

Consultas SQL resueltas
Base de Datos
FIUBA

Pablo Musumeci
pmusumeci@fi.uba.ar

Revisado por:

Florencia Bosch
Martín Buchwald

10 de febrero de 2014

1. Sueldos y pilotos

1.1. Esquemas

Vuelo(VueloNro, Desde, HAcia, Distancia)
Aeronave (aID, aNombre, Rango)
Certificado(eID, aID)
Empleado(eID, eNombre, Sueldo)

1.2. Enunciado

La relacion empleado contiene datos de todos los empleados de la compañía. En la relacion certificado solo figuran los pilotos certificados para volar una determinada aeronave.

Listar los nombres y sueldos de los empleados que no son pilotos pero que su sueldo es superior al promedio de los pilotos.

1.3. Consulta

```
SELECT e.enombre, e.sueldo
FROM empleado e
WHERE
    e.eid NOT IN (
        SELECT c.eid
        FROM certificado c
    )
AND e.sueldo > (
    SELECT AVG(e2.sueldo)
    FROM empleado e2
    WHERE e2.eid IN (
        SELECT c2.eid
        FROM certificado c2
    )
);
```

1.4. Enunciado

Listar los nombres de los pilotos que pueden volar aeronaves con rango de crucero mayor a 5000 millas pero que no están certificados en ningún avión Boeing.

```
SELECT DISTINCT e.eNombre
FROM Empleado e, Aeronave a, Certificado c
WHERE e.eID = c.eID AND a.aID = c.aID
AND a.rango > 5000 AND NOT EXISTS (
    SELECT *
    FROM Certificado c2, Aeronave a2
    WHERE a2.aID = c2.aID AND e.eID = c2.eID
    AND a2.aNombre = "BOEING"
```

```
);
```

1.5. Enunciado

Listar los nombres de los pilotos que pueden **solo** vuelen aeronaves con rango de crucero mayor a 2000 millas y que esten cerficiados en al menos un avión Boeing.

```
SELECT DISTINCT e.eNombre
FROM Empleado e,
WHERE EXISTS(
    SELECT *
    FROM Certificado c2, Aeronave a2
    WHERE a2.aID = c2.aID AND e.eID = c2.eID
    AND a2.aNombre = "BOEING"
)
AND NOT EXISTS(
    SELECT *
    FROM Certificado c3, Aeronave a3
    WHERE a3.aID = c3.aID AND e.eID = c3.eID
    AND a3.rango <= 2000
);
```

2. Partes de aeronaves

2.1. Esquemas

InfoParte(nroParte, nombreParte, subparteDe)

UsadoEn(nroParte, tipoAvion, cantUsada)

EnStock(nroParte, lugar, cantidad)

2.2. Enunciado

Para cuales partes el total en stock es al menos la cantidad usada por un B737 y por que monto el total excede la cantidad usada

2.3. Consulta

```
SELECT s.NroParte, s.Cantidad - e.CantUsada
FROM EnStock s, UsadoEn e,
WHERE ( s.NroParte = e.NroParte
    AND e.TipoAvion = "B737"
    AND s.Cantidad => e.CantUsada );
```

3. Estudiantes

3.1. Esquemas

Estudiantes(eNro, eNombre, carrera, anioCursada, edad)
Curso(cNombre, horario, aula, pID)
Cursa(eNro, cNombre)
Profesor(pID, pNombre, departamento)

3.2. Enunciado

Hallar los nombres de los estudiantes que cursan el maximo numero de cursos.

3.3. Consulta

```
SELECT e.eNombre
FROM Estudiantes e, Cursa c
WHERE e.eNro = c.eNro
GROUP BY e.eNro
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*)
    FROM Cursa c2
    GROUP BY c2.eNro
);
```

3.4. Enunciado

Escribir una consulta SQL que permita, para cada valor de edad que aparece en Estudiante, hallar el valor de año que aparece más frecuentemente. Por ejemplo, si hay más estudiantes de 20 años de edad en 2do año que en cualquier otro año para estudiantes de 20 años, listar el par (20,2do). Nota: No deben listarse duplicados y la consulta debe ser tan concisa como resulte posible.

3.5. Consulta

```
SELECT DISTINCT e.edad, e.anioCursada
FROM Estudiantes e
GROUP BY e.edad, e.anioCursada
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*) FROM Estudiantes e2
    WHERE e2.edad = e1.edad
    GROUP BY e2.anioCursada
);
```

3.6. Enunciado

Hallar los nombres de todos los profesores que enseñan en todas las aulas en las que se dicta algún curso.

3.7. Consulta

```
SELECT p.pNombre
FROM Profesor p
WHERE NOT EXISTS
(
    (SELECT Aula
     FROM Curso c1
     WHERE NO EXISTS
        (SELECT *
         FROM Curso c2
         WHERE c1.aula = c2.aula AND c2.pID = p.pID)
    )
);
```

4. Hipódromo

4.1. Esquemas

Encuentros(nroEncuentro, fecha, nombreHipo)

Carreras(nroEncuentro, nroCarrera, tipoPista, estadoPista)

Caballos(nombreCaballo, fechaNac, pelaje)

Corredores(nroEncuentro, nroCarrera, nombreCaballo, nroPosicion)

4.2. Enunciado

Obtener los datos de aquellos caballos que hayan conseguido llegar a la meta en más de 3 oportunidades.

4.3. Consulta

```
SELECT c.nombreCaballo, c.fechaNac, c.pelaje
FROM Caballos c, Corredores corr
WHERE c.nombreCaballo = corr.nombreCaballo
      AND corr.nroPosicion <= 3
GROUP BY corr.nombreCaballo
HAVING COUNT(*) > 3;
```

4.4. Enunciado

Obtener la cantidad de veces que, en carreras de *San Isidrio*, hayan competido los caballos *Matuk* y *Laca* juntos.

4.5. Consulta

```
SELECT COUNT(*)
FROM Corredores c1, Corredores c2
WHERE c1.nombreCaballo = "Matuk" AND c2.nombreCaballo = "Laca"
AND c1.nroCarrera = c2.nroCarrera
AND c1.nroEncuentro = c2.nroEncuentro
AND c1.nroEncuentro IN (
    SELECT e.nroEncuentro
    FROM Encuentros e
    WHERE e.nombreHipo = "San Isidrio"
);
```

5. Apendice

5.1. Valores Nulos

Las comparaciones en la presencia de valores nulos, devuelven uno de los 3 valores posibles: **True**, **False** y **Unknown**

- Comparación de un valor cualquiera con NULL, el resultado es **Unknown**.
- Comparación de dos valores NULL, el resultado es **Unknown**.
- True **OR** Unknown evalua True
- False **OR** Unknown evalua Unknown
- True **AND** Unknown evalua Unknown
- False **AND** Unknown evalua False
- Operación aritmética con un resultado NULL evalua NULL
- Las funciones MAX, SUM y AVG ignoran los valores nulos, a menos que todos los valores de la columna involucrada sean nulos, en este último caso el resultado es nulo.
- La función COUNT cuenta la cantidad de filas independientemente de su contenido; pero si la columna a la que se aplica COUNT solo contiene valores nulos, el resultado es cero. Si la columna contiene valores nulos, y otros valores, el resultado es la cantidad de valores ignorando los nulos.