# Department of Computing

**CS220: Database Systems**

**Class: BSCS-9A**

**Lab 03: Introduction to SQL**

**Time: 09000-1200**

**Lab Engineer: Ms. Sadia Amir**

# 

# 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.

CREATE TABLE Student

(snum integer,

sname char(30),

major char(25),

level char(2),

Primary key(snum)

);

CREATE TABLE Faculty

(fid integer, fname char(30), deptid integer,Primary key(fid));

CREATE TABLE Class(cname char(40),

meets\_at char(20),

room char(10),

fid integer ,

Primary key(cname),

Foreign key(fid) references faculty(fid)

);

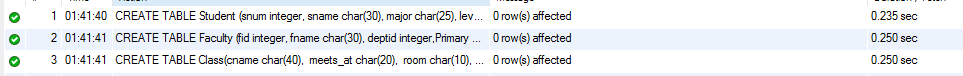
CREATE TABLE Enrollment (snum INTEGER,

cname CHAR(20),

PRIMARY KEY (snum,cname),

FOREIGN KEY (snum) REFERENCES Student(snum),

FOREIGN KEY (cname) REFERENCES Class(cname));



1. Add a new attribute age in STUDENT table. ~~.~~

ALTER TABLE STUDENT ADD Column Age int;



1. Modify data type of attribute: NAME (i.e. cname, sname, fname) in all tables to varchar data type.

ALTER TABLE enrollment MODIFY cname varchar(30);

ALTER TABLE student MODIFY sname varchar(30);

ALTER TABLE faculty MODIFY fname varchar(30);

1. Add a new NOT NULL constraint to DEPTID in FACULTY table.

ALTER table faculty ADD CHECK(deptid is NOT NULL);



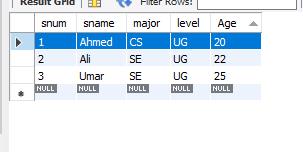
1. After creating the database using your SQL statements, populate the database according to the data given in text files using the SQL INSERT commands.

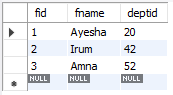
INSERT INTO enrollment(snum,cname) VALUES(1,'CS9A'),(2,'CS9B'),(3,'CS9B');

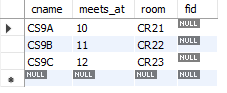
INSERT INTO

student(snum,sname,major,level,age)VALUES(1,'Ahmed',CS’,’UG’,20),(2,'Ali',’SE’ ’UG’,22,),(3,'Umar',’CS,’UG’,25);

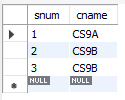
**Student Table**

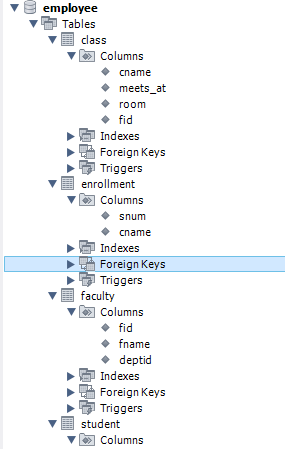


**Faculty Table Class Table**



**Enrollment Table**





# 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.