Bases de datos y JDBC

Java y Servicios Web I
Master en Ingeniería Matemática

Manuel Montenegro
Dpto. Sistemas Informáticos y Computación

Desp. 467 (Mat)

montenegro@fdi.ucm.es

Bases de datos

- Una base de datos (DB) es una colección de datos interrelacionados, pertenecientes a un mismo contexto, y almacenados sistemáticamente para su posterior uso.
- Un sistema gestor de bases de datos (DBMS) es un programa que almacena y accede a la información contenida en las bases de datos.
- Una base de datos modela la información sobre ciertas entidades, y sobre relaciones entre las mismas.

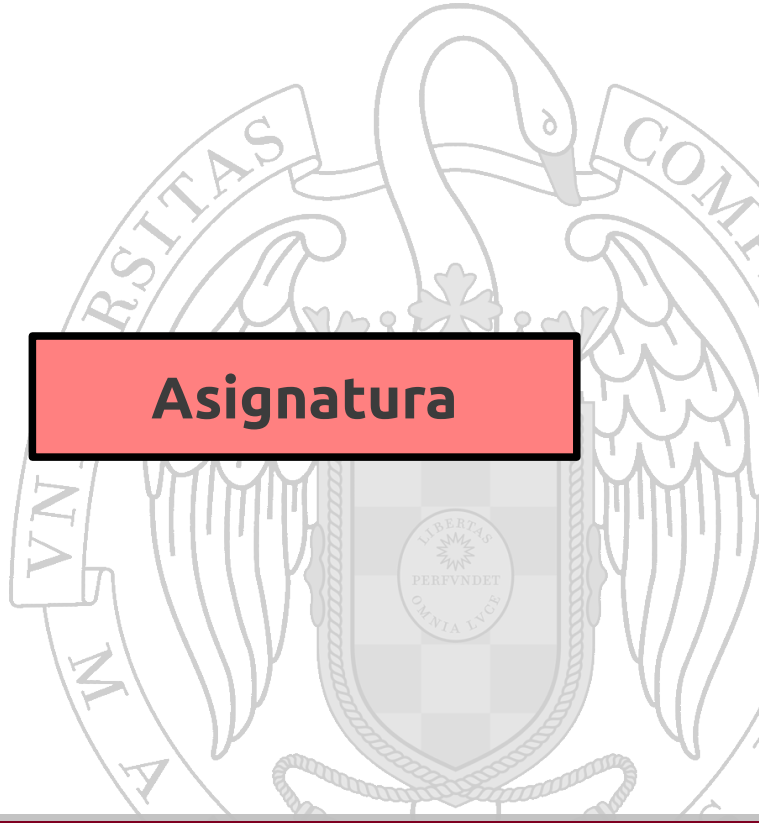
Contenidos

- Bases de datos relacionales
- Sistemas gestores de bases de datos
- Acceso a bases de datos
- Consultas SQL
- Acceso a bases de datos con JDBC

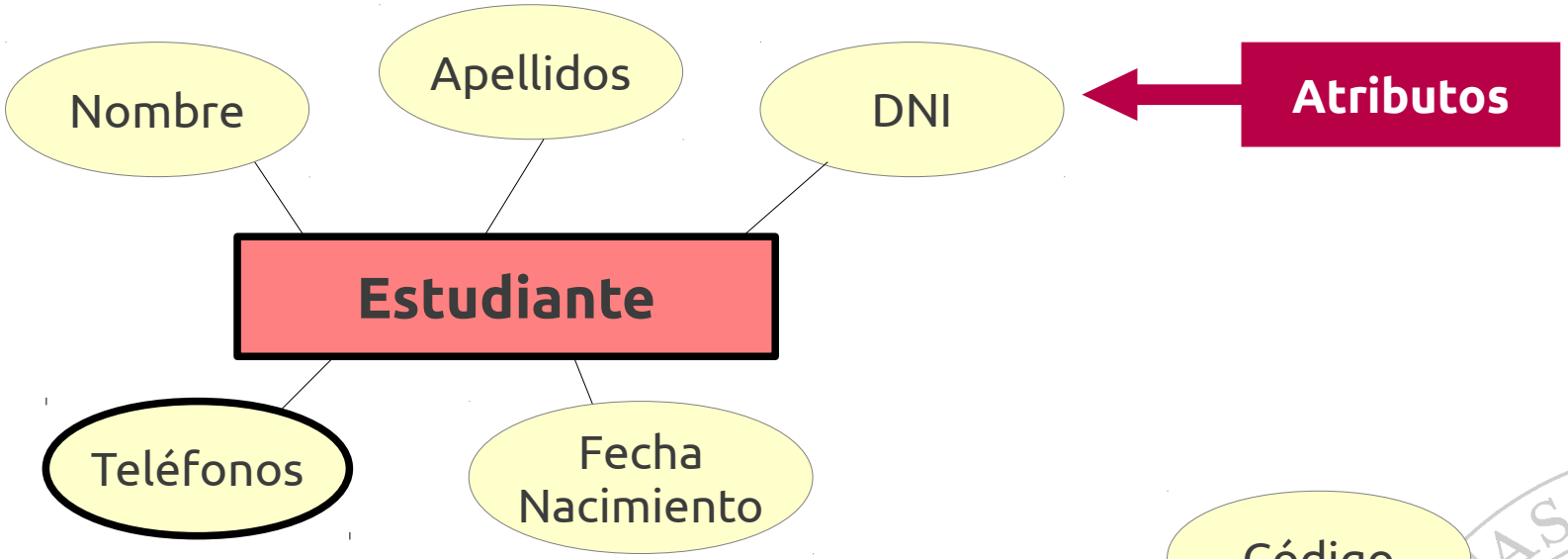
Entidades

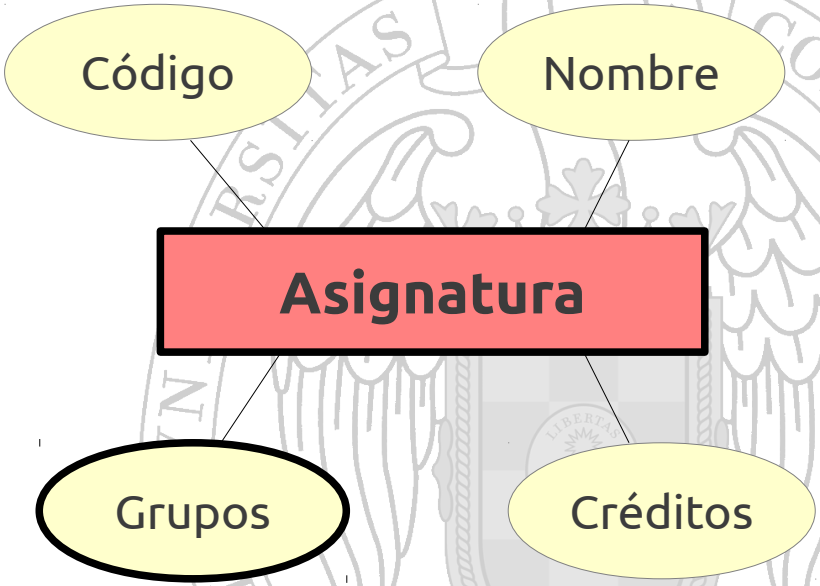
Estudiante

Asignatura

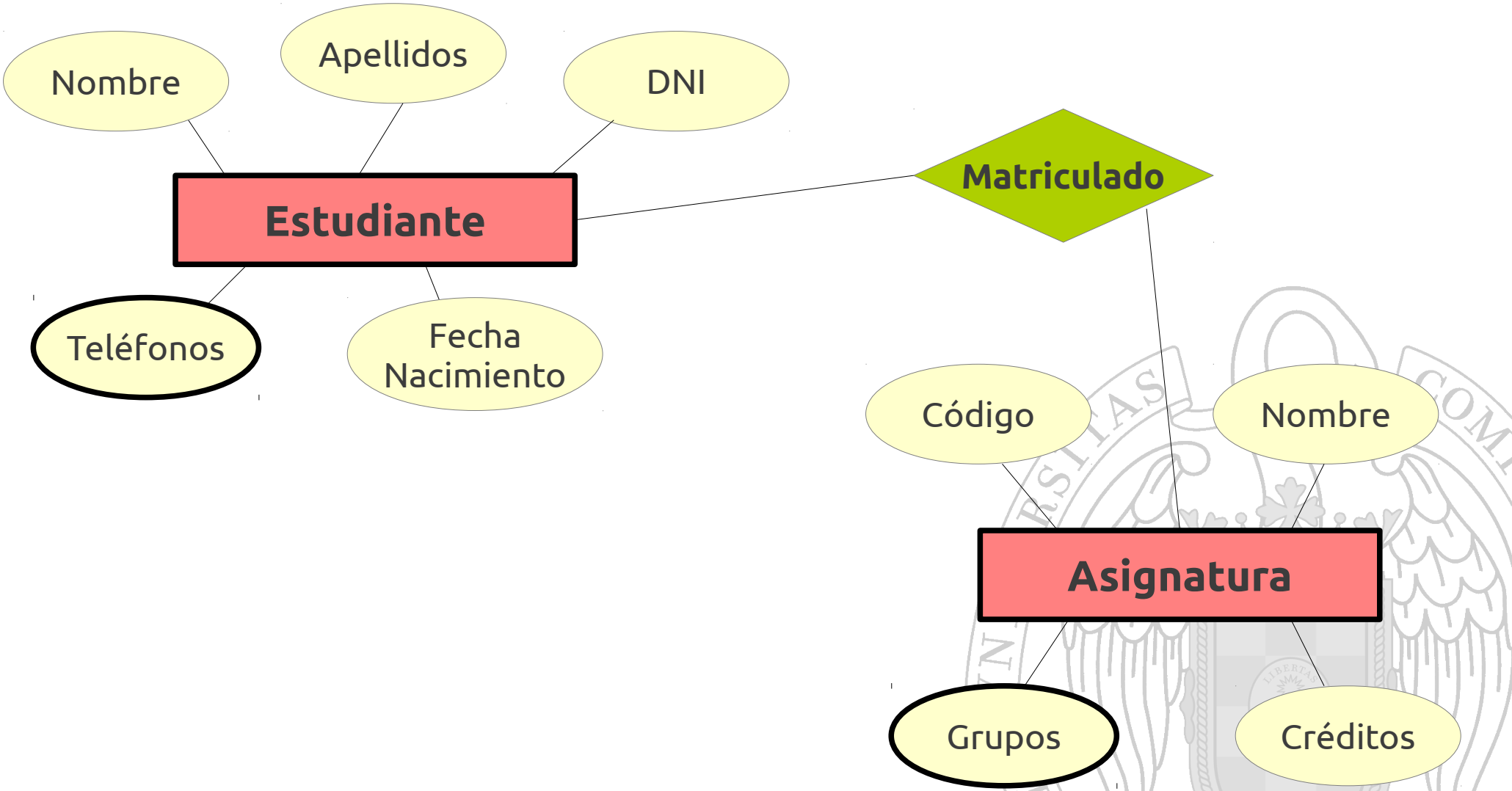


Entidades

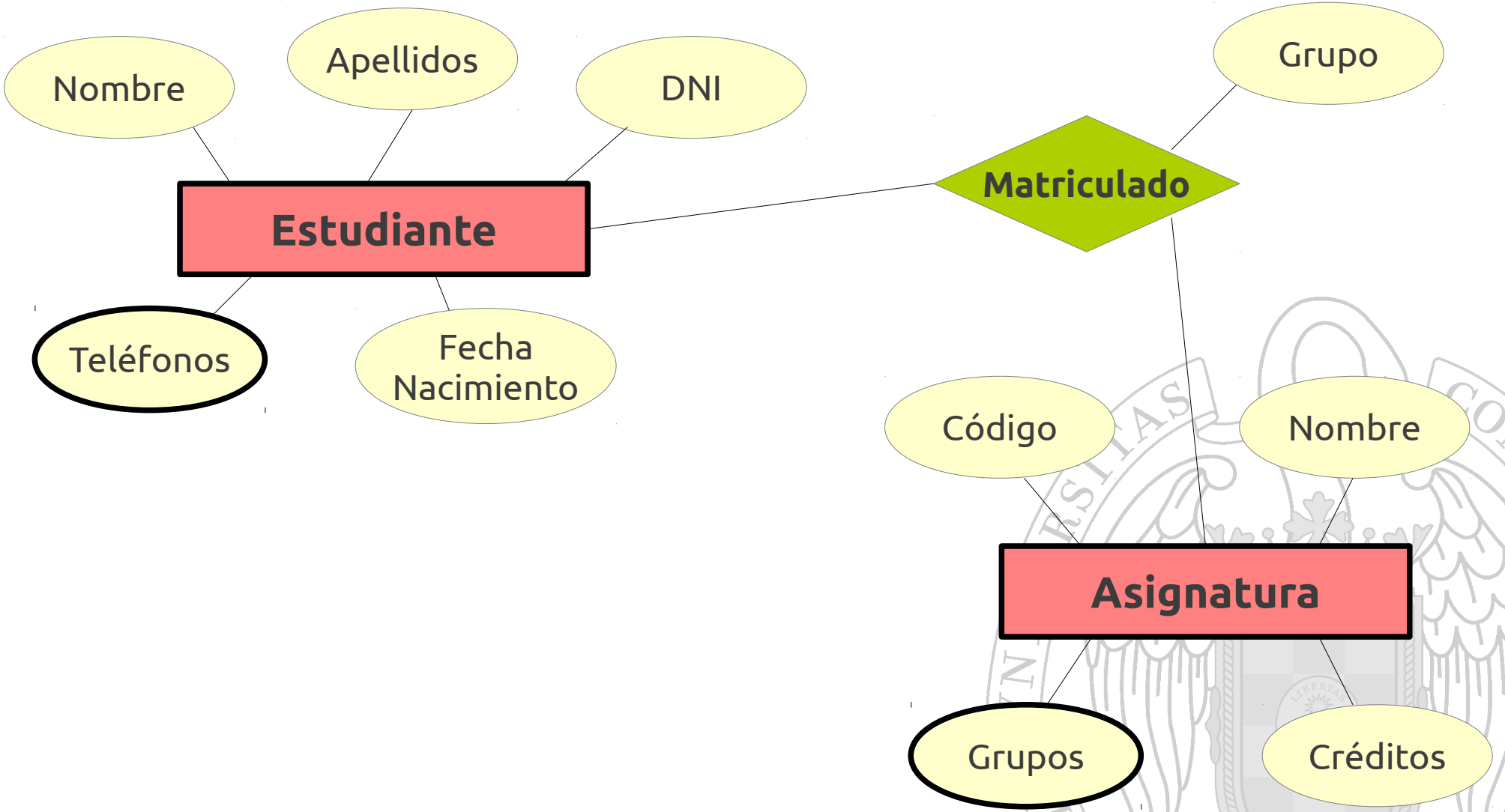

Nombre
Apellidos
DNI
Atributos
Estudiante
Teléfonos
Fecha
Nacimiento


Código
Nombre
Asignatura
Grupos
Créditos

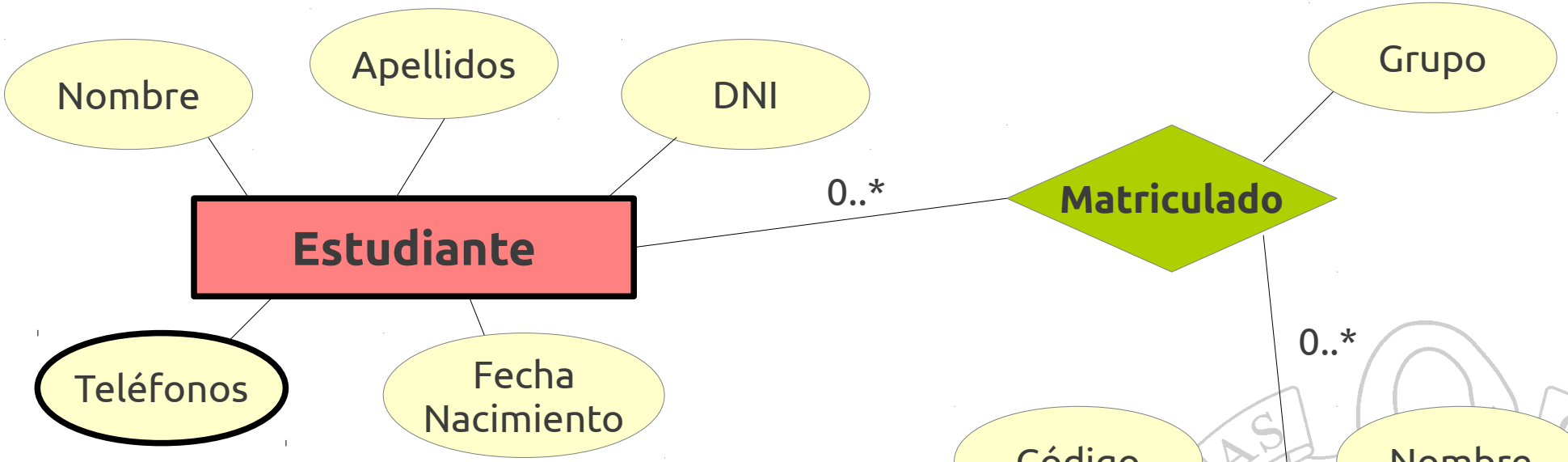
Relaciones


Nombre
Apellidos
DNI
Estudiante
Matriculado
Teléfonos
Fecha Nacimiento
Código
Nombre
Asignatura
Grupos
Créditos

Relaciones


Nombre
Apellidos
DNI
Grupo
Estudiante
Matriculado
Teléfonos
Fecha Nacimiento
Código
Nombre
Asignatura
Grupos
Créditos

Relaciones

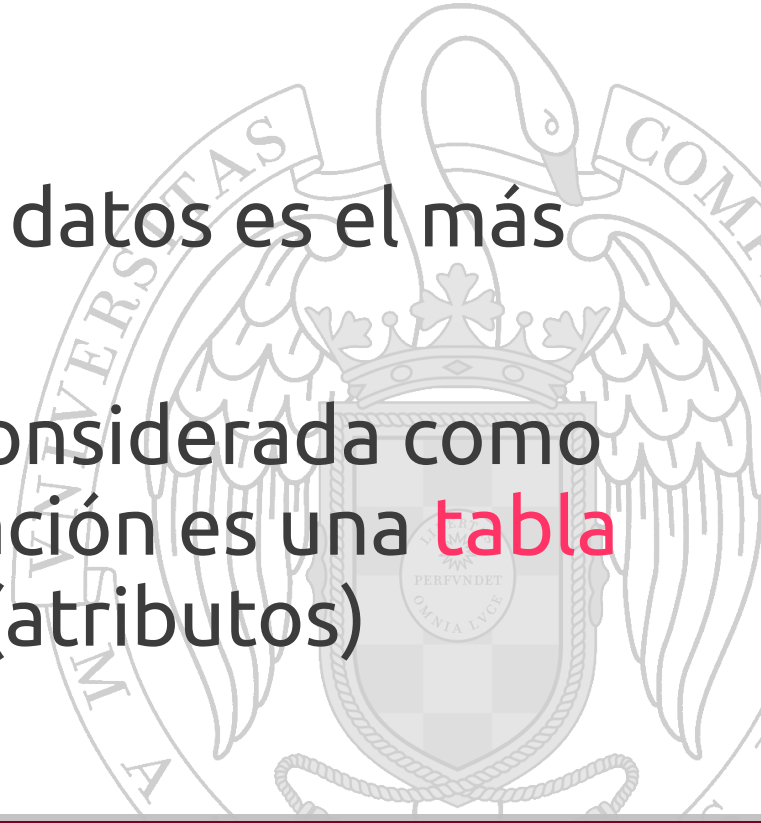


Relaciones:

– Uno a uno
– Uno a varios
– Varios a varios.

Bases de datos relacionales

- Modelos de bases de datos:
  - Relacionales.
  - Jerárquicos.
  - Orientado a objetos.
  - etc.
- El modelo relacional de base de datos es el más usado en la actualidad.
- Se basa en la idea de relación, considerada como un conjunto de tuplas. Cada relación es una tabla con filas (registros) y columnas (atributos)



Bases de datos relacionales

Estudiante

DNI	Nombre	Apellidos	Fecha Nac.	Teléfonos
12345673V	Ricardo	Fernández Aguinaga	20/04/1980	912421124 617293744
82122314X	Luis	Díaz Castro	25/04/1978	913111564
...	...	...	...	...

Asignatura

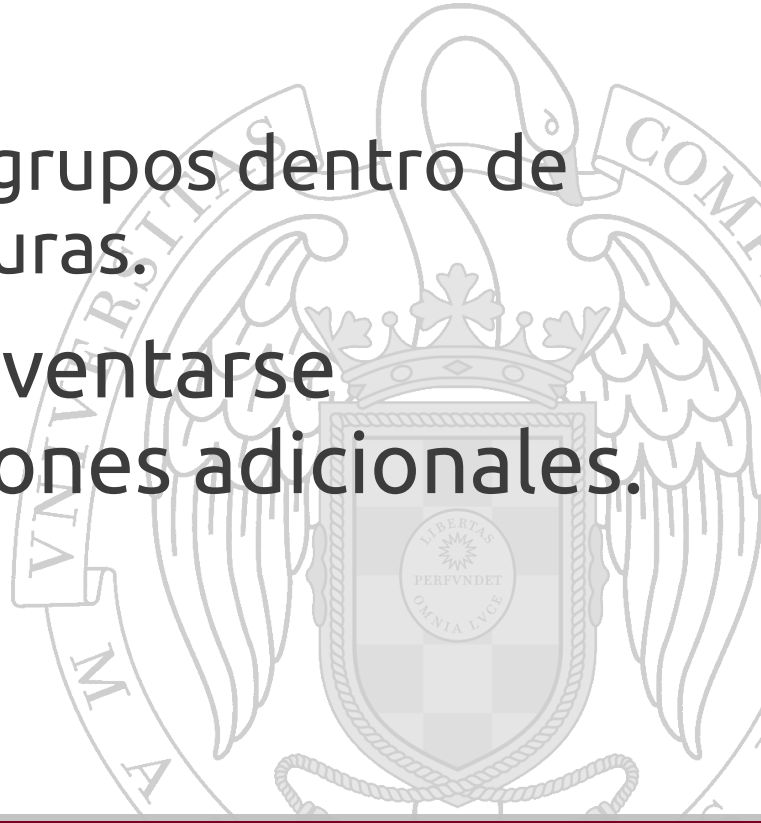
Código	Nombre	Créditos	Grupos
101	Álgebra	15	A,B,C
102	Funciones de una variable	12	A,B,C
...	...	...	...

Bases de datos relacionales

DNI Estudiante	Cod. Asignatura	Grupo
12345673V	101	A
12345673V	102	A
82122314X	101	B
...	...	...

Atomicidad

- Las columnas sólo pueden tener valores atómicos. En particular:
  - No podemos tener una lista de números de teléfono dentro de una celda de la tabla de estudiantes.
  - No podemos tener una lista de grupos dentro de una celda de la tabla de asignaturas.
- Estas restricciones pueden solventarse mediante la creación de relaciones adicionales.



Atomicidad

DNI	Nombre	Apellidos	Fecha Nac.	Teléfonos
12345673V	Ricardo	Fernández Aguinaga	20/04/1980	912421124 617293744
82122314X	Luis	Díaz Castro	25/04/1978	913111564
...	...	...	...	...

DNI	Nombre	Apellidos	Fecha Nac.	Teléfonos
12345673V	Ricardo	Fernández Aguinaga	20/04/1980	912421124
12345673V	Ricardo	Fernández Aguinaga	20/04/1980	617293744
82122314X	Luis	Díaz Castro	25/04/1978	913111564
...	...	...	...	...

Redundancia

Atomicidad

DNI	Nombre	Apellidos	Fecha Nac.	Teléfonos
12345673V	Ricardo	Fernández Aguinaga	20/04/1980	912421124 617293744
82122314X	Luis	Díaz Castro	25/04/1978	913111564
...	...	...	...	...

DNI	Nombre	Apellidos	Fecha Nac.
12345673V	Ricardo	Fernández Aguinaga	20/04/1980
82122314X	Luis	Díaz Castro	25/04/1978
...	...	...	...

DNI	Teléfono
12345673V	912421124
12345673V	617293744
82122314X	913111564
...	...

Atomicidad

Código	Nombre	Créditos	Grupos
101	Álgebra	15	A,B,C
102	Funciones de una variable	12	A,B,C
...	...	...	...

Código	Nombre	Créditos
101	Álgebra	15
102	Funciones de una variable	12
...	...	...

Código	Grupo
101	A
101	B
101	C
102	A
102	B
102	C
...	...

Bases de datos relacionales

- El modelo relacional de nuestra base de datos quedaría:
  - Estudiante(DNI, Nombre, Apellido, FechaNac)
  - Asignatura(Codigo, Nombre, NumCreditos)
  - Matriculado(DNIEstud, CodigoAsig, Grupo)
  - TieneTlf(DNIEstud, Telefono)
  - TieneGrupo(CodigoAsig, Grupo)

Claves

- Las claves de una tabla es el conjunto de campos que identifican unívocamente a cada registro de la tabla.
  - Debe tener siempre un valor en cada registro.
  - El valor debe permanecer inalterable.
  - No pueden existir dos registros dentro de la misma tabla con la misma clave.

Claves

- Las claves de cada tabla se representan mediante el subrayado en el modelo relacional.
  - Estudiante(DNI, Nombre, Apellido, FechaNac)
  - Asignatura(Codigo, Nombre, NumCreditos)
  - Matriculado(DNIEstud, CodigoAsig, Grupo)
  - TieneTlf(DNIEstud, Telefono)
  - TieneGrupo(CodigoAsig, Grupo)

Claves

- Una clave externa es la representación de la clave de una tabla en otra.
  - Estudiante(DNI, Nombre, Apellido, FechaNac)
  - Asignatura(Codigo, Nombre, NumCreditos)
  - Matriculado(DNIEstud, CodigoAsig, Grupo)
  - TieneTlf(DNIEstud, Telefono)
  - TieneGrupo(CodigoAsig, Grupo)

Contenidos

- Bases de datos relacionales
- Sistemas gestores de bases de datos
- Acceso a bases de datos
- Consultas SQL
- Acceso a bases de datos con JDBC

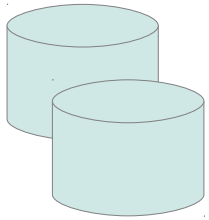
Gestores de bases de datos

- Existe una gran cantidad de gestores de bases de datos relacionales.
  - Oracle
  - Microsoft SQL Server
  - Microsoft Access
  - MySQL
  - PostgreSQL

Gestores de bases de datos

Aplicación

Consultas
Modificaciones



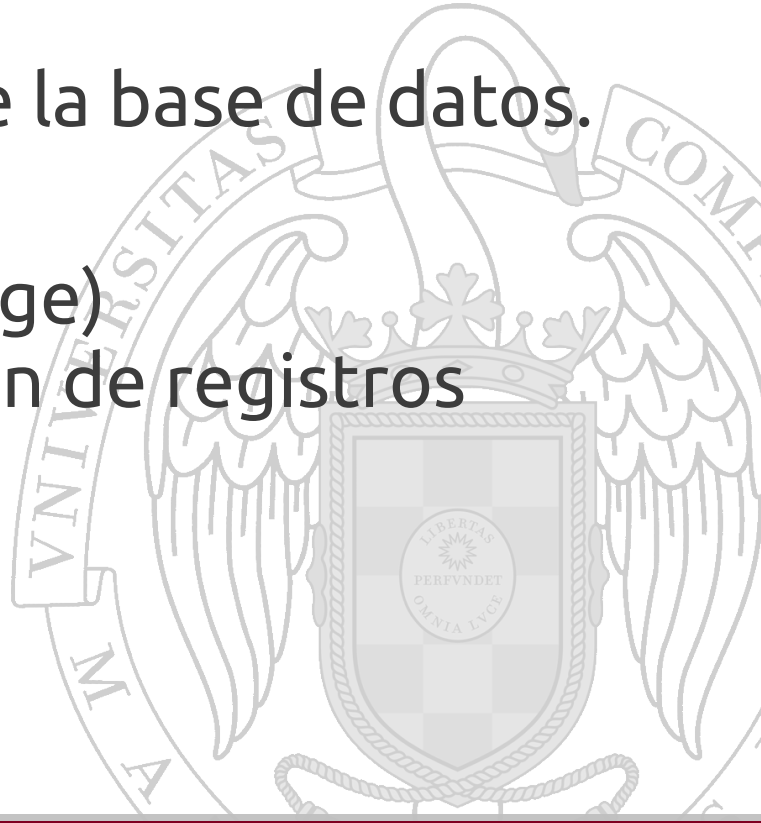
DBMS **Datos**

Programador

Lenguaje *SQL*

Lenguaje *SQL*

- SQL (Structured Query Language)
- Es un lenguaje declarativo de acceso a bases de datos relacionales.
  - DDL (Data definition language)
    Modificación de la estructura de la base de datos.
    Creación de tablas.
  - DML (Data manipulation language)
    Consulta, inserción, y eliminación de registros dentro de una tabla.



Contenidos

- Bases de datos relacionales
- Sistemas gestores de bases de datos
- Acceso a bases de datos
- Consultas SQL
- Acceso a bases de datos con JDBC

Acceso a bases de datos

Aplicación

- Desde Java: JDBC + driver MySQL

Programador

- Consola de MySQL.
- Herramienta gráfica (*phpMyAdmin*)

Herramienta *phpMyAdmin*

- Interfaz web para MySQL.
- Situada en `http://dalila.sip.ucm.es/phpmyadmin`
- Nombre de usuario y contraseña:
  ¡Pregunta al profesor!

Herramienta *phpMyAdmin*

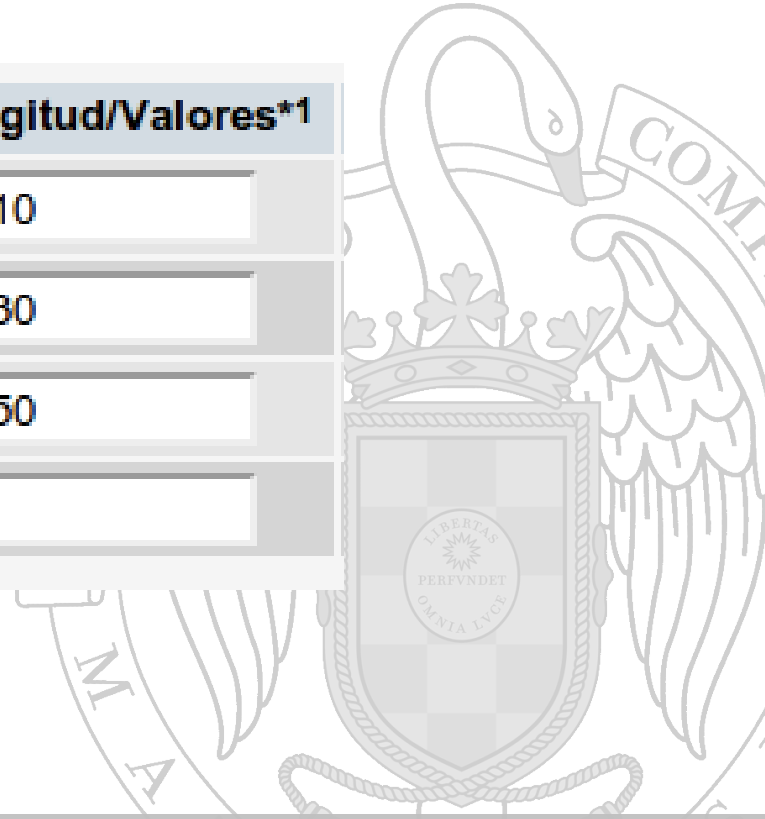
- Lista de bases de datos

Crear tablas

Crear nueva tabla en la base de datos **AlumnoJSW1**

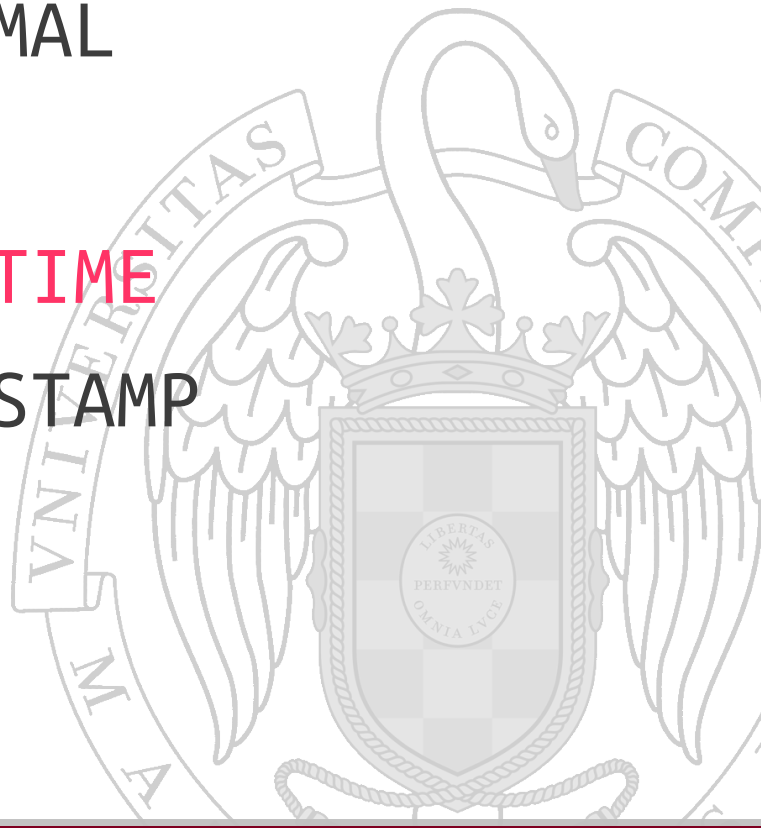
Nombre: Estudiantes Número de campos: 4

Campo	Tipo	Longitud/Valores*1
DNI	VARCHAR	10
Nombre	VARCHAR	30
Apellidos	VARCHAR	50
FechaNacimien	DATE	

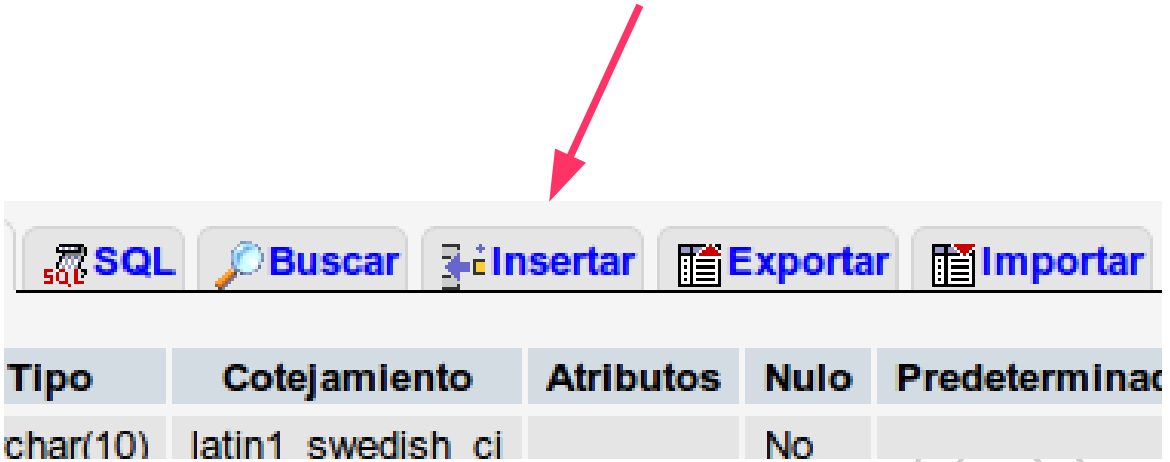


Tipos de datos

- CHAR(Longitud)
- VARCHAR(Longitud)
- TINYTEXT
- TEXT
- MEDIUMTEXT
- TINYINT
- SMALLINT
- MEDIUMINT
- INT
- BIGINT
- FLOAT
- DOUBLE
- DECIMAL
- DATE
- DATETIME
- TIMESTAMP
- TIME
- ENUM



Insertar registros en una tabla



Campo	Tipo	Función	Nulo	Valor
DNI	varchar(10)			12345673V
Nombre	varchar(30)			Ricardo
Apellidos	varchar(50)			Fernández Aguinaga
FechaNacimiento	date			1978-04-25

Continuar

Contenidos

- Bases de datos relacionales
- Sistemas gestores de bases de datos
- Acceso a bases de datos
- Consultas SQL
- Acceso a bases de datos con JDBC

Consultas en SQL

- Obtener todas las filas y columnas de una tabla.

```
SELECT * FROM Estudiantes
```

DNI	Nombre	Apellidos	FechaNac
12345673V	Ricardo	Fernández Aguinaga	1980-04-25
82122314X	Luis	Díaz Castro	1978-04-15
20358182T	Marta	Domínguez Iborra	1975-12-21
29377172M	Roberto	Blanco Rodrigo	1981-12-22
45219913T	Marta	Díaz Agrela	1983-01-20
40231491M	Martín	Montes Di Cesare	1979-03-15
23195991S	Sandra	González de Castro	1982-11-21
29918481X	Francisca	Montes Trujillo	1907-10-19

Consultas en SQL

- Obtener ciertas columnas de una tabla

```
SELECT DNI, Nombre
FROM Estudiantes
```

DNI	Nombre
12345673V	Ricardo
82122314X	Luis
20358182T	Marta
29377172M	Roberto
45219913T	Laura
40231491M	Martín
23195991S	Sandra
29918481X	Francisca

Consultas en SQL

- Imponer condiciones a las filas resultado.

```
SELECT DNI, Nombre, Apellidos
FROM Estudiantes
WHERE Nombre = 'Marta'
```

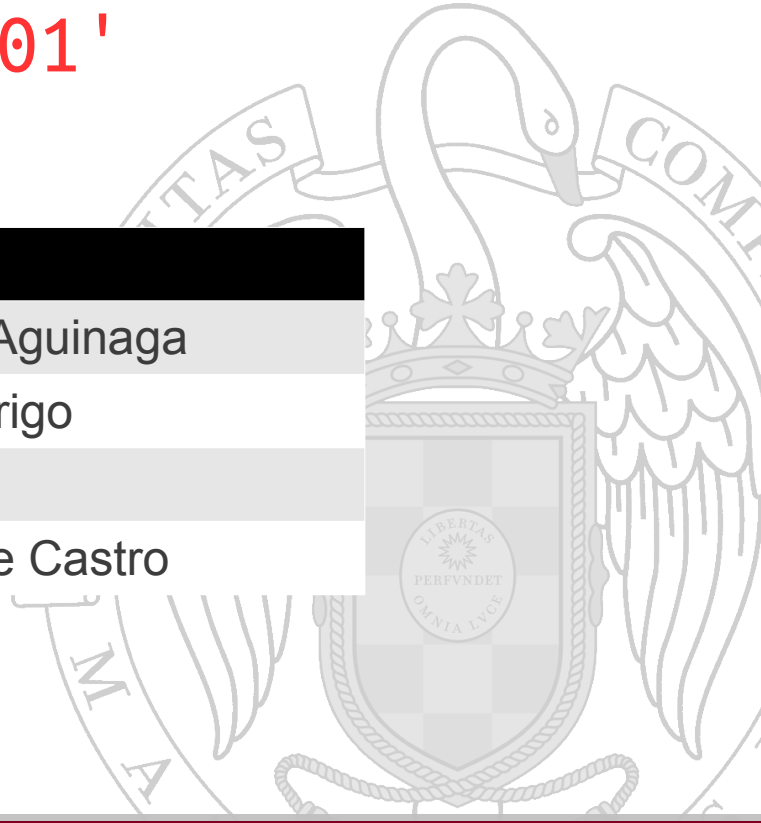
DNI	Nombre	Apellidos
20358182T	Marta	Domínguez Iborra
45219913T	Marta	Díaz Agrela

Consultas en SQL

- Imponer condiciones a las filas resultado.

```
SELECT DNI, Nombre, Apellidos
FROM Estudiantes
WHERE FechaNac >= '1980-01-01' AND
      FechaNac <  '1990-01-01'
```

DNI	Nombre	Apellidos
12345673V	Ricardo	Fernández Aguinaga
29377172M	Roberto	Blanco Rodrigo
45219913T	Marta	Díaz Agrela
23195991S	Sandra	González de Castro

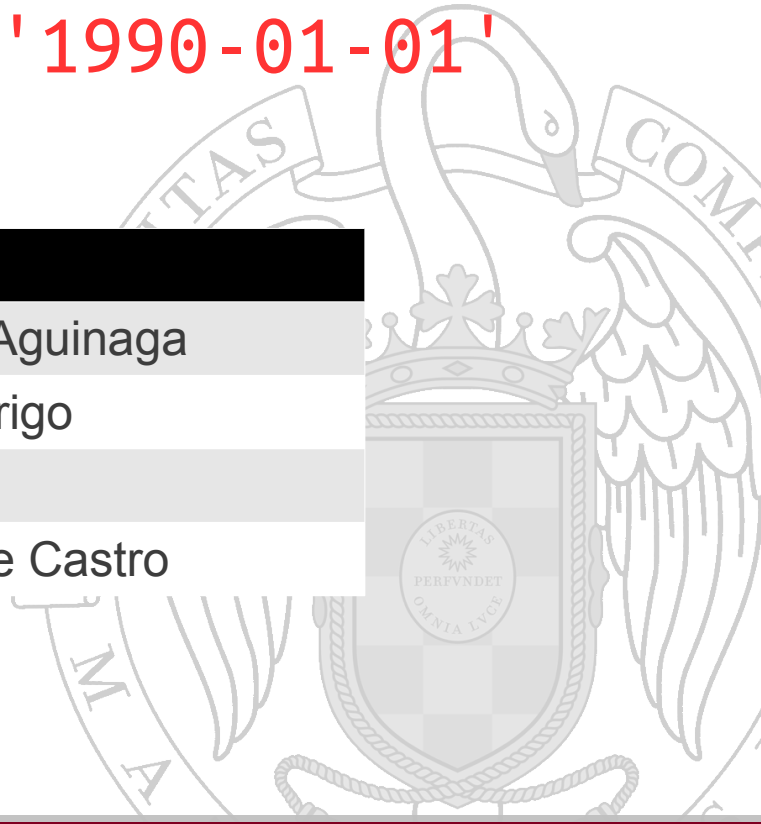


Consultas en SQL

- Imponer condiciones a las filas resultado.

```
SELECT DNI, Nombre, Apellidos
FROM Estudiantes
WHERE FechaNac BETWEEN '1980-01-01' AND
                       '1990-01-01'
```

DNI	Nombre	Apellidos
12345673V	Ricardo	Fernández Aguinaga
29377172M	Roberto	Blanco Rodrigo
45219913T	Marta	Díaz Agrela
23195991S	Sandra	González de Castro

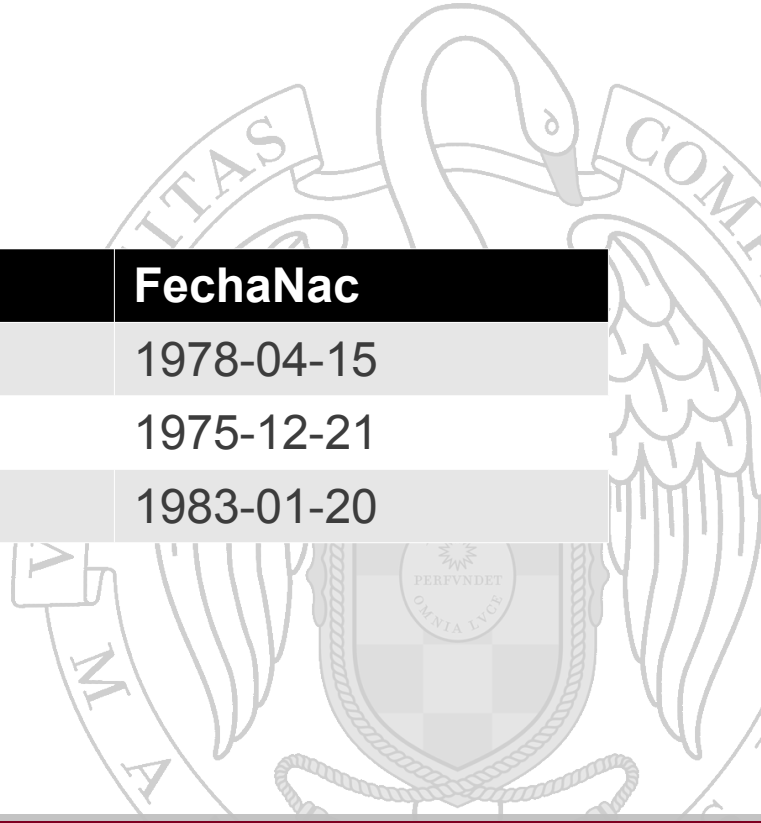


Consultas en SQL

- Imponer condiciones a las filas resultado.

```
SELECT DNI, Nombre, Apellidos
FROM Estudiantes
WHERE Apellidos LIKE 'D%'
```

DNI	Nombre	Apellidos	FechaNac
82122314X	Luis	Díaz Castro	1978-04-15
20358182T	Marta	Domínguez Iborra	1975-12-21
45219913T	Marta	Díaz Agrela	1983-01-20

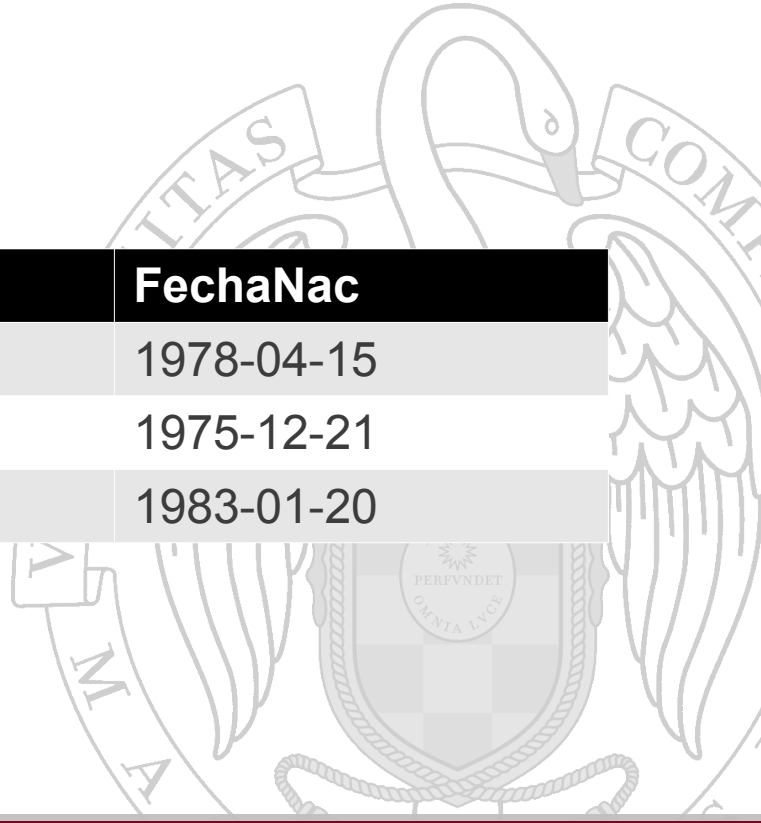


Consultas en SQL

- Imponer condiciones a las filas resultado.

```
SELECT DNI, Nombre, Apellidos
FROM Estudiantes
WHERE Apellidos LIKE 'D%'
```

DNI	Nombre	Apellidos	FechaNac
82122314X	Luis	Díaz Castro	1978-04-15
20358182T	Marta	Domínguez Iborra	1975-12-21
45219913T	Marta	Díaz Agrela	1983-01-20



Consultas en SQL

- Producto cartesiano

```
SELECT DNI, Apellidos, DNIEstud, CodigoAsig
FROM Estudiantes, Matriculado
```

DNI	Apellidos	DNIEstud	CodigoAsig
12345673V	Fernández Aguinaga	12345673V	803261
12345673V	Fernández Aguinaga	12345673V	803265
12345673V	Fernández Aguinaga	12345673V	803262
12345673V	Fernández Aguinaga	29377172M	803265
12345673V	Fernández Aguinaga	29377172M	803261
12345673V	Fernández Aguinaga	23195991S	803261
82122314X	Díaz Castro	12345673V	803261
82122314X	Díaz Castro	12345673V	803265
…	…	…	…

Consultas en SQL

- Producto cartesiano

```
SELECT DNI, Apellidos, DNIEstud, CodigoAsig
FROM Estudiantes, Matriculado
WHERE DNI = DNIEstud
```

DNI	Apellidos	DNIEstud	CodigoAsig
12345673V	Fernández Aguinaga	12345673V	803261
12345673V	Fernández Aguinaga	12345673V	803265
12345673V	Fernández Aguinaga	12345673V	803262
29377172M	Blanco Rodrigo	29377172M	803265
29377172M	Blanco Rodrigo	29377172M	803265
23195991S	González de Castro	23195991S	803261

Consultas en SQL

- Producto cartesiano

```
SELECT Estudiantes.DNI, Estudiantes.Apellidos,
       Matriculado.CodigoAsig
FROM Estudiantes, Matriculado
WHERE Estudiantes.DNI = Matriculado.DNIEstud
```

DNI	Apellidos	CodigoAsig
12345673V	Fernández Aguinaga	803261
12345673V	Fernández Aguinaga	803265
12345673V	Fernández Aguinaga	803262
29377172M	Blanco Rodrigo	803265
29377172M	Blanco Rodrigo	803265
23195991S	González de Castro	803261

Consultas en SQL

- Uso de funciones

```
SELECT Estudiantes.DNI, Estudiantes.Apellidos,
       LENGTH(Asignaturas.Nombre)
FROM Estudiantes, Matriculado, Asignaturas
WHERE Estudiantes.DNI = Matriculado.DNIEstud
  AND Asignaturas.Codigo = Matriculado.CodigoAsig
```

DNI	Apellidos	LENGTH(Nombre)
12345673V	Fernández Aguinaga	41
12345673V	Fernández Aguinaga	36
12345673V	Fernández Aguinaga	27
29377172M	Blanco Rodrigo	41
29377172M	Blanco Rodrigo	27
23195991S	González de Castro	41

Consultas en SQL

- Renombramientos de columnas

```
SELECT Estudiantes.DNI, Estudiantes.Apellidos,
       LENGTH(Asignaturas.Nombre) AS Longitud
FROM Estudiantes, Matriculado, Asignaturas
WHERE Estudiantes.DNI = Matriculado.DNIEstud
  AND Asignaturas.Codigo = Matriculado.CodigoAsig
```

DNI	Apellidos	Longitud
12345673V	Fernández Aguinaga	41
12345673V	Fernández Aguinaga	36
12345673V	Fernández Aguinaga	27
29377172M	Blanco Rodrigo	41
29377172M	Blanco Rodrigo	27
23195991S	González de Castro	41

Funciones de SQL

Cadenas

- LENGTH(col)
- LEFT(col,n)
- RIGHT(col,n)
- TRIM(col)
- UPPER(col)
- LOWER(col)
- SUBSTRING(col,ini,fin)
- …

Fechas/Horas

- HOUR(col)
- MINUTE(col)
- SECOND(col)
- DAYNAME(col)
- CURDATE()
- NOW()
- …

Consultas en SQL

- Agrupamientos

```
SELECT DNIEstud, CodigoAsig
FROM Matriculado
```

DNIEstud	CodigoAsig
12345673V	803261
12345673V	803265
12345673V	803262
29377172M	803265
29377172M	803261
23195991S	803261

Consultas en SQL

- Agrupamientos

```
SELECT COUNT(*) AS NumAls, CodigoAsig
FROM Matriculado
GROUP BY DNIEstud
```

NumAls	CodigoAsig
3	803261
2	803265
1	803262

Funciones de agrupamiento

- `MIN(col)`
- `MAX(col)`
- `SUM(col)`
- `COUNT(col)`
- `AVG(col)`

Consultas en SQL

- Agrupamientos + Condiciones

```
SELECT COUNT(*) AS NumAls, CodigoAsig
FROM Matriculado
GROUP BY DNIEstud
HAVING COUNT(*) >= 3
```

NumAls	CodigoAsig
3	803261

Inserciones en SQL

```
INSERT INTO Estudiantes (DNI, Nombre, Apellidos, FechaNac)
VALUES ('12314561F', 'Carolina', 'Rodriguez Yagüe', '1983-05-20')
```

DNI	Nombre	Apellidos	FechaNac
12345673V	Ricardo	Fernández Aguinaga	1980-04-25
82122314X	Luis	Díaz Castro	1978-04-15
20358182T	Marta	Domínguez Iborra	1975-12-21
29377172M	Roberto	Blanco Rodrigo	1981-12-22
45219913T	Marta	Díaz Agrela	1983-01-20
40231491M	Martín	Montes Di Cesare	1979-03-15
23195991S	Sandra	González de Castro	1982-11-21
29918481X	Francisca	Montes Trujillo	1907-10-19
12314561F	**Carolina**	**Rodríguez Yagüe**	**1983-05-20**

Actualizaciones en SQL

```
UPDATE Estudiantes SET Nombre='Javier'
WHERE DNI='82122314X'
```

- Si no se especifica cláusula WHERE, se actualizarán todas las filas.

DNI	Nombre	Apellidos	FechaNac
12345673V	Ricardo	Fernández Aguinaga	1980-04-25
82122314X	**Javier**	Díaz Castro	1978-04-15
20358182T	Marta	Domínguez Iborra	1975-12-21
29377172M	Roberto	Blanco Rodrigo	1981-12-22
45219913T	Marta	Díaz Agrela	1983-01-20
40231491M	Martín	Montes Di Cesare	1979-03-15
23195991S	Sandra	González de Castro	1982-11-21
29918481X	Francisca	Montes Trujillo	1907-10-19
12314561F	Carolina	Rodríguez Yagüe	1983-05-20

Borrar registros en SQL

```
DELETE FROM Estudiantes
WHERE FechaNac BETWEEN '1980-01-01' AND '1990-01-01'
```

- Si no se especifica cláusula WHERE, se borrarán todas las filas.

DNI	Nombre	Apellidos	FechaNac
12345673V	**Ricardo**	**Fernández Aguinaga**	**1980-04-25**
82122314X	Javier	Díaz Castro	1978-04-15
20358182T	Marta	Domínguez Iborra	1975-12-21
29377172M	**Roberto**	**Blanco Rodrigo**	**1981-12-22**
45219913T	**Marta**	**Díaz Agrela**	**1983-01-20**
40231491M	Martín	Montes Di Cesare	1979-03-15
23195991S	**Sandra**	**González de Castro**	**1982-11-21**
29918481X	Francisca	Montes Trujillo	1907-10-19
12314561F	**Carolina**	**Rodríguez Yagüe**	**1983-05-20**

Contenidos

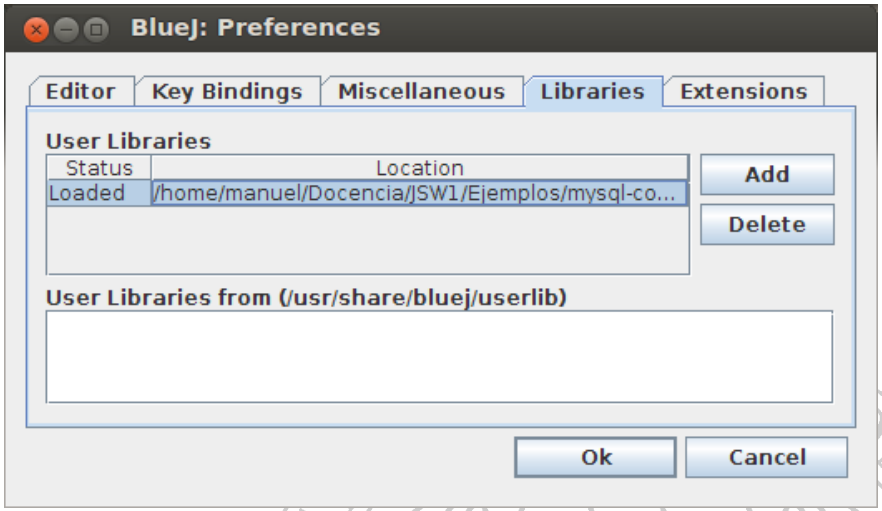
- Bases de datos relacionales
- Sistemas gestores de bases de datos
- Acceso a bases de datos
- Consultas SQL
- Acceso a bases de datos con JDBC

JDBC

- JDBC: *Java Database Connectivity*
- Paquete `java.sql`
- API de acceso a bases de datos desde Java.
- Interfaz independiente del sistema gestor de base de datos (DBMS) que se utilice.
- Cada DBMS con soporte para JDBC proporciona un driver que implementa esa interfaz.
- MySQL
  - http://dev.mysql.com/downloads/connector/j/

Instalación

- Extraer `mysql-connector-java-5.1.20-bin.jar`
- Desde *BlueJ:*
  - *Tools → Preferences*
  - Pestaña *Libraries*
  - Botón *Add*
  - Seleccionar el archivo .jar
- Requiere reiniciar la JVM
  - *Tools → Reset Java Virtual Machine*



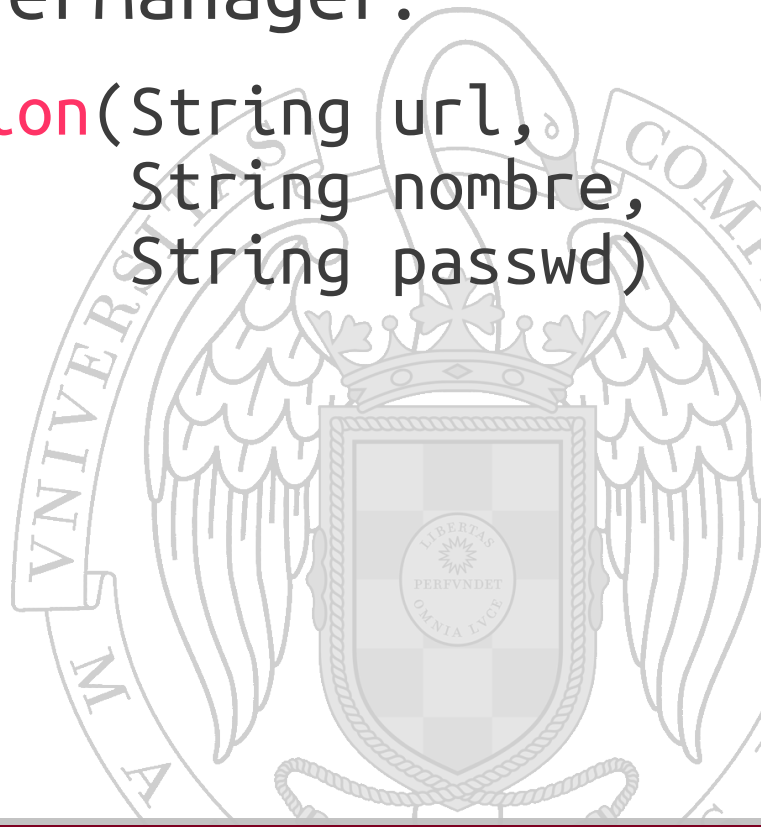
URL de una base de datos

- Se utiliza para especificar a qué base de datos acceder, qué driver utilizar, y en qué servidor se encuentra el gestor de bases de datos.


```
jdbc:mysql://localhost/AlumnoJSW1
```

La clase Connection

- Sus instancias representan conexiones a una base de datos.
- Se construye mediante el método estático `getConnection` de la clase `DriverManager`.
  - `static Connection getConnection(String url, String nombre, String passwd)`



Ejemplo

```
public class ConsultaSQL {

    static final String DATABASE_URL = "jdbc:mysql://dalila.sip.ucm.es/AlumnoJSW1";
    static final String USER = "AlumnoJSW1";
    static final String PASSWORD = "jsw1";


    public static void main(String[] args) {

        Connection con = null;

        try {

            con = DriverManager.getConnection(DATABASE_URL, USER, PASSWORD);
            System.out.println("Conexión creada correctamente");

        } catch(SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                con.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

    }
}
```

La clase Statement

- Los objetos de la clase Connection pueden crear objetos de la clase Statement.
  - Statement createStatement()
- Los objetos Statement también han de cerrarse mediante su método close()

```
Statement st = null;
try {
    con = DriverManager.getConnection(DATABASE_URL, USER, PASSWORD);
    st = con.createStatement();
} catch(SQLException e) { … }
finally {
    try {
       st.close();
       con.close();
    } catch (Exception e) {
       e.printStackTrace();
    }
}
```

Ejecutar consultas

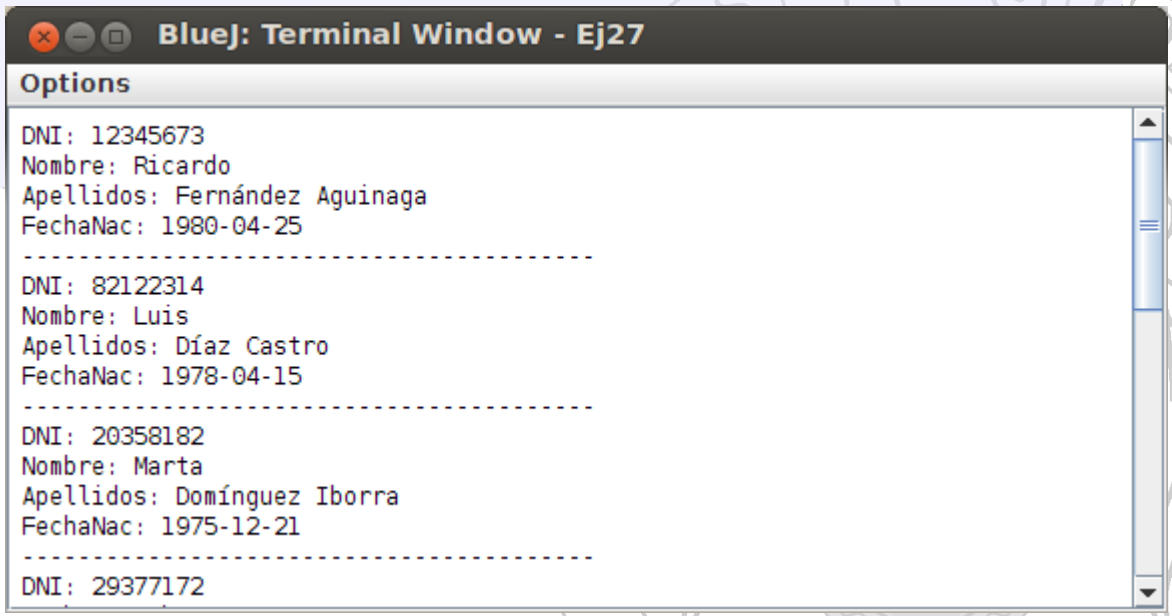
- Las consultas a la base de datos se realizan mediante el método `executeQuery` de la clase `Statement`.
  - `ResultSet executeQuery(String sql)`
- Devuelve un objeto de la clase `ResultSet`, que permite iterar sobre los resultados de la consulta.
  - `boolean next()`
  - `int getInt(int indiceCol)`
    `int getInt(String nombreCol)`
  - `String getString(int indiceCol)`
    `String getString(String nombreCol)`
  - `Date getDate(int indiceCol)`
    `Date getDate(String nombreCol)`
  - `void close()`

Ejemplo

```
public class ConsultaSQL
{
    static final String DATABASE_URL = "jdbc:mysql://dalila.sip.ucm.es/AlumnoJSW1";
    static final String USER = "AlumnoJSW1";
    static final String PASSWORD = "jsw1";


    public static void main(String[] args) {
        Connection con = null;
        Statement st = null;
        ResultSet rs = null;

        try {

            con = DriverManager.getConnection(DATABASE_URL, USER, PASSWORD);
            st = con.createStatement();
            rs = st.executeQuery(
                      "SELECT DNI, Nombre, Apellidos, FechaNac FROM Estudiantes");
            while (rs.next()) {
                System.out.println("DNI: " + rs.getInt(1));
                System.out.println("Nombre: " + rs.getString(2));
                System.out.println("Apellidos: " + rs.getString(3));
                System.out.println("FechaNac: " + rs.getDate(4));
                System.out.println("----------------------------------------");
            }
            …
```

Ejemplo

```
            …
        } catch(SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                rs.close();
                st.close();
                con.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```



Actualizar una BD

- Para realizar modificaciones a una tabla de la base de datos (`INSERT`, `UPDATE` o `DELETE`), se utiliza el método `executeUpdate` de la clase `Statement`.
  - `int executeUpdate(String sql)`

Ejemplo

```
try {
  con = DriverManager.getConnection(DATABASE_URL, USER, PASSWORD);
  st = con.createStatement();
  st.executeUpdate("INSERT INTO Estudiantes (DNI, Nombre, Apellidos, FechaNac)"
            + "VALUES (\'99999999X\', \'Nueva\', \'Persona\', \'20100910\')");
} catch(SQLException e) {
      e.printStackTrace();
} finally {
    try {
       st.close();
       con.close();
    } catch (Exception e) {
       e.printStackTrace();
}
```

```
INSERT INTO Estudiantes (DNI, Nombre, Apellidos, FechaNac)
VALUES ('99999999X', 'Nueva', 'Persona', '20100910')
```

Consultas paramétricas

- Son consultas con marcadores (*placeholders*), donde se pueden colocar elementos en tiempo de ejecución.

```
SELECT Dni, Nombre, Apellidos, FechaNac
FROM Estudiantes
WHERE DNI = ?
```

- Se manejan con los métodos de la clase `PreparedStatement`

Ejemplo

```
Connection con = null;
PreparedStatement pst = null;
ResultSet rs = null;
Scanner sc = new Scanner(System.in);
System.out.print("Dime el número de DNI: ");
int dni = sc.nextInt();

try {

   con = DriverManager.getConnection(DATABASE_URL, USER, PASSWORD);
   pst = con.prepareStatement(
       "SELECT DNI, Nombre, Apellidos, FechaNac FROM Estudiantes WHERE DNI = ?"
         );
   pst.setInt(1, dni);
   rs = pst.executeQuery();
   if (rs.next()) {

       System.out.println("DNI: " + rs.getInt(1));
       System.out.println("Nombre: " + rs.getString(2));
       System.out.println("Apellidos: " + rs.getString(3));
       System.out.println("FechaNac: " + rs.getDate(4));

    } else {

       System.out.println("No se han encontrado entradas");

    }
 } catch (…) { … }
```

Referencias

- L. Ullman
  MySQL. Guía de Aprendizaje.
  Prentice Hall
- A. Silberschatz, H. F. Korth, S. Sudarshan
  Fundamentos de Bases de Datos (5ª Edición)
  McGraw Hill
- P. Deitel, H. Deitel
  Java. How to Program (9th Edition)
  Cap. 28