

# Práctica II

SQL: Lenguaje de Consulta

**Grado en  
Ingeniería  
Informática**



**Bases de  
Datos**

**2022/23**

Departamento de Tecnologías de la Información  
**Universidad de Huelva**

## Objetivo

- ☐ Saber utilizar, de forma eficiente, el lenguaje SQL para consultar bases de datos relacionales

## Duración

- ☐ 5 clases

## Bibliografía

- ☐ Capítulo 8 [Elmasri 02]
- ☐ Capítulo 3 [Silberschatz 06]
- ☐ Capítulo 5 [Connolly 05]

## SQL: Lenguaje de Consulta

### Tema 4 (parte 2/3): Sesión 1

#### Resumen de conceptos

### Consultas básicas

- ❑ Forma básica de instrucción SELECT:

**SELECT**    <lista de atributos>  
**FROM**     <lista de tablas>  
**WHERE**    <condición>

- <lista de atributos>: nombres de los atributos cuyos valores va a obtener la consulta
  - <lista de tablas>: nombres de las tablas requeridas para procesar la consulta
  - <condición>: expresión lógica que identifica las tuplas que obtendrá la consulta
- 
- ❑ Es habitual asignar **alias** a los nombres de las tablas y **cualificar** los atributos, sobre todo cuando se realizan consultas a varias tablas, con objeto de eliminar la ambigüedad y facilitar la escritura
  - ❑ Pueden **renombrarse** los nombres de los atributos en el resultado, escribiendo en la cláusula SELECT el nuevo nombre justo detrás del atributo o bien utilizando la sentencia AS

- ☐ Se puede **omitir** la cláusula WHERE para indicar una selección de tuplas incondicional
- ☐ Para obtener los valores de **todos** los atributos de las tuplas seleccionadas se utiliza el carácter asterisco (\*)
- ☐ SQL trata las tablas como **multiconjuntos**  $\Rightarrow$  no elimina las tuplas repetidas del resultado
- ☐ Hay que indicárselo explícitamente mediante la cláusula **DISTINCT**.
- ☐ Se pueden **calcular** valores y mostrarlos en el resultado de la consulta, mediante operaciones aritméticas realizadas sobre los campos
- ☐ La cláusula ORDER BY y las palabras reservadas DESC y ASC permiten mostrar el resultado de la consulta de forma **ordenada** según los campos que se elijan como criterio de ordenación
  - Por defecto, ordena **ascendentemente**
- ☐ BETWEEN: compara un valor **entre** otros dos valores
- ☐ IN: comprueba que un valor **pertenece** a un conjunto valores.

## Patrones de búsqueda

- ☐ En atributos de tipo cadena, la selección se puede realizar utilizando un **patrón** de búsqueda
- ☐ El operador de comparación es LIKE (**como**)
- ☐ Para la descripción de **patrones** se utilizan dos caracteres especiales:
  - % (tanto por ciento): sustituye a un número arbitrario de caracteres
  - \_ (subrayado): sustituye a un solo carácter
- ☐ NOT delante del operador LIKE: **complementario** de la búsqueda

## Definición y Manejo de variables de tipo FECHA

### FUNCIONES DE FECHA

- **SYSDATE:** Devuelve la fecha actual del sistema
- **CURRENT\_DATE:** Devuelve la fecha del sistema en la zona horaria de la sesión
- **SYSTIMESTAMP:** Devuelve la fecha actual del sistema en el formato del tipo de datos TIMESTAMP

### FUNCIONES DE MANIPULACIÓN DE FECHAS

**TO\_CHAR:** Convierte una fecha en cadena de caracteres

**TO\_DATE:** Convierte en fecha una cadena de caracteres

**BETWEEN:** Hace comparación entre fechas, aplicado sobre un atributo de tipo fecha (DATE)

**IN:** Comprueba el valor de un atributo de tipo fecha dentro de un conjunto de valores

**EXTRACT:** Selecciona partes de los valores de un atributo de tipo fecha (DATE y TIMESTAMP)

El esquema completo de la base de datos “*MoviFacturas*” es el siguiente:

<b>COMPAÑIA</b> (cif, nombre, web) CP: cif Único: nombre VNN: nombre	<b>TARIFA</b> (tarifa,compañia,descripcion,coste) CP: (tarifa,compañia) CAj: compañía → COMPAÑIA (cif) VNN: coste
<b>LLAMADA</b> (tf_origen,tf_destino,fecha_hora,duracion) CP: (tf_origen,fecha_hora) CAj: tf_origen → TELEFONO(numero) CAj: tf_destino→ TELEFONO(numero) VNN: duracion VNN: tf_destino,duracion Unico: (tf_destino, fecha_hora)	<b>TELEFONO</b> (numero,f_contrato,tipo,puntos,compañia,tarifa,cliente) CP: numero CAj: (tarifa,compañia) → TARIFA CAj: compañía → COMPAÑIA (cif) CAj: cliente → CLIENTE (dni) VNN: compañía, tarifa, cliente
<b>CLIENTE</b> (dni, nombre, f_nac, direccion, cp, ciudad, provincia) CP: dni VNN: nombre	

Este esquema está creado en la base de datos “**ETSI**” y su propietario es el usuario “**mf**”. Todos vosotros tenéis permiso de lectura (SELECT) sobre las tablas del esquema. Puesto que el propietario es otro usuario, en las consultas debéis anteponer el nombre de las tablas al nombre del propietario.

**Ejemplo:** SELECT \*  
FROM mf.cliente  
WHERE provincia = 'Huelva';

**Sobre la BD de MoviFacturas, resolver las siguientes consultas:**

Consultas Sesión 1

- S1.1 Obtener el nombre de las compañías cuya dirección web contenga la cadena 'et' y acabe en 'com'.
- S1.2 Obtener el nombre y dirección de los clientes nacidos en 1973 o 1985 y cuyo código postal comience por 15, ordenado ascendentemente por el nombre y, en caso de igualdad, descendientemente por la dirección
- S1.3 Obtener el teléfono de destino de las llamadas realizadas desde el número "666010101", en el año 2006.
- S1.4 Obtener los números de teléfono que han llamado alguna vez al "666010101", entre las 10:00 y las 12:00
- S1.5 Obtener las tarifas distintas de aquellos clientes cuyo dni contiene el número 2 y tienen entre 10000 y 20000 puntos, pero que son de contrato, no tarjeta de prepago.
- S1.6 Obtener los números de teléfonos junto a su tarifa, cuya fecha de contrato es en mayo (cualquier día y año), con tarifa distinta de joven, y que acaban en 9, ordenados por puntos descendientemente.
- S1.7 Obtener los distintos teléfonos a los que se han llamado desde el '654345345' durante el mes de octubre y noviembre, cuyas llamadas hayan tenido una duración superior a los 250 segundos.
- S1.8 Obtener los nombres de los clientes que nacieron entre 1970 y 1985 y que pertenezcan a la provincia de Huelva, ordenados ascendentemente por ciudad y descendientemente por provincia.



## Consultas a más de una tabla

### Tema 4 (parte 2/3): Sesión 2

#### Resumen de conceptos

- ❑ A partir de la versión SQL2, la concatenación de tablas se realiza utilizando la cláusula **INNER JOIN** e indicando explícitamente la condición de relación entre las tablas.
- ❑ Si coincide el nombre del atributo en ambas tablas, se puede utilizar la cláusula **USING**, indicando el atributo de combinación
- ❑ La cláusula **NATURAL JOIN** se utiliza para combinar tablas según todos los atributos cuyos nombre coincidan en ambas tablas

EI-8. Obtener los nombres de las asignaturas junto con el nombre del profesor responsable

```
SELECT A.nombre as NomAsig, P.nombre as NomProf  
FROM EI.ASIGNATURA A INNER JOIN EI.PROFESOR P ON A.prof = P.nPr;
```

EI-9. Obtener los números de los alumnos que se han matriculado en Bases de Datos I en el curso 2002-03

```
SELECT alum  
FROM EI.MATRICULA M INNER JOIN EI.ASIGNATURA A USING (idAsig)  
WHERE A.nombre = 'Bases de Datos I' AND año = 2002;
```

- ❑ Operaciones de la **Teoría de Conjuntos**:
  - unión (UNION)
  - intersección (INTERSECT)
  - diferencia (MINUS)
- ❑ Las tres operaciones tratan a las tablas como conjuntos, es decir, **eliminan** las tuplas repetidas del resultado
- ❑ Las tablas deben ser **compatibles**: mismo número de atributos y mismos tipos de datos
- ❑ Para preguntar por atributos con valores nulos, SQL utiliza las cláusulas **IS NULL** (es nulo) e **IS NOT NULL** (no es nulo) en lugar de usar los operadores = y ≠

**Sobre la BD de MoviFacturas, resolver las siguientes consultas:**

Consultas Sesión 2

S2.1 Mostrar el código y coste de las tarifas junto con el nombre de la compañía que las ofrecen, de aquellas tarifas cuya descripción indique que otras personas deben estar también en la misma compañía

S2.2 Nombre y número de teléfonos de aquellos abonados con contrato que tienen tarifas inferiores a 0,20 €.

S2.3 Obtener el código de las tarifas, el nombre de las compañías, los números de teléfono y los puntos, de aquellos teléfonos que se contrataron en el año 2006 y que hayan obtenido más de 200 puntos.

S2.4 Obtener los números de teléfono (origen y destino), así como el tipo de contrato, de los clientes que alguna vez hablaron por teléfono entre las 8 y las 10 de la mañana

S2.5 Interesa conocer los nombres y números de teléfono de los clientes (origen y destino) que, perteneciendo a compañías distintas, mantuvieron llamadas que superaron los 15 minutos. Se desea conocer, también, la fecha y la hora de dichas llamadas así como la duración de esas llamadas

## Consultas anidadas o subconsultas

### Tema 4 (parte 2/3): Sesión 3

#### Resumen de conceptos

- ☐ A veces resulta necesario seleccionar valores de la base de datos para que actúen en la condición de comparación
- ☐ La forma más simple de realizar esta operación es mediante las **consultas anidadas** o **subconsultas**:  
“Consultas *SELECT* completas dentro de la cláusula *WHERE* de otra consulta (**consulta exterior**)”
- ☐ Podemos utilizar los operadores de comparación (=, >, >=, <, <= y <>) para comparar un valor de la tupla que se está examinando con un único valor producido por la subconsulta
- ☐ Cuando se utilicen operadores de comparación, la subconsulta debe devolver un único valor

- ☐ El operador IN se utiliza para comparar la **pertenencia** de un valor a un conjunto de valores
- ☐ La **no pertenencia** de un valor a un conjunto se expresa mediante NOT IN
- ☐ El resultado de la subconsulta puede contener más de un valor
  
- ☐ Operador (=ANY): devuelve verdadero si el valor que se está comparando es igual a **algún** valor del conjunto (equivalente a IN)
- ☐ El operador = se puede sustituir por cualquier otro operador de comparación (>, >=, <, <=, <>)
  
- ☐ Operador (=ALL): devuelve verdadero si el valor que se está comparando es igual a **todos** los valores devueltos por la subconsulta
- ☐ El operador = se puede sustituir por cualquier otro operador de comparación (>, >=, <, <=, <>)

**Sobre la BD de MoviFacturas, resolver las siguientes consultas:**

Consultas Sesión 3

S3.1 Obtener la fecha (día-mes-año) en la que se realizó la llamada de mayor duración

S3.2 Obtener el nombre de los abonados de la compañía 'Aotra' con el mismo tipo de tarifa que la del telefono "654123321"

S3.3 Mostrar, utilizando para ello una subconsulta, el número de teléfono, fecha de contrato y tipo de los abonados que han llamado a teléfonos de clientes de fuera de la provincia de La Coruña durante el mes de octubre de 2006.

S3.4 Se necesita conocer el nombre de los clientes que tienen teléfonos con tarifa "dúo" pero no "autónomos". Utiliza subconsultas para obtener la solución.

S3.5 Obtener mediante subconsultas los nombres de clientes y números de teléfono que aquellos que hicieron llamadas a teléfonos de la compañía Petafón pero no Aotra

S3.6 Nombre de los clientes de la compañía Kietostar que hicieron las llamadas de mayor duración en septiembre de 2006

S3.7 Se necesita conocer el nombre de los clientes que tienen teléfonos con fecha de contratación anterior a alguno de los teléfonos de Ramón Martínez Sabina, excluido, claro, el propio Ramón Martínez Sabina.

## La función EXISTS

### Tema 4 (parte 2/3): Sesión 4

#### Resumen de conceptos

- ❑ Cuando una consulta anidada o subconsulta puede existir por sí sola se dice que es una subconsulta independiente
- ❑ **Subconsultas correlacionadas:** subconsultas cuyos valores de ejecución dependen de la tupla que está siendo examinada en la consulta exterior
  - La consulta anidada se evalúa una sola vez para cada tupla de la consulta externa
  - La función **EXISTS** se utiliza para comprobar si el resultado de una consulta correlacionada está o no vacío, es decir, contiene o no tuplas (la función **NOT EXISTS** realiza la operación contraria):
    - Devuelve *Verdad* si la consulta anidada contiene alguna tupla
    - Devuelve *Falso* en caso contrario
- ❑ **Funcionamiento (EXISTS):**
  - Por cada tupla de la consulta externa, se evalúa la consulta interna con dicha tupla
  - En la consulta anidada (interna) se seleccionan las tuplas que cumplen la condición de la cláusula WHERE para la tupla de la consulta externa que se esté evaluando
  - Si el resultado de la consulta correlacionada devuelve alguna tupla, entonces se selecciona la tupla de la consulta externa evaluada

**Sobre la BD de MoviFacturas, resolver las siguientes consultas:**

Consultas Sesión 4

- S4.1 Utilizando consultas correlacionadas, mostrar el nombre de los abonados que han llamado el día '16/10/06'
- S4.2 Utilizando consultas correlacionadas, obtener el nombre de los abonados que han realizado llamadas de menos de 1 minuto y medio
- S4.3 Utilizando consultas correlacionadas, obtener el nombre de los abonados de la compañía 'KietoStar' que no hicieron ninguna llamada el mes de septiembre
- S4.4 Utilizando consultas correlacionadas, mostrar todos los datos de los telefonos que hayan llamado al número 654234234 pero no al 666789789
- S4.5 Utilizando consultas correlacionadas, obtener el nombre y número de teléfono de los clientes de la compañía Kietostar que no han hecho llamadas a otros teléfonos de la misma compañía



## Concatenación externa

### Tema 4 (parte 2/3): Sesión 5

#### Resumen de conceptos

- ❑ SQL, al igual que el álgebra relacional, permite la agrupación de tablas incluso si los campos de unión no son coincidentes o uno de sus valores es nulo (concatenación externa – OUTER JOIN)
  - Izquierda: LEFT OUTER JOIN
  - Derecha: RIGHT OUTER JOIN
  - Ambos: FULL OUTER JOIN
  
- ❑ Funciones de Agrupación más habituales:
  - **COUNT**: cuenta (COUNT(\*) devuelve el número de filas en el resultado de la consulta)
  - **SUM**: suma
  - **MAX**: máximo
  - **MIN**: mínimo
  - **AVG**: promedio
  
- ❑ Se puede utilizar la función COUNT para contar los valores (no nulos) de una columna en vez de las tuplas. Necesitaremos utilizar la cláusula DISTINCT para eliminar los duplicados

- ❑ Algunas veces resulta necesario utilizar estas funciones para seleccionar tuplas que cumplen alguna condición
- ❑ En este caso, se especifica una consulta anidada (correlacionada) con la función deseada y se usará dicha consulta en la cláusula WHERE de una consulta exterior

EI-24. Obtener, mediante una consulta correlacionada, los nombres de las asignaturas que tienen recomendadas 2 o más asignaturas.

```
SELECT A.nombre  
FROM EI.ASIGNATURA A  
WHERE (SELECT COUNT(*)  
        FROM EI.RECOMENDACIONES R  
        WHERE A.idAsig = R.idAsig1) >= 2;
```

- ❑ Suele ser habitual aplicar estas funciones integradas a subgrupos de tuplas de una relación, agrupadas por uno o varios atributos
- ❑ La cláusula que realiza la agrupación según unos determinados atributos es **GROUP BY** (agrupar por)
- ❑ En la cláusula SELECT es correcto que se especifiquen los atributos de agrupación (que aparezcan en la cláusula GROUP BY) así como funciones de agrupación, pero no atributos cuyos valores puedan variar para un mismo grupo

EI-26. Para cada asignatura y año académico, mostrar el nombre de la asignatura, el año, el número de alumnos que se han presentado y la nota media obtenida en la convocatoria de febrero\_junio

```
SELECT A.nombre, M.año, COUNT(feb_jun) AS Presentados, AVG (feb_jun) AS Media
FROM EI.ASIGNATURA A INNER JOIN EI.MATRICULA M USING (idAsig)
GROUP BY A.nombre, M.año;
```

- ❑ A veces ocurre que solamente queremos utilizar estas funciones con aquellos grupos de tuplas que satisfagan ciertas condiciones
- ❑ SQL proporciona la cláusula **HAVING** (que tiene), la cual aparece junto con la cláusula GROUP BY
- ❑ HAVING especifica una condición, y sólo aquellos grupos que satisfagan dicha condición, entrarán en el resultado de la consulta

EI-27. Obtener un listado con el nombre del alumno, el nombre de la asignatura y el número de veces que se ha matriculado en esa asignatura, pero sólo cuando se haya matriculado 3 o más años

```
SELECT A.nombre, ASIG.nombre, COUNT(*)  
FROM (EI.ALUMNO A INNER JOIN EI.MATRICULA M ON A.nAl = M.alum)  
     INNER JOIN EI.ASIGNATURA ASIG USING (idAsig)  
GROUP BY A.nombre, ASIG.nombre  
HAVING COUNT(*) >=3;
```

Sobre la BD de MoviFacturas, resolver las siguientes consultas:

Consultas Sesión 5

S5.1 Mostrar la compañía con la que se realizaron más llamadas durante el día 16/10/06

División

**S5.2** Obtener los números de teléfono y los nombres de los abonados que han llamado a todos los números a los que se llamó desde el 654345345 en octubre de 2006

S5.3 Mostrar el nombre de cada cliente junto con coste total de las llamadas que realiza con cada compañía. El resultado debe mostrarse ordenado descendientemente por cliente y ascendentemente por compañía

S5.4 Para cada cliente residente en la provincia de 'La Coruña', mostrar la duración de todas las llamadas realizadas a clientes residentes en 'Jaén'

S5.5 Mostrar los nombres de todos los clientes que hayan realizado más de 5 llamadas

S5.6 Mostrar el nombre de aquellos clientes que hayan contratado con cualquier compañía, de media, una tarifa superior a la tarifa media total.

S5.7 Mostrar los nombres de los clientes que hayan realizado llamadas a telefonos de la compañía 'Kietostar' por un coste menor de 100€