National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

Laboratory Manual 03:

DDL & DML Operations

CS-220: Database Systems

Fall 2017

Class: BS(CS)-6B

Dr. Amanullah Yasin

Mr. Ahmed Shahzaib Abid Bhatti

Date:29-Sep-17

Timings:1415-1700

Introduction

- SQL DDL (Data Definition Language) commands are used to create and modify the databases. Data Manipulation Language (DML) commands are used to query the databases.

Objectives

After performing this lab students should be able to:

1. Create tables in SQL using DDL commands.
2. Perform DML operations on created tables.

Tools/Software Requirement

- MySQL Community Server 5.6
- MySQL Workbench 6.1

Description

CREATE TABLE Syntax

```
CREATE TABLE [IF NOT EXISTS] tbl_name
(create_definition,...)

create_definition:
col_name column_definition
| [CONSTRAINT [symbol]] PRIMARY KEY [index_type]
(index_col_name,...)
[index_option] ...
| {INDEX|KEY} [index_name] [index_type] (index_col_name,...)
[index_option] ...
| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
[index_name] [index_type] (index_col_name,...)
[index_option] ...
| [CONSTRAINT [symbol]] FOREIGN KEY
[index_name] (index_col_name,...) reference_definition
| CHECK (expr)
```

```
column_definition:
data_type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
[COMMENT 'string']
[COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
[STORAGE {DISK|MEMORY|DEFAULT}]
[reference_definition]

data_type:
INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE
| TIME[(fsp)]
| TIMESTAMP[(fsp)]
| CHAR[(length)] [BINARY]
| VARCHAR(length) [BINARY]
| BINARY[(length)]
| VARBINARY(length)
| BLOB
| TEXT [BINARY]


reference_definition:
REFERENCES tbl_name (index_col_name,...)
[ON DELETE reference_option]
[ON UPDATE reference_option]

reference_option:
RESTRICT | CASCADE | SET NULL | NO ACTION
```

Schema Evaluation: ALTER TABLE Syntax

```
ALTER  TABLE tbl_name
[alter_specification [, alter_specification] ...]
[partition_options]
alter_specification:
table_options
| ADD [COLUMN] col_name column_definition
[FIRST | AFTER col_name ]
| ADD [COLUMN] (col_name column_definition,...)
```

```
| ADD {INDEX|KEY} [index_name]
[index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY
[index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
UNIQUE [INDEX|KEY] [index_name]
[index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]]
FOREIGN KEY [index_name] (index_col_name,...)
reference_definition
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition
[FIRST|AFTER col_name]
| MODIFY [COLUMN] col_name column_definition [FIRST | AFTER
col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO|AS] new_tbl_name
| ORDER BY col_name [, col_name] ...
```

Lab Task

Given the following database schema:

Student (snum: integer, sname: char(30), major: char(25), level: char(2))

Faculty (fid: integer, fname: char(30), deptid: integer)

Class (cname: char(40), meets_at: char(20), room: char(10), fid: integer | fid REFS Faculty.fid)

Enrolled (snum: integer, cname: char(40) | snum REFS student.snum, cname REFS class.name)

Write SQL expressions for each of the following queries and execute them:

1. Create a database for these four relations. You need to define the primary keys and foreign keys in your statement. After creating the database, evolve it as follows.
2. Add a new attribute age in STUDENT table.
3. In table STUDENT, the age should be not less than 18 and not greater than 35.

4. Modify data type of attribute: NAME (i.e. cname, sname, fname) in all tables to varchar data type.
5. Add a new NOT NULL constraint to DEPTID in FACULTY table.
6. After creating the database using your SQL statements, populate the database according to the data given in text files using the SQL INSERT commands.

Deliverables

1. Complete your lab tasks in SQL workbench and submit a word file in with queries along with the screenshots of the results to all the questions attempted. Upload it on LMS. The marking will be based on viva/lab task submitted.