

EXERCISE 1

1.Create a database;

```
mysql>create database employee;
```

Query OK, 1 row affected (0.01 sec)

2.Create table Employee with the following filed – empno int primary key,ename varchar(30),JOB varchar(10),managerid int,hiredate date,salary int,commision int,deptno int

```
mysql> CREATE TABLE Employee (
```

```
->  empno INT PRIMARY KEY,
```

```
->  ename VARCHAR(30),
```

```
->  JOB VARCHAR(10),
```

```
->  managerid INT,
```

```
->  hiredate DATE,
```

```
->  salary INT,
```

```
->  commision INT,
```

```
->  deptno INT
```

```
-> );
```

Query OK, 0 rows affected (0.04 sec)

3.describe schema.

```
mysql>DESCRIBE Employee;
```

Field	Type	Null	Key	Default	Extra
empno	int	NO	PRI	NULL	
ename	varchar(30)	YES		NULL	
JOB	varchar(10)	YES		NULL	
managerid	int	YES		NULL	
hiredate	date	YES		NULL	
salary	int	YES		NULL	
commision	int	YES		NULL	
deptno	int	YES		NULL	

8 rows in set (0.00 sec)

4.create a table department with the following field -deptno int primary key,deptname varchar(15),location varchar(10)

```
mysql>CREATE TABLE Department (  
-> deptno INT PRIMARY KEY,  
-> deptname VARCHAR(15),  
-> location VARCHAR(10)  
-> );
```

Query OK, 0 rows affected (0.02 sec)

5.describe schema

```
mysql>DESCRIBE Department;
```

Field	Type	Null	Key	Default	Extra
deptno	int	NO	PRI	NULL	
deptname	varchar(15)	YES		NULL	
location	varchar(10)	YES		NULL	

3 rows in set (0.00 sec)

6. create a table salarygrade with the following field - (grade int primary key,lowsalary int,highsalary int);

```
mysql>CREATE TABLE SalaryGrade (  
-> grade INT PRIMARY KEY,  
-> lowsalary INT,  
-> highsalary INT  
-> );
```

Query OK, 0 rows affected (0.02 sec)

7.describe schema

```
mysql>DESCRIBE SalaryGrade;
```

Field	Type	Null	Key	Default	Extra
grade	int	NO	PRI	NULL	
lowsalary	int	YES		NULL	
highsalary	int	YES		NULL	

3 rows in set (0.00 sec)

EXERCISE 2

.....

1.Insert atleast 5 values to each table(must include the following values)

a.job-clerk,salesman,manager etc

b.department name-accounting,research,sales etc

c.employee name – allen,smith,ward etc.

d.dept number -10,20

e.Location-US

```
mysql> INSERT INTO Department VALUES
```

```
-> (10, 'Accounting', 'US'),
```

```
-> (20, 'Research', 'US'),
```

```
-> (30, 'Sales', 'US'),
```

```
-> (40, 'Marketing', 'US'),
```

```
-> (50, 'IT', 'US');
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO Employee VALUES
```

```
-> (1001, 'Allen', 'Salesman', 1005, '1981-06-01', 1600, 300, 30),
```

```
-> (1002, 'Smith', 'Clerk', 1006, '1987-12-09', 800, NULL, 20),
```

```
-> (1003, 'Ward', 'Salesman', 1005, '1981-02-22', 1250, 500, 30),
```

```
-> (1004, 'Jones', 'Manager', 1007, '1981-04-02', 2975, NULL, 20),
```

```
-> (1005, 'Martin', 'Salesman', 1006, '1981-09-28', 1250, 1400, 30);
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO SalaryGrade VALUES
```

```
-> (1, 700, 1200),
```

-> (2, 1201, 1400),

-> (3, 1401, 2000),

-> (4, 2001, 3000),

-> (5, 3001, 9999);

Query OK, 5 rows affected (0.00 sec)

Records: 5 Duplicates: 0 Warnings: 0

2. Select all information from table employee.

```
mysql> SELECT * FROM Employee;
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1001	Allen	Salesman	1005	1981-06-01	1600	300	30
1002	Smith	Clerk	1006	1987-12-09	800	NULL	20
1003	Ward	Salesman	1005	1981-02-22	1250	500	30
1004	Jones	Manager	1007	1981-04-02	2975	NULL	20
1005	Martin	Salesman	1006	1981-09-28	1250	1400	30

5 rows in set (0.00 sec)

3. Select all information from table department.

```
mysql> SELECT * FROM Department;
```

deptno	deptname	location
10	Accounting	US
20	Research	US
30	Sales	US
40	Marketing	US
50	IT	US

5 rows in set (0.00 sec)

4.Select all information from table salarygrade.

```
mysql> SELECT * FROM SalaryGrade;
```

grade	lowsalary	highsalary
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

5 rows in set (0.00 sec)

4.Select empno,ename form table employee.

```
mysql> SELECT empno, ename FROM Employee;
```

empno	ename
1001	Allen
1002	Smith
1003	Ward
1004	Jones
1005	Martin

5 rows in set (0.00 sec)

5.List all employees having a salary range between 1000 and 2000

```
mysql> SELECT * FROM Employee WHERE salary BETWEEN 1000 AND 2000;
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1001	Allen	Salesman	1005	1981-06-01	1600	300	30
1003	Ward	Salesman	1005	1981-02-22	1250	500	30
1005	Martin	Salesman	1006	1981-09-28	1250	1400	30

3 rows in set (0.00 sec)

6.List dname and department number in department name order.

```
mysql> SELECT deptname, deptno FROM Department ORDER BY deptname;
```

deptname	deptno
Accounting	10
IT	50
Marketing	40
Research	20
Sales	30

5 rows in set (0.00 sec)

7.List the employee details in department 10 and 20

```
mysql> SELECT * FROM Employee WHERE deptno IN (10, 20);
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1002	Smith	Clerk	1006	1987-12-09	800	NULL	20
1004	Jones	Manager	1007	1981-04-02	2975	NULL	20

2 rows in set (0.00 sec)

8.List names and jobs of all clerks in dept 20

```
mysql> SELECT ename, job FROM Employee WHERE job='Clerk' AND deptno=20;
```

ename	job
Smith	Clerk

1 row in set (0.00 sec)

9.Display all employee names which have TH or LL in name

```
mysql> SELECT ename FROM Employee WHERE ename LIKE '%TH%' OR ename LIKE '%LL%';
```

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

```
|  ename  |
```

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

```
|  Allen  |
```

```
|  Smith  |
```

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

2 rows in set (0.00 sec)

10. List name, job, and salary of all employees who have a manager.

```
mysql> SELECT ename, job, salary FROM Employee WHERE managerid IS NOT NULL;
```

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

```
|  ename  |  job      |  salary  |
```

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

```
|  Allen  |  Salesman |    1600  |
```

```
|  Smith  |   Clerk   |     800  |
```

```
|  Ward   |  Salesman |    1250  |
```

```
|  Jones  |  Manager  |    2975  |
```

```
|  Martin |  Salesman |    1250  |
```

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

5 rows in set (0.00 sec)

11. Display name and annual remuneration for all employees.

```
mysql> SELECT ename, (salary * 12 + IFNULL(commision,0)) AS annual_remuneration FROM Employee;
```

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

```
|  ename  | annual_remuneration |
```

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

```
|  Allen  |                19500 |
```

```
|  Smith  |                 9600 |
```

```
|  Ward   |                15500 |
```

```
|  Jones  |                35700 |
```

```
|  Martin |                16400 |
```

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

5 rows in set (0.00 sec)

12. Display all employees hired during 1987.


```
mysql> SELECT * FROM Employee WHERE YEAR(hiredate) = 1987;
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1002	Smith	Clerk	1006	1987-12-09	800	NULL	20

1 row in set (0.00 sec)

13.Display name,job,annual sal,commission of all sales peoples whose monthly salary greater than commission.The output should be order by salary highest first.

```
mysql> SELECT ename, job, (salary*12) AS annual_sal, commision
```

-> FROM Employee

-> WHERE job='Salesman' AND salary > IFNULL(commision, 0)

-> ORDER BY salary DESC;

ename	job	annual_sal	commision
Allen	Salesman	19200	300
Ward	Salesman	15000	500

2 rows in set (0.00 sec)

14.List the employee name and salary increased by 12.5%.Express has a whole number

```
mysql> SELECT ename, ROUND(salary * 1.125) AS increased_salary FROM Employee;
```

ename	increased_salary
Allen	1800
Smith	900
Ward	1406
Jones	3347
Martin	1406

5 rows in set (0.00 sec)

15. Produce the following output

EMPLOYEE AND JOB

SMITH CLERK

ALLEN SALESMAN

```
mysql> SELECT CONCAT(ename, ' ', job) AS "EMPLOYEE AND JOB" FROM
Employee WHERE ename IN ('SMITH', 'ALLEN') order by ename desc;
```

```
+-----+
| EMPLOYEE AND JOB |
+-----+
| Smith      Clerk  |
| Allen      Salesman |
+-----+
```

2 rows in set (0.00 sec)

16. Produce the following output

EMPLOYEE AND JOB

SMITH(Clerk)

ALLEN(Salesman)

```
mysql> SELECT CONCAT(ename, '(', job, ')') AS "EMPLOYEE AND JOB" FROM
Employee WHERE ename IN ('SMITH', 'ALLEN') order by ename desc;
```

```
+-----+
| EMPLOYEE AND JOB |
+-----+
| Smith(Clerk)      |
| Allen(Salesman)   |
+-----+
```

2 rows in set (0.00 sec)

17. Find the minimum, maximum, and average salaries of all employees.

```
mysql> SELECT MIN(salary), MAX(salary), AVG(salary) FROM Employee;
```

```
+-----+-----+-----+
| MIN(salary) | MAX(salary) | AVG(salary) |
+-----+-----+-----+
|          800 |          2975 |    1575.0000 |
+-----+-----+-----+
```

18. List the minimum and maximum salary for each job.

```
mysql> SELECT job, MIN(salary), MAX(salary) FROM Employee GROUP BY job;
```

```
+-----+-----+-----+
| job          | MIN(salary) | MAX(salary) |
+-----+-----+-----+
| Salesman    |          1250 |          1600 |
| Clerk       |           800 |           800 |
| Manager     |          2975 |          2975 |
+-----+-----+-----+
```

3 rows in set (0.00 sec)

19. Find how many managers are there without listing them.

```
mysql> SELECT COUNT(*) FROM Employee WHERE job = 'Manager';
```

```
+-----+
| COUNT (*) |
+-----+
|          1 |
+-----+
```

1 row in set (0.00 sec)

20. Find the average salary and average total remuneration for each job.

```
mysql> SELECT job, AVG(salary) AS avg_salary,
-> AVG(salary + IFNULL(commision, 0)) AS avg_total
-> FROM Employee
-> GROUP BY job;
```

job	avg_salary	avg_total
Salesman	1366.6667	2100.0000
Clerk	800.0000	800.0000
Manager	2975.0000	2975.0000

3 rows in set (0.00 sec)

21. Find the difference between highest and lowest salaries.

```
mysql> SELECT MAX(salary) - MIN(salary) AS salary_difference FROM Employee;
```

salary_difference
2175

1 row in set (0.00 sec)

22. Find all departments having more than 3 employees.

```
mysql> SELECT deptno, JOB, COUNT(*) as emp_count
```

```
-> FROM Employee
```

```
-> GROUP BY deptno
```

```
-> HAVING COUNT(*) > 2;
```

deptno	JOB	emp_count
30	Salesman	3

1 row in set (0.00 sec)

23. Check whether all employee numbers are unique.

```
mysql> SELECT empno, COUNT(*)
```

```
-> FROM Employee
```

```
-> GROUP BY empno
```

```
-> HAVING COUNT(*) > 1;
```

Empty set (0.00 sec)

EXERCISE 3

.....

1. Display all employee names and their department names in department name order.

```
mysql> SELECT e.ename, d.deptname
-> FROM Employee e
-> JOIN Department d ON e.deptno = d.deptno
-> ORDER BY d.deptname;
```

```
+-----+-----+
| ename  | deptname |
+-----+-----+
| Smith  | Research |
| Jones  | Research |
| Allen  | Sales    |
| Ward   | Sales    |
| Martin | Sales    |
+-----+-----+
```

5 rows in set (0.00 sec)

2. Display the name, location, and department name of all employees whose salary is more than 1500.

```
mysql> SELECT e.ename, d.location, d.deptname
-> FROM Employee e
-> JOIN Department d ON e.deptno = d.deptno
-> WHERE e.salary > 1500;
```

```
+-----+-----+-----+
| ename | location | deptname |
+-----+-----+-----+
| Allen | US       | Sales    |
| Jones | US       | Research |
+-----+-----+-----+
```

2 rows in set (0.00 sec)

3. Produce a list showing employee's salary grade.

```
mysql> SELECT e.ename, e.salary, s.grade
```

```
-> FROM Employee e
```

```
-> JOIN SalaryGrade s ON e.salary BETWEEN s.lowsalary AND s.highsalary;
```

```
+-----+-----+-----+
| ename  | salary | grade |
+-----+-----+-----+
| Smith  |      800 |      1 |
| Ward   |     1250 |      2 |
| Martin |     1250 |      2 |
| Allen  |     1600 |      3 |
| Jones  |     2975 |      4 |
+-----+-----+-----+
```

5 rows in set (0.00 sec)

4. List employees in grade 3.

```
mysql> SELECT e.*
```

```
-> FROM Employee e
```

```
-> JOIN SalaryGrade s ON e.salary BETWEEN s.lowsalary AND s.highsalary
```

```
-> WHERE s.grade = 3;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| empno | ename | JOB       | managerid | hiredate   | salary | commision | deptno |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1001  | Allen | Salesman  | 1005      | 1981-06-01 | 1600   | 300      | 30     |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

1 row in set (0.00 sec)

5. Show all employees in US.

```
mysql> SELECT e.*
```

```
-> FROM Employee e
```

```
-> JOIN Department d ON e.deptno = d.deptno
```

```
-> WHERE d.location = 'US';
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1001	Allen	Salesman	1005	1981-06-01	1600	300	30
1002	Smith	Clerk	1006	1987-12-09	800	NULL	20
1003	Ward	Salesman	1005	1981-02-22	1250	500	30
1004	Jones	Manager	1007	1981-04-02	2975	NULL	20
1005	Martin	Salesman	1006	1981-09-28	1250	1400	30

5 rows in set (0.00 sec)

6. List employee name, job, salary, grade and department name for all except clerk. Sort on salary descending order..

```
mysql> SELECT e.ename, e.job, e.salary, s.grade, d.deptname
-> FROM Employee e
-> JOIN Department d ON e.deptno = d.deptno
-> JOIN SalaryGrade s ON e.salary BETWEEN s.lowsalary AND s.highsalary
-> WHERE e.job != 'Clerk'
-> ORDER BY e.salary DESC;
```

ename	job	salary	grade	deptname
Jones	Manager	2975	4	Research
Allen	Salesman	1600	3	Sales
Ward	Salesman	1250	2	Sales
Martin	Salesman	1250	2	Sales

4 rows in set (0.00 sec)

7. List the following details for all employees who earn 36000 a year or who are clerk.

```
mysql> SELECT ename, job, salary
-> FROM Employee
-> WHERE salary * 12 = 36000 OR job = 'Clerk';
```



```
+-----+-----+-----+
|  ename  | job    | salary |
+-----+-----+-----+
| Smith   | Clerk  |    800  |
+-----+-----+-----+
```

1 row in set (0.00 sec)

EXERCISE 4

.....

1.To display employees who earn morethan the lowest salary in department 30.

```
mysql> SELECT * FROM Employee
```

```
-> WHERE salary > (
```

```
->  SELECT MIN(salary) FROM Employee WHERE deptno = 30
```

```
-> );
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1001	Allen	Salesman	1005	1981-06-01	1600	300	30
1004	Jones	Manager	1007	1981-04-02	2975	NULL	20

2 rows in set (0.00 sec)

2.Find the employees who earn morethan every employees in department 30.

```
mysql> SELECT * FROM Employee
```

```
-> WHERE salary > ALL (
```

```
->  SELECT salary FROM Employee WHERE deptno = 30
```

```
-> );
```

empno	ename	JOB	managerid	hiredate	salary	commision	deptno
1004	Jones	Manager	1007	1981-04-02	2975	NULL	20

1 row in set (0.00 sec)

3.To find the job with the highest average salary.

```
mysql> SELECT job
```

```
-> FROM Employee
```

```
-> GROUP BY job
```

```
-> ORDER BY AVG(salary) DESC
```

```
-> LIMIT 1;
```

```

+-----+
| job    |
+-----+
| Manager |
+-----+

```

1 row in set (0.00 sec)

4. Find the departments not having any employees

```
mysql> SELECT * FROM Department
```

```
-> WHERE deptno NOT IN (SELECT DISTINCT deptno FROM Employee);
```

```

+-----+-----+-----+
| deptno | deptname | location |
+-----+-----+-----+
|      10 | Accounting | US      |
|      40 | Marketing  | US      |
|      50 | IT         | US      |
+-----+-----+-----+

```

3 rows in set (0.00 sec)

5. Display the name and salary of the top three earners in the company

```
mysql> SELECT ename, salary
```

```
-> FROM Employee
```

```
-> ORDER BY salary DESC
```

```
-> LIMIT 3;
```

```

+-----+-----+
| ename  | salary |
+-----+-----+
| Jones  | 2975   |
| Allen  | 1600   |
| Martin | 1250   |
+-----+-----+

```

3 rows in set (0.00 sec)

