# Department of Computing

**CS220: Database Systems**

**Class: BSCS-9**

**Lab 05: Grouping & Aggregation in SQL**

**Time: 0900-1200**

**Lab Engineer: Miss 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

**Aggregate Functions**

The functions are used to summarize information from multiple tuples into a single-tuple summary. Well known built-in aggregate functions are COUNT, SUM, MAX, MIN, and AVG.

**Grouping – Group By** clause:

It creates subgroups of tuples before summarizing. Grouping is based on grouping attribute(s).

**HAVING** clause

It provides a condition on the summary information, i.e. grouping.

**Note**: Aggregate functions can be used in the SELECT clause or in a HAVING clause.

Given the following **database schema**:

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

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)

1. **Find the numbers of class rooms.**

select COUNT(\*) from class

1. **Find student strength in each class**.

select c.cname, COUNT(\*)

from class c, enrolled e

where c.cname = e.cname

group by c.cname

1. **Find the class names, and their rooms of all classes that have five or more students enrolled in it.**

Select C.cname, C.room

From class C, enrolled E

Where C.cname = E.cname

Group by E.cname

Having COUNT (\*) >= 5

# Lab Task

**TABLE CREATION**

CREATE TABLE student

(sno int,sname char(20),major char(25),level char(2),age int );

CREATE TABLE faculty

(fid int,fname char(30),deptid int(11));

CREATE TABLE enrolled(snum int,cname char(40) );

CREATE TABLE class(cname char(40),meets\_at char(20),room char(20),fid int);

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

1. **Find average age of students.**

QUERY

Select avg(age) from student ;

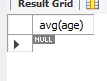
OUTPUT



1. **Find the average age of student in course: ‘Organic Chemistry’.**

QUERY

select avg(age) from student S,enrolled E where E.cname= "Organic Chemistry" and E.snum=S.sno

OUTPUT

1. **Find eldest student.**

QUERY

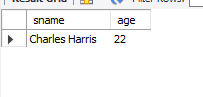
SELECT S.sname, S.age

FROM Student S

WHERE (SELECT max(S2.age)

FROM student S2)

= S.age

OUTPUT

1. **Find youngest student in ‘Electrical Engineering’ major.**

Query

SELECT S.sname, S.age

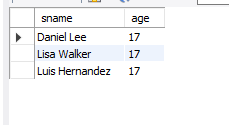
FROM Student S

WHERE(SELECT min(S2.age)

FROM student S2 Where s2.major='Electrical Engineering'

= S.age

OUTPUT

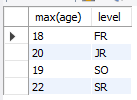


1. **Find eldest student in each level.**

Query

SELECT max(age),level from student GROUP BY level;

OUTPUT

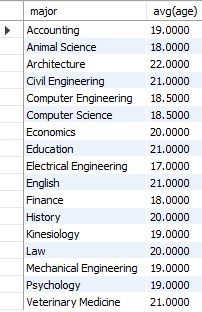


1. **Find average age of students in each level**

Query

Select major,avg(age) from student group by major

OUTPUT

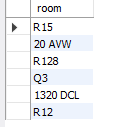


1. **Find the number of distinct class rooms**

QUERY

Select distinct room from class;

OUTPUT

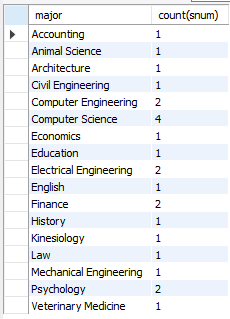


1. **Find the strength of students in each major.**

QUERY

Select major,count(snum) from student group by major

OUTPUT

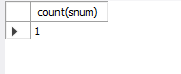


1. **Find strength of students in course: ‘Urban Economics’**

QUERY

select count(snum) from enrolled where cname='Urban Economics'

OUTPUT



1. **Find the number of courses for student: ‘Karen Scott’.**

QUERY

SELECT count(cname),sname from student S, enrolled E

where sname ="Karen Scott" AND S.sno=E.snum ;

OUTPUT

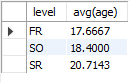


1. **Print the level and the average age of students for that level for all levels except ‘JR’.**

QUERY

Select level,avg(age) from student group by level having level!='JR';

OUTPUT

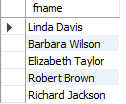


1. **For each faculty member that has taught classes only in room R128, print the faculty**

QUERY

Select fname from faculty F,class C where F.fid=C.fid AND C.room='R128';

OUTPUT

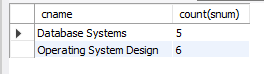


**13.Find the names of all classes and their enrollment strength that have enrollment greater than 2**

QUERY

Select cname,count(snum) from enrolled group by cname having count(snum)>2;

OUTPUT



**14.Find the names and strength of classes taught by ‘Ivana Teach’.**

QUERY

Select count(snum),C.cname from enrolled E,faculty F,class C where F.fname='IVANA TEACH' AND f.fid=c.fid AND C.cname=E.cname;

OUTPUT



**15.Find the number of faculty members that teach to class ‘database Systems’.**

QUERY

Select count(fid) from class where cname='database systems' ;

OUTPUT

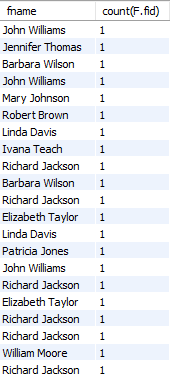


**16.Find the names, strength of faculty members for each class.**

QUERY

Select fname,count(F.fid) from class C,faculty F where c.fid=F.fid group by C.cname;

OUTPUT



**17.Find the age of youngest student in ‘Database Systems’ class.**

QUERY

select min(s.age) from student s, enrolled E

where E.cname='Database Systems' AND E.snum=S.sno;

OUTPUT

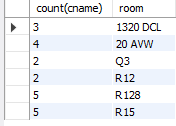


**18.Find the number of classes that occurred in the same room.**

QUERY

Select count(cname),room from class group by room;

OUTPUT



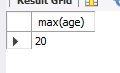
**19.Find the name and strength of students taught by each faculty member.**

**20.Find the age of eldest student in the class taught by ‘Ivana Teach’.**

QUERY

select max(age) from student s, class c , faculty f, enrolled E where f.fname='Ivana Teach' AND f.fid=c.fid AND c.cname=E.cname AND s.sno=E.snum

OUTPUT



# 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.