

## 24303 BASES DE DADES

### TUTORIAL

### Primera sessió de pràctiques

Introducció a MySQL i al Llenguatge Estructurat de Consultes

**Objectius:** Els objectius d'aquesta pràctica són els d'oferir una introducció a:

- 1) MySQL en línia de comanda i entorn gràfic
- 2) Llenguatge SQL. Aprendre a escriure ordres senzilles per crear, editar i manipular bases de dades, taules i registres en aquestes taules.

En finalitzar la pràctica s'haurà de lliurar **individualment usant el link a l'aula global (metacurs de base de dades) els exercicis desenvolupats. Els exercicis a lliurar són E1, E2, E3, E4, E5, E6, i E7.** El material lliurat individualment per l'alumne s'utilitzarà com a nota de concepte.

**Metodologia:** L'enunciat de la pràctica cal anar llegint-lo en ordre, seguint pas a pas les instruccions que es van donant.

## 1. INTRODUCCIÓ

MySQL és un **Sistema Gestor de Bases de Dades** (SGBD) que ens permet: crear, mantenir i gestionar bases de dades (BBDD). Una BD relacional és una BD que organitza la informació en taules i relacions entre elles.

MySQL utilitza el llenguatge SQL (Structured Query Language) per a gestionar les BBDD. Es tracta d'un llenguatge estructurat declaratiu amb el que s'especifica "què" es vol fer amb la base de dades i no "com" es fa.

El llenguatge SQL proporciona funcionalitat per a:

- a- La definició de dades (**DDL** - Data Definition Language)
- b- La manipulació de dades (**DML** - Data Manipulation Language)
- c- La gestió de rols i permisos d'usuari de BD (**DCL** - Data Control Language).

La part de definició de dades (DDL) s'ocupa de la **creació dels objectes** de la BD (taules, quins atributs, tipus de dades, procediments, funcions, packages, índex, restriccions de dades, etc.).

La part de manipulació de dades (DML) s'ocupa de la **selecció, inserció, modificació i esborrat** d'informació de les taules.

Finalment, el sub-llenguatge DCL conté les sintaxis SQL relatives a la creació, edició i esborrat de

permisos d'usuari als objectes existents de la BD. Aquests permisos poden ser definits a diferents nivells: a nivell de taula, de columna, de procediment, etc i que permeten diferents privilegis: de lectura, d'escriptura, de creació, d'esborrat, d'edició...

L'SQL té una sintaxi estructurada de fàcil comprensió, per exemple:

```
SELECT surname
```

```
FROM person
```

```
WHERE name LIKE 'Joan%';
```

Aquesta sentència selecciona com a **font de dades** (FROM) la taula "person" de la BD i en concret **en filtra** (WHERE) només aquells registres (o *files*) els noms dels quals comencen per Joan. Finalment, d'entre totes les columnes de la taula *person*, **en mostrarà** (SELECT) només el cognom.

**\*Nota1:** A les pràctiques farem servir els noms de taules i tuples en anglès per convenció i per evitar caràcters especials tals com accents.

**\*Nota2 MOLT IMPORTANT:** Al llarg de l'enunciat de la pràctica es mostraran exemples de sintaxi SQL que incorporen comes simples '. Si intenteu copiar i enganxar aquestes sentències directament la BD del vostre ordinador, molt possiblement fallaran perquè el caràcter de coma simple del pdf o Word no és el mateix que el usa MySQL.

## E1: Què és SQL?

### 2. TREBALLANT AMB LA LÍNIA DE COMANDES MySQL

Es pot invocar el motor de MySQL des de la línia de comandes del sistema operatiu o mitjançant el "Command Line Client" des del menú de programes (en Windows).



Per a instal·lar MySQL en el vostre ordinador personal, versió línia de comandes, haureu de descarregar: MySQL Community Server. (Per a la instal·lació podreu trobar una guia a l'aula global)

Es comença una sessió amb MySQL entrant en el terminal:

```
shell> mysql -u root -p
```

-u especifica l'usuari, que és generalment "root". Es demana llavors la contrasenya (Password), que és generalment "root" també. El password de l'administrador *root* (DBA – Database Administrator) es pot haver canviat en el moment d'instal·lació del SGBD.

Després d'un missatge de benvinguda que comença amb "Welcome to the MySQL monitor", apareix el prompt `mysql>` que indica que el programa està llest per rebre i respondre a ordres SQL.

En la línia d'ordres es poden ingressar comandaments en una o diverses línies, en la majoria dels casos **la comanda s'acaba amb un punt i coma (";")**.

Algunes comandes de l'entorn MySQL:

Comanda	Explicació
<b>QUIT</b>	S'utilitza per abandonar l'entorn (en línia d'ordres).
<b>SHOW DATABASES;</b>	Mostra les BDs disponibles en el servidor.
<b>USE &lt;database&gt;;</b>	Especifica quina BD a usar a partir d'aleshores.
<b>SHOW TABLES;</b>	Mostra les taules de la BD en ús.
<b>DESCRIBE &lt;table&gt;;</b>	Mostra l'estructura de la taula <table> (camps, tipus de dades, etc.) de la BD seleccionada.
<b>SHOW CREATE TABLE &lt;table&gt;;</b>	Mostra la sintaxi SQL usada per a crear la taula en especificada.
<b>CREATE DATABASE &lt;database&gt;;</b>	Crea la base de dades amb el nom especificat <database>.

El subconjunt de llenguatge SQL anomenat DML conté les següents instruccions per a la consulta i manipulació de dades:

- a. **SELECT** → Consultar dades de la BD.
- b. **INSERT** → Inserir dades dins de les taules de la BD.
- c. **UPDATE** → Actualitzar dades de les taules.
- d. **DELETE** → Eliminar dades de les taules.

A aquest tipus d'accions se les anomena CRUD: Create-Read-Update-Delete. Només apliquen accions sobre les dades, però no modifiquen l'estructura dels objectes de la BD.

També disposa de les comandes que ofereix el subconjunt DDL:

- a. **CREATE** → Per a crear objectes nous dins de la BD.
- b. **DROP** → Per a eliminar objectes existents en la BD.
- c. **ALTER** → Per a modificar l'estructura o definició d'objectes existents en la BD.

Aquestes sintaxis es poden aplicar a diferents objectes de la BD: taules, columnes, índexs, bases de dades en sí mateixes, procediments, funcions, packages, events...

Per exemple: CREATE DATABASE; ALTER DATABASE; CREATE TABLE; ALTER TABLE; DROP TABLE; CREATE INDEX; DROP INDEX



### Majúscules o minúscules?

És bona pràctica, i molt recomanada, posar les paraules claus que pertanyen al llenguatge SQL com **SELECT** o **FROM** en **majúscula** (presumptament millora la llegibilitat), tot i que les sentències s'executarien igualment en minúscules.

Al respecte als noms de taules, columnes i altres dades, Windows no distingeix entre majúscules i minúscules mentre que Linux i MacOS sí que ho fan.

Per tant, és recomanable assumir que la distinció entre majúscules i minúscules sí que afecta en l'execució de les nostres sintaxis SQL.

Si en Unix-Like Systems ens referim a un nom de columna usant les majúscules i minúscules de manera errònia, el SGBD ens retornaria un error especificant que la taula no existeix en la BD seleccionada.

Per tant, es recomana escriure tant els noms de taules i columnes en minúscula sempre que sigui possible.

Per a determinar si les majúscules i minúscules seran rellevants en les dades emmagatzemades dins de les taules, usarem el COLLATION adequat en el moment de creació de la BD tal i com veurem més endavant.



### El punt i coma ;

Una comanda SQL consisteix en una sentència (statement) seguida d'un punt i coma. Per a alguns (pocs) ordres, com QUIT, el punt i coma és opcional. Si entrem per exemple la comanda següent sense punt i coma:

```
mysql> SHOW DATABASES
```

La resposta serà ->

que indica que s'espera una altra cosa. A l'entrar el punt i coma després del "->", s'executarà la comanda correctament.

Es recomana, posar sempre al final de la proposició (statement) punt i coma (";"), per tal d'evitar possibles confusions i/o errades.

**E2: Quines instruccions SQL hem d'usar per a crear una base de dades?**

## 2.1 Creant la primera base de dades

Crearem una base de dades per a modelar i emmagatzemar dades sobre una empresa. En la línia de comandes de mysql crearem la base de dades “company” entrant:

```
mysql> CREATE DATABASE company;
```

Podem comprovar que la nova base de dades ha esta creada:

```
mysql> SHOW DATABASES;
```

Haurem veure “company” com una de les bases de dades al servidor. Ara anem a començar a utilitzar aquesta BD, per això introduïrem la comanda:

```
mysql> USE company;
```

La resposta serà: Database changed

Ara la BD activa i sobre la qual treballarem serà “company”.

**CHARACTER SET i COLLATION**

Quan creem una base de dades, li podem especificar quin conjunt de caràcters i de quin idioma (humà) s'usarà per a guardar la informació dins de les taules. Això s'anomena el CHARACTER SET de la BD.

D'entre els diferents jocs de caràcters que ofereix el SGBD, disposem de: BINARY, ASCII, UTF-8, LATIN1, BIG5, GREEK, HEBREW, etc.

Això ens servirà per tal de que la nostra BD a crear accepti caràcters d'un determinat alfabet humà.

Podem consultar els jocs de caràcters disponibles dins del SGBD on estem connectats usant la següent instrucció: SHOW CHARACTER SET;

```
mysql> SHOW CHARACTER SET;
```

Charset	Description	Default collation	Maxlen
armscii8	ARMSSCII-8 Armenian	armscii8_general_ci	1
ascii	US ASCII	ascii_general_ci	1
big5	Big5 Traditional Chinese	big5_chinese_ci	2
binary	Binary pseudo charset	binary	1
cp1250	Windows Central European	cp1250_general_ci	1
cp1251	Windows Cyrillic	cp1251_general_ci	1
cp1256	Windows Arabic	cp1256_general_ci	1
cp1257	Windows Baltic	cp1257_general_ci	1
cp850	DOS West European	cp850_general_ci	1
cp852	DOS Central European	cp852_general_ci	1
cp866	DOS Russian	cp866_general_ci	1
cp932	SJIS for Windows Japanese	cp932_japanese_ci	2
dec8	DEC West European	dec8_swedish_ci	1
eucjpms	UJIS for Windows Japanese	eucjpms_japanese_ci	3
euckr	EUC-KR Korean	euckr_korean_ci	2
gb18030	China National Standard GB18030	gb18030_chinese_ci	4
gb2312	GB2312 Simplified Chinese	gb2312_chinese_ci	2
gbk	GBK Simplified Chinese	gbk_chinese_ci	2
geostd8	GEOSTD8 Georgian	geostd8_general_ci	1
greek	ISO 8859-7 Greek	greek_general_ci	1

Afegidament, podem aplicar el modificador de compaginació sobre el joc de caràcters prèviament escollit. La compaginació d'un joc de caràcters determina quina versió del joc de caràcters es vol usar i si les majúscules i minúscules són rellevants en el moment d'emmagatzemar les dades dins de les taules. Això ens servirà si per exemple volem que la BD contempli els noms "Anna" i "anna" com el mateix nom o no. En general, els més usats són: CI (Case Insensitive) i CS (Case Sensitive).

Per a visualitzar les compaginacions disponibles dins del SGBD podem usar:  
SHOW COLLATION;

```
mysql> show collation like 'latin1%';
```

Collation	Charset	Id	Default	Compiled	Sortlen	Pad_attribute
latin1_bin	latin1	47		Yes	1	PAD SPACE
latin1_danish_ci	latin1	15		Yes	1	PAD SPACE
latin1_general_ci	latin1	48		Yes	1	PAD SPACE
latin1_general_cs	latin1	49		Yes	1	PAD SPACE
latin1_german1_ci	latin1	5		Yes	1	PAD SPACE
latin1_german2_ci	latin1	31		Yes	2	PAD SPACE
latin1_spanish_ci	latin1	94		Yes	1	PAD SPACE
latin1_swedish_ci	latin1	8	Yes	Yes	1	PAD SPACE

8 rows in set (0.04 sec)

Un exemple de creació de BD amb la combinació adequada de CHARACTER SET i COLLATION per a una BD amb dades en català/castellà, seria:

```
CREATE DATABASE IF NOT EXISTS company DEFAULT CHARACTER SET  
'latin1' DEFAULT COLLATE 'latin1_general_cs';
```

Això ens permetrà crear la BD *company* amb un joc de caràcters vàlid pel català i castellà i distingirà entre majúscules i minúscules. Per tant, haurem de ser rigorosos en el moment d'introducció de dades.

A més a més, si en el servidor ja existia una BD amb el nom *company*, la instrucció no produirà cap efecte però tampoc cap error gràcies a l'afegit de: IF NOT EXISTS.

## 2.2 Creant una taula

Volem representar dades sobre els departaments, els projectes i els empleats d'una empresa. Per això crearem les tres taules 'department', 'project' i 'employee' a la BD 'company'. Per a crear la taula 'department' introduïrem la següent comanda en MySQL:

```
CREATE TABLE department (
  id integer(4),
  name varchar2(18),
  floor integer(2),
  building varchar2(20),
  city varchar2(15),
  constraint pk_dept primary key (id)
);
```

Aquesta comanda crea una taula anomenada 'department' com taula de la BD activa. S'especifiquen els noms dels camps i els seus tipus de dades.

En aquest cas són 5 camps, els tipus utilitzats són **integer** (per a dominis sencers com l'identificador departament i planta) i **varchar2** (per alfanumèrics com a nom del departament, nom de l'edifici i ciutat).

També s'especifiquen les mides. La comanda també fa referència a **una restricció** (anomenada pk\_dept) sobre **la clau primària** de la taula, en aquest cas diu que id és una clau primària.

Vegem que la taula ha estat creada utilitzant la comanda:

```
SHOW TABLES;
```

S'hauria veure la taula 'departament' com una de les taules de la base de dades 'company'.

Vegem l'estructura de la taula creada utilitzant la comanda:

```
DESCRIBE departament;
```

Això ens mostrarà els camps i els seus tipus, les restriccions (clau), valors per defecte, etc.

També en podem observar la sintaxi SQL de creació de la taula:

```
SHOW CREATE TABLE departament;
```



### Canviar el nom d'una columna

Si s'ha entrat el nom d'un camp de forma incorrecta, per exemple "foor" en comptes de "floor", es pot rectificar així:

```
ALTER TABLE department CHANGE foor floor integer(2);
```

És a dir, es demana alterar la taula 'department', canviant el nom del camp 'foor' per 'floor' i també n'haurem d'especificar el seu tipus de dades el qual podrem aprofitar per intentar canviar-lo si ho volem, si no, el tornarem a escriure igual. És possible que el tipus de dades no es pugui canviar si la taula ja conté informació.



### Convencions de noms

Els noms de les taules estan en singular, tot i que emmagatzemen un o més registres (files), per exemple 'employee'.

Els noms compostos es recomana separar-los o bé seguint ***CamelCase*** o bé ***snake\_case***, per exemple, si haguéssim d'anomenar un camp que representi un codi d'identificació podríem fer-ho:

- CamelCase: IdentificationCode ó identificationCode
- snake\_case: identification\_code



**E3: Quina és la comanda SQL per a crear una taula?**

**E4: Com es pot saber quines són totes les taules de la base de dades activa?**

2.3 Inserint un registre en la base de dades

Per inserir un registre a la taula 'department', entrar la comanda següent:

```
INSERT INTO department (id, name, floor, building, city)
```

```
VALUES (1, 'COMPTABILITAT', 2, 'La Castellana', 'MADRID');
```

S'utilitza per a inserir un registre amb camps determinats en la taula 'departaments'.

**Aconseguir el número de registres d'una taula**

Entrada la comanda següent per aconseguir el nombre de registres a la taula 'departament':

```
SELECT count(*) FROM department;
```

**E5: Quina és la comanda de SQL per a inserir dades en una taula?****2.4 Creant més taules i inserint més registres**

S'ha de completar la BD 'company' amb la creació de les taules 'project' i 'employee', i la inserció de registres en les tres taules.

**Nota:** El fitxer "empresa.sql" a l'Aula Global conté ordres per a això, incloent les ordres de creació de la taula 'department' de l'apartat 2.2 i del registre que inserim en l'apartat 2.3.

Ingresseu un a un (o en bloc) les **ordres SQL** a l'indicador per executar-(donant al <ENTER> després de cada comanda). Es pot fer "copiar i enganxar" des del fitxer al **prompt de SQL**. Tingueu cura de no ingressar CAP comandament que s'hagi ingressat en els apartats anteriors.

**E6: Inseriu dades a la taula employee, a partir del contingut del fitxer empleats.csv**

Al fitxer empleats.csv teniu les dades de 10 empleats de l'empresa, on s'informa del seu codi, nom, sou, codi del departament al que pertanyen i el codi del projecte on treballa.

A partir d'aquesta informació, creeu les instruccions d'inserció de dades a la taula employees.

Ho podeu fer escrivint les instruccions de insert manualment per a cada fila, generant les instruccions de INSERT automàticament usant fórmules de l'excel o important les dades massivament usant l'script de mysqlimport.



**Script mysqlimport (Nota: En Unix-Like Systems pot no funcionar d'igual manera degut a restriccions en els permisos de fitxers)**

Aquest script es troba dins de la carpeta d'instal·lació del SGBD MySQL i ens permet carregar dades massivament des d'un fitxer pla de dades (.csv, .txt).

Per a què funcioni, el format del fitxer haurà de ser l'adequat i contenir informació per a totes les columnes de la taula de destí.

Passos a seguir:

1. Copiar l'arxiu de dades que volem inserir a la carpeta de datafiles del SGBD. Per a trobar la carpeta dins del nostre disc dur, podem usar: **SELECT @@datadir;**
2. Editar l'arxiu de manera que es digui igual que la taula de destí on voleu carregar les dades. Usant un editor de text avançat (com per exemple notepad++), elimineu la capçalera de dades i separeu els valors de les columnes per un tabulador sense usar el punt i coma. És possible que també hagueu de canviar la codificació del fitxer i el format de salt de línia a Unix (LF).
3. Obrir un terminal de sistema nou i ubicar-nos a la ruta d'instal·lació del SGBD, En Windows, ho trobarem a una ruta similar a: C:\Program Files\MySQL\MySQL Server 8.0\bin
4. Executar la comanda: **mysqlimport --user root nom\_bd nom\_document**

## 2.5 Executant algunes comandes

Proveu les següents comandes sobre la base de dades acabada de crear. Entendre i explicar què fa cada un d'ells. Assegureu-vos que la base de dades està activa abans d'executar qualsevol comanda SELECT utilitzant la següent instrucció:

**USE company;**

Comandes:

1. `SELECT * FROM employee;`
2. `SELECT * FROM department;`
3. `SELECT * FROM project;`
4. `SELECT name FROM employee;`
5. `SELECT id, name FROM employee;`
6. `SELECT name, id FROM employee;`
7. `SELECT name as NOMBRE FROM employee;`
8. `SELECT name as 'Employee Name' FROM employee;`
9. `SELECT name FROM employee WHERE name LIKE 'Josep';`
10. `SELECT id, name FROM employee WHERE name LIKE 'Josep';`
11. `SELECT name FROM department;`
12. `SELECT name FROM department WHERE city LIKE 'Barcelona';`
13. `SELECT name, budget FROM project;`
14. `SELECT name, budget FROM project WHERE budget < 100000;`

Sortir del MySQL Command line amb la comanda:

**EXIT;**

o la comanda

**QUIT;**

La informació creada (base de dades, relacions, tuples, etc.) quedaran protegida en disc. Per verificar-les pot ingressar novament al MySQL per línia d'ordres, seleccionar la base de dades *company* i executar algunes ordres SELECT com abans.

### 3. L'ENTORN GRÀFIC



Per a instal·lar l'IDE Workbench en el vostre ordinador personal, haureu de descarregar: MySQL Workbench client. (Per a la instal·lació podreu trobar una guia a l'aula global).

Es pot iniciar el MySQL Workbench des del menú d'inici.

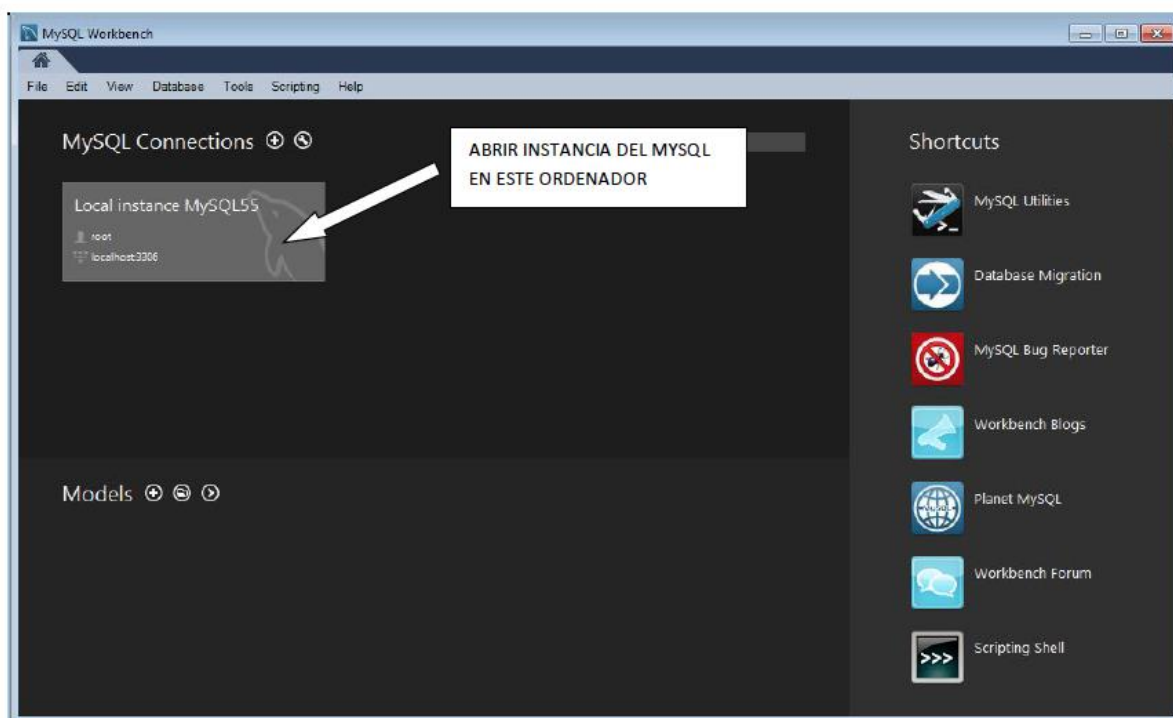
En Windows:

- Start -> All Programs -> MySQL -> MySQL Workbench

En Linux:

- Aplicacions -> Programació -> MySQL

En la Figura 1 es mostra la pantalla d'inici del MySQL Workbench.



#### 3.1 Accedir a la base de dades

En ingressar a una instància local de MySQL (usant la contrasenya donada pel vostre docent de pràctiques) es podrà veure una pantalla com la de la Figura 2.

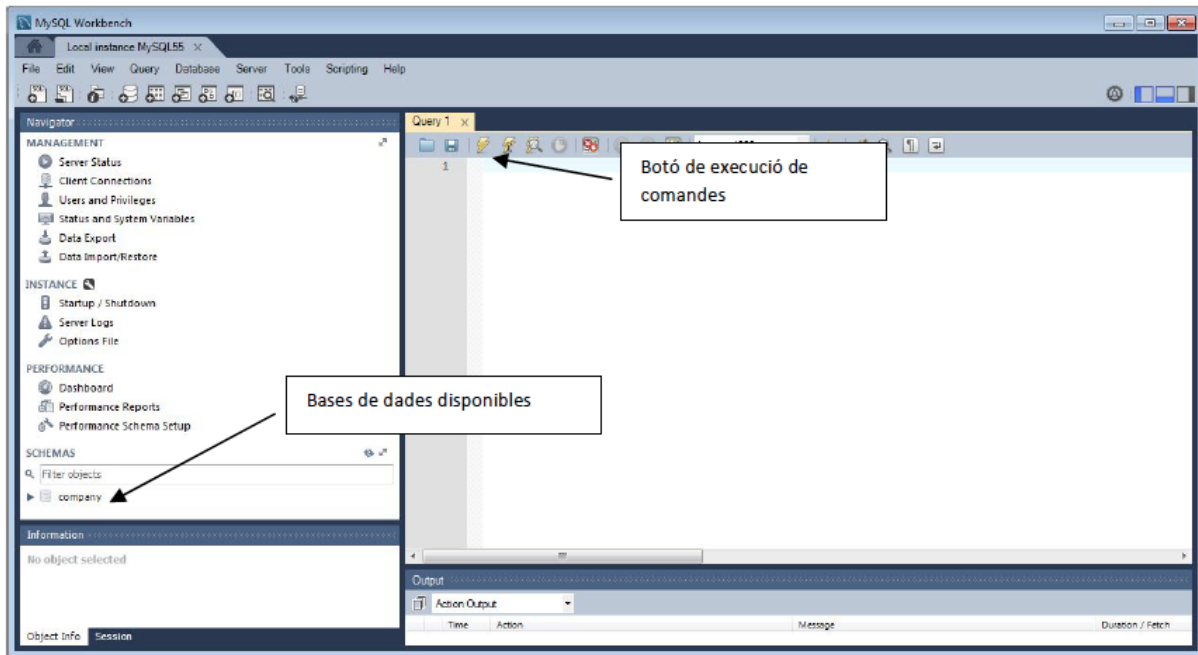
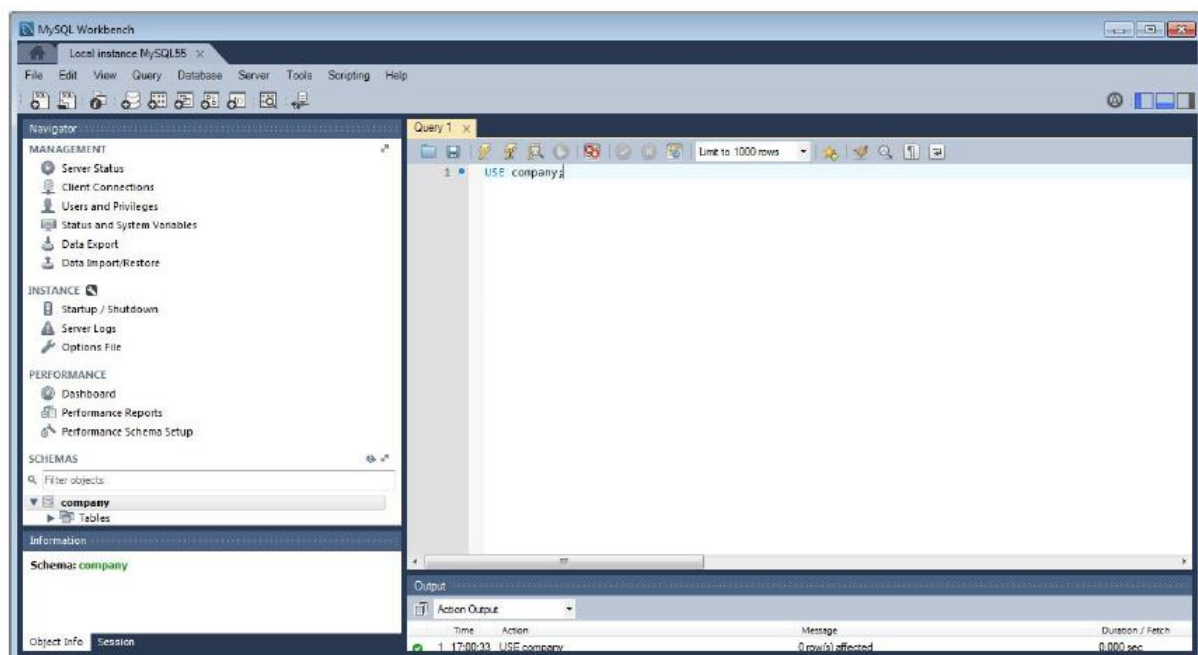
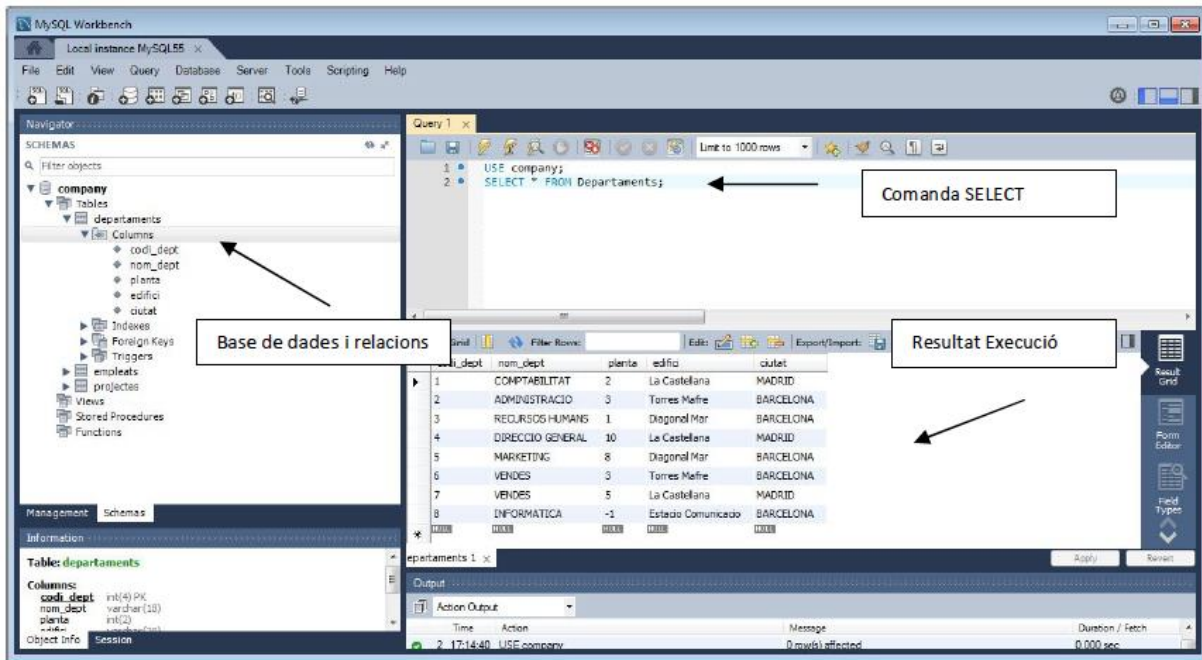


Figura 2 - Instància Local MySQL

A l'entrar a la instància local de MySQL s'haurà veure entre altres coses la base de dades *company*. Seguiu les instruccions que us passi el professor de pràctiques per treballar amb aquesta base de dades.





Practicar en la interfície gràfica totes les ordres de l'apartat anterior.

Un cop hagueu provat totes les ordres elimineu la base de dades usant la comanda:

**DROP SCHEMA company;**

Ara useu l'opció "Open SQL Script" per obrir l'script "empresa.sql". Es podran veure tots els comandaments a la pestanya de l'SQL Script. Executeu totes les comandes per crear i poblar novament la base de dades "companya".

Després d'executar les ordres proveu algun dels comandaments SELECT per comprovar que la base de dades conté dades.

### 3.2 Inserir un nou registre mitjançant l'editor SQL

En la Figura 5 es mostra com inserir una tupla usant l'editor. La tupla a inserir conté els valors

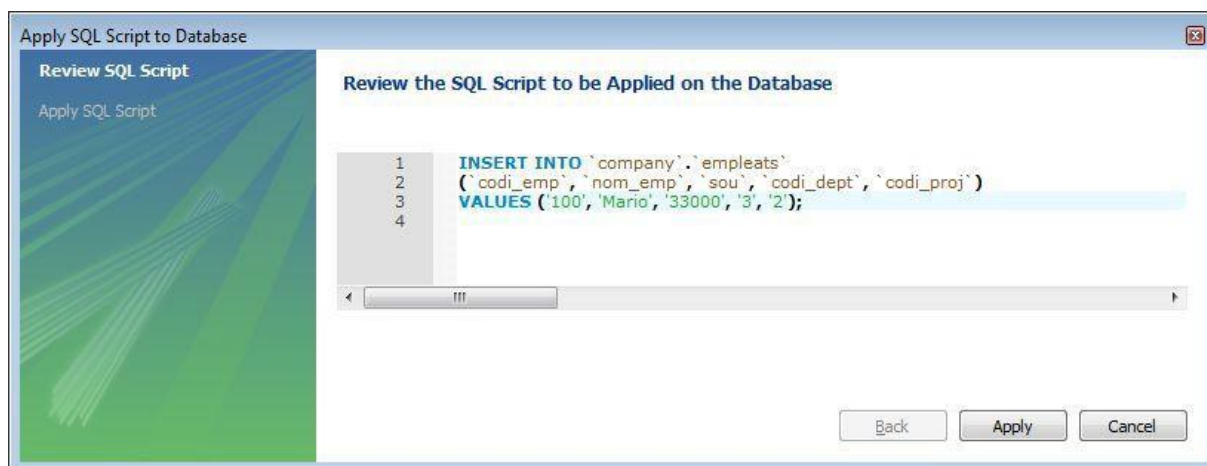


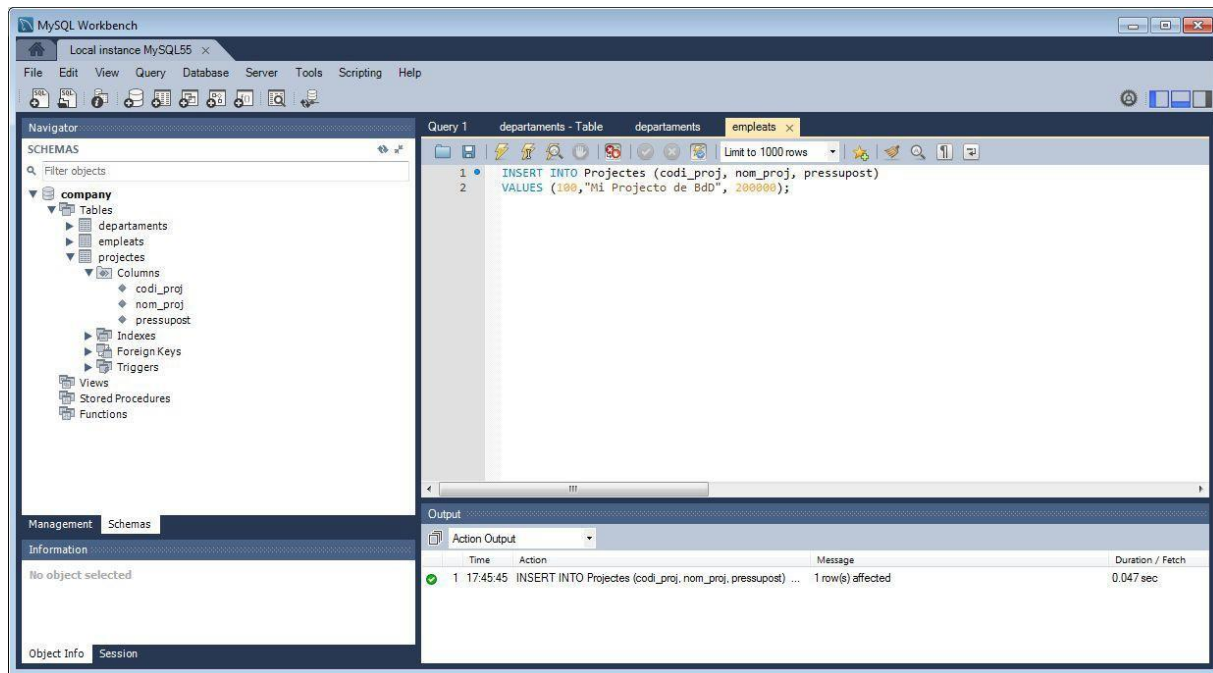
Figura 5 - Comanda per inserir una tupla

```
id = 100,  
name = "Mario",  
department_id = 3 i  
project_id = 2
```

Inseriu aquesta tupla a la taula d'empleats i faci un "commit" usant botó "apply". Noteu que MySQL genera una ordre d'inserció de tupla corresponent amb la informació ingressada a la interfície gràfica, això pot resultar útil en alguns casos per generar instruccions d'inserció de dades.

Executar una comanda SELECT per verificar que la tupla està efectivament en la taula d'empleats.





Ara inseriu "manualment" un nou registre / tupla a la taula 'projectes' codificant una instrucció INSERT.

```
INSERT INTO project (id, name, budget) VALUES (100, "El meu Projecte de BdD", 200000);
```

Amb l'indicador en la comanda, triar al menú "Query" la comanda "Execute Current Statement" (o Ctrl + Enter).

Llistar les tuples de la taula de Projectes i per tal de verificar que hi ha un nou registre pel projecte número 100.

### 3.3 Consultes a la BD mitjançant l'editor SQL

Escriure i executar les ordres següents amb l'editor SQL, observant els resultats de l'execució. Traduir les ordres al català / català. Per exemple:

```
SELECT * FROM department;
```

Es tradueix a "Selecciona tots els registres de la taula 'departament'". Retornarà una taula amb 8 registres que són tots els registres que hi ha a la taula 'departament'.

Observar que la taula que resulta d'aquesta comanda només es pot navegar, no es pot editar.

A l'apartat 4, es resumeixen les paraules claus de les ordres SQL.

#### 4.1

```
SELECT name FROM employee;  
SELECT name, budget FROM project;
```

Què s'observa?

- Descriu els resultats obtinguts.
- Quines diferències trobeu entre les consultes del punt 4.1 i el punt 4.2?
- Per a què creieu que pot ser útil l'esquema de consulta del punt 4.2?

#### 4.2

```
SELECT name, salary FROM employee ORDER BY name;  
SELECT name, salary FROM employee ORDER BY salary asc;  
SELECT name, salary FROM employee ORDER BY name, salary;
```

Què s'observa?

- Descriu els resultats obtinguts.
- Quines diferències trobeu entre aquestes tres consultes?

#### 4.3

```
SELECT name FROM employee WHERE salary = 18000;  
SELECT name FROM employee WHERE salary > 24000;
```

Què s'observa?

- Descriu els resultats obtinguts.
- Per a què creieu que pot ser útil aquest esquema de consulta?

#### 4.4

```
SELECT a.name, b.name FROM project AS a, employee AS b  
WHERE b.project_id = a.id;
```

Què s'observa?

- Descriu els resultats obtinguts.

**4.5**

Preneu la consulta del punt 4.4 i modifiqueu-la perquè:

- a. Imprimiu els resultats ordenats pel nom del projecte.
- b. En lloc de mostrar els resultats relacionats amb el nom del projecte, mostreu els resultats relacionats amb el nom del departament al qual pertany l'empleat.

**4.6**

Penseu una pregunta complexa que es pugui fer a la BD y traduïu-la a SQL.

Executeu-la. Fa el que es pretenia?

**E7: Entregar totes les respostes 4.1 a 4.6**

## 4. 4. Resum comandes SQL

<b>Comandes DDL</b> (per a la manipulació de taules i objectes de BD)	
<b>CREATE</b>	Utilitzat per crear noves taules, camps, índexs i altres objectes de BD
<b>DROP</b>	Empleat per eliminar taules, índexs i altres objectes de BD
<b>ALTER</b>	Utilitzat per modificar les taules i columnes agregant camps o canviant la definició dels mateixos.

<b>Comandes DML</b> (per a la manipulació de dades)	
<b>SELECT</b>	Utilitzat per consultar i mostrar per pantalla registres de la base de dades que satisfacin un criteri determinat.
<b>INSERT</b>	Utilitzat per carregar lots de dades a la base de dades en una única operació.
<b>UPDATE</b>	Utilitzat per modificar els valors dels camps i registres especificats.
<b>DELETE</b>	Utilitzat per eliminar registres (files) d'una taula d'una base de dades.

Les clàusules, per la seva banda, són condicions de modificació utilitzades per definir les dades que es vol seleccionar o manipular:

<b>Clàusules</b>	
<b>FROM</b>	Utilitzada per especificar les taules que seran la font d'informació de les nostres consultes.
<b>WHERE</b>	Utilitzada per especificar les condicions que han de complir els registres que es van a seleccionar. Per a filtrar les dades resultants a mostrar.
<b>GROUP BY</b>	Utilitzada per a agrupar els registres seleccionats en grups específics en funció dels valors repetits
<b>ORDER BY</b>	Utilitzada per ordenar els registres seleccionats i mostrar-los d'acord amb un ordre específic