

UNIVERSIDAD DEL CAUCA



LABORATORIO III DE SISTEMAS DE TELECOMUNICACIONES

Énfasis en Aplicaciones y Servicios Telemáticos

Ing. Javier Alexander Hurtado
Departamento de Telemática

Popayán, Agosto de 2002

Fundamentos de Bases de Datos y MySQL



<http://www.mysql.com>

Aplicaciones y Servicios Telemáticos
Ing. Javier Alexander Hurtado

Popayán, Agosto de 2002

Objetivos

- Conocer el entorno de desarrollo de MySql como SGBD utilizando línea de comandos y una interfaz gráfica (phpMyAdmin o MySQLGUI)
- Estar en capacidad de desarrollar aplicaciones cliente/servidor usando Java como lenguaje de programación.

Referencias

- JDBC

Web: <http://java.sun.com/products/jdbc/>

- Manual de MySQL

Web: <http://www.mysql.com/documentation/index.html>

- phpMyAdmin

Web: <http://www.phpmyadmin.com>



Definición de Base de Datos

- Una base de datos (BD) es un conjunto de datos interrelacionados almacenados en conjunto, sin redundancias innecesarias, de forma independiente de los programas que acceden a ellos.



MySQL

¿Qué es MySQL?

Servidor de Bases de Datos SQL multiusuario y multitarea. Es una aplicación cliente / servidor conformada por un demonio servidor *mysqld* y algunos programas clientes, al igual que librerías.

Busca velocidad, robustez y fácil uso.

Excelente desempeño

Pruebas con mas de 40 BD que contienen 10.000 tablas de las cuales más de 500 tienen más de 7 millones de registros.



El Sistema Gestor de Bases de Datos SGBD

- Sirve para:
 - definir y crear datos
 - manipular esos datos
 - seguridad e integridad de los datos
 - recuperar los datos: lenguaje SQL
 - mantenimiento de un diccionario de datos
 - alto rendimiento: se debe asegurar que todas estas funciones se ejecuten lo más rápidamente posible.
- Gratuitos como: MySQL, PostgreSQL. Válidos hasta ~500.000 accesos diarios
- Otros (comerciales): Oracle, SQLServer, Informix,...



El modelo Entidad - Relación

- Se usa para diseñar la BD
- La mayoría de BD actuales son de tecnología **relacional**
- Conceptos base:
 - entidades (y atributos y elementos)
 - relaciones

Entidades

- Los objetos que aparece en la vida real, es lo que llamamos **entidad**. Por ejemplo, alumnos, empleados, aviones, teléfonos, hoteles, ...
- Una entidad da lugar a una **tabla** en la BD.

Atributos

- Estas entidades están compuestas por varios **atributos**, también llamados **campos**, que vienen a ser sus propiedades. Por ejemplo, la entidad alumnos, tendrá los atributos nombre, ID, sexo, fecha de nacimiento, ...
- Los atributos también reciben el nombre de **columnas** en la terminología de BD

Elementos

- Cada entidad tendrá un número ilimitado de elementos. Por ejemplo, un elemento de la entidad alumnos será un alumno en sí; así el alumno Pedro será un elemento, José será otro, ...
- Cada uno de esos elementos también recibe el nombre de **fila** o **registro** en la terminología de BD

Identificador o Clave primaria

- Dada una entidad, de entre todos sus atributos, uno o varios son suficientes para identificar unívocamente a un elemento de los demás.
- Este atributo o conjunto de atributos es lo que denominamos **identificador** o **clave primaria**
- A menudo es un código creado expresamente para tal fin

Tablas

- Combinando estos tres conceptos tenemos una estructura del tipo tabla, la base de las BD.

Tabla ALUMNOS:

| Nombre | DNI | Nacionalidad | Fecha nacim. |
|--------|------------|--------------|--------------|
| Joseph | 99.999.999 | Irlanda | 2-2-1969 |
| Pepe | 88.888.888 | España | 8-8-1968 |
| José | 77.777.777 | Chile | 7-7-1967 |
| ... | | | |

Diagram illustrating the structure of a table (Tabla ALUMNOS) with columns and rows.

The table has 4 columns: **Nombre**, **DNI**, **Nacionalidad**, and **Fecha nacim.**.

The table has 4 rows (including the header row):

- Row 1: Joseph, 99.999.999, Irlanda, 2-2-1969
- Row 2: Pepe, 88.888.888, España, 8-8-1968
- Row 3: José, 77.777.777, Chile, 7-7-1967
- Row 4: ...

Arrows indicate the structure:

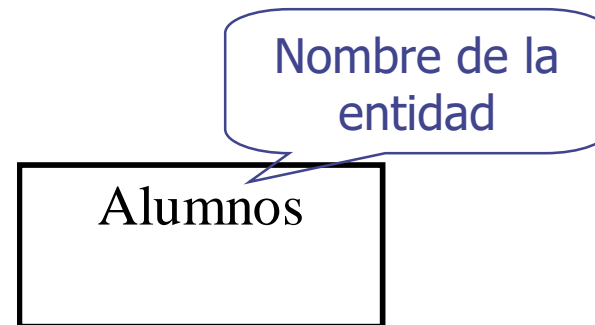
- Arrows pointing to the rows are labeled **Filas**.
- Arrows pointing to the columns are labeled **Columnas**.

Relaciones

- Las entidades no están aisladas sino que están relacionadas entre sí.
Alumnos vs Matrículas, Central vs Abonado
- Estas **relaciones** pueden ser de tres tipos diferentes:
 - 1 a 1
 - 1 a muchos (1 a N)
 - Muchos a muchos (M a N)

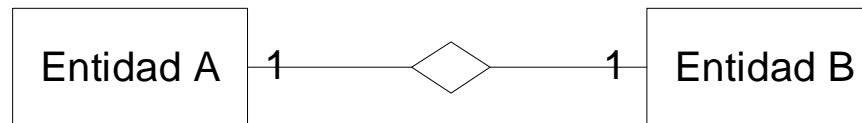
Representación del modelo

- Representaremos las entidades con cuadros con su nombre en el interior

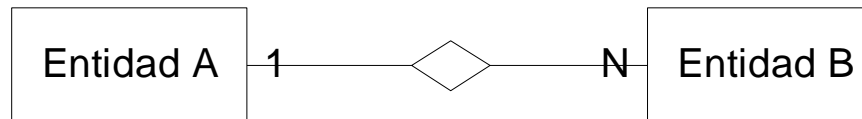


Representación del modelo

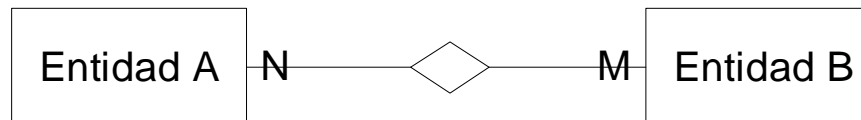
- Representaremos las relaciones así:
 - Relación 1:1



- Relación 1:N

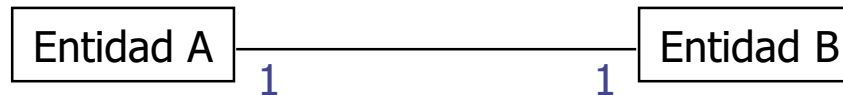


- Relación M:N

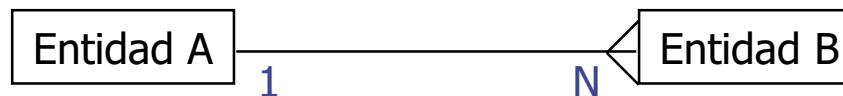


Representación del modelo

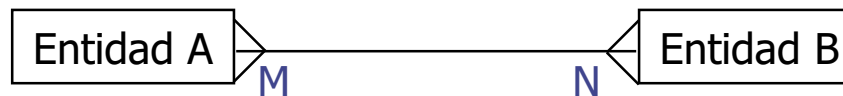
- Otra manera de representarlas es:
 - Relación 1:1



- Relación 1:N



- Relación M:N

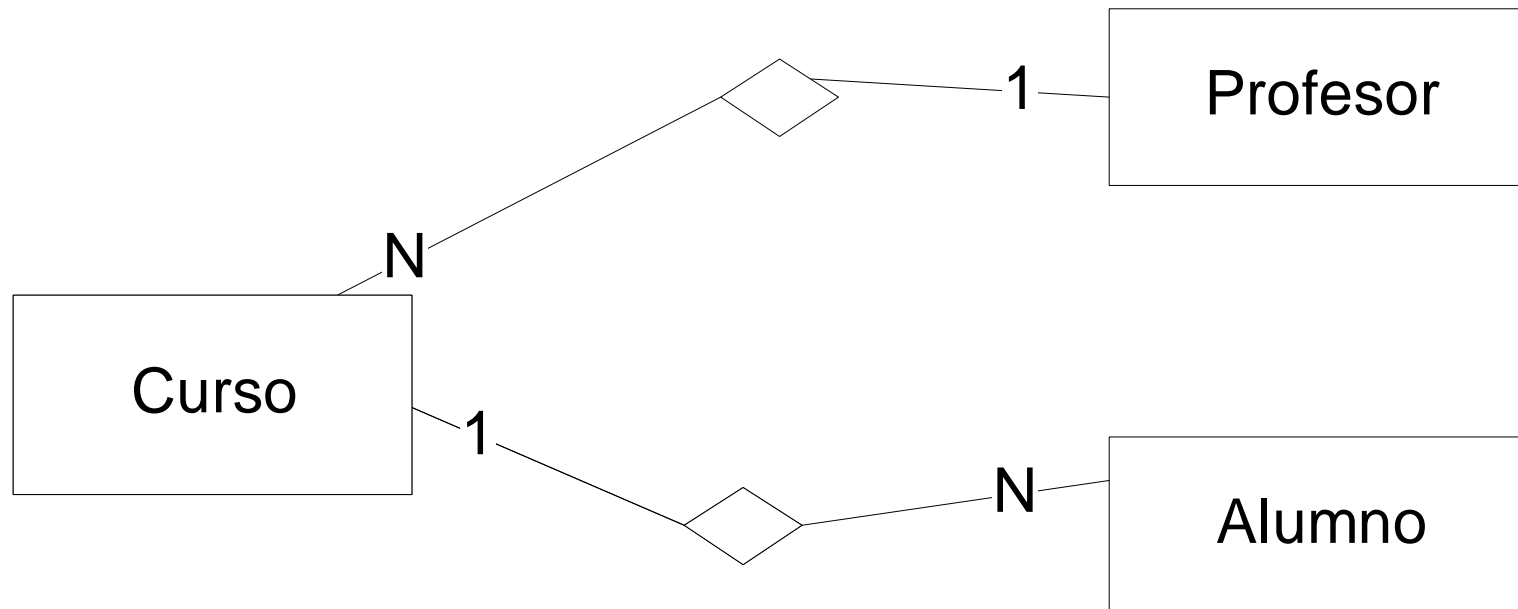




Ejercicio 1

- Hacer un modelo Entidad-Relación para la siguiente situación:
 - Tenemos una universidad, en la que hay varios cursos. Cada curso está dirigido por un profesor, el cual puede dirigir varios cursos. Los cursos son subvencionados, por lo que sólo se permite que un alumno se matricule de un curso.

Solución: Diagrama E/R





Solución: las tablas

Profesor:

| Código | Nombre | DNI | Fecha nac. |
|--------|------------|------------|------------|
| 1 | Don Pepito | 11.111.111 | 1-1-1950 |
| 2 | Don José | 22.222.222 | 2-2-1950 |

Curso:

| Código | Nombre | Horas | Código Profesor |
|--------|-----------|-------|-----------------|
| 1 | Word | 10 | 2 |
| 2 | Photoshop | 10 | 2 |
| 3 | Director | 10 | 2 |
| 4 | Internet | 30 | 1 |

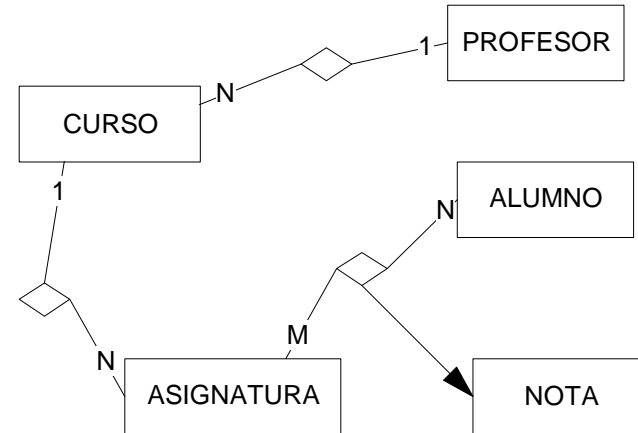
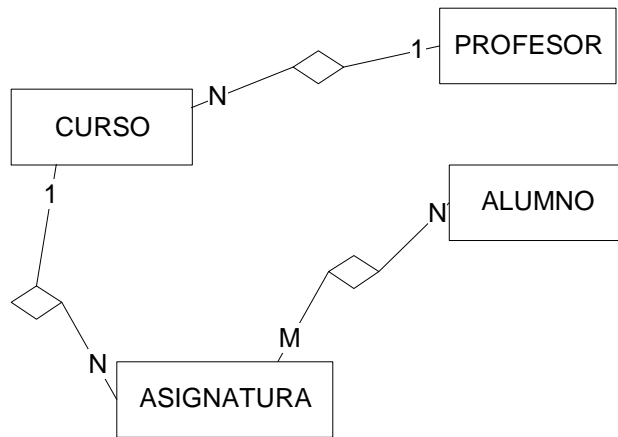
Alumno:

| Código | Nombre | DNI | Fecha nacim. | Código Curso |
|--------|---------|------------|--------------|--------------|
| 1 | Juan | 33.333.333 | 3-3-1970 | 1 |
| 2 | Pedro | 44.444.444 | 4-4-1970 | 1 |
| 3 | Antonio | 55.555.555 | 5-5-1970 | 2 |
| ... | ... | ... | ... | ... |

Ejercicio 2

- Compliquemos un poco la situación anterior:
 - Ahora suponemos que un curso está compuesto por varias asignaturas. Los alumnos se matriculan en las asignaturas que quieren (similar a la facultad). Por último, el alumno recibe una nota (calificación) para cada asignatura al final del curso.

Solución: Diagrama E/R



Las relaciones N:M implican la creación de una nueva entidad

Ejercicios para la casa

- Ejercicio 1

- Queremos hacer una base de datos con los CD's que tenemos en casa. Un CD puede tener un cantante o grupo, o varios. Además un CD tiene una lista de canciones.

Ejercicios para la casa

- Ejercicio 2
 - Vamos a complicar un poco el ejemplo anterior: ahora hemos de tener en cuenta que un disco está compuesto por canciones que pueden estar escritas por la misma persona que las canta, pero a menudo se trata de personas diferentes.

Ejercicios para la casa

- Ejercicio 3

- Imaginemos que hemos de hacer un sistema para el Mundial de Alemania 2006. 32 selecciones nacionales se han clasificado, y en cada una hay 22 jugadores. Hemos de poder reflejar todos los partidos que se disputan (con los resultados) y qué jugadores lo hacen por cada equipo, así como quien marca los goles.

Ya descansamos, continuemos...



Ojo!, sigue lo mas
importante!!!

Fundamentos de SQL



<http://www.mysql.com>

Aplicaciones y Servicios Telemáticos
Ing. Javier Alexander Hurtado

Popayán, Agosto de 2002



Fundamentos de SQL

- SQL es el lenguaje estándar utilizado para consultar las bases de datos relacionales
- Permite (además de opciones más avanzadas) crear, modificar o borrar tablas, así como insertar, eliminar, modificar o consultar los elementos de las tablas
- Lo más común es realizar consultas:
 - sentencia SELECT



SQL: Sentencia SELECT

```
SELECT nombre_de_columna  
FROM nombre_de_tabla  
WHERE condición;
```

- Ejemplo:

Queremos saber el código del profesor "Pepito Perez":

```
SELECT codigo  
FROM profesor  
WHERE nombre="Pepito";
```



SQL: SELECT ejemplos 1 y 2

- Queremos saber el código de la asignatura "lab3telco":
 - `SELECT asignatura_id FROM Asignatura WHERE nombre='lab3telco';`
- Queremos saber los nombres de todas las asignaturas que se dictan en los primeros 6 semestres:
 - `SELECT nombre FROM Asignatura WHERE semestre < 7;`

SQL: SELECT ejemplo 3

- Queremos saber los nombres de todos los alumnos que se matricularon durante los '80, es decir, que su fecha de matrícula está entre 1980 y 1989
 - `SELECT nombre FROM Alumno WHERE (anyo_matricula >= 1980) and (anyo_matricula <= 1989);`
- Nota: los paréntesis no son obligatorios, pero ayudan a la lectura.

SQL: SELECT ejemplos 4 y 5

- Queremos saber los nombres de todos los alumnos que se matricularon tanto durante la década de los '60 como durante los '80. Para ello necesitamos utilizar una OR entre las condiciones:
 - `SELECT nombre FROM Alumno WHERE ((anyo_matricula >= 1960) and (anyo_matricula <= 1969)) or ((anyo_matricula >= 1990) and (anyo_matricula <= 1989));`
- Queremos saber qué alumnos (todos los datos) están activos:
 - `SELECT * FROM Alumno WHERE activo='SI';`

SQL: SELECT ejemplo 6

- Queremos saber qué alumnos se encuentran matriculados en Ing. Electrónica (todos los datos). Primero buscaremos el programa de Ing. Electrónica y luego los alumnos cuyo programa_id coincida con este:
 - ```
SELECT Alumno.* FROM Alumno, Programa
WHERE
(Programa.nombre = 'Electronica') AND
(Alumno.programa_id = Programa.programa_id);
```

## SQL: otras sentencias

- **Insertar, borrar y modificar los datos de una tabla**

1) Insertar:

```
INSERT INTO Tipo VALUES ('AG','Agroturismo','Centros de
agroturismo y turismo rural');
```

(Inserta todos los valores de una fila de Tipo)

```
INSERT INTO Hotel (hotel_id, nombre) VALUES (7,'Arts');
```

(Inserta sólo algunos valores de una fila de Hotel)

2) Eliminar:

```
DELETE FROM Hotel;
```

(Borrar todas las filas de la tabla avión)

```
DELETE FROM Hotel WHERE codigo=5;
```

(Borra sólo las filas que cumplan una condición)

3) Modificar:

```
UPDATE Hotel SET nombre='Les Arts' WHERE hotel_id=7';
```

(Modificar el atributo nombre en todas las filas que cumplan la condición)



# SQL: Tablas

## Sintaxis SQL básica

- **Crear una tabla**

`CREATE TABLE [IF NO EXIST] nombre_tabla [()]`

**Ejemplo:**

`CREATE TABLE prueba`

- **Eliminar una tabla**

`DROP TABLE [IF EXIST] nombre_tabla`

**Ejemplo:**

`DROP TABLE prueba`

## SQL: Tablas

- Cómo crear una tabla

```
CREATE TABLE Municipio
(
 municipio_id CHAR(4) PRIMARY KEY,
 nombre VARCHAR(20) NOT NULL,
 provincia_id CHAR(2) REFERENCES
 Provincia(provincia_id)
)
```

- También hay sentencias para borrar (Drop table) una tabla y también para modificarla (Alter table), pero no las veremos aquí.



## Otras sentencias SQL

### Sintaxis SQL básica

- **Crear una BD**

`CREATE DATABASE [IF NO EXIST] nombre_bd`

**Ejemplo:**

`CREATE DATABASE prueba`

- **Eliminar una BD**

`DROP DATABASE [IF EXIST] nombre_bd`

**Ejemplo:**

`DROP DATABASE prueba`

# **Fundamentos de Bases de Datos y MySQL**

## **GRACIAS POR SU ATENCIÓN FIN**

### **Preguntas??**