

1. Introducción

1.1.- ¿Qué es SQL?

1.2.- Estado actual y futuro del SQL

2. Sintaxis Básica del SQL

2.1.- Lenguaje de Definición de Datos

2.1.1.- Definición del esquema

2.1.2.- Evolución del esquema

2.2.- Lenguaje de Manipulación de Datos

2.2.1.- Actualizaciones

2.2.2.- Consultas

2.3.- Lenguaje de Control de Datos

2.3.1.- Recuperación y concurrencia

2.3.2.- Seguridad y confidencialidad

2. Sintaxis Básica del SQL: Lenguaje de Definición de Datos - LDD

		Nivel Lógico Global	Nivel Externo	Nivel Físico
LENGUAJE DE DEFINICIÓN DE DATOS - LDD	Definición del Esquema (CREATE)	DOMAIN, TABLE, ASSERTION...	VIEW	INDEX
	Evolución del Esquema (ALTER, DROP)	DOMAIN, TABLE, ASSERTION...	VIEW	INDEX
LENGUAJE DE MANIPULACIÓN DE DATOS – LMD	Actualizaciones: Altas, Bajas, Modificaciones	INSERT ... DELETE... UPDATE...		
	Consultas	SELECT...		
LENGUAJE DE CONTROL DE DATOS – LCD	Recuperación y control de concurrencia	COMMIT, ROLLBACK...		
	Seguridad y Protección	GRANT, REVOKE...		

LMD: Actualizaciones

Altas

```
INSERT INTO <nombre_tabla> [(lista_columnas)]  
VALUES (lista_columnas_inserción);
```

```
INSERT INTO grupo VALUES ('I11', 1, DEFAULT);
```

⇒ Todas las columnas de la tabla y en el mismo orden.

```
INSERT INTO grupo (curso, cod_grupo) VALUES (2,'I12');
```

⇒ Clave primaria tiene que estar incluida.

GRUPO

cod_grupo	curso	turno
I11	1	M
I12	2	M

LMD: Actualizaciones

Altas

GRUPO_M
debe existir

```
INSERT INTO GRUPO_M  
SELECT * FROM Grupo  
WHERE turno='M';
```

GRUPO_M

cod_grupo	curso	turno
I11	1	M
I12	2	M

LMD: Actualizaciones

Altas

```
INSERT INTO GRUPO_Curso1 (Turno,Codigo)  
SELECT turno, cod_grupo FROM Grupo  
WHERE curso = 1;
```

GRUPO_Curso1

Turno	Codigo
M	I11

LMD: Actualizaciones

Altas

Alumno:

num_mat	nombre	ciudad	cod_grupo	precio
1	Juan	Madrid	I11	25000
3	Ana	Leganés	I21	80000
8	María	Leganés	I22	30000
2	Pedro	Getafe	I21	20000
5	Salomé	Madrid	I21	25000

Grupo:

cod_grupo	curso	turno
I11	1	M
I12	2	M
I13	3	M
I21	1	T
I22	2	T
I31	3	T

LMD: Actualizaciones

Bajas

**DELETE [FROM] tabla
[WHERE condición];**

**DELETE FROM grupo
WHERE curso = 1;**

¿Qué ocurre en la tabla ALUMNO?

cod_grupo	curso	turno
I12	2	M
I13	3	M
I22	2	T
I23	3	T

DELETE FROM grupo;

cod_grupo	curso	turno
-----------	-------	-------

LMD: Actualizaciones

Modificaciones

UPDATE tabla
SET columna1=expresión1 [, columna2=expresión2...]
[WHERE condición];

UPDATE alumno
SET cod_grupo = 'I22', precio = precio +50
WHERE nombre > 'Ramiro';

Alumno:

num_mat	nombre	ciudad	cod_grupo	precio
1	Juan	Madrid	I11	25000
3	Ana	Leganés	I21	80000
8	María	Leganés	I22	30000
2	Pedro	Getafe	I21	20000
5	Salomé	Madrid	I21	25000

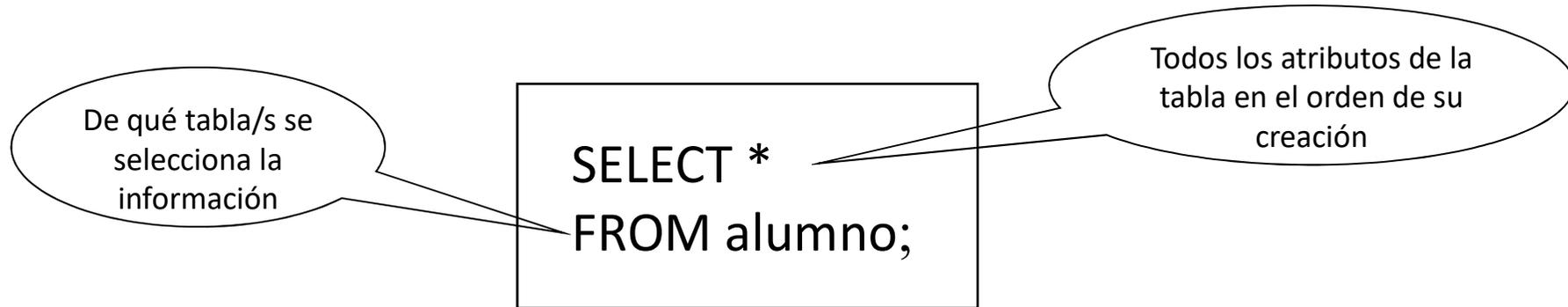
↓	↓
I22	25050

Ejercicio 1

□ LMD: Consultas

```
SELECT [ALL|DISTINCT] {lista-atributos | *}  
FROM nombre_tabla [, nombre_tabla, ...]  
[WHERE condición]  
[cláusula GROUP BY]  
[cláusula HAVING]  
[cláusula ORDER BY];
```

LMD: Consultas



⇒ *Todos* los atributos de la tabla ALUMNO (todas las tuplas).

num_mat	nombre	ciudad	cod_grupo	precio
1	Juan	Madrid	I11	25000
3	Ana	Leganés	I21	80000
8	María	Leganés	I22	30000
2	Pedro	Getafe	I21	20000
5	Salomé	Madrid	I22	25000

LMD: Consultas

Proyección

```
SELECT nombre, ciudad  
FROM alumno;
```

⇒ Selección de los atributos nombre y ciudad de la tabla ALUMNO.

nombre	ciudad
Juan	Madrid
Ana	Leganés
María	Leganés
Pedro	Getafe
Salomé	Madrid

LMD: Consultas

ALL: valor que se
asume por omisión

DISTINCT:
Eliminación de
duplicados

```
SELECT ciudad
FROM alumno;
```

ciudad
<i>Madrid</i>
Leganés
Leganés
Getafe
<i>Madrid</i>

¿Proyección?

```
SELECT DISTINCT ciudad
FROM alumno;
```

ciudad
Getafe
Leganés
Madrid

¿Proyección!

π ciudad Alumno

LMD: Consultas

Posición: nº de orden de la columna dentro de la lista de columnas seleccionadas (*izda a dcha*)

Por omisión: **ASC**
Valores nulos los más altos

ORDER BY {columna | posición} [DESC|ASC]
[, {columna | posición} [DESC|ASC]]...

LMD: Consultas

```
SELECT ciudad  
FROM alumno  
ORDER BY ciudad ASC;
```

ciudad
Getafe
Leganés
Leganés
Madrid
Madrid

```
SELECT ciudad  
FROM alumno  
ORDER BY nombre;
```

ciudad
Leganés
Madrid
Leganés
Getafe
Madrid

```
SELECT DISTINCT ciudad  
FROM alumno  
ORDER BY nombre;
```

LMD: Consultas

```
SELECT 1 nombre, 2 precio*0.10  
FROM alumno  
ORDER BY 2, nombre;
```

Ordenación por una
columna calculada y por
nombre

nombre	precio*0.10
PEDRO	2000
JUAN	2500
SALOME	2500
MARIA	3000
ANA	8000

LMD: Consultas

Selección

$\sigma_{\text{cod_grupo}='I21'}$ ALUMNO

```
SELECT *
FROM ALUMNO
WHERE cod_grupo='I21';
```

Condición: combinación de *una o más expresiones* (usando operadores lógicos) que da como resultado: CIERTO, FALSO o DESCONOCIDO

num_mat	nombre	ciudad	cod_grupo	precio
3	Ana	Leganés	I21	80000
2	Pedro	Getafe	I21	20000

LMD: Consultas

Operadores

OPERADORES DE COMPARACIÓN:

- igual ("="), distinto ("<>"), menor que ("<"), mayor que (">"), menor o igual a ("<=") y mayor o igual a (">=")
(⇒ No hay espacio entre los símbolos)

OPERADORES ARITMÉTICOS:

- Suma ("+"), resta ("-"), multiplicación ("*") y división ("/")

• OPERADORES LÓGICOS:

- **AND, OR y NOT**

```
SELECT * FROM Alumno WHERE precio >20000 AND (ciudad = 'MADRID' OR ciudad=('GETAFE');  
SELECT * FROM Alumno WHERE precio >20000 AND ciudad = 'MADRID' OR ciudad=('GETAFE');
```

Primero
condiciones con
AND y luego OR

1º

2º

• VALORES NULOS (desconocido):

- **IS [NOT] NULL**

```
SELECT * FROM Alumno WHERE ciudad IS NULL;  
SELECT * FROM Alumno WHERE cod_grupo IS NOT NULL;
```

LMD: Consultas

RANGO DE VALORES: BETWEEN ... AND

```
SELECT * FROM Alumno WHERE precio BETWEEN 20000 AND 25000;  
= SELECT * FROM Alumno WHERE precio >=20000 AND precio <= 25000;  
SELECT * FROM Alumno WHERE precio NOT BETWEEN 20000 AND 25000;  
= SELECT * FROM Alumno WHERE NOT (precio BETWEEN 20000 AND 25000);
```

OPERADOR LIKE:

Se emplea para comparar el contenido de una columna con una serie de caracteres.

Caracteres comodín:

- Subrayado (): sustituye a un carácter en la misma posición
- Tanto por ciento (%): sustituye a n caracteres, donde n puede ser 0.

```
SELECT * FROM Alumno WHERE nombre LIKE '_E%O':
```

OPERADOR IN:

Permite comprobar si un valor pertenece a un conjunto de valores determinados.

• expresión IN (lista de valores)

```
SELECT * FROM Alumno WHERE ciudad IN ('BARCELONA','MADRID');  
= SELECT * FROM ALUMNO WHERE ciudad = 'BARCELONA' OR ciudad = 'MADRID';  
SELECT * FROM Alumno WHERE ciudad NOT IN ('BARCELONA','MADRID');
```

EJEMPLOS:

```
SELECT *  
FROM Alumno  
WHERE cod_grupo LIKE 'I2%'; (o, en este caso, 'I2_')
```

num_mat	nombre	ciudad	cod_grupo	precio
3	Ana	Leganés	I21	80000
8	María	Leganés	I22	30000
2	Pedro	Getafe	I21	20000
5	Salomé	Madrid	I22	25000

2. Sintaxis Básica del SQL

2.2. Lenguaje de Manipulación de Datos

EJEMPLOS:

```
INSERT INTO Alumno  
VALUES (9, 'Eva', 'Fuenlabrada', NULL, 60000);
```

```
SELECT *  
FROM Alumno  
WHERE cod_grupo IS NULL; (=NULL no siempre funciona)
```

```
SELECT *  
FROM Alumno  
WHERE precio BETWEEN 15000 AND 30000;
```