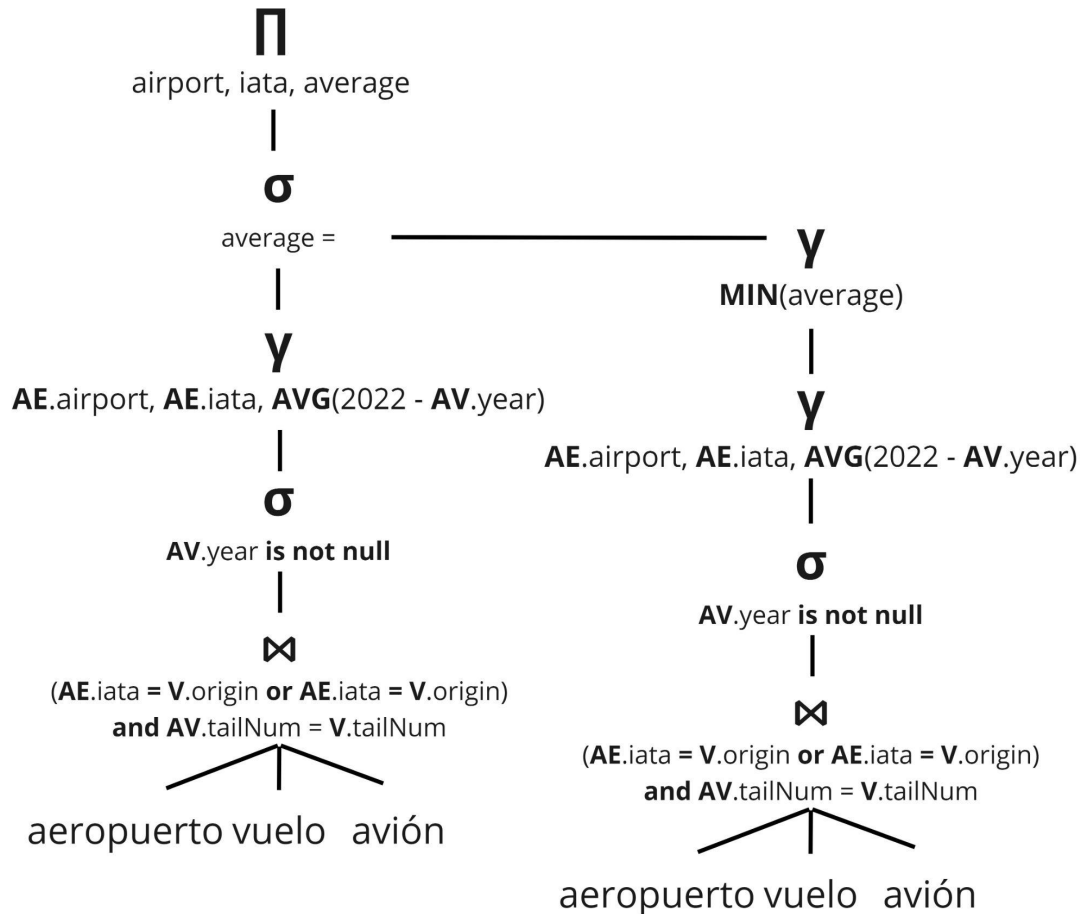
```
create view porcentajeSalidasLlegadasMismoEstado as
select A.carrier, ROUND((A.num / B.num * 100), 2) as porcentaje
from (
    select S.carrier, COUNT(*) as num
    from (
        select V.carrier, A1.state as origin, A2.state as destination
        from vuelo V, aeropuerto A1, aeropuerto A2
        where V.origin = A1.iata and V.destination = A2.iata and A1.state = A2.state
    ) S
    group by S.carrier
) A, (
    select T.carrier, COUNT(*) as num
    from (
        select V.carrier, A1.state as origin, A2.state as destination
        from vuelo V, aeropuerto A1, aeropuerto A2
        where V.origin = A1.iata and V.destination = A2.iata
    ) T
    group by T.carrier
) B
where A.carrier = B.carrier

select A.name, B.porcentaje
from aerolinea A, porcentajeSalidasLlegadasMismoEstado B
where A.code = B.carrier and B.porcentaje = (
    select MAX(porcentaje)
    from porcentajeSalidasLlegadasMismoEstado A
);
```

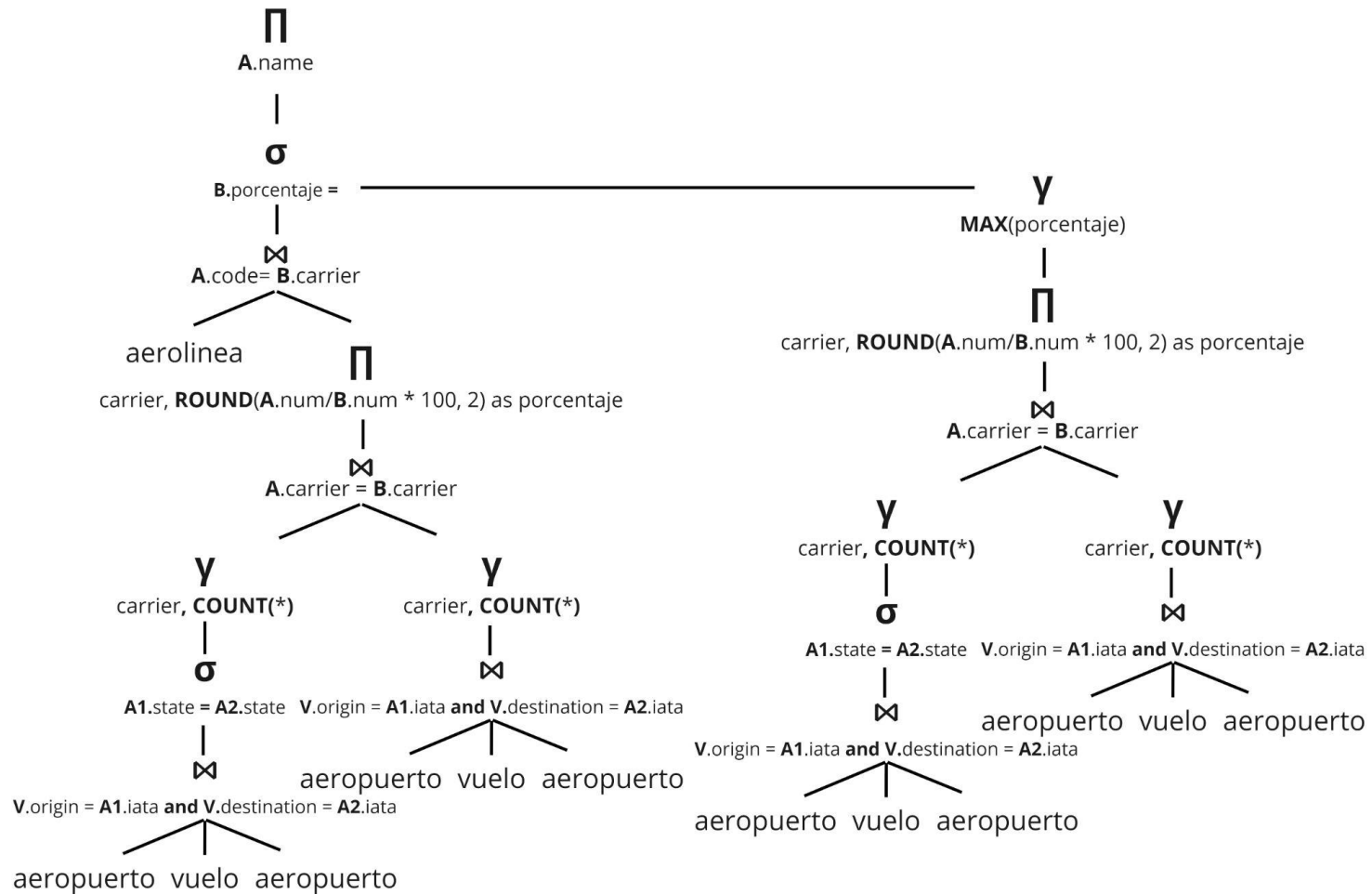
1 - Número de compañías que operan en al menos cinco aeropuertos de Alaska



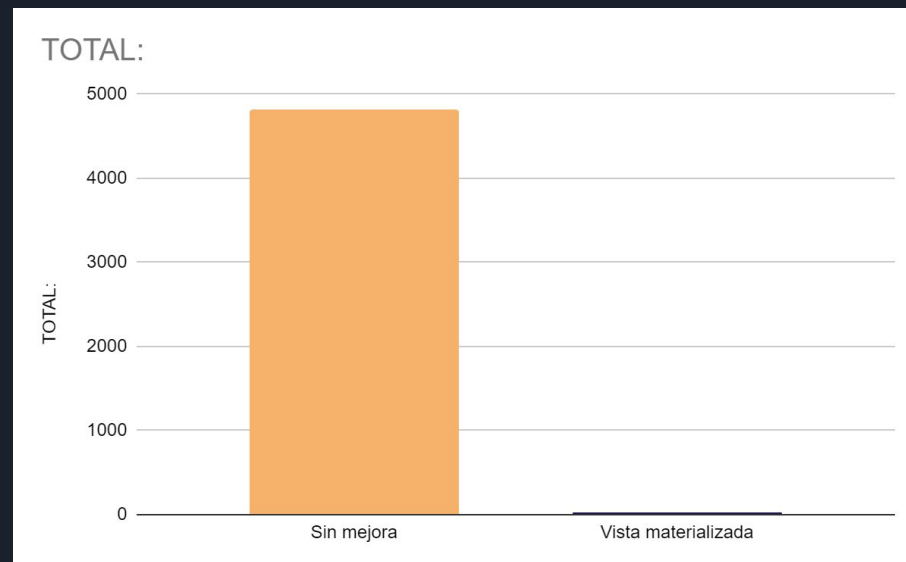
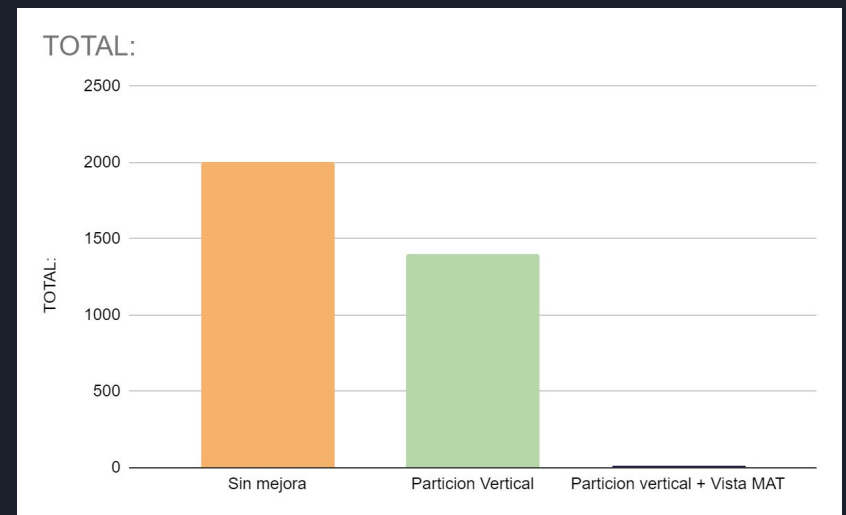
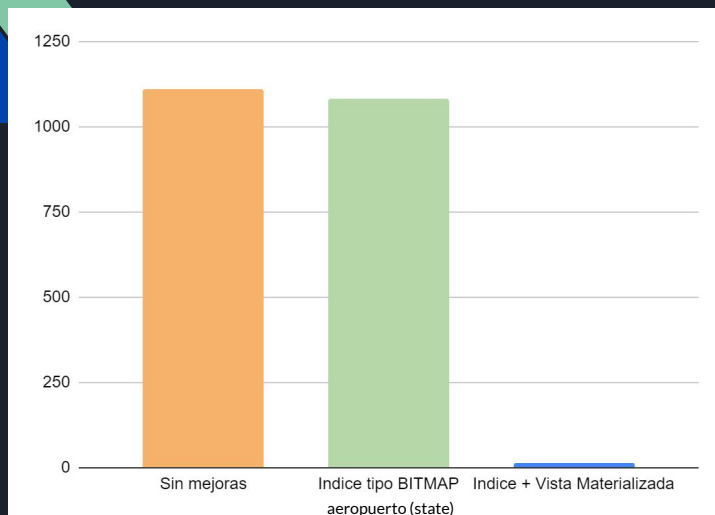
2- Aeropuerto en el que operan los aviones más modernos



3- Compañía con el mayor porcentaje de vuelos que despegan y aterrizan en un mismo estado.



Diseño físico





Trigger 1

```
-- Un vuelo no puede ser desviado si ha sido cancelado
CREATE OR REPLACE TRIGGER EXC_DESVIOS
BEFORE INSERT ON desvios
FOR EACH ROW
DECLARE
    flag NUMBER;
BEGIN
    SELECT COUNT(*) INTO flag
    FROM cancelaciones
    WHERE idVuelo = :NEW.idVuelo;

    IF flag >= 1 THEN
        RAISE_APPLICATION_ERROR (-20000, 'No puede haber desvio puesto que el vuelo ha sido cancelado');
    END IF;
END;
/
```

Trigger 2

```
--Verifica que el primer desvio de un vuelo parta del mismo aeropuerto de origen
CREATE OR REPLACE TRIGGER EXT_PRIMER_DESVIO
BEFORE INSERT ON desvios
FOR EACH ROW
DECLARE
    flag NUMBER;
    origin VARCHAR2(4);
BEGIN
    SELECT origin INTO origin
    FROM vuelo
    WHERE id = :NEW.idVuelo;

    SELECT COUNT(*) INTO flag
    FROM desvios
    WHERE idVuelo = :NEW.idVuelo;

    IF origin <> :NEW.origin and flag = 0 THEN
        -- No es el mismo avion que el del vuelo original vuelo
        RAISE_APPLICATION_ERROR (-20002, 'No existe el primer desvio del vuelo: ' || :NEW.idVuelo || ',
con matricula: ' || origin);
    END IF;
END;
/
```

Trigger 3

–Se encarga de verificar que la fecha de vuelo es posterior a la fecha de montaje del avión

```
CREATE OR REPLACE TRIGGER FECHAS
BEFORE INSERT ON vuelo
FOR EACH ROW
DECLARE
    flag NUMBER;
BEGIN
    SELECT COUNT(*) INTO flag
    FROM avion
    WHERE year > SUBSTR(:NEW.flightDate,1,4);
    IF flag >= 1 THEN
        RAISE_APPLICATION_ERROR (-20000, 'Error al añadir el vuelo: la fecha del vuelo no puede ser
anterior a la de creación del avion');
    END IF;
END;
/
```