

Unidad de Trabajo 4  
Realización de consultas

# Subconsultas

IES Palomeras Vallecas

Curso 2020/2021

Profesor: Alberto Ruiz

# Subconsulta

---

- Una subconsulta es una sentencia SELECT que aparece dentro de otra expresión SELECT, a la que llamamos consulta principal
- Son necesarias cuando para realizar una consulta necesitamos los datos devueltos por otra
- Se corresponde con frases subordinadas...
  - Obtener el nombre del empleado **que** menos cobra
  - Obtener nombre y continente del país **que** más población tiene
- ... pero no siempre aparecerá la conjunción "que":
  - Obtener el nombre del jugador más alto (**que** es más alto)
  - Obtener el DNI de Alberto (del profesor **que** se llama Alberto)

# Ejemplo

 Empleados : Tabla

	Id	Nombre	Apellido	Edad	Departamento	Oficio
	1	Álvaro	García	35	20	Analista
	2	Pedro	López	30	20	Analista
	3	Julio	Álvarez	45	20	Programador
	4	Ana	Pérez	37	20	Jefe Proyecto
	5	Susana	Fernández	48	30	Director
	6	Ángela	González	29	30	Programador
	7	Sergio	López	40	40	Jefe Proyecto
	8	Olga	Gil	31	50	Analista

*Obtener el nombre de los empleados que tienen el mismo oficio **que Gil***

# Por separado es fácil:

- *Averiguar el oficio de Gil:*

```
SELECT oficio  
FROM empleados  
WHERE apellido = 'Gil';
```



oficio
Analista

- *Averiguar el nombre de los empleados que son analistas:*

```
SELECT nombre  
FROM empleados  
WHERE oficio = 'Analista';
```



nombre
Álvaro
Pedro
Olga

# Combinamos las dos:

- Averiguar el nombre de los empleados que tienen el mismo oficio que Gil*

```
SELECT nombre  
FROM empleados  
WHERE oficio =  
    (SELECT oficio  
     FROM empleados  
     WHERE apellido = 'Gil');
```



	nombre
	Álvaro
	Pedro
	Olga

# ¿Cómo se resuelven?

Igual que las operaciones matemáticas

$$50 - (4 * 5) \longrightarrow 50 - 20 \longrightarrow 30$$

SELECT nombre  
FROM empleados  
WHERE oficio =

SELECT nombre  
FROM empleados  
WHERE oficio = 'Analista'

(SELECT oficio  
FROM empleados  
WHERE apellido='Gil');



	nombre
	Álvaro
	Pedro
	Olga

# ¿Cómo se usan?

---

- Se colocan **siempre** en el WHERE (o en el HAVING dentro de un GROUP BY)

```
SELECT nombre  
FROM empleados  
WHERE oficio =  
    (SELECT oficio  
     FROM empleados  
     WHERE apellido = 'Gil');
```

# Operadores de comparación

---

- Si el resultado de la subconsulta es numérico o de texto, podemos usar los operadores que ya conocemos:

< > <= >= = <> !=

SELECT pais, renta

FROM paises

WHERE renta >

(SELECT AVG(renta) FROM paises)



# Operadores de conjuntos

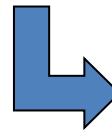
- *Obtener el nombre de los empleados cuyos oficios sean los **que** realizan los empleados del departamento 20*
- Si la subconsulta devuelve **varios** valores, ya no sirven los operadores de comparación

```
SELECT nombre  
FROM empleados  
WHERE oficio =
```

Error Code: 1242. Subquery returns more than 1 row

¡ERROR!


```
(SELECT oficio  
FROM empleados  
WHERE departamento = '20');
```



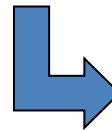
	oficio
	Analista
	Analista
	Programador
	Jefe Proyecto

# Operadores de conjuntos

- En este caso utilizamos el operador de **pertenencia a conjunto, IN...**
  - O de no pertenencia, NOT IN

```
SELECT nombre  
FROM empleados  
WHERE oficio IN   
      (SELECT oficio  
      FROM empleados  
      WHERE departamento = '20');
```

“empleados cuyo oficio  
esté incluido en los  
resultados de la  
subconsulta”



	oficio
	Analista
	Analista
	Programador
	Jefe Proyecto

# Operadores de conjuntos

## Operadores disponibles:

TIPO	OPERADOR
Operadores relacionales	> , >= , < , <= , <> , =
Predicados cuantificados	ANY, SOME, ALL
Predicado de inclusión	IN
Predicado de existencia	EXISTS

*ALL: Todos, ANY/SOME: Alguno*

SELECT pais, renta

FROM paises

WHERE renta > ALL

(SELECT renta FROM paises

WHERE continente="Europa")



"países cuya renta sea mayor que la de todos los países europeos"

# Ejemplos

---

- Muestra los países cuya renta es mayor que la media de renta de todos los países.
  - Resuelve primero la subconsulta: ¿cuál es la media de renta de todos los países?
  - `SELECT pais, renta FROM paises`  
`WHERE renta >`  
`(SELECT AVG(renta)`  
`FROM paises)`

*En MySQL puedes seleccionar con el ratón únicamente la subconsulta y ejecutarla pulsando el icono del rayo, para tener claro qué resultado da*

# Ejemplos

---

- Obtener el nombre y la población del país más poblado.
  - Resuelve primero la subconsulta: ¿cuál es la población del país más poblado?
  - `SELECT Name, Population`  
`FROM country`  
`WHERE Population =`  
`(SELECT MAX(Population)`  
`FROM Country)`

*En este caso no pasaría nada si hubiese varios países con la misma población máxima, ya que el valor máximo sigue siendo uno y por tanto la subconsulta devuelve 1 valor*

# Ejemplos

---

- Obtener el nombre y el equipo de los jugadores que juegan en la división South East
  - Resuelve primero la subconsulta: ¿cuáles son los equipos que juegan en la división South East?
  - `SELECT Nombre, NombreEquipo`  
`FROM jugadores`  
`WHERE NombreEquipo IN`  
`(SELECT Nombre`  
`FROM equipos`  
`WHERE división="SouthEast")`

# Ejemplos

---

- ¡A veces hay más de una forma de realizar una consulta!
- Obtener el nombre y el equipo de los jugadores que juegan en la división South East
  - Este último ejemplo se podía resolver también con consulta multitable:
  - `SELECT jugadores.Nombre, jugadores.NombreEquipo  
FROM jugadores, equipos  
WHERE jugadores.NombreEquipo=equipos.Nombre  
AND equipos.División = "SouthEast"`

# Ejemplos

---

- Las consultas "en negativo" suelen requerir soluciones más complicadas, como las subconsultas
- *Equipos que **no** tengan jugadores españoles:*

```
SELECT equipos.nombre  
FROM equipos  
WHERE equipos.nombre NOT IN  
    (SELECT NombreEquipo  
     FROM Jugadores  
     WHERE procedencia="Spain")
```



# Subconsultas correlacionadas

---

- Imagina esta consulta: "códigos de jugadores que hayan duplicado la media de puntos por partido *del jugador con código 66*"

```
SELECT DISTINCT estadísticas.CódigoJugador
FROM estadísticas
WHERE estadísticas.PuntosPorPartido >
      2*(SELECT AVG(PuntosPorPartido)
         FROM estadísticas
         WHERE CódigoJugador = 66);
```

# Subconsultas correlacionadas

---

- Ahora queremos: "códigos de jugadores que hayan duplicado **su propia** media de puntos por partido

```
SELECT DISTINCT estadísticas.CódigoJugador
FROM estadísticas
WHERE estadísticas.PuntosPorPartido >
      2*(SELECT AVG(PuntosPorPartido)
        FROM estadísticas
        WHERE CódigoJugador = estadísticas.CódigoJugador);
```

Son más complicadas de entender, porque la subconsulta no es independiente: en la subconsulta se hace relación a la tabla de la consulta original

- Buscamos a un jugador cuyos puntos por partido hayan sido el doble de la media de puntos por partido de...
  - Antes: el jugador de código 66.
  - Ahora: él mismo → **aprendemos a hacer referencia a "él mismo"**

# Subconsultas correlacionadas

---

- Módulos que aparecen en más de dos ciclos formativos:

```
SELECT Nombre
```

```
FROM Módulo
```

```
WHERE 2 < (
```

```
    SELECT COUNT(CódigoCiclo)
```

```
    FROM Pertenece
```

```
    WHERE Módulo.Código = Pertenece.CódigoMódulo)
```

# Subconsultas correlacionadas

---

- Mecánicos que han reparado más de tres coches:

SELECT \*

FROM Mecánicos

WHERE 3 < (

SELECT COUNT(DISTINCT Matrícula)

FROM Revisiones

WHERE Mecánicos.DNI = Revisiones.DNIMecánico)

*En este caso la misma matrícula puede aparecer muchas veces en distintas revisiones: ponemos DISTINCT*

# Subconsultas correlacionadas

---

- Con EXISTS / NOT EXISTS nos quedamos con los valores con los que la subconsulta dé / no dé resultados:

```
SELECT equipos.Nombre  
FROM equipos  
WHERE NOT EXISTS  
      (SELECT *  
       FROM jugadores  
       WHERE equipos.Nombre = jugadores.NombreEquipo  
             AND jugadores.Procedencia="Spain")
```

¿Cómo funcionan? **Por cada** uno de los resultados de la consulta principal (equipos) se realiza la subconsulta y se ve si devuelve o no resultados

# Subconsultas correlacionadas

---

- Mecánicos que **no** han revisado el coche 1234ABC:

```
SELECT *
```

```
FROM Mecánicos
```

```
WHERE NOT EXISTS (
```

```
    SELECT *
```

```
    FROM Revisiones
```

```
    WHERE Mecánicos.DNI = Revisiones.DNIMecánico
```

```
        AND Revisiones.Matrícula = "1234ABC")
```

*Las consultas de EXISTS / NOT EXISTS suelen poder formularse con IN / NOT IN, que resulta más intuitivo*

# Subconsultas correlacionadas

---

- Empleados que tienen un salario mayor que la media de su departamento:

```
SELECT *  
FROM Empleados E1  
WHERE Salario > (  
    SELECT AVG(Salario)  
    FROM Empleados E2  
    WHERE E1.Departamento = E2.Departamento)
```

*Recuerda: por cada uno de los empleados E1, se realiza la subconsulta (media de todos los empleados cuyo departamento es el de E1)*

# Subconsultas correlacionadas

---

- Empleados que tienen al menos una persona a su cargo:

```
SELECT *  
FROM Empleados E1  
WHERE EXISTS (  
    SELECT *  
    FROM Empleados E2  
    WHERE E1.DNI = E2.DNISuperior)
```

*Por cada uno de los empleados E1, vemos si existe algún empleado que tenga a E1 como superior*



# Subconsultas correlacionadas

---

- Departamentos sin empleados:

```
SELECT *  
FROM Departamentos  
WHERE NOT EXISTS (  
    SELECT *  
    FROM Empleados  
    WHERE Departamento.Número = Empleados.NumDepto)
```

*Por cada uno de los departamentos, vemos si existe algún empleado asignado al mismo*