



DBMS LAB – 4

Aggregate functions, Grouping,
String and Numeric functions

NAME : RAHUL KATINNI

ROLL NO : S2020010091

EXERCISE 1:

1. Calculate the total number of employees name available in the table
2. Display the maximum salary of each department and also all departments put together
3. Find the employees whose salary is between 100000 and 500000 but not exactly 120000.
4. Get the count of employees whose income is more than 1 lakh.
5. List the employees according to ascending order of salary
6. For each department, retrieve the department name, the number of employees in the department, and Maximum income for the department.
7. List the number of employees in each place.
8. List the number of employee in each country sorted high to low
9. List the number of employees in each place. (Only include places with more than 1 employee)
10. List the number of employees in each place, except the California, sorted high to low. Only include places with 2 or more employees

Creating tables

```
mysql> CREATE TABLE employee(emp_id int AUTO_INCREMENT, emp_name varchar(20) ,emp_dept varchar(20), emp_age int, place varchar(20), income int,doj date, PRIMARY KEY(emp_id));
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> desc employee;
```

Field	Type	Null	Key	Default	Extra
emp_id	int	NO	PRI	NULL	auto_increment
emp_name	varchar(20)	YES		NULL	
emp_dept	varchar(20)	YES		NULL	
emp_age	int	YES		NULL	
place	varchar(20)	YES		NULL	
income	int	YES		NULL	
doj	date	YES		NULL	

```
7 rows in set (0.01 sec)
```

```
mysql> ALTER TABLE employee AUTO_INCREMENT=2505;
```

```
Query OK, 0 rows affected (0.04 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

Load data into table in workbench

```
mysql> select * from employee;
```

emp_id	emp_name	emp_dept	emp_age	place	income	doj
2505	peter	Finance	32	Newyork	100000	2002-08-25
2506	Mark	HR	32	California	120000	1980-03-25
2507	Donald	Finance	28	Arizona	100000	1995-12-26
2508	Obama	Management	35	Florida	500000	1990-10-30
2509	Linklon	HR	25	Georgia	25000	2008-08-08
2510	Kane	Sales	29	Alaska	30000	2000-01-01
2511	Adam	Management	38	California	54000	2020-10-25
2512	Mac	Finance	40	Florida	280000	1970-06-09
2513	Manas	Accounts	29	India	600000	1990-12-11
2514	Vasin	Accounts	30	India	800000	1989-10-10
2515	peter	Finance	32	Newyork	100000	1989-10-10
2516	Mark	HR	32	California	120000	1990-12-11
2517	Donald	Finance	28	Arizona	100000	1970-06-09
2518	Obama	Management	35	Florida	500000	2020-10-25
2519	Linklon	HR	25	Georgia	25000	2000-01-01
2520	Kane	Sales	29	Alaska	30000	2008-08-08
2521	Adam	Management	38	California	54000	1990-10-30
2522	Mac	Finance	40	Florida	280000	1995-12-26
2523	Manas	Accounts	29	India	600000	1980-03-25
2524	Vasin	Accounts	30	India	800000	2002-08-25

```
20 rows in set (0.01 sec)
```

Questions 1,2

```
mysql> select count(*) from employee;
```

```
+-----+  
| count(*) |  
+-----+  
|      20 |  
+-----+
```

```
1 row in set (0.01 sec)
```

```
mysql> select emp_dept, MAX(income) from employee group by emp_dept;
```

```
+-----+-----+  
| emp_dept | MAX(income) |  
+-----+-----+  
| Finance  |      280000 |  
| HR       |      120000 |  
| Management |     500000 |  
| Sales    |       30000 |  
| Accounts |     800000 |  
+-----+-----+
```

```
5 rows in set (0.01 sec)
```


Questions 3,4

```
mysql> select * from employee where income between 100000 and 500000 and income != 120000;
```

emp_id	emp_name	emp_dept	emp_age	place	income	doj
2505	peter	Finance	32	Newyork	100000	2002-08-25
2507	Donald	Finance	28	Arizona	100000	1995-12-26
2508	Obama	Management	35	Florida	500000	1990-10-30
2512	Mac	Finance	40	Florida	280000	1970-06-09
2515	peter	Finance	32	Newyork	100000	1989-10-10
2517	Donald	Finance	28	Arizona	100000	1970-06-09
2518	Obama	Management	35	Florida	500000	2020-10-25
2522	Mac	Finance	40	Florida	280000	1995-12-26

```
8 rows in set (0.01 sec)
```

```
mysql> select count(*) from employee where income >= 100000;
```

count(*)
14

```
1 row in set (0.01 sec)
```

Question 5

```
mysql> select * from employee order by income asc;
```

emp_id	emp_name	emp_dept	emp_age	place	income	doj
2509	Linklon	HR	25	Georgia	25000	2008-08-08
2519	Linklon	HR	25	Georgia	25000	2000-01-01
2510	Kane	Sales	29	Alaska	30000	2000-01-01
2520	Kane	Sales	29	Alaska	30000	2008-08-08
2511	Adam	Management	38	California	54000	2020-10-25
2521	Adam	Management	38	California	54000	1990-10-30
2505	peter	Finance	32	Newyork	100000	2002-08-25
2507	Donald	Finance	28	Arizona	100000	1995-12-26
2515	peter	Finance	32	Newyork	100000	1989-10-10
2517	Donald	Finance	28	Arizona	100000	1970-06-09
2506	Mark	HR	32	California	120000	1980-03-25
2516	Mark	HR	32	California	120000	1990-12-11
2512	Mac	Finance	40	Florida	280000	1970-06-09
2522	Mac	Finance	40	Florida	280000	1995-12-26
2508	Obama	Management	35	Florida	500000	1990-10-30
2518	Obama	Management	35	Florida	500000	2020-10-25
2513	Manas	Accounts	29	India	600000	1990-12-11
2523	Manas	Accounts	29	India	600000	1980-03-25
2514	Vasin	Accounts	30	India	800000	1989-10-10
2524	Vasin	Accounts	30	India	800000	2002-08-25

```
20 rows in set (0.00 sec)
```

Question 6,7

```
mysql> select count(*), emp_dept, MAX(income) from employee group by emp_dept;
```

count(*)	emp_dept	MAX(income)
6	Finance	280000
4	HR	120000
4	Management	500000
2	Sales	30000
4	Accounts	800000

```
5 rows in set (0.00 sec)
```

```
mysql> select count(*), place from employee group by place;
```

count(*)	place
2	Newyork
4	California
2	Arizona
4	Florida
2	Georgia
2	Alaska
4	India

```
7 rows in set (0.00 sec)
```


Question 8,9

```
mysql> select count(*), place from employee group by place order by count(place) desc;
```

count(*)	place
4	California
4	Florida
4	India
2	Newyork
2	Arizona
2	Georgia
2	Alaska

```
7 rows in set (0.01 sec)
```

```
mysql> select count(*), place from employee group by place having count(place) > 1 order by count(place) desc;
```

count(*)	place
4	California
4	Florida
4	India
2	Newyork
2	Arizona
2	Georgia
2	Alaska

```
7 rows in set (0.00 sec)
```

Question 10

```
mysql> select count(*), place from employee where place != "California" group by place having count(place) >= 2 order by count(place) desc;
```

count(*)	place
4	Florida
4	India
2	Newyork
2	Arizona
2	Georgia
2	Alaska

6 rows in set (0.00 sec)

EXERCISE 2:

1. Create the tables for above schema and load data from the respective .csv files
2. For all customers who have loan from the bank, find their names, loan numbers and loan amount(with and without renaming tables)
3. Find the customer names, loan numbers and loan amounts for all loans at perryridge branch.
4. Find the names of all branches that have assets greater than at least one branch located at Brooklyn.
5. List in alphabetical order all customers who have loans at the perryridge branch.
6. Print the entire Loan relation in descending order of amount.If several loans have the same amount,order them in ascending order by loan number.
7. Find the average balance for all accounts.
8. Find no.of tuples in customer relation.
9. Find the total of all loan amounts.
10. Find the average account balance at the Perryridge branch.
11. Find the average account balance at each branch.
12. Find the average account balance at each branch ,where the account balance is more than 1200.
13. Find the number of depositors for each branch.
14. Find the average balance for each customer who lives in "Harrison" and has at least 3 accounts

Creating tables

```
mysql> Create table customer (customer_name char(20),customer_street char(30),customer_city  
-> char(30),PRIMARY KEY(customer_name));  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> Create table branch (branch_name char(15),branch_city char(30),assets  
-> numeric(16,2),PRIMARY KEY(branch_name));  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> Create table account (account_number char(15),branch_name char (15),balance  
-> numeric(12,2),PRIMARY KEY(account_number),FOREIGN KEY (branch_name)  
-> REFERENCES branch(branch_name));  
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> Create table depositor(customer_name char(20),account_number char(10),PRIMARY  
-> KEY(customer_name,account_number),FOREIGN KEY (customer_name)  
-> REFERENCES customer(customer_name),FOREIGN KEY (account_number)  
-> REFERENCES account(account_number));  
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> Create table loan(loan_number varchar(6),branch_name char(15),amount int,PRIMARY  
-> KEY(loan_number),FOREIGN KEY (branch_name) REFERENCES  
-> branch(branch_name));  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> Create table borrower(customer_name char(20),loan_number varchar(6),PRIMARY  
-> KEY(customer_name,loan_number),FOREIGN KEY (customer_name) REFERENCES  
-> customer(customer_name),FOREIGN KEY (loan_number) REFERENCES  
-> loan(loan_number));  
Query OK, 0 rows affected (0.07 sec)
```


1. Load data into table in workbench and show tables

```
mysql> select * from account;
```

account_number	branch_name	balance
A-101	Downtown	500.00
A-102	Perryridge	400.00
A-201	Brighton	900.00
A-215	Mianus	700.00
A-217	Brighton	750.00
A-222	Redwood	700.00
A-305	Round Hill	350.00

```
7 rows in set (0.00 sec)
```

```
mysql> select * from borrower;
```

customer_name	loan_number
Smith	L-11
Hayes	L-15
Adams	L-16
Jones	L-17
Williams	L-17
Smith	L-23
Curry	L-93

```
7 rows in set (0.00 sec)
```

```
mysql> select * from branch;
```

branch_name	branch_city	assets
Brighton	Brooklyn	7100000.00
Downtown	Brooklyn	9000000.00
Mianus	Horseneck	400000.00
North Town	Rye	3700000.00
Perryridge	Horseneck	1700000.00
Pownal	Bennington	300000.00
Redwood	Palo Alto	2100000.00
Round Hill	Horseneck	8000000.00

```
8 rows in set (0.01 sec)
```

```
mysql> select * from customer;
```

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

```
12 rows in set (0.00 sec)
```

```
mysql> select * from depositor;
```

customer_name	account_number
Johnson	A-101
Hayes	A-102
Johnson	A-201
Smith	A-215
Jones	A-217
Lindsay	A-222
Turner	A-305

```
7 rows in set (0.02 sec)
```

```
mysql> select * from loan;
```

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

```
7 rows in set (0.00 sec)
```

Question 2

```
mysql> select distinct customer_name, amount, borrower.loan_number as loan_id from borrower, loan where borrower.loan_number = loan.loan_number;
```

customer_name	amount	loan_id
Smith	900	L-11
Hayes	1500	L-15
Adams	1300	L-16
Jones	1000	L-17
Williams	1000	L-17
Smith	2000	L-23
Curry	500	L-93

```
7 rows in set (0.00 sec)
```

Questions 3,4

```
mysql> select distinct customer_name, amount, borrower.loan_number as loan_id from borrower, loan where b  
orrower.loan_number = loan.loan_number and branch_name = "Perryridge";
```

customer_name	amount	loan_id
Hayes	1500	L-15
Adams	1300	L-16

2 rows in set (0.03 sec)

```
mysql> select distinct M.branch_name from branch as M, branch as N where M.assets > N.assets and N.branch  
_city = "Brooklyn";
```

branch_name
Downtown
Round Hill

2 rows in set (0.00 sec)

Question 5,6

```
mysql> select distinct customer_name from borrower, loan where borrower.loan_number = loan.loan_number and  
branch_name = "Perryridge" order by customer_name;
```

```
+-----+  
| customer_name |  
+-----+  
| Adams         |  
| Hayes         |  
+-----+
```

```
2 rows in set (0.00 sec)
```

```
mysql> select * from loan order by amount desc, loan_number asc;
```

```
+-----+-----+-----+  
| loan_number | branch_name | amount |  
+-----+-----+-----+  
| L-23       | Redwood     | 2000   |  
| L-14       | Downtown    | 1500   |  
| L-15       | Perryridge  | 1500   |  
| L-16       | Perryridge  | 1300   |  
| L-17       | Downtown    | 1000   |  
| L-11       | Round Hill  | 900    |  
| L-93       | Mianus      | 500    |  
+-----+-----+-----+
```

```
7 rows in set (0.00 sec)
```

Question 7,8

```
mysql> select avg (balance) from account;
```

```
+-----+  
| avg (balance) |
```

```
+-----+  
|      614.285714 |
```

```
+-----+
```

```
1 row in set (0.03 sec)
```

```
mysql> select count(*) from customer;
```

```
+-----+  
| count(*) |
```

```
+-----+  
|        12 |
```

```
+-----+
```

```
1 row in set (0.03 sec)
```

Question 9,10

```
mysql> select sum(amount) from loan;
```

```
+-----+  
| sum(amount) |  
+-----+  
|          8700 |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> select avg(balance) from account where branch_name = "Perryridge";
```

```
+-----+  
| avg(balance) |  
+-----+  
| 400.000000 |  
+-----+
```

```
1 row in set (0.03 sec)
```

Question 11,12

```
mysql> select branch_name, avg(balance) from account group by branch_name;
```

branch_name	avg(balance)
Brighton	825.000000
Downtown	500.000000
Mianus	700.000000
Perryridge	400.000000
Redwood	700.000000
Round Hill	350.000000

```
mysql> select branch_name, avg(balance) from account group by branch_name having avg(balance) > 1200;  
Empty set (0.00 sec)
```


Question 13,14

```
mysql>
mysql> select branch_name, count(distinct customer_name) from depositor, account where depositor.account_
number = account.account_number group by branch_name;
+-----+-----+
| branch_name | count(distinct customer_name) |
+-----+-----+
| Brighton   | 2 |
| Downtown   | 1 |
| Mianus      | 1 |
| Perryridge | 1 |
| Redwood     | 1 |
| Round Hill  | 1 |
+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> select avg(balance), customer_name from depositor natural join account natural join customer where
customer_city = "Harrison" group by customer_name having count(customer_name)>=3;
Empty set (0.00 sec)
```