UNIZAR – UNIVERSIDAD DE ZARAGOZA

BASES DE DATOS

Universidad Zaragoza

DANIEL RUEDA
JORGE SANZ
ALVARO MONTEAGUDO

INGENIERIA INFORMATICA

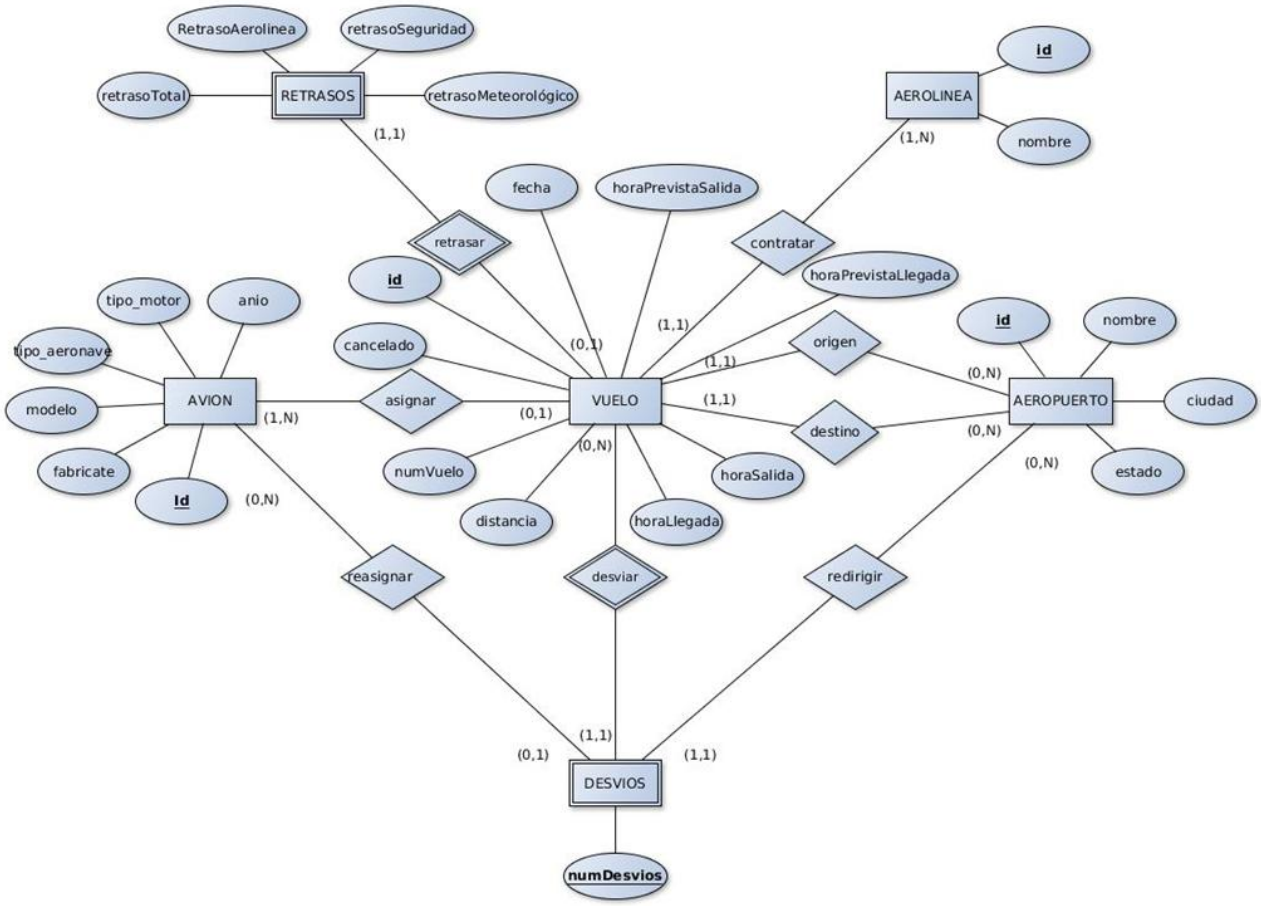
SEGUNDO AÑO

SEGUNDO CUATRIMESTRE

10/09/2015

PARTE 1: DISEÑO DE LA BASE DE DATOS

MODELO ENTIDAD RELACIÓN

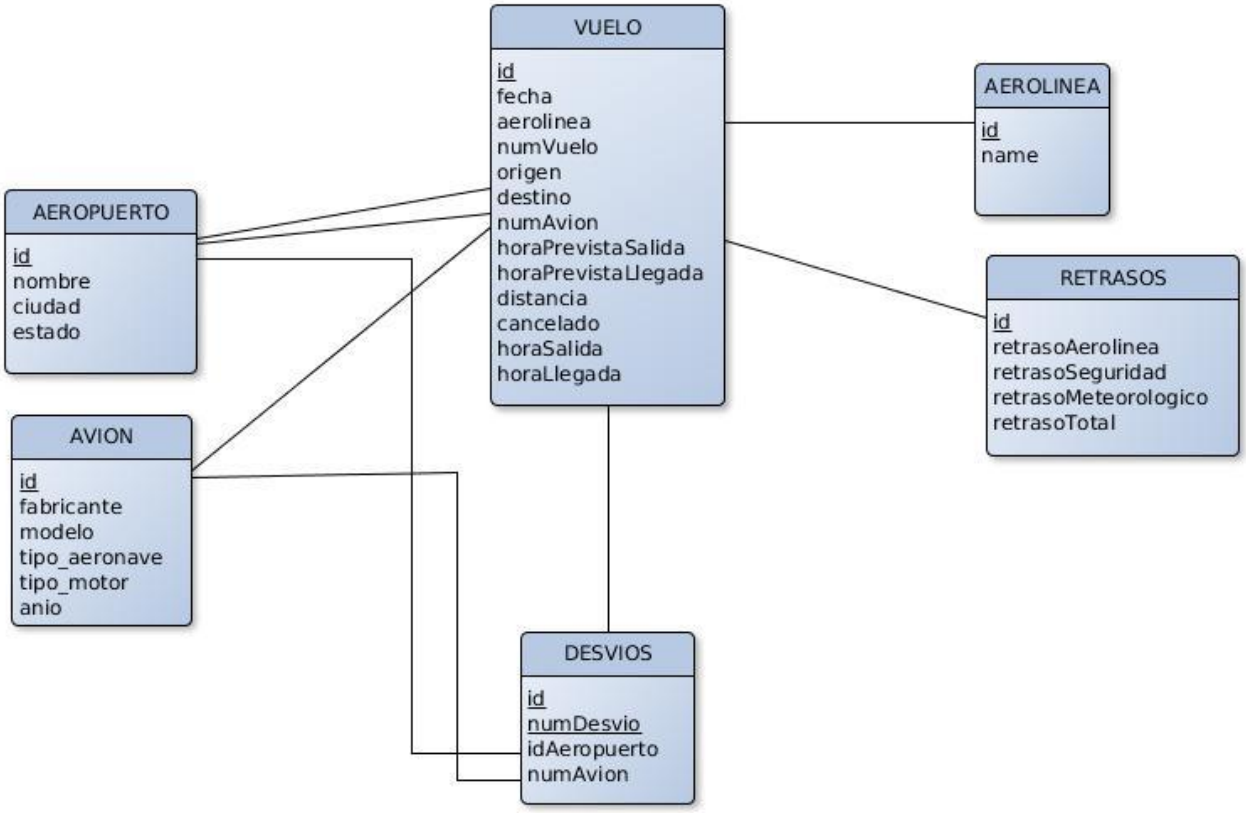


Hemos considerado a vuelo como una entidad fuerte, de forma que su existencia e identificación no dependen de otras identidades. La entidad vuelo tiene un entero como clave primaria, tiene también un origen y un destino que son relaciones con la entidad aeropuerto y otras relaciones con las entidades avión y aerolínea que representan el avión y la aerolínea empleados en el vuelo respectivamente. Las entidades retrasos y desvíos son entidades dependientes de vuelo. La entidad desvío tiene además otras dos relaciones, la primera con avión debido a que cuando un avión es desviado es posible un cambio de avión y otra con aeropuerto que es el aeropuerto al que ha sido desviado.

Hubiera sido posible juntar las entidades vuelo, desvíos y retrasos dado que la cardinalidad entre vuelo y retrasos es de 1 a 1 y la de vuelo y desvíos es de 1 a 0,1 o 2, sin embargo consideramos que es información de ámbitos diferentes y por eso lo hacemos así.

También podríamos haber utilizado fecha,aerolínea,origen,destino y numVuelo como clave primaria y considerar a vuelo como entidad dependiente, pero hubiera sido necesario repetir todas esos atributos en las entidades retrasos y desvíos.

MODELO RELACIONAL



Para la transformación del esquema E/R al relacional no ha sido necesaria la creación de ninguna tabla debido a que la cardinalidad de las relaciones es como máximo de 1 a N.

NORMALIZACIÓN

El esquema relacional cumple la 1ª FN porque esta es inherente al modelo relacional.

Cumple la 2ª FN porque esta en 1ª FN y la única entidad con más de un atributo identificador es desvíos, pero sus atributos dependen de la clave completa.

El esquema relacional cumple la 3ª FN porque esta en 2ª FN y ningún atributo no perteneciente a la clave depende de otro atributo no perteneciente a la clave.

El esquema relacional cumple la FNBC porque esta en 3ª FN y las únicas dependencias son entre un atributo y su clave.

El esquema relacional cumple la 4ª FN porque está en FNBC y no existen dependencias multivaluadas.

El fichero sql correspondiente es el siguiente:

```
CREATE TABLE AVION(
id CHAR(7) PRIMARY KEY,
fabricante CHAR(30),
modelo CHAR(20),
tipo_aeronave CHAR(25),
tipo_motor CHAR(15),
anio NUMBER(4));

CREATE TABLE AEROLINEA(
id CHAR(7) PRIMARY KEY,
nombre CHAR(100));

CREATE TABLE AEROPUERTO(
id CHAR(4) PRIMARY KEY,
nombre CHAR(45),
ciudad CHAR(35),
estado CHAR(2));

CREATE TABLE VUELO(
id NUMBER(5) PRIMARY KEY,
fecha CHAR(11) NOT NULL,
aerolinea CHAR(7) NOT NULL,
numVuelo NUMBER(4) NOT NULL,
origen CHAR(4) NOT NULL,
destino CHAR(4) NOT NULL,
numAvion CHAR(7),
horaPrevistaSalida CHAR(4),
horaPrevistaLlegada CHAR(4),
distancia NUMBER(4),
cancelado NUMBER(1) NOT NULL,
horaSalida CHAR(4),
horaLlegada CHAR(4),
FOREIGN KEY (aerolinea) REFERENCES AEROLINEA(id),
FOREIGN KEY (numAvion) REFERENCES AVION(id),
FOREIGN KEY (origen) REFERENCES AEROPUERTO(id),
FOREIGN KEY (destino) REFERENCES AEROPUERTO(id));

CREATE TABLE RETRASOS(
id NUMBER(5),
retrasoAerolinea NUMBER(4),
retrasoSeguridad NUMBER(4),
retrasoMeteorologico NUMBER(4),
retrasoTotal NUMBER(4),
FOREIGN KEY (id) REFERENCES VUELO(id),
PRIMARY KEY (id));

CREATE TABLE DESVIOS(
id NUMBER(5),
numDesvio NUMBER(1),
idAeropuerto CHAR(4) NOT NULL,
numAvion CHAR(7),
FOREIGN KEY (id) REFERENCES VUELO(id),
PRIMARY KEY (id,numDesvio));
```

DATOS SOBRE REUNIONES

La primera parte de esta práctica se realizo a finales de junio por las mañanas en una biblioteca. No surgieron demasiados problemas y se hizo con relativa rapidez, debido a nuestra experiencia con las otras dos. En total se emplearon cerca de 3 horas, aunque posteriormente se realizaron algunos cambios.

PARTE 2: POBLACIÓN BASE DE DATOS Y CONSULTAS

INSERCIÓN

Para poblar la base de datos se han utilizado los datos de la base de datos en mysql. Mediante consultas se obtuvieron los datos deseados. Estos fueron guardados en ficheros con los atributos separados por ",". Con otro programa java seleccionaron los atributos adecuados para cada una de las tablas para luego añadirlos al fichero de inserción.
Debido a que muchos de los atributos utilizaban caracteres especiales (´ , ' , & ...) y a que estos no eran admitidos en oracle, eliminamos los caracteres especiales con el editor de texto sublime text y su sustitución de patrones.

ESTADÍSTICAS DE LAS TABLAS

AEROPUERTO: 3376 tuplas, 252.3 Kb. Tuplas de ejemplo:

ID

NOMBRE

CIUDAD

ESTADO

00M
Thigpen
Bay Springs
MS

ID

NOMBRE

CIUDAD

ESTADO

00R
Livingston Municipal
Livingston
TX

ID

NOMBRE

CIUDAD

ESTADO

00V
Meadow Lake
Colorado Springs
CO

ID

NOMBRE

CIUDAD

ESTADO

01G
Perry-Warsaw
Perry
NY

ID

NOMBRE

CIUDAD

ESTADO

01J
Hilliard Airpark
Hilliard
FL

La consulta utilizada para obtener sus datos fue:
SELECT iata, airport, city, state FROM airports;

AVIONES: 5333 tuplas, 523.6 Kb. Tuplas de ejemplo:

ID

FABRICANTE

MODELO

TIPO_AERONAVE

TIPO_MOTOR ANIO
--- ---
N11155
EMBRAER
EMB-145XR

ID

FABRICANTE

MODELO

TIPO_AERONAVE

TIPO_MOTOR ANIO
--- ---
Fixed Wing Multi-Engine
Turbo-Fan 2004

ID

FABRICANTE

MODELO

TIPO_AERONAVE

TIPO_MOTOR ANIO
--- ---
N11164
EMBRAER
EMB-145XR

ID

FABRICANTE

MODELO

TIPO_AERONAVE

TIPO_MOTOR ANIO
--- ---
Fixed Wing Multi-Engine
Turbo-Fan 2004

ID

FABRICANTE

MODELO

TIPO_AERONAVE

TIPO_MOTOR ANIO
--- ---
N11165
EMBRAER
EMB-145XR

La consulta utilizada para obtener sus datos fue:
SELECT * FROM planes;

AEROLINEAS: 1495 tuplas, 90 Kb. Tuplas de ejemplo:

ID

NOMBRE

02Q
Titan Airways

04Q
Tradewind Aviation

05Q

ID

NOMBRE

Comlux Aviation, AG

06Q
Master Top Linhas Aereas Ltd.

07Q
Flair Airlines Ltd.

La consulta utilizada para obtener sus datos fue:
SELECT * FROM carriers;

El principal problema en la inserción de datos fue la tabla vuelos, debido a que nosotros la dividimos en tres tablas y le asignamos un identificador único. Utilizamos un programa en java para asignar un identificador único y otro para escoger los atributos específicos en cada una de las tablas.

Todos los datos de las 3 tablas vienen de la siguiente consulta:

```
SELECT flightDate, carrier, flightNum, origin, dest, tailNum, crsDepTime, crsArrTime, distance, cancelled, depTime,
arrTime, carrierDelay,
securityDelay, weatherDelay, nasDelay, lateAircraftDelay, div1airport, div1TailNum, div2airport, div2TailNum
FROM fights200810;
```

VUELO: 48983 tuplas, 5.4MB. Tuplas de ejemplo:

```
         ID FECHA                          AEROLINEA             NUMVUELO
---------- -------------------------------- -------------------- ----------
ORIGEN             DESTINO      NUMAVION             HORAPREVISTA HORAPREVISTA
------------ ------------ -------------------- ------------ ------------
 DISTANCIA  CANCELADO HORASALIDA   HORALLEGADA
---------- ---------- ------------ ------------
         1 2008-10-25                      AA                    2
LAX        JFK    N335AA               935         1802
    2475           0 935           2153

         2 2008-10-25                      US                   12
PHX        JFK    N664AW               903         1700
    2153           0 1005          2218

         ID FECHA                          AEROLINEA             NUMVUELO
```

---------- -------------------------------- -------------------- ----------
ORIGEN DESTINO NUMAVION HORAPREVISTA HORAPREVISTA
---------- ---------- -------------------- ---------- ----------
DISTANCIA CANCELADO HORASALIDA HORALLEGADA
---------- ---------- ---------- ----------

3 2008-10-25 CO 16
LAX EWR N57111 1020 1837
2454 0 1033 2206

4 2008-10-25 AA 24
SFO JFK N352AA 810 1640

ID FECHA AEROLINEA NUMVUELO
---------- -------------------------------- -------------------- ----------
ORIGEN DESTINO NUMAVION HORAPREVISTA HORAPREVISTA
---------- ---------- -------------------- ---------- ----------
DISTANCIA CANCELADO HORASALIDA HORALLEGADA
---------- ---------- ---------- ----------
2586 0 823 2123

5 2008-10-27 WN 41
DAL HOU N433LV 1630 1735
239 0 1752

En la tabla original de vuelos existían 5 tipos de retrasos, pero como algunos de ellos apenas se usaban solo cogimos 3 de ellos y el tiempo total de los 5, que fue calculado con un programa en java.
Las tuplas cuyo retraso era <=0 fueron eliminadas.

VUELO: 7528 tuplas, 348.2KB. Tuplas de ejemplo:

ID RETRASOAEROLINEA RETRASOSEGURIDAD RETRASOMETEOROLOGICO RETRASOTOTAL
---------- ---------- ---------- ---------- ----------
132 21 0 0 21
133 37 0 0 37
137 0 0 0 28
140 0 0 0 21
144 0 0 0 22

Añadimos un identificador mas a desvíos, de forma que este se identificase con el id del vuelo y el

número de desvío. Para ello seleccionamos primero los primeros desvíos y les pusimos identificador 1, para luego coger a los segundos con identificador 2.

VUELO: 145 tuplas, 7.2KB. Tuplas de ejemplo:

La tabla desvíos consta de 145 tuplas, un tamaño de 7.2 kB y tiene el siguiente formato:

ID	NUMDESVIO	IDAEROPUERTO	NUMAVION
1	1	SYR	N335AA
2	1	MDT	N664AW
3	1	SYR	N57111
4	1	IAD	N352AA
5	1	HRL	

CONSULTAS

1) La ruta entre dos ciudades cualesquiera con un mayor de cruces reales en el aire. Se entiende por cruce que dos aviones estén volando simultáneamente entre dos ciudades en direcciones opuestas. Se piden cruces reales, por tanto se tienen que tener en cuenta los desvíos.

En la consulta se ha utilizado una vista para obtener los vuelos que realmente se han producido, es decir, sustituyendo en los vuelos los destinos por sus respectivos desvíos si es que tienen. También se han eliminado los vuelos cancelados.
Para cada ruta de esa vista se ha seleccionado aquella cuyo número de vuelos con ruta contraria sea máxima.

2) Lista las ciudades con vuelos que sólo están servidas con aviones con hélices

Se ha seleccionado las ciudades que tienen algún vuelo con avión de hélices y se le ha restado aquellas ciudades que tengan algún vuelo sin avión de hélices.

3) Lista los estados ordenados por el ratio de conexión entre sus aeropuertos (de mayor a menor ratio). Se entiende por ratio al cociente entre el número de conexiones diferentes existentes entre ciudades de un mismo estado y el máximo total de conexiones posibles

Se ha creado una vista con los estados y sus conexiones, para ello se ha seleccionado distintas rutas ordenadas, es decir, el destino de la ruta es siempre aquel menor alfabéticamente.
Se ha creado otra vista agrupando las rutas en función de su estado para contar el numero de rutas por estado.
Matemáticamente el número máximo de conexiones por estado es igual a:
(NumAer*(NumAer-1))/2
Por ello se ha creado otra vista con el estado y el número máximo de conexiones en el.
Se ha seleccionado de ambas vistas el estado y el ratio, es decir,
numConexiones/numMaxConexiones.

CONSULTAS OPCIONALES

1) Seleccionar al estado con mayor número de aeropuertos desiertos, es decir, sin vuelos.

Se ha creado una vista con los aeropuertos desiertos seleccionando todos los aeropuertos y descontando aquellos que hayan actuado como origen o como destino en un vuelo.
Luego se han agrupado en función del estado y se ha seleccionado aquel con mayor numero de aeropuertos.

2) Seleccionar modelo, id del avión, conexión y número de vuelos en esta conexión, de los aviones que solo han realizado una conexión en toda su existencia.

Como en otra de las consultas, se ha creado una vista con el id de vuelo, el id del avión y la conexión (ordenando alfabéticamente) de aquellos vuelos cuyo id del avión fuera distinto de NULL. De esa vista se ha agrupado en función del id del avión, y se han cogido aquellos cuyo número de conexiones fuese igual a 1. Se ha seleccionado el id del avión, el origen, el destino y el número de vuelos en esa conexión, para luego hacer un join con la tabla aviones y sacar también el modelo de avión para cada id.

3) Seleccionar las aerolíneas junto con el retraso medio provocado por ellas en sus vuelos, es decir, retrasoTotalAerolinea/numVuelosAerolinea. La selección debe estar ordenada en función del retraso medio.

Se ha creado una vista con las aerolíneas y su retraso total provocado por ellas, mediante un group by y la función sum().
Se ha realizado un join entre la vista y las aerolíneas con su número de vuelos para luego proyectar solo los atributos aerolínea y retraso medio. Mediante un order by se han ordenado en función del retraso medio

ALGEBRA RELACIONAL

1)

$R1 = \sigma_{cancelado = 0}(VUELOS)$ // vuelos no cancelados

$R2 = \Pi_{id}(R1) - \Pi_{id}(DESVIOS)$ // vuelos no cancelados ni desviados

$R3 = \Pi_{id, origen, destino, fecha, horaSalida, horaLlegada}(R7 \mid X \mid VUELOS)$ // vuelos no cancelados ni desviados con mas información

$R4(id, max) := AGRUPAR_{MAX(numDesvio)}(DESVIO, id)$ // vuelos desviados con su numero de desvios

$R5 = \sigma_{numDesvio = \Pi_{max}(\sigma_{DESVIOS.id = R4.id}(R4))}(DESVIOS)$ // ultimos desvios con su información de vuelo

$R6 = \Pi_{id, origen, desvio, fecha, horaSalida, horaLlegada}(R5)$ // ultimos desvios junto con la información necesaria de vuelo

$R7 = R3 \cup R6$ // Vuelos reales junto con su información de vuelo

$R8 = R7$

$R9 = R7 \mid X \mid R8$ // parejas de cruces reales con su información y con repeticiones

R7.origen = R8.destino AND
R8.origen = R7.destino AND
R7.fecha = R8.fecha AND
((R7.horaSalida BETWEEN
R8.horaSalida AND R8.horaLlegada)
OR (R8.horaSalida BETWEEN
R7.horaSalida AND R7.horaLlegada))

$R10(origen, destino, numero) := AGRUPAR_{count(*)}(R9.origen, destino)$ // rutas con su numero de cruces reales y con repeticiones

$R11 = R10$

$R12 = \sigma_{R10.numero = \Pi_{MAX(R11.numero)}(R11)}(R10)$ // ruta de ida y vuelta con el maximo numero de cruces reales

$R13 = \sigma_{origen > destino}(R12)$ // ruta con el maximo numero de cruces reales

2)

$R1 = \Pi_{ciudad}(\sigma_{tipo\_aeronave = Rotorcraft}((VUELO \mid X \mid AVION) \mid X \mid AEROPUERTO))$
origen = AEROPUERTO.id OR
destino = AEROPUERTO.id

// ciudades que tienen vuelos con aviones de helices

$R2 = \Pi_{ciudad}(\sigma_{tipo\_aeronave != Rotorcraft}((VUELO \mid X \mid AVION) \mid X \mid AEROPUERTO))$
origen = AEROPUERTO.id OR
destino = AEROPUERTO.id

// ciudades que tienen vuelos sin aviones de helices

$R3 = R1 - R2$ // ciudades que tienen vuelos unicamente con aviones de helice

(3)

R1 = AEROPUERTO

R2 = AEROPUERTO ⋈ (origen = AEROPUERTO.id) VUELO ⋈ (destino = R1.id) R1 // vuelos con información de su aeropuerto origen y destino

R3 (estado, origen, destino) := $\Pi_{R1.estado, destino, origen}$ ($\sigma_{AEROPUERTO.estado = R1.estado}$ (R2)) AND destino > origen
// estados y rutas de ese estado con origen y destino intercambiados para aquellos en los que destino > origen alfabéticamente

R4 = R3 ∪ ($\sigma_{destino < origen}$ AND AEROPUERTO.estado = R1.estado (R1)) // estados y conexiones con atributos ordenados alfabéticamente y con repeticiones

R5 (estado, origen, destino) := AGRUPAR (R3, R1.estado, origen, destino)
// estados y conexiones con atributos ordenados alfabéticamente y sin repeticiones

R6 (estado, numConexiones) := AGRUPAR $_{count(*)}$ (R5, estado) // estado y numConexiones locales

R7 (estado, numMax) := AGRUPAR $_{\frac{count(*)(count(*)-1)}{2}}$ (AEROPUERTO, estado)
// estado y número máximo de conexiones locales

R8 = ρ $\Pi_{R6.estado, \frac{R6.num}{R7.numMax}}$ (R6 ⋈ (R6.estado = R7.estado) R7) // ratios de conexión para cada estado

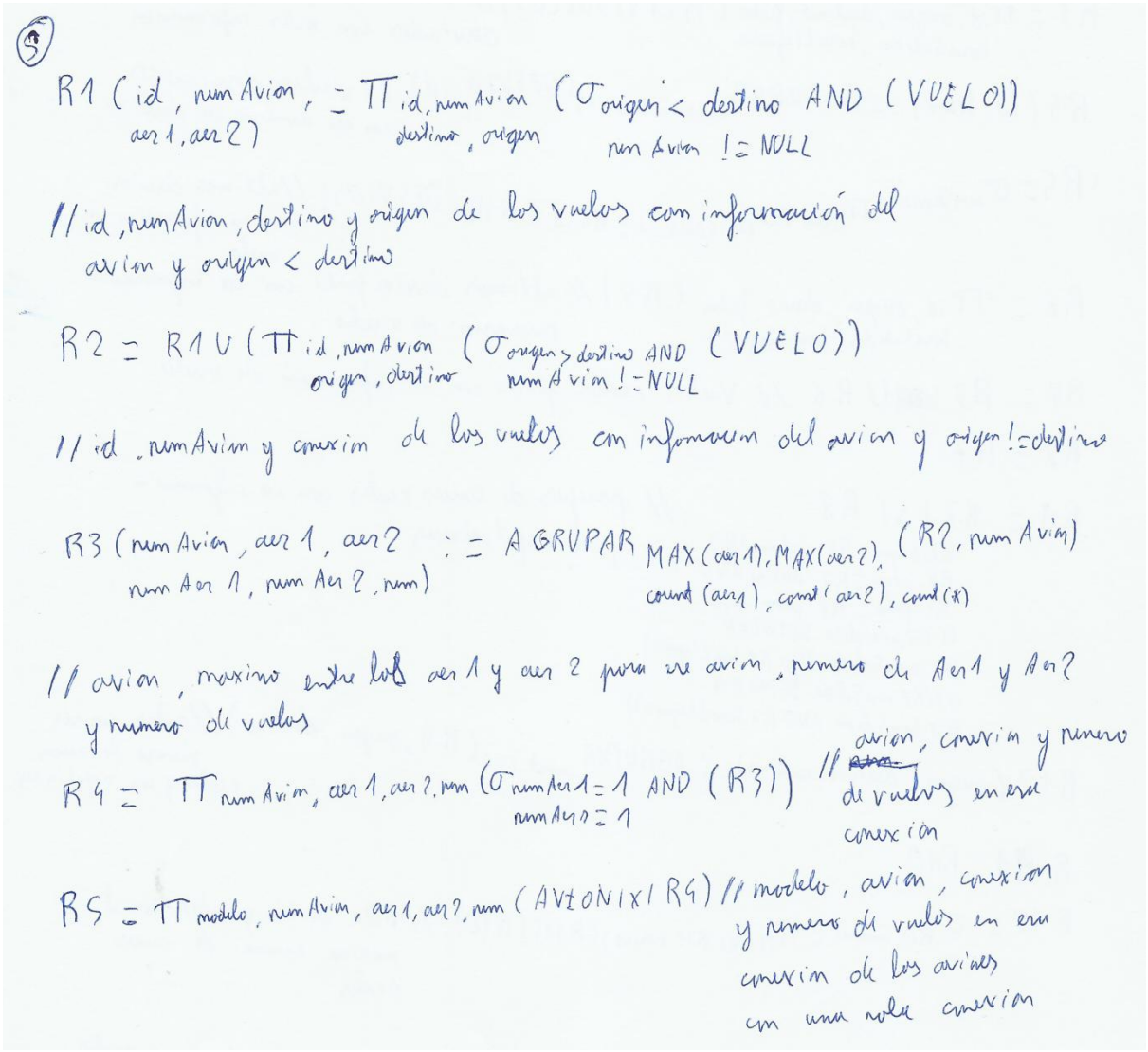
(4)

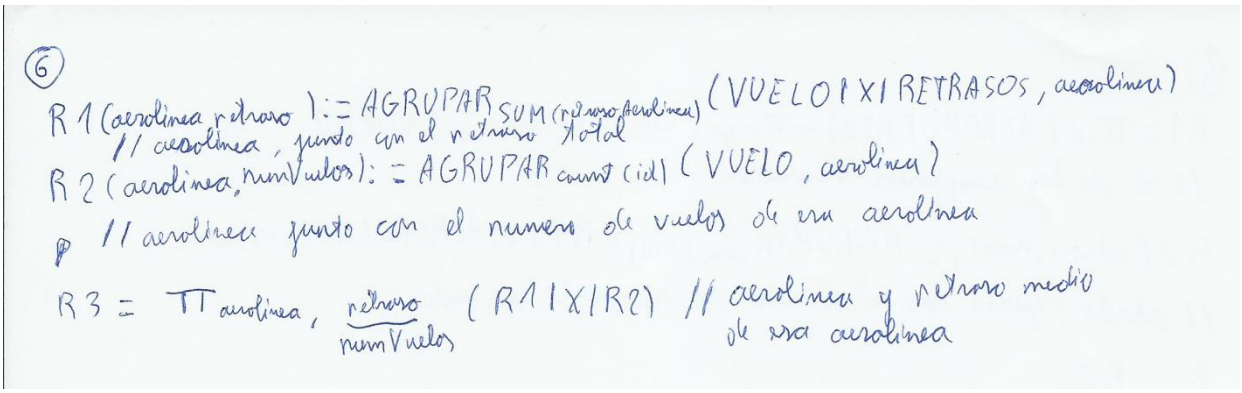
R1 = Π id (AEROPUERTO) − Π_{origen} (VUELO) − $\Pi_{destino}$ (VUELO)
// id de los aeropuertos sin vuelos

R2 (estado, num) := AGRUPAR $_{count(id)}$ (R1 ⋈ AEROPUERTO, estado)
// estados, junto con el número de aeropuertos sin vuelos

R3 = R2

R4 = $\sigma_{R2.num = \Pi_{MAX(R3.num)}(R3)}$ (R2)
// estado con mayor número de aeropuertos sin vuelos





DATOS SOBRE REUNIONES

La segunda parte de la practica la dividimos en dos partes, la primera (población de la base) se realizo entre finales de junio y principios de julio en la biblioteca. Tampoco tuvo demasiada dificultad debido a nuestra experiencia y a que los datos eran bastante limpios. De la segunda parte de la practica (consultas) se encargo Jorge Sanz, las consultas fueron realizadas a principios de agosto y el álgebra relacional a mediados de agosto en solitario. En total se emplearon cerca de 7 horas en la primera parte y 10 en la segunda.

MEMORIA PARTE 3: DISEÑO FÍSICO

ESTADISTICAS PRE_DISEÑO

1)
Statistics
---
344 recursive calls
0 db block gets
6982 consistent gets
0 physical reads
0 redo size
534 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
29 sorts (memory)
0 sorts (disk)
1 rows processed

2)
Statistics
---
656 recursive calls
0 db block gets
3044 consistent gets
0 physical reads
0 redo size
275 bytes sent via SQL*Net to client
373 bytes received via SQL*Net from client
1 SQL*Net roundtrips to/from client
24 sorts (memory)
0 sorts (disk)
0 rows processed

3)
Statistics
---
61 recursive calls
0 db block gets
880 consistent gets
0 physical reads
0 redo size
1305 bytes sent via SQL*Net to client
395 bytes received via SQL*Net from client
3 SQL*Net roundtrips to/from client
4 sorts (memory)
0 sorts (disk)
24 rows processed

4)
Statistics
---
44 recursive calls
0 db block gets
2614 consistent gets
0 physical reads
124 redo size
484 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
9 sorts (memory)
0 sorts (disk)
1 rows processed

5)
Statistics
---
53 recursive calls
0 db block gets
991 consistent gets
0 physical reads
0 redo size
8017 bytes sent via SQL*Net to client
527 bytes received via SQL*Net from client
15 SQL*Net roundtrips to/from client
6 sorts (memory)
0 sorts (disk)
208 rows processed

6)
Statistics
---
20 recursive calls
0 db block gets
1191 consistent gets
0 physical reads
0 redo size
1250 bytes sent via SQL*Net to client
395 bytes received via SQL*Net from client
3 SQL*Net roundtrips to/from client
3 sorts (memory)
0 sorts (disk)
18 rows processed

DISEÑO FÍSICO

Las claves utilizadas ya aportan información por sí mismas, la única que no aporta demasiada información es la de vuelo, pero no es adecuado cambiar su clave puesto que se necesitan demasiados campos para aportar información.

Se ha considerado la opción de realizar particiones sobre la tabla de aviones en función de los valores de tipo_aeronave para acelerar la consulta número 2, pero se ha optado por utilizar índices en su lugar.

La partición vertical ya se realizó en la fase conceptual sobre la tabla vuelo debido a que los atributos básicos del vuelo, sus retrasos y sus desvíos no se consideraban de la misma entidad además de que a menudo se usan por separado.

No se han realizado precálculos de joins debido a que supondría un coste muy alto por el tamaño.

Se ha creado una vista materializada con los últimos desvíos de cada vuelo si es que tienen y así algunos joins en la consulta numero 1. La tabla de desvíos es pequeña, por lo que no supone un coste demasiado alto. Esta vista se actualiza cada día.

```
CREATE MATERIALIZED VIEW UD (id,destino,avion)
REFRESH COMPLETE
START WITH TO_DATE('02-09-2015 19:00:00','DD-MM-YYYY HH24:MI:SS')
NEXT SYSDATE + 1
as select D1.id,D1.idAeropuerto,D1.numAvion from desvios D1
where D1.numDesvio=(select max(D2.numDesvio) from desvios D2
                    where D2.id=D1.id);
```

Se han creado también índices sobre aquellos atributos que han sido usados frecuentemente en condiciones de selección.

```
CREATE INDEX tipo_idx ON AVION(tipo_aeronave);
CREATE INDEX modelo_idx ON AVION(modelo);
CREATE INDEX estado_idx ON AEROPUERTO(estado);
CREATE INDEX aerolinea_idx ON VUELO(aerolinea);
CREATE INDEX numAvion_idx ON VUELO(numAvion);
```

ESTADISTICAS POST_DISEÑO

1)
Statistics
---
93 recursive calls
0 db block gets
5994 consistent gets
0 physical reads
124 redo size
534 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
25 sorts (memory)
0 sorts (disk)
1 rows processed

2)
Statistics
---
29 recursive calls
0 db block gets
150957 consistent gets
129 physical reads
0 redo size
275 bytes sent via SQL*Net to client
373 bytes received via SQL*Net from client
1 SQL*Net roundtrips to/from client
6 sorts (memory)
0 sorts (disk)
0 rows processed

3)
Statistics
---
38 recursive calls
0 db block gets
879 consistent gets
0 physical reads
0 redo size
1305 bytes sent via SQL*Net to client
395 bytes received via SQL*Net from client
3 SQL*Net roundtrips to/from client
4 sorts (memory)
0 sorts (disk)
24 rows processed

4)
Statistics
---
44 recursive calls
0 db block gets
2612 consistent gets
0 physical reads
0 redo size
484 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
9 sorts (memory)
0 sorts (disk)
1 rows processed

5)
Statistics
---
44 recursive calls
0 db block gets
2612 consistent gets
0 physical reads
0 redo size
484 bytes sent via SQL*Net to client
384 bytes received via SQL*Net from client
2 SQL*Net roundtrips to/from client
9 sorts (memory)
0 sorts (disk)
1 rows processed

6)
Statistics
---
19 recursive calls
0 db block gets
827 consistent gets
130 physical reads
0 redo size
1250 bytes sent via SQL*Net to client
395 bytes received via SQL*Net from client
3 SQL*Net roundtrips to/from client
3 sorts (memory)
0 sorts (disk)
18 rows processed

TRIGGERS

1) El primer trigger se encarga de que si se realiza un retraso en la hora real de salida en la tabla VUELO, este cambio se tenga en cuenta también en la tabla RETRASOS.
Ejemplo:

El vuelo con id=48983, tiene de hora prevista las 9:25 y de hora real de salida las 9:25, por tanto no tiene retraso.

Si realizamos la siguiente operación, el trigger se encarga de añadir una tupla a la tabla de retrasos con retraso total=1

update vuelo set horaSalida=0926 where id=48983;

Si luego realizamos la siguiente operación, el trigger se encarga de actualizar su tupla en la tabla retrasos a 5.

update vuelo set horaSalida=0930 where id=48983;

2) Este trigger se encarga de que no se puedan modificar los atributos horaPrevistaSalida y horaPrevistaLlegada, puesto que estos valores deben permanecer constantes para que sea posible calcular el retraso.
Ejemplo:

Al intentar ejecutar la siguiente operación aparece lo siguiente.

update vuelo set horaPrevistaSalida=1111 where id=11112;

ERROR at line 1:
ORA-20001: Campos no modificables
ORA-06512: at "RUEDINES.MODIFHORA", line 2
ORA-04088: error during execution of trigger '
RUEDINES.MODIFHORA'

3) Este trigger se encarga de que siempre que se modifique el retrasoAerolinea, el retrasoSeguridad o el retrasoMeteorologico en la tabla RETRASOS, la suma de los tres siga siendo inferior al retraso total.
Ejemplo:

Al intentar ejecutar la siguiente operación aparece lo siguiente.

update retrasos set retrasoAerolinea=15 where id=48983;

ERROR at line 1:
ORA-20002: La suma de los retrasos específicos no puede ser superior al retraso total
ORA-06512: at "RUEDINES.RETESPEC", line 4
ORA-04088: error during execution of trigger 'RUEDINES.RETESPEC'

DATOS SOBRE REUNIONES

La tercera parte de la práctica fue realizada en solitario por Jorge Sanz a finales de agosto y principios de septiembre. En total se emplearon alrededor de 15 horas.