

East West University

Department of Computer Science & Engineering

**Course Title:** Database System

**Course Code:** CSE301  
**Experiment No:** 04

**Semester:** Fall, 2015

**Section:** 01

**Submitted By:**

S. M. Amirul Islam

ID: 2013-1-60-048

**Introduction**

These functins operate on the multiset of values of a column of a relation, and return a value.

avg: average value

min: minimum value

max: maximum value

sum: sum of values

count: number of values

**Lab Work**

Create Table & Data Insert Query:

create table borrower(

Loan\_no int,

Customer\_id int,

primary key(loan\_no)

);

create table loan(

loan\_no int,

amount int,

branch\_name varchar(50),

primary key(loan\_no)

);

create table depositor(

acc\_no int,

customer\_id int,

primary key(acc\_no)

);

create table customer(

customer\_id int,

name varchar(50),

address varchar(50),

primary key(customer\_id)

);

create table account(

acc\_no int,

balance int,

branch\_name varchar(50),

primary key(acc\_no)

);

insert into borrower(Loan\_no,Customer\_id) values(245,105)

insert into borrower(Loan\_no,Customer\_id) values(310,925)

insert into borrower(Loan\_no,Customer\_id) values(356,820)

insert into borrower(Loan\_no,Customer\_id) values(450,820)

insert into loan(loan\_no,amount,branch\_name) values(245,10000,'AB')

insert into loan(loan\_no,amount,branch\_name) values(310,15000,'BD')

insert into loan(loan\_no,amount,branch\_name) values(350,8000,'BD')

insert into loan(loan\_no,amount,branch\_name) values(450,7000,'CD')

insert into depositor(acc\_no,customer\_id) values(101,307)

insert into depositor(acc\_no,customer\_id) values(103,925)

insert into depositor(acc\_no,customer\_id) values(106,820)

insert into customer(customer\_id,name,address) values(105,'Asif','Dhaka')

insert into customer(customer\_id,name,address) values(925,'Ana','Sylhet')

insert into customer(customer\_id,name,address) values(820,'Dona','Raj')

insert into customer(customer\_id,name,address) values(207,'Alif','Bogra')

insert into account(acc\_no,balance,branch\_name) values(101,20000,'BD')

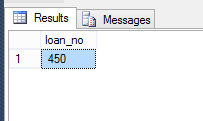
insert into account(acc\_no,balance,branch\_name) values(103,50000,'CD')

insert into account(acc\_no,balance,branch\_name) values(206,70000,'AD')

Q1.

select loan\_no from loan where amount=(select min(amount) from loan)

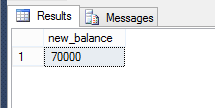
*Output:*



Q2.

(select max(balance) as new\_balance from account)

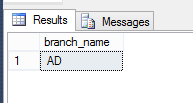
*Output:*



Q3.

select branch\_name from account where balance=(select max(balance) from account)

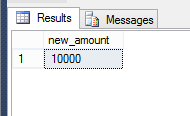
*Output:*



Q4.

select avg(amount) as new\_amount from loan

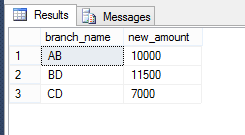
*Output:*



Q5.

select branch\_name,avg(amount) as new\_amount from loan group by branch\_name

*Output:*

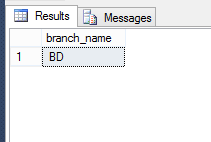


Q6.

select branch\_name from loan group by branch\_name

having avg (amount ) >= all(select avg(amount) from loan group by branch\_name)

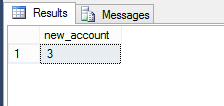
*Output:*



Q7.

select count(acc\_no) as new\_account from account

*Output:*

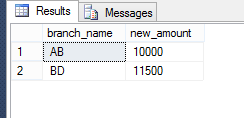


Q8.

select branch\_name,avg(amount) as new\_amount from loan group by branch\_name

having avg(amount)>8000

*Output:*



**Lab Exam**

1. What is aggregation function? Explain about it with examples.

Aggregate functions are used to compute against a "returned column of numeric data" from your SELECT statement. They basically summarize the results of a particular column of selected data. We are covering these here since they are required by the next topic, "GROUP BY". Although they are required for the "GROUP BY" clause, these functions can be used without the "GROUP BY" clause.

Example:

select avg(salary) from employee;

|  |  |
| --- | --- |
| MIN | returns the smallest value in a given column |
| MAX | returns the largest value in a given column |
| SUM | returns the sum of the numeric values in a given column |
| AVG | returns the average value of a given column |
| COUNT | returns the total number of values in a given column |
| COUNT(\*) | returns the number of rows in a table |

create table employee(

empid int,

empname varchar(50),

projectname varchar (50),

salary int,

primary key(empid)

);

insert into employee(empid,empname,projectname,salary) values(741,'asif','tista',10000)

insert into employee(empid,empname,projectname,salary) values(852,'rana','tista',12000)

insert into employee(empid,empname,projectname,salary) values(412,'rajib','padma',15000)

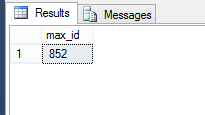
insert into employee(empid,empname,projectname,salary) values(842,'tara','tista',17000)

insert into employee(empid,empname,projectname,salary) values(558,'tamim','padma',8000)

Q4.

select max(empid) as max\_id from employee

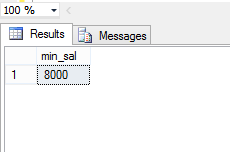
*Output:*



Q5.

select min(salary) as min\_sal from employee

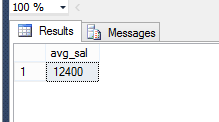
*Output:*



Q6.

select avg(salary) as avg\_sal from employee

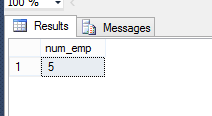
*Output:*



Q7.

select count(empid) as num\_emp from employee

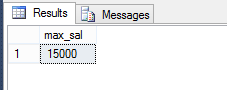
*Output:*



Q8.

select max(salary) as max\_sal from employee where projectname='padma'

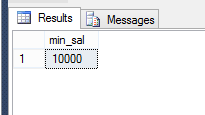
*Output:*



Q9.

select min(salary) as min\_sal from employee where projectname='tista'

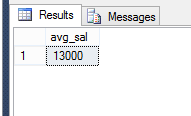
*Output:*



Q10.

select avg(salary) as avg\_sal from employee where projectname='tista'

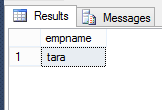
*Output:*



Q11.

select empname from employee where salary=(select max(salary) from employee)

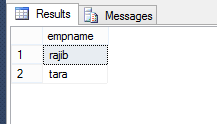
*Output:*



Q12.

select empname from employee where salary>(select avg(salary) from employee)

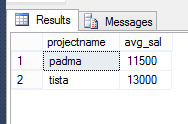
*Output:*



Q13.

select projectname,avg(salary)as avg\_sal from employee group by projectname

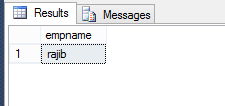
*Output:*



Q14.

select empname from employee where empid=(select min(empid) from employee)

*Output:*



**Discussion**

In this lab, we learned how to use aggregation functions using SQL query.