

Practical 01- Part 02

Terminology of MySQL

☐ Database

- Is a collection of tables where related data is stored in a systematic format.

☐ Table

- Is a matrix of rows and columns that hold data.

☐ Column

- Columns are stack of cells, defined to hold a specific type of data, such as dates, numeric, or textual data.

☐ Row

- Data is stored horizontally as records in rows/tuples.
- A row is composed of fields and contains all the data about one particular person, company, or item in a database.

Terminology

☐ Primarykey

- A primary key is a field in a table which uniquely identifies each row/record in a database table.

- A primary key column cannot have NULL values.

- A table can have only one primary key.

☐ Foreign Key

- A foreign key is a column (or columns) that refer a column (most often the primary key) of another table.

- The purpose of the foreign key is to ensure referential integrity of the data.

☐ Composite key

- Composite key, or compound primary key, refers to cases where more than one column is used to specify the primary key of a table.

☐ Candidate Key

- A Candidate Key can be any column or a combination of columns that can qualify as unique key in database.

- There can be multiple Candidate Keys in one table.

- Each Candidate Key can qualify as Primary Key.

☐ Index

- Indexes are special lookup tables that the database search engine can use to speed up data retrieval.

- It is a pointer to data in a table.

- An index in a database is very similar to an index in the back of a book.

☐ Redundancy

- The appearance of the same data factor in more than one field or table of data.

Install MySQL on Ubuntu 16.04.2

- ☐ We can install MySQL by using the apt package manager.

ICT1222

Database Management Systems Practicum

☐ First make sure your packages lists are up to date.

☐ Open the terminal and run this apt command.

sudo apt-get update

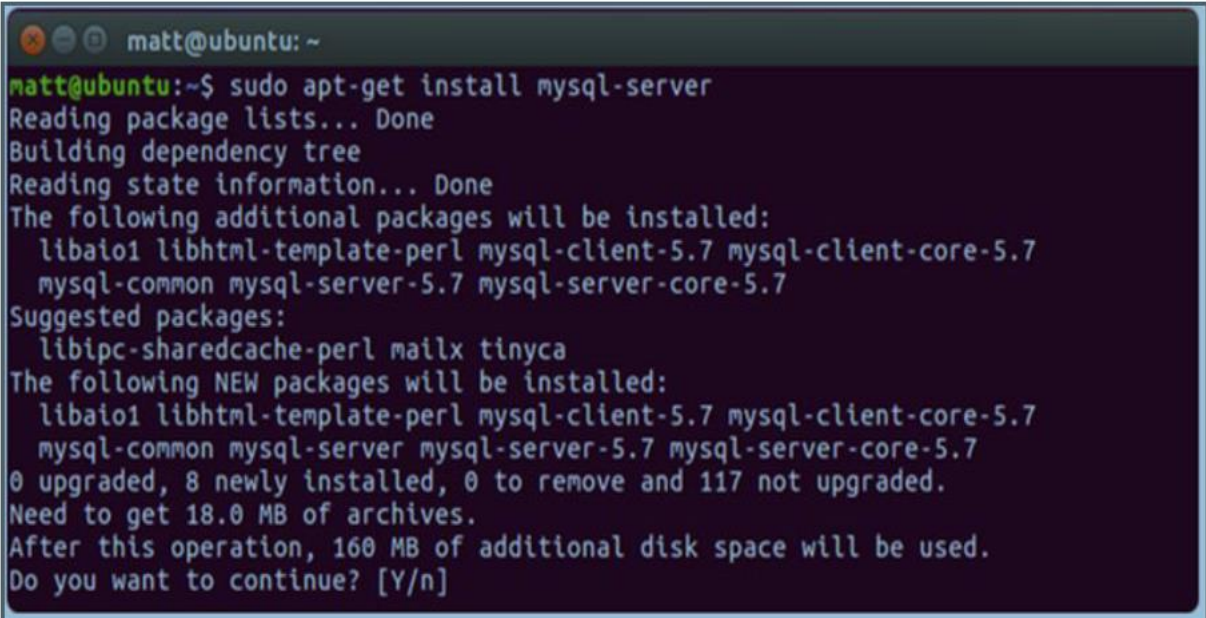
Install MySQL on Ubuntu 16.04.2

- ☐ We need to install the mysql-server package,
 - which downloads the required files,
 - configures the initial database set up
 - and handles running MySQL as a system service.
- ☐ Run this apt command to get the process started.

sudo apt-get install mysql-server

Install MySQL on Ubuntu 16.04.2

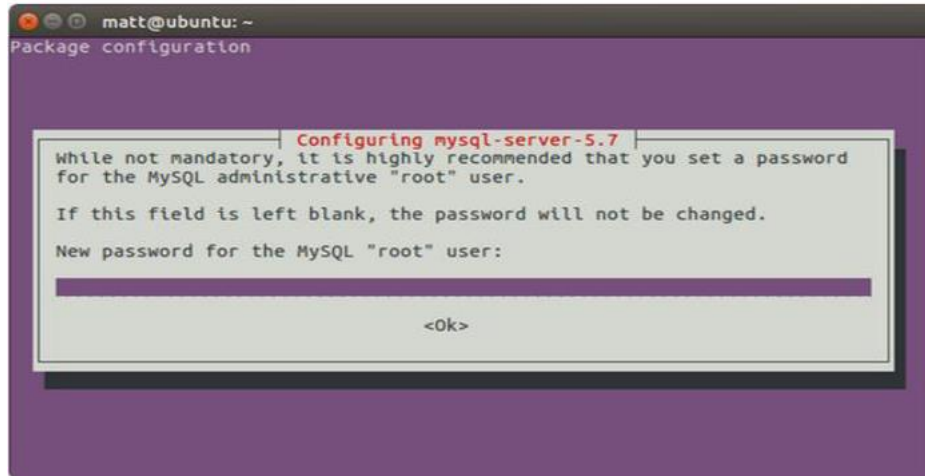
- ☐ Enter 'y' when prompted with whether or not you want to install the new package.



```
matt@ubuntu: ~  
matt@ubuntu:~$ sudo apt-get install mysql-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libaio1 libhtml-template-perl mysql-client-5.7 mysql-client-core-5.7  
  mysql-common mysql-server-5.7 mysql-server-core-5.7  
Suggested packages:  
  libipc-sharedcache-perl mailx tinycd  
The following NEW packages will be installed:  
  libaio1 libhtml-template-perl mysql-client-5.7 mysql-client-core-5.7  
  mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7  
0 upgraded, 8 newly installed, 0 to remove and 117 not upgraded.  
Need to get 18.0 MB of archives.  
After this operation, 160 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

☐ An administrative screen asking for a new root password will appear several times in the middle of the package installation process.

☐ Enter your chosen new password and the installation will continue



In a moment the installation will finish and you'll be back at the command prompt.

```
Setting up mysql-client-core-5.7 (5.7.12-0ubuntu1) ...
Setting up mysql-client-5.7 (5.7.12-0ubuntu1) ...
Setting up mysql-server-core-5.7 (5.7.12-0ubuntu1) ...
Setting up mysql-server-5.7 (5.7.12-0ubuntu1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my
.cnf) in auto mode
Setting up libhtml-template-perl (2.95-2) ...
Setting up mysql-server (5.7.12-0ubuntu1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (229-4ubuntu4) ...
matt@ubuntu:~$
```

SQL Data Types

- ☐ MySQL uses many different data types broken into three categories.
- Numeric
 - ☐ Uses all the standard ANSI SQL numeric data types
- Date and Time
- String Types.

Numeric Data Types

- ☐ INT
 - A normal-sized integer that can be signed or unsigned.
 - If signed, the allowable range is from -2147483648 to 2147483647.
 - If unsigned, the allowable range is from 0 to 4294967295.
 - You can specify a width of up to 11 digits.
- ☐ TINYINT
 - A very small integer that can be signed or unsigned.
 - If signed, the allowable range is from -128 to 127.
 - If unsigned, the allowable range is from 0 to 255.

ICT1222

Database Management Systems Practicum

- You can specify a width of up to 4 digits.

□ SMALLINT

- A small integer that can be signed or unsigned.
- If signed, the allowable range is from -32768 to 32767.
- If unsigned, the allowable range is from 0 to 65535.
- You can specify a width of up to 5 digits.

□ MEDIUMINT

- A medium-sized integer that can be signed or unsigned.
- If signed, the allowable range is from -8388608 to 8388607.
- If unsigned, the allowable range is from 0 to 16777215.
- You can specify a width of up to 9 digits.

□ BIGINT

- A large integer that can be signed or unsigned.
- If signed, the allowable range is from -9223372036854775808 to 9223372036854775807.
- If unsigned, the allowable range is from 0 to 18446744073709551615.
- You can specify a width of up to 20 digits.

□ FLOAT(M,D)

- A floating-point number that cannot be unsigned.
- You can define the display length (M) and the number of decimals (D).
- This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals).
- Decimal precision can go to 24 places for a FLOAT.

□ DOUBLE(M,D)

- A double precision floating-point number that cannot be unsigned.
- You can define the display length (M) and the number of decimals (D).
- This is not required and will default to 16,4, where 4 is the number of decimals.
- Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.

□ DECIMAL(M,D)

- An unpacked floating-point number that cannot be unsigned. In the unpacked decimals, each decimal corresponds to one byte.
- Defining the display length (M) and the number of decimals (D) is required.
- NUMERIC is a synonym for DECIMAL.

Date and Time Types

□ DATE

- A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31.
- For example, December 30th, 1973 would be stored as 1973-12-30.

ICT1222

Database Management Systems Practicum

□ DATETIME

- A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59.
- For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.

□ TIMESTAMP

- A timestamp between midnight, January 1st, 1970 and sometime in 2037.
- This looks like the previous DATETIME format, only without the hyphens between numbers
- 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 (YYYYMMDDHHMMSS).

□ TIME

- Stores the time in a HH:MM:SS format.

□ YEAR(M)

- Stores a year in a 2-digit or a 4-digit format.
- If the length is specified as 2 (for example YEAR(2)), YEAR can be between 1970 to 2069 (70 to 69).
- If the length is specified as 4, then YEAR can be 1901 to 2155. The default length is 4.

String Types

□ CHAR(M)

- A fixed-length string between 1 and 255 characters in length (for example CHAR(5)),
- Right-padded with spaces to the specified length when stored.
- Defining a length is not required, but the default is 1.

□ VARCHAR(M)

- A variable-length string between 1 and 255 characters in length.
- For example, VARCHAR(25).
- You must define a length when creating a VARCHAR field.

□ BLOB or TEXT

- A field with a maximum length of 65535 characters.
- BLOBs are "Binary Large Objects" and are used to store large amounts of binary data,
- such as images or other types of files.
- Fields defined as TEXT also hold large amounts of data.
- The difference between the two is that the sorts and comparisons on the stored data are **case sensitive** on BLOBs and are **not case sensitive** in TEXT fields.
- You do not specify a length with BLOB or TEXT.

□ TINYBLOB or TINYTEXT

- A BLOB or TEXT column with a maximum length of 255 characters.

ICT1222

Database Management Systems Practicum

- You do not specify a length with TINYBLOB or TINYTEXT.
- MEDIUMBLOB or MEDIUMTEXT
- A BLOB or TEXT column with a maximum length of 16777215 characters.
- You do not specify a length with MEDIUMBLOB or MEDIUMTEXT.

- LONGBLOB or LONGTEXT
- A BLOB or TEXT column with a maximum length of 4294967295 characters.
- You do not specify a length with LONGBLOB or LONGTEXT.

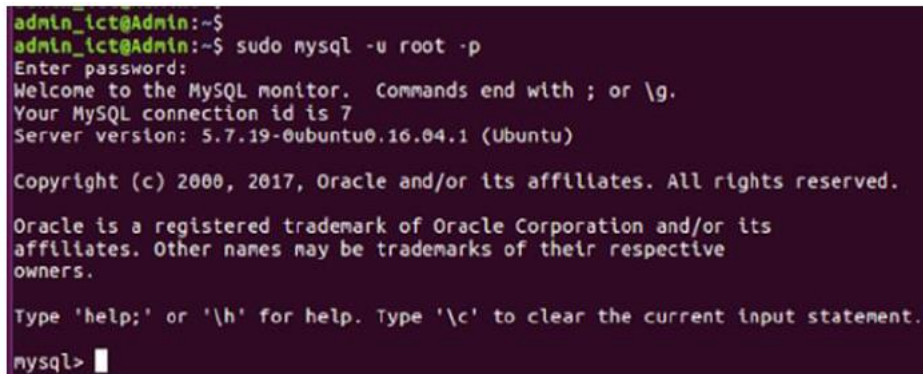
- ENUM
- An enumeration, which is a fancy term for list.
- When defining an ENUM, you are creating a list of items from which the value must be selected (or it can be NULL).
- For example, if you wanted your field to contain "A" or "B" or "C", you would define your ENUM as ENUM ('A', 'B', 'C') and only those values (or NULL) could ever populate that field.

Hands-On session

Connect to MySQL

- Connect to the MySQL instance with the MySQL command line client. Enter the password when asked.

sudo mysql -u root -p

A terminal window with a dark background and light green text. The user is at the prompt 'admin_lct@Admin:~\$'. They enter 'sudo mysql -u root -p'. The prompt changes to 'Enter password:'. After an invisible password entry, the prompt changes to 'mysql>'. The terminal displays the MySQL welcome message: 'Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 7. Server version: 5.7.19-0ubuntu0.16.04.1 (Ubuntu)'. It also shows the copyright notice for Oracle and its affiliates, and a prompt for help: 'Type \'help;\' or \'\\h\' for help. Type \'\\c\' to clear the current input statement.'

Check Version

- This will return the versions of MySQL and Ubuntu on your machine.
- SELECT VERSION();**


```
mysql> SELECT VERSION();
+-----+
| VERSION() |
+-----+
| 5.7.19-0ubuntu0.16.04.1 |
+-----+
1 row in set (0.00 sec)
```

List Databases

◦ This will show available databases with the SHOW DATABASES command

◦

SHOW DATABASES;

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

Create a Database

◦ You can create your own new database using following command.

◦ Example: **CREATE DATABASE company;**

CREATE DATABASE db_name;

Select a Database

◦ You can connect to the newly created database with the USE command. **USE db_name;**

◦ Example: **USE company;**

Create New Tables

◦ You can create new tables with the CREATE TABLE command.

CREATE TABLE table_name

(

Column-definitions, primaryKey-definition

);

Example:

CREATE TABLE employee

(

emp_no INT AUTO_INCREMENT,

name VARCHAR(20) NOT NULL,

gender CHAR(1),

dob DATE,

```
salary FLOAT(7,2),  
PRIMARY KEY(emp_no)  
);
```

View structure of a table

- You can use the following command to view the structure of a table.
- This will describe the table columns, data types and other specifications.
- Example: **DESC employee;**
DESC table_name;

List Tables

- View the available tables within the database using following command.
SHOW TABLES;

View Data in a table

- Use following commands to view all the data in a table or view data column wise.
SELECT * FROM table_name;
SELECT column1, column2
FROM table_name;

Insert values into tables

- Use following commands to insert values into tables.
INSERT INTO table_name
(field1, field2, ...)
VALUES
(value1, value2, ...) ;

Insert values into tables

- Example 01:
INSERT INTO employee
(emp_no, name, gender, dob, salary) VALUES
(1, 'Silva', 'M', '1980-03-25', 55000.00);
- Example 02:

```
INSERT INTO employee  
VALUES  
(2, 'Zoysa', 'F', '1983-06-07', 48000.00);
```

Insert values into tables

- Example 03:
INSERT INTO employee
(name, gender, dob, salary)
VALUES
('Fonseka','M','1979-04-19',64000.00);

□ Example 04:

```
INSERT INTO employee
(name, gender, dob, salary)
VALUES
('Fernando', 'F', '1975-05-28', 50000.00) ,
('Perera', 'F', '1985-11-14', 72000.00) ,
('Liyanage', 'M', '1989-04-10', 80000.00);
```

Delete table data

- Using this command you can delete only the data of a table.
- Example: **DELETE FROM employee;**
DELETE FROM table_name;

Remove table

- This command will remove the table completely.
- Example: **DROP TABLE employee;**
DROP TABLE table_name;

Remove Database

- In MySQL, the phrase most often used to delete objects is Drop. You would delete a MySQL database with this command.
- Example: **DROP DATABASE company;**
DROP DATABASE db_name;

Exit MySQL

- You can exit from MySQL server using one of following commands.
QUIT;
EXIT;