

# PRACTICAS CON MySQL

Se verán los siguientes apartados:

1. Introducción
2. El entorno de MySQL
3. Tablas de ejemplo y ejercicios de laboratorio

## 1.- INTRODUCCIÓN

El objetivo de los trabajos prácticos que se sugieren es que el alumno aplique los conocimientos teóricos que ha aprendido en clase y que se familiarice con los recursos que ofrece una base de datos relacional en el entorno MySQL haciendo uso del software MySQL Workbench.

Para ello se presenta primero un apartado sobre el manejo de instrucciones y sentencias en MySQL y a continuación un conjunto de tablas de ejemplo y ejercicios sencillos para que el alumno vaya practicando los conocimientos que va adquiriendo.

## 2.- ENTORNO MySQL

**MySQL** es un gestor de base de datos relacional. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.

Los datos de una BD relacional están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación **SQL** (Structured Query Language).

**MySQL Server** es un servidor de bases de datos muy rápido, seguro y fácil de usar. Gracias a la colaboración de muchos usuarios la base de datos se ha ido mejorando optimizando su velocidad. Por eso es una de las bases de datos más usadas en Internet. Existe una gran cantidad de software que la usa.

**MySQL Workbench**, por su parte, es una herramienta visual que nos permite conectarnos a un servidor MySQL y realizar consultas o ejecutar sentencias contra él además de administrar el servidor.

### 2.2.1. INSTALAR MySQL SERVER

La instalación de MySQL Server es muy sencilla. Primero accederemos al apartado de descargas del sitio web oficial de MySQL <http://www.mysql.com/downloads/> y elegimos la primera opción, que es **MySQL Community Server** (es suficiente para lo que de momento vamos a hacer). Existen versiones más complejas e incluso herramientas visuales, pero primero debemos entender la programación y utilización básica de una base de datos para luego ir mejorando.

En esta página descargamos la versión apropiada para nuestro Sistema Operativo (Windows, generalmente de 64 bits) optando por el instalador MSI. A partir de aquí la instalación es muy sencilla a través de un asistente. Para que nuestra instalación tenga éxito debemos estar atentos a los siguientes pasos.

- En la pantalla de bienvenida pulsamos **Next**.
- Aceptamos las condiciones de uso y **Next**.
- Elegimos la instalación **Typical**.
- En la siguiente ventana confirmamos que estamos preparados para la instalación pulsando **Install**.
- Esperamos mientras se instala el sistema. Windows puede solicitar permiso de administrador para proceder a la instalación.
- Se abre una pantalla para registrarnos en MySQL.com, podemos ignorarlo cerrando con la X. En caso de que queramos registrarnos ya lo haremos más adelante.
- Una vez que aparece la pantalla de instalación completada nos aseguramos de que la casilla **Launch the MySQL Instance Configuration Wizard** está seleccionada y hacemos clic en **Finish**.

En la pantalla siguiente empieza lo realmente importante.

- Comenzamos el asistente, hacemos clic en **Next**.
- Pasamos a la ventana de configuración.
- Seleccionamos **Standard Configuration** y hacemos clic en **Next**.
- Estamos en la ventana de las opciones, en la que tendremos que seleccionar **Install as a Windows Service**, mantenemos el nombre del servicio como **MySQL**, verificamos que **Launch the MySQL Server automatically** está marcado, y por ultimo seleccionamos **Include Bin Directory in Windows PATH**. Hacemos clic en **Next**.
- Llegamos a la ventana de las opciones de seguridad, escogemos la contraseña del usuario *root*, por ejemplo *eside*, que deberemos recordar, y hacemos clic en **Next**.
- En la ventana de ejecutar clicamos en **Execute**.
- Por ultimo clicamos en **Finish**, y ya tenemos nuestro servidor MySQL.

Los usuarios del Sistema Operativo GNU/Linux, en el caso de Ubuntu simplemente deberán buscar en el “Centro de Software de Ubuntu” o en el gestor de paquetes Synaptic un paquete con el nombre **MySQL Server** e instalarlo. Al finalizar la instalación pedirá un usuario y contraseña, o ejecutar el comando “**mysql -u root -p**” e introducirla cuando nos la pida.

## 2.2.2. INSTALAR MySQL WORKBENCH

La instalación de MySQL Workbench es aún más sencilla. En la misma web <http://www.mysql.com/downloads/mysql/> seleccionamos **MySQL Workbench** y descargamos la versión apropiada para nuestro Sistema Operativo y plataforma.

La instalación es incluso más rápida que la del servidor MySQL.

Nota. En caso de que el instalador falle porque falta algún software requerido (*Microsoft .NET Framework 4 Client Profile* o *Microsoft Visual C++ 2010 Redistributable*

*Package*) podemos descargar lo que nos falte desde los enlaces que nos facilita el sitio web de MySQL.

- En la pantalla de bienvenida pulsamos Next.
- Dejamos la ruta por defecto. Next.
- Escogemos la instalación completa. Next.
- Pulsamos en Install y esperamos a que se termina de instalar. Windows puede solicitar permiso de administrador para proceder a la instalación.
- Una vez instalado pulsamos en Finish para cerrar el asistente. La aplicación se iniciará por defecto si no hemos desmarcado la casilla *Launch MySQL Workbench now*.

### 2.2.3. PRIMEROS PASOS CON MySQL DESDE LA LÍNEA DE COMANDOS

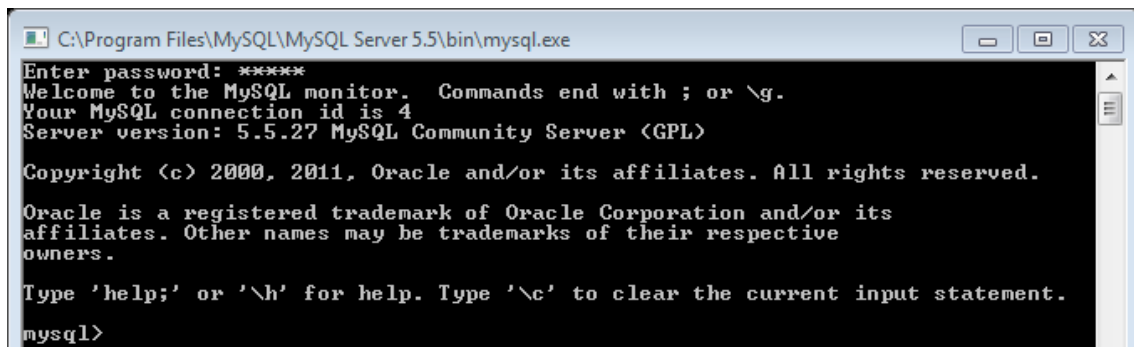
En MySQL todas las sentencias se pueden ejecutar desde la línea de comandos.

Podemos abrir una consola tecleando Windows+R y ejecutando:

*C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql.exe -uroot -p*

Otra opción más cómoda es escribir directamente el comando *mysql.exe -uroot -p* desde la consola de Windows.

La contraseña del usuario *root* (que se inicia por defecto) en los laboratorios es *eside*.



Siempre al final de cada comando escribiremos **punto y coma (;)** para que sepa que le estamos haciendo una **petición de ejecución**. Por esto mismo un comando no necesita ser escrito en una sola línea: MySQL determina que la sentencia finaliza cuando encuentra el punto y coma, no cuando encuentra el fin de línea.

Lo primero que debemos hacer es mirar las bases de datos que MySQL nos crea por defecto para acordarnos que nunca debemos borrarlas accidentalmente. Para consultar las bases de datos almacenadas escribimos en la línea de comandos: **SHOW DATABASES;**

El servidor nos muestra lo siguiente:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| BaseDatosWeb |
| mysql |
+-----+
3 rows in set (0.02 sec)
```

Para crear otra base de datos para que podamos usarla para meter nuestras tablas usaremos el comando **CREATE DATABASE** seguido del nombre de la base de datos que queramos crear.

```
mysql> CREATE DATABASE INGENIERIA;
Query OK, 1 row affected (0.08 sec)
```

Una vez creada nuestra base de datos el siguiente paso es seleccionarla para poder usarla y crear tablas dentro; esto se hace desde el comando **USE** seguido del nombre de la base de datos.

```
mysql> USE INGENIERIA;
Database changed
```

Ahora ya podemos crear tablas dentro de nuestra base de datos y consultarlas. Pero primero podemos hacer una consulta para saber la versión que hemos instalado, y el día y la hora que es. Escribimos el siguiente comando: **SELECT VERSION(), CURRENT\_DATE, NOW();**

```
mysql> SELECT VERSION(), CURRENT_DATE, NOW();
+-----+-----+-----+
| VERSION() | CURRENT_DATE | NOW() |
+-----+-----+-----+
| 5.1.41-3ubuntu12.10 | 2011-04-21 | 2011-04-21 10:54:53 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

A través de la línea de comandos podemos introducir cualquier sentencia de SQL estándar y obtendremos el resultado por pantalla.

MySQL también incluye una librería para hacer operaciones matemáticas y trigonométricas. Por ejemplo, podríamos escribir **SELECT COS(PI()/5), (19\*7)/3;**

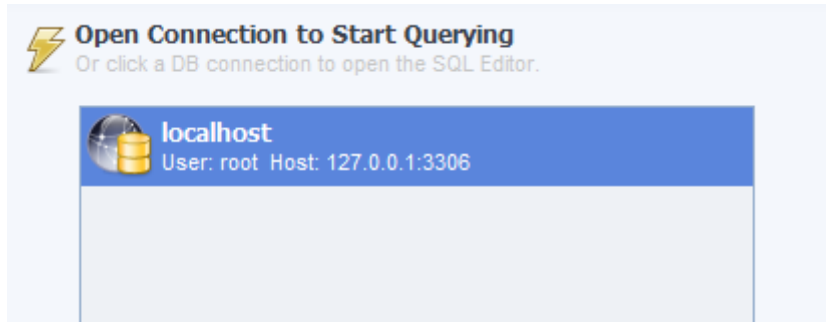
```
mysql> SELECT COS(PI()/5), (19*7)/3;
+-----+-----+
| COS(PI()/5) | (19*7)/3 |
+-----+-----+
| 0.809016994374947 | 44.3333 |
+-----+-----+
1 row in set (0.00 sec)
```

## 2.2.4. PRIMEROS PASOS CON MySQL Workbench

En los ordenadores de los laboratorios se puede abrir MySQL Workbench entrando en *Inicio > Todos los programas > MySQL > MySQL Workbench 5.2 CE.*

Lo primero que debemos hacer es crear una **conexión al servidor MySQL**. Encontramos la opción *New Connection* abajo a la izquierda, y en la pantalla de configuración establecemos un nombre para la conexión (por ejemplo *localhost*), pulsamos en “*Store in Vault...*” para guardar la contraseña del usuario *root* (por defecto *eside* en los laboratorios) y pulsamos en OK dejando los demás parámetros sin modificar.

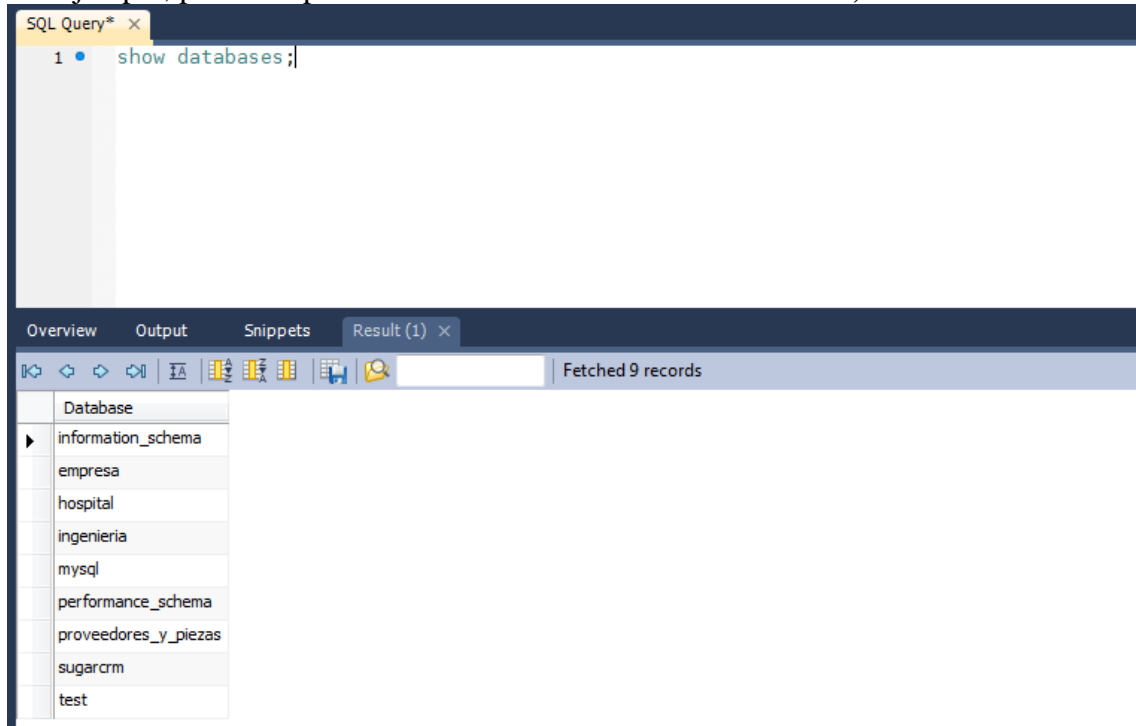
Ahora ya tenemos la conexión creada en la parte izquierda de la pantalla principal de la aplicación.



Con doble clic abrimos la pantalla *SQL Editor* que proporciona una **interfaz mucho más potente que la línea de comandos** para trabajar con sentencias SQL contra el mismo servidor.

La zona de *SQL Query* es la que permite introducir una o varias sentencias SQL.

Por ejemplo, podemos probar el comando **SHOW DATABASES;**



Desde aquí se pueden ejecutar las mismas sentencias que en la línea de comandos pero con una interfaz mucho más potente que nos permite ejecutar una serie de sentencias seguidas, abrir archivos SQL, guardar nuestros propios ficheros SQL...

## 2.2.5. EL DICCIONARIO DE DATOS

El soporte para **INFORMATION\_SCHEMA** está disponible en MySQL 5.0.2 y posterior. Proporciona acceso a los metadatos de la base de datos.

*Metadatos* son datos acerca de los datos, tales como el nombre de la base de datos o tabla, el tipo de datos de una columna o permisos de acceso. Otros términos que a veces se usan para esta información son *diccionario de datos* o *catálogo del sistema*.

**INFORMATION\_SCHEMA** es la base de datos de información, que almacena información acerca de todas las otras bases de datos que mantiene el servidor MySQL. Dentro del **INFORMATION\_SCHEMA** hay varias tablas de solo lectura. En realidad son vistas, no tablas, así que no se puede ver ningún fichero asociado con ellas.

Cada usuario MySQL tiene derecho a acceder a estas tablas, pero sólo a los registros que se corresponden a los objetos a los que tiene permiso de acceso.

El comando **SELECT ... FROM INFORMATION\_SCHEMA.[TABLE]** es una forma más consistente de proporcionar acceso a la información proporcionada por los comandos **SHOW** que soporta MySQL (**SHOW DATABASES**, **SHOW TABLES** y demás). Usar **SELECT** tiene las siguientes ventajas, en comparación a **SHOW**:

- Cumple las reglas de Codd. Esto es, todo acceso se hace por tabla.
- Nadie necesita aprender una nueva sintaxis. Una vez conocen cómo funciona **SELECT** los usuarios solo necesitan aprender los nombres de los objetos.
- El implementador no tiene que preocuparse de palabras clave.
- Hay millones de variaciones de la salida en lugar de solo una. Esto proporciona flexibilidad a las aplicaciones con requerimientos cambiantes acerca de los metadatos que necesitan.
- La migración es más fácil ya que todos los DBMS funcionan así.

Sin embargo, como **SHOW** es popular entre los empleados y usuarios de MySQL, y como puede ser confuso si desaparece, las ventajas de una sintaxis convencional no es razón para eliminar **SHOW**.

### Comando SHOW

**SHOW** puede usarse para obtener información acerca de la estructura de **INFORMATION\_SCHEMA**.

Varios comandos **SHOW** aceptan una cláusula **WHERE** que proporciona más flexibilidad al especificar qué registros mostrar.

```
mysql> SHOW TABLES FROM INFORMATION_SCHEMA;
```

**SHOW COLUMNS** y **DESCRIBE** pueden mostrar información acerca de las columnas en tablas **INFORMATION\_SCHEMA** individuales.

```
SHOW COLUMNS
SHOW DATABASES
SHOW TABLES
```

## 2.2.6. UTILIZACIÓN DE ARCHIVOS

### 2.2.6.1. Guardar la salida de una sesión MySQL en un fichero

Algunas veces conviene guardar la salida de alguna consulta a la base de datos MySQL para poder consultar el resultado posteriormente. Vamos a ver cómo usar el comando **tee**.

Para empezar a almacenar la salida de los comandos SQL en un fichero podemos usar indistintamente **\T** o bien **tee**:

```
mysql> \T ejemplo
Logging to file 'ejemplo'
```

Es decir, se crea un fichero de texto con el nombre *ejemplo.txt*.

A continuación podemos ejecutar los comandos que queramos. Todo lo que tecleemos y lo que se visualice por pantalla, quedará registrado como texto en el fichero.

Para deshabilitar el **tee** podemos usar el comando **notee** o **\t**:

```
mysql> \t
Outfile disabled.
```

Cuando acabe con una sesión, todos los archivos creados con la instrucción **tee**, se cerrarán automáticamente. Podremos consultar estos ficheros de texto desde el sistema operativo.

### 2.2.6.2. Ejecutar sentencias SQL desde un fichero de texto

#### ¿Qué es un archivo de comandos?

Un conjunto de comandos de SQL almacenados en un archivo del sistema operativo, y que se ejecutan secuencialmente al lanzar la ejecución del archivo.

El cliente **mysql** se usa normalmente de forma interactiva, como se muestra a continuación:

```
shell> mysql nombre_base_de_datos
```

Sin embargo, es posible poner comandos SQL en un fichero y decirle a **mysql** que lea las entradas de ese fichero. Para ello, cree un fichero de nombre **fichero\_de\_texto** que contenga el comando que quiera ejecutar. Luego invoque **mysql** como se muestra aquí:

```
shell> mysql nombre_base_de_datos < fichero_de_texto
```

Puede comenzar su fichero de texto con un comando **USE nombre\_base\_de\_datos**. En este caso, no es necesario especificar el nombre de la base de datos en la línea de comandos:

```
shell> mysql < text_file
```

Si está ejecutando **mysql**, puede ejecutar un script SQL en un fichero usando el comando **source** o **\.** :

```
mysql> source nombre_de_fichero
mysql> \. nombre_de_fichero
```

Si desea que el script muestre información de progreso al usuario, puede insertar algunas líneas como

```
SELECT '<información>' AS ' ';
```

que muestra <información>.



### 3.- TABLAS DE EJEMPLO Y EJERCICIOS

#### 3.3.1.- DESCRIPCIÓN DE LAS TABLAS

##### Base de datos de EMPRESA

**Tabla EMP**

EMP_NO	Número del empleado (4 dígitos numéricos) PK
APELLIDO	Apellido (16 caracteres)
OFICIO	Oficio (10 caracteres)
DIR	Número del director o jefe del empleado (4 dígitos numéricos)
FECHA_ALT	Fecha de alta en la empresa (tipo fecha)
SALARIO	Salario mensual (9 dígitos numéricos)
COMISION	Comisión (9 dígitos numéricos)
DEPT_NO	Número de departamento donde trabaja (2 dígitos numéricos, dato obligatorio) FK

**Tabla DEPT**

DEPT_NO	Número del departamento PK
DNOMBRE	Nombre del departamento (14 caracteres)
LOC	Localización del departamento (13 caracteres)

##### Base de datos de HOSPITAL

**Tabla HOSPITAL**

HOSPITAL_COD	Código del Hospital (2 dígitos numéricos) PK
NOMBRE	Nombre (12 caracteres)
DIRECCION	Dirección (20 caracteres)
TELEFONO	Teléfono (8 caracteres)
NUM_CAMA	Número de camas del Hospital (4 dígitos numéricos)

**Tabla DOCTOR**

HOSPITAL_COD	Código del Hospital donde trabaja el médico (2 dígitos numéricos) FK
DOCTOR_NO	Número del doctor (3 dígitos numéricos) PK
APELLIDO	Apellido (16 caracteres)
ESPECIALIDAD	Especialidad (16 caracteres)

**Tabla ENFERMO**

INSCRIPCION	Número de inscripción del enfermo (5 dígitos numéricos) PK
APELLIDO	Apellido (12 caracteres)
DIRECCION	Dirección (20 caracteres)
FECHA_NAC	Fecha de nacimiento (tipo fecha)
S	Sexo (1 carácter) Puede ser: M: Masculino F: Femenino
NSS	Número de Seguridad Social
DOCTOR_NO	Número del doctor (3 dígitos numéricos) FK

**Tabla SALA**

HOSPITAL_COD	Código del Hospital (2 dígitos numéricos) PK -FK
SALA_COD	Código de la sala (2 dígitos numéricos) PK
NOMBRE	Nombre de la sala (20 caracteres)
NUM_CAMA	Número de camas de la sala (4 dígitos numéricos)

**Tabla PLANTILLA**

HOSPITAL_COD	Código del Hospital (2 dígitos numéricos) FK – PK
SALA_COD	Código de la sala (2 dígitos numéricos) FK - PK
EMPLEADO_NO	Número del empleado trabajando en esa sala (4 dígitos numéricos) PK
APELLIDO	Apellido del empleado (16 caracteres)
FUNCION	Función desempeñada por el empleado (10 caracteres)
TURNOS	Turno de trabajo (1 carácter). Puede ser: M: Mañana T: Tarde N: Noche
SALARIO	Salario anual (10 dígitos numéricos)

### 3.3.2.- EJERCICIOS DE SQL

#### 1<sup>er</sup> LABORATORIO

##### **Objetivos del primer laboratorio:**

- Aprender a crear tablas e índices
- Aprender a cargar datos en una tabla
- Aprender a solucionar los errores producidos durante la creación y carga de una tabla
- Aprender a utilizar el diccionario de datos y comprender su utilidad

##### **CREACION DE TABLAS**

1. Crear una base de datos DOCENCIA y a continuación crear una tabla PROFESOR con dos columnas:
  - DNI: numérico, clave primaria
  - NOM\_P: alfanumérico (10)

Una vez creada la tabla, usar el comando DESCRIBE para consultar su estructura.

---

---

---

---

2. Realizar las siguientes modificaciones en la tabla profesor:
  - cambiar la longitud del campo NOM\_P a 30 caracteres.
  - añadirle una nueva columna, N\_HORAS, de tipo numérico
  - añadirle una nueva columna SALARIO de tipo numérico.

A continuación consultar la estructura de la tabla para comprobar que realmente se ha alterado.

---

---

---

---

3. Crear una nueva tabla ASIG con 4 columnas:

- COD\_A numérico y clave primaria
- NOM\_A alfanumérico 10
- CRED 1 entero y 1 decimal
- DNI clave externa que referencia a PROFESOR, con borrado en cascada

---

---

---

---

4. Usar el comando SHOW para ver que tablas tenemos en la base de datos DOCENCIA, y si entre ellas está la nueva tabla creada.

---

---

---

5. Intentar eliminar de la BD la tabla PROFESOR. ¿Se realiza correctamente? En caso de error ¿por qué se produce? En este último caso, realizar las operaciones necesarias para eliminar la tabla, y comprobar que efectivamente ha desaparecido.

---

---

---

6. Crear un fichero de comandos CREAM.SQL conteniendo dos instrucciones SQL que vuelvan a crear las tablas PROFESOR y ASIG. Ejecutar el fichero de comandos CREAM. Comprobar el resultado de su ejecución.

---

---

---

---

## CREACION DE INDICES

7. Emplear la sentencia `SHOW` para ver qué índices hay actualmente definidos y sobre las tablas `PROFESOR` Y `ASIG`.

---

---

8. Crear un nuevo índice, `X_NOM_P`, sobre el nombre del profesor de la tabla `PROFESOR`. A continuación comprobar que se ha incorporado su definición usando la sentencia `SHOW`.

---

---

## CARGA DE LA BD (inserción de filas en las tablas)

9. Insertar en la tabla `PROFESOR` las siguientes filas:

- 111/J.García/10/700
- 222/P.Jimenez/12/1200
- 333/M.Alvarez/8/500

Consulta luego la tabla para comprobar su contenido.

---

---

---

10. Insertar en la tabla `ASIG` las siguientes filas:

- 01/BD/6/111
- 02/EMPRESA/6/444
- 03/ESTADISTICA/4.5/333

¿Se ha producido algún error? ¿Por qué? ¿Cómo solucionarlo?

Consultar ambas tablas para comprobar que se han cargado correctamente.

---

---

---

---

11. Crear una nueva tabla PRACTICAS con las siguientes columnas:
- COD\_P numérico clave primaria (código de práctica)
  - NOM\_P alfanumérico 15 (nombre de la práctica)
  - COD\_A numérico clave externa que referencia a ASIGNATURA (código de asignatura)

Se deberá asignar un nombre a cada restricción de integridad definida

---

---

---

12. Insertar las siguientes filas en la tabla PRACTICAS

- 001/MANEJO SQL/01
- 002/EJ.CONTAB/02

---

---

---

13. Eliminar la asignatura BD. ¿Se realiza correctamente? ¿Por qué? Realizar los pasos adecuados para conseguirlo.

---

---

---

---

14. Eliminar el profesor 333. ¿Se realiza correctamente? ¿Qué pasa con las asignaturas de dicho profesor, continúan en la BD o han desaparecido también? ¿Por qué?

---

---

---

---

## **CONTROL DE LOS CAMBIOS (COMMIT y ROLLBACK)**

15. Lanza una sentencia COMMIT y consulta el contenido de la tabla PROFESOR. A continuación elimina el profesor cuyo nombre es J.GARCIA. Consulta la tabla PROFESOR para comprobar que efectivamente la fila correspondiente desaparece. Después deshaz la transacción emitiendo ROLLBACK. Consulta de nuevo la tabla PROFESOR para comprobar que la transacción ha quedado deshecha.

---

---

---

## **CONSULTAS AL DICCIONARIO DE DATOS**

16. Consultar en el diccionario de datos los nombres de las tablas de la base de datos DOCENCIA y sus características.

---

---

17. Consultar los nombres y tipos de datos de las columnas de la tabla Profesor.

---

---

18. Consultar los índices definidos sobre las tablas de la base de datos DOCENCIA visualizando el nombre del índice y el nombre de la tabla.

---

---

19. Consultar las restricciones de integridad definidas sobre las tablas de la base de datos DOCENCIA.

---

---

## **2º LABORATORIO**

### **Objetivos del segundo laboratorio:**

- Aprender a realizar operaciones de consulta sencillas
- Aprender a utilizar funciones y la cláusula ORDER BY

### **RECUPERACION DE DATOS (SELECT) USANDO LAS TABLAS DE EJEMPLO**

20. Obtener todos los oficios de empleados de la BD Empresa.

---

---

21. Obtener todos los oficios de empleados de la BD Empresa sin valores repetidos.

---

---

22. Obtener los datos de todos los empleados del hospital cuyo apellido empiece por H.

---

---

23. Obtener todos los datos de los empleados del hospital cuyo número de empleado empiece por 3.

---

---

24. Obtener todos los datos de los empleados del hospital cuyo número de empleado no empiece por 3.

---

---



## **FUNCIONES ARITMETICAS**

25. Obtener el valor absoluto de  $-3$ .

---

---

26. Calcular el valor de la operación  $5+5$

---

---

27. Obtener entero igual o inmediatamente superior a 4

---

---

28. Obtener el entero igual o inmediatamente superior a 4,5

---

---

29. Obtener el valor de  $2^4$

---

---

## **FUNCIONES PARA CADENAS DE CARACTERES**

30. Obtener cuántos caracteres tiene el literal 'hola'

---

---

31. Poner a mayúsculas el literal 'hola'

---

---

### **FUNCIONES PARA MANEJO DE FECHAS**

32. Obtener la fecha de hoy del sistema

---

---

33. Obtener el nombre del día de la fecha de hoy del sistema

---

---

34. Obtener cuántos meses hay entre el 1/4/99 y el 2/5 del 99

---

---

### **FUNCIONES DE COLUMNA**

35. Calcular el salario medio de los empleados del hospital mediante una función.

---

---

36. Calcular el salario medio de los empleados del hospital como cociente de 2 funciones.

---

---

37. Obtener cuál es el salario máximo y el salario mínimo de los empleados del hospital.

---

---

### **OTRAS FUNCIONES**

38. Obtener la comisión de los empleados visualizando el valor “N/A” para aquellos que no tengan.

---

---

### **USO DE ORDER BY**

39. Obtener el apellido y salario de los empleados de la tabla EMP ordenando el resultado en descendente por salario.

---

---

40. Obtener el apellido y el salario anual de todos los empleados de la tabla EMP, dando el resultado ordenado en ascendente por salario anual. En esta columna deberá ponerse el título SALARIO ANUAL.

---

---

---

### **CONSULTAS VARIADAS**

41. Seleccionar el nombre y código de todos los hospitales.

---

---

42. Recuperar todo el personal ordenado por el turno en que trabaja, indicando el nombre, función y turno.

---

---

---

43. Mostrar los apellidos y salarios en orden descendente de salario y ascendente de función.

---

---

44. Mostrar los apellidos y salarios de los enfermeros cuyo sueldo esté entre 2 y 2.5 millones, ordenados por código de hospital descendente.

---

---

45. Hallar el número de días que han pasado desde el 28/03/64.

---

---

46. Encontrar todos los enfermeros y enfermeras con indicación del salario mensual de cada uno.

---

---

47. Hallar la fecha de hace 3 semanas.

---

---

48. Encontrar a todos los miembros de la plantilla del hospital cuyo nombre contenga al menos una "a".

---

---

49. ¿Quiénes son las enfermeras y enfermeros que trabajan en turnos de tarde o mañana?

---

---

50. Encontrar el salario medio de los internos.

---

---

51. Obtener las funciones distintas que tienen los empleados de la plantilla del hospital.

---

---

52. Hallar las funciones y turnos distintos del hospital.

---

---

53. Mostrar, para todos los hospitales, el código de hospital, el nombre completo del mismo, y su nombre abreviado de 3 letras. Ordenar la recuperación por esta abreviatura.

---

---

54. Hacer un listado de los nombres de los pacientes y la posición de la primera letra "A" que aparezca en su apellido.

---

---

55. Encontrar el salario más alto y más bajo de la tabla de plantilla, y la diferencia entre ambos.

---

---

56. Obtener cuántas funciones distintas que hay en total en las salas 1 y 2 del hospital 22.

---

---

### **3<sup>er</sup> LABORATORIO**

#### **Objetivos del tercer laboratorio:**

- Aprender a realizar consultas que incluyan el acceso a varias tablas (uso de operadores de conjunto, join y subconsultas)

#### **OBTENCIÓN DE DATOS DE VARIAS TABLAS (OPERADOR UNION)**

57. Obtener los empleados que sean directores o tengan comisión utilizando la unión de dos select, y mediante una única select.

---

---

#### **OBTENCION DE DATOS DE VARIAS TABLAS (JOIN) Y COMBINACION DE DATOS DE UNA SOLA TABLA**

58. Obtener los apellidos de los empleados que ganen más que Fernández.

---

---

59. Obtener los apellidos de los doctores que trabajen en hospitales que tengan más de 600 camas

---

---

#### **USO DE SUBQUERIES o SUBCONSULTAS**

60. Obtener los apellidos de empleados que trabajan en el dpto de *VENTAS*. Plantearlo de 2 formas, como join y como subconsulta.

---

---

61. Obtener los apellidos de los enfermos que hayan sido atendidos por doctores especialistas en pediatría. Plantearlo de 2 formas, como join y como subconsulta.

---

---

62. Sacar los nombres de la plantilla cuyo sueldo sea superior a la media.

---

---

63. Conseguir los nombres y oficios de los empleados del departamento 20, cuyo trabajo sea idéntico al de cualquiera de los empleados del departamento de *VENTAS*.

---

---

### **CONSULTAS VARIADAS CON VARIAS TABLAS**

64. Mostrar todos los apellidos de los empleados que trabajan en *MADRID* o *BARCELONA*, ordenados por apellido.

---

---

65. Obtener la lista de los empleados de la empresa con el mismo oficio y salario que *JIMENEZ*. Hacerlo de varias formas.

---

---

66. Obtener los apellidos y oficios de los empleados que ganan más que cualquier otro del departamento 30.

---

---



67. Mostrar los apellidos de los empleados con el mismo oficio que *JIMENEZ* y salario mayor o igual al de *FERNANDEZ*. Hacerlo de varias formas.

---

---

68. Listar los nombres de los hospitales que tengan sala de *RECUPERACION*.

---

---

69. Obtener un listado con los nombres de los hospitales y los apellidos y especialidades de los doctores que trabajan en cada uno, ordenando el resultado por nombre de hospital en ascendente y apellido en descendente.

---

---

70. Actualizar los datos del paciente *10995* asignándole el mismo apellido que el almacenado para el doctor *453*.

---

---

71. En la tabla *EMP* recuperar todas las personas que trabajan para *NEGRO*, excepto *TOVAR* y *JIMENO*.

---

---

72. Obtener el n° de inscripción y el apellido de los enfermos que han sido atendidos por un especialista en *CARDIOLOGIA*. Hacer la codificación de 3 formas distintas.

---

---

## **4º LABORATORIO**

Objetivos del cuarto laboratorio:

- Codificar consultas que incluyan GROUP BY
- Aprender a utilizar vistas y sinónimos
- Experimentar los problemas que pueden surgir en la actualización de vistas

### **USO DE GROUP BY**

73. Obtener el n°dpto y el salario medio de cada dpto.

---

---

74. Obtener el n°dpto, nombre de dpto y el salario medio de cada dpto.

---

---

75. Obtener el salario medio de cada oficio de los empleados

---

---

76. Obtener el salario medio de cada oficio de los empleados, siempre que el salario medio sea superior a 2000

---

---

77. Obtener el salario medio de cada oficio de los empleados, siempre que el salario medio sea superior a 2000, excluyendo a los directores

---

---

78. Calcular el número de personas de la plantilla del hospital que realizan cada función en cada turno.

---

---

79. Buscar qué turnos tienen más de dos personas realizando una función concreta, mostrando turnos y funciones.

---

---

80. Obtener los apellidos de los doctores que tienen 2 o más pacientes a su cargo.

---

---

### **CREACION DE TABLAS COMO RESULTADO DE UNA CONSULTA**

81. De la tabla de *ENFERMOS*, crear una tabla *VARON* que contenga los enfermos varones, conservando la misma estructura de la tabla.

---

---

---

82. Crear una tabla *VENDEDOR* con todos los empleados que sean vendedores. Consultar después su estructura y contenido.

---

---

---

## VISTAS

83. Crear una vista e1 que contenga el nºemp, apellido, oficio y salario de los empleados que ganan más de 2500. Consultar luego el diccionario de datos para comprobar que se ha incluido en la BD. Consultar también el contenido de la vista.

---

---

84. Crear una vista e2 que contenga el apellido, oficio y salario de los empleados que ganan más de 2500

---

---

85. Insertar una fila en e2 con los valores ARANA/VENDEDOR/3000. ¿Se ejecuta correctamente? ¿Por qué?

---

---

---

86. Eliminar de la BD la vista e2.

---

---

87. Insertar las siguientes filas en e1:

- 1111/ARANA/VENDEDOR/3000
- 7934/AGUIRRE/VENDEDOR/4000
- 2222/AGUIRRE/ANALISTA/1500

¿Se produce algún error? ¿Cuál es el motivo?

Al final consulta el contenido de e1 y el de emp. Compáralos.

---

---

---

## ACTUALIZACIÓN DE VISTAS

88. Modifica el salario del empleado 1111 a 2400 en la vista e1. ¿Qué efectos tiene sobre la vista e1 y sobre la tabla emp? Consúltalas para averiguarlo.

---

---

---

## CONSULTAS CON VISTAS

89. Obtener cuántos enfermos tiene a su cargo cada uno de los doctores que trabajan en el hospital de la paz: Crear primero una vista con la información de los doctores de La Paz y los enfermos a su cargo. Consultar luego la vista obteniendo el n° y apellido del doctor, y el n° de enfermos a su cargo.

---

---

---

90. Obtener por cada hospital, cuántos empleados trabajan en el turno de tarde y cuál es su sueldo medio dando n° hospital nombre de hospital, n° empleados en el turno T y salario medio. Crear 1° una vista con los hospitales y los empleados que trabajan en el turno de tarde. Consultar luego la vista obteniendo los datos pedidos.

---

---

---

## 5º LABORATORIO

### Objetivos del sexto laboratorio:

- Que el/la estudiante sea capaz de dar todos los pasos necesarios para utilizar de una base de datos a partir de un diseño inicial: creación, carga y utilización.

Sea la siguiente base de datos para llevar el control de las ventas en una cadena de supermercados durante el último año. La cadena consta de varios supermercados ubicados en diferentes ciudades del país, pudiendo haber varios en la misma ciudad.

CIUDAD (COD\_CIUDAD, NOM\_CIUDAD)

SUPERMERCADO (COD\_SUP, DIRECCION, TFNO, COD\_CIUDAD)

PRODUCTO (COD\_PROD, DENOMINACION, PRECIO)

VENTAS (COD\_SUP, COD\_PROD, CANT)

Se pide codificar en SQL las siguientes operaciones. En caso de error, justifica por qué se ha producido e intenta solucionarlo:

- Creación de la base de datos de VENTAS y creación de las tablas correspondientes, haciendo cumplir las reglas de integridad.
- Inserción de los siguientes datos en las tablas:

#### CIUDAD

Cod Ciudad	Nom Ciudad
C1	Bilbao
C2	Vitoria
C3	San Sebastián

#### SUPERMERCADO

Cod Sup	Dirección	Tfno	Cod Ciudad
S1	Gran Vía 1	111111111	C1
S2	Alda. Recalde 7	222222222	C1
S3	Ondarreta 7	333333333	C2
S4	Dato 3	444444444	C3
S5	Avda. de Vitoria 8	555555555	C3

#### PRODUCTO

Cód Prod	Denominac	Precio
P1	Nocilla	3
P2	Tomate Orlando	1,5
P3	Cereales Nestlé	3,5
P4	Cerales Crunch	3
P5	Pan Bimbo	2
P6	Pan Panrico	1,9
P7	Bonito Zallo	2,75

#### VENTAS

Cód Prod	Cód Sup	Cantidad
P1	S1	10
P1	S2	15
P1	S3	8
P2	S1	20
P2	S2	25
P2	S3	15
P3	S1	25
P3	S2	18
P3	S3	20

- Aumentar las ventas del producto P1 en 5 unidades para el supermercado 1 realizando las operaciones necesarias.
- Eliminar el producto P3 de la base de datos

- e) Obtener un listado en el que aparezca el importe total de ventas por cada supermercado. El importe de cada venta se calcula multiplicando precio por cantidad.
- f) Obtener qué supermercado ha realizado un importe mayor de ventas durante el periodo visualizando el supermercado y el total de ventas.
- g) Obtener cuál ha sido el producto más vendido durante el periodo visualizando el código y nombre del producto, así como el número de unidades totales vendidas.
- h) Aumentar en un 5% el precio de todos los productos que hayan tenido alguna venta.
- i) Obtener un listado en el que aparezcan todos los productos y el importe total de ventas de cada uno, indicando código del producto, su denominación y el importe total.