**a. Dar los nombres de los aeropuertos por los cuales hayan pasado pasajeros de nacionalidad japonesa y de sexo femenino.**

El primer select trae los aeropuertos desde los cuales despegaron pasajeras japonesas, y el segundo select trae los aeropuertos donde aterrizaron.

SELECT a.nombre, a.codigo\_internacional FROM ((((pasajeros p

INNER JOIN pasajeros\_vuelos pv ON p.id = pv.id\_pasajero)

INNER JOIN vuelos v ON pv.id\_vuelo = v.id)

INNER JOIN programas\_de\_vuelo pdv ON v.id\_programa = pdv.id\_programa)

INNER JOIN aeropuertos a ON pdv.origen = a.codigo\_internacional)

WHERE p.nacionalidad = 'JAPON' AND p.sexo = 'F'

UNION

SELECT a.nombre, a.codigo\_internacional FROM ((((pasajeros p

INNER JOIN pasajeros\_vuelos pv ON p.id = pv.id\_pasajero)

INNER JOIN vuelos v ON pv.id\_vuelo = v.id)

INNER JOIN programas\_de\_vuelo pdv ON v.id\_programa = pdv.id\_programa)

INNER JOIN aeropuertos a ON pdv.destino = a.codigo\_internacional)

WHERE p.nacionalidad = 'JAPON' AND p.sexo = 'F'

**b. ¿A cuántos kilómetros de su casco urbano se encuentran los aeropuertos por los cuales despegaron vuelos sin plazas vacías (completos) y que pertenecen a programas de vuelo de la aerolínea "Aerolineas Argentinas"?**

Primero se agrupan los pasajeros por vuelo, y se usa la función agregada COUNT para saber la cantidad total de pasajeros por vuelo. Luego se filtran los vuelos donde ese total es igual a la capacidad máxima del modelo de avión usado.

SELECT count(\*) as total\_pasajeros, m.capacidad, pavu.id\_vuelo

FROM (((((pasajeros\_vuelos pavu

INNER JOIN vuelos v ON pavu.id\_vuelo = v.id)

INNER JOIN programas\_de\_vuelo pv ON pv.id\_programa = v.id\_programa )

INNER JOIN aviones a ON v.id\_avion = a.id\_avion)

INNER JOIN modelos m ON m.id\_modelo = a.id\_modelo)

INNER JOIN aeropuertos ap ON ap.codigo\_internacional = pv.origen)

WHERE pv.id\_aerolinea = 'Aerolineas Argentinas'

GROUP BY pavu.id\_vuelo, m.capacidad

HAVING count(\*) = m.capacidad

**c. Determinar la cantidad de pistas que existen por provincia.**

SELECT COUNT(1) AS nro\_pistas, a.provincia

FROM pistas p

INNER JOIN aeropuertos a ON p.id\_aeropuerto = a.codigo\_internacional

GROUP BY a.provincia

**d. Determinar la cantidad de controles que se le hayan efectuado a los aviones modelo Boeing 747 durante el 2013-2014, para aquellos aviones que hayan obtenido un puntaje promedio mayor a 7.**

Primero se seleccionan los controles filtrando por modelo de avion. Luego se agrupa por tipo de test, y para cada grupo de tests se cuentan los controles realizados, filtrando por promedio.

SELECT COUNT(\*) as cantidad\_controles, t.nombre

FROM ((controles c

INNER JOIN aviones a ON c.id\_avion = a.id\_avion AND c.id\_modelo = a.id\_modelo)

INNER JOIN tests t ON t.id\_test = c.id\_test)

WHERE c.id\_modelo = 747 AND YEAR(c.fecha) BETWEEN 2013 AND 2014

GROUP BY t.id\_test, t.nombre

HAVING AVG(c.puntaje) > 7

**e. Determinar cuáles fueron los aeropuertos que hayan obtenido el peor puntaje entre los controles que se realizaron durante el año 2014 a los aviones modelo Boeing 747**

Primero (líneas 07-11) se encuentran los peores puntajes por aeropuerto, para los aviones del modelo requerido. Luego (línea 6) se selecciona el mínimo de todos esos mínimos, es decir, el peor puntaje absoluto. Luego (líneas 1-6) se agrupan los controles por aeropuerto, y nos quedamos con los aeropuertos cuyo peor puntaje sea igual al mínimo absoluto encontrado anteriormente.

1 SELECT MIN(c.puntaje) as peor\_puntaje, c.id\_aeropuerto AS aeropuerto

2 FROM (controles c

3 INNER JOIN aviones a ON c.id\_avion = a.id\_avion AND c.id\_modelo = a.id\_modelo)

4 WHERE a.id\_modelo = 747 AND YEAR(c.fecha) = 2014

5 GROUP BY c.id\_aeropuerto

6 HAVING MIN(c.puntaje) = (SELECT MIN(puntajes.peores) FROM

7 (SELECT MIN(c.puntaje) AS peores

8 FROM (controles c

9 INNER JOIN aviones a ON c.id\_avion = a.id\_avion

AND c.id\_modelo = a.id\_modelo)

10 WHERE a.id\_modelo = 747 AND YEAR(c.fecha)= 2014

11 GROUP BY c.id\_aeropuerto) puntajes)

**f. Pensar 3 consultas**

**f.i. Encontrar a los pasajeros que hayan hecho vuelos inter-provinciales (para ofrecerles descuento por viajeros frecuentes).**

SELECT pavu.id\_pasajero, p.nombre

FROM (((((pasajeros\_vuelos pavu

INNER JOIN pasajeros p ON p.id = pavu.id\_pasajero)

INNER JOIN vuelos v ON pavu.id\_vuelo = v.id)

INNER JOIN programas\_de\_vuelo pv ON pv.id\_programa = v.id\_programa)

INNER JOIN aeropuertos a\_origen ON a\_origen.codigo\_internacional = pv.origen)

INNER JOIN aeropuertos a\_destino ON a\_destino.codigo\_internacional = pv.destino)

WHERE a\_origen.provincia <> a\_destino.provincia

GROUP BY pavu.id\_pasajero, p.nombre

**f.ii. Armar un ranking de los modelos de avión que pesen menos de 200 toneladas y que más se hayan usado en vuelos realizados durante el año pasado.**

SELECT COUNT(\*) AS cantidad\_de\_vuelos, a.id\_modelo AS modelo, a.id\_avion AS avion

FROM ((vuelos v

INNER JOIN aviones a ON v.id\_avion = a.id\_avion AND v.id\_modelo = a.id\_modelo)

INNER JOIN modelos m ON m.id\_modelo = a.id\_modelo)

WHERE m.peso\_toneladas <= 200

GROUP BY a.id\_avion, a.id\_modelo

ORDER BY COUNT(\*) DESC

**f.iii. Listar los vuelos en los que haya viajado el (o la) más joven de los viajeros más frecuentes.**

Primero se cuentan los vuelos por pasajero (líneas 9-12); se encuentra la cantidad máxima de vuelos entre ellos (línea 8); nos quedamos con los pasajeros que tengan la cantidad de vuelos máxima (líneas 5-8); a esos pasajeros los ordenamos por fecha de nacimiento y encontramos la máxima (i.e., el más joven) (líneas 3-13); finalmente, se buscan los vuelos de los pasajeros que tengan esa fecha de nacimiento más reciente (porque podría haber más de uno que haya nacido el mismo día).

1. SELECT pv.id\_vuelo AS vuelo, p.nombre AS viajero

2. FROM (pasajeros\_vuelos pv INNER JOIN pasajeros p ON p.id = pv.id\_pasajero)

3. WHERE p.fecha\_nacimiento = (select max(p.fecha\_nacimiento) from

4. (SELECT p.id, p.fecha\_nacimiento FROM

5. (SELECT COUNT(\*) AS cant\_vuelos,

p.fecha\_nacimiento, p.id

6. FROM (pasajeros\_vuelos pv INNER JOIN pasajeros p

ON pv.id\_pasajero = p.id)

7. GROUP BY p.id, p.fecha\_nacimiento

8. HAVING COUNT(\*) = (SELECT MAX(cant\_vuelos) FROM

9. (SELECT COUNT(\*) AS cant\_vuelos, p.id

FROM

10. (pasajeros\_vuelos pv INNER JOIN pasajeros

p ON pv.id\_pasajero = p.id)

11. GROUP BY p.id, p.fecha\_nacimiento

12. ORDER BY COUNT(\*) DESC)))

13. ORDER BY p.fecha\_nacimiento, p.id DESC))