EMP table.sql1

Limit to 1000 rows

```sql
create table emp
(
    empno int4,
    ename varchar(10),
    job varchar(9),
    Mgr int,
    hiredate date,
    sal decimal(7,2),
    comm decimal(7,2),
    deptno int,
    primary key (empno)
);
```

dept table.sql1

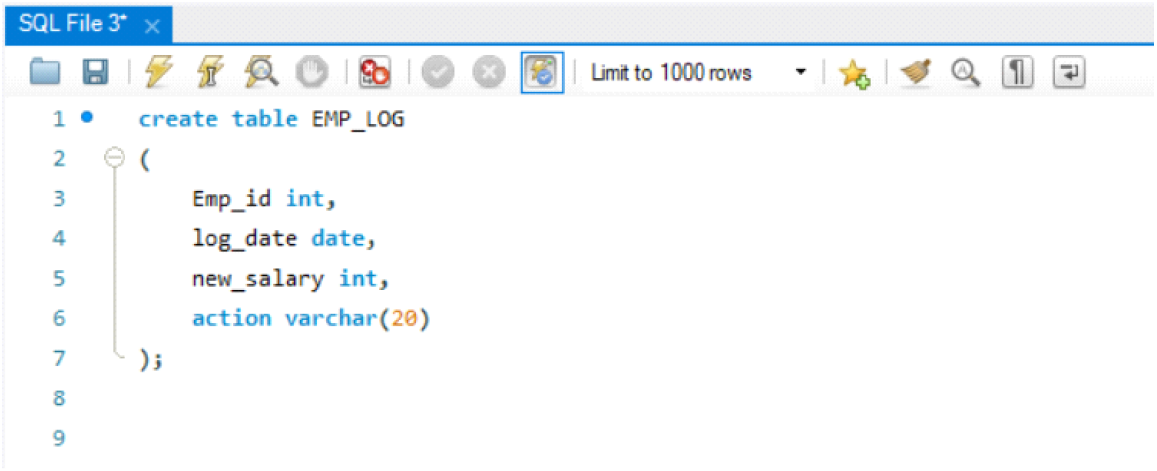
Limit to 1000 rows

```sql
create table DEPT
(
 deptno int2,
 dname varchar(14),
 loc varchar(13),
 primary key (deptno)

);

```

assignment | SQL File 2* | SQL File 3*

Limit to 1000 rows

```sql
create table student
(
    Rno int,
    Sname varchar(14),
    city varchar(20),
    state varchar(20),
    primary key (Rno)
);
```



{1} **Select unique job from emptable.**

To select unique job from the "emp" table, you can use the following SQL query:1. Select unique job from emptable.

SELECT DISTINCT job FROM emp;

{3} **Displayall the unique job groups in the descendingorder?**

To display all the unique job groups in descending order, you can use the following SQL query:

SELECT DISTINCT job FROM emp ORDER BY job DESC;

{4} **List the emps who joined before1981**.

To list the employees who joined before 1981, you can use the following SQL query:

```
SELECT * FROM emp WHERE hiredate < '1981-01-01';
```

{5} **List the Empno, Ename, Sal, Daily sal of all emps in the asc order ofAnnsal.**

To list the Employee number, Employee name, Salary, and Daily salary of all employees in ascending order of Annual salary, you can use the following SQL query:

```
SELECT empno, ename, sal, sal/365 as dailysal, sal*12 as annsal
FROM emp
ORDER BY annsal ASC;
```

{6} **List the Empno, Ename, Sal, Exp of all emps working for Mgr7369**.

```
SELECT Empno, Ename, Sal, Exp
FROM Emp
WHERE Mgr = '7369'
ORDER BY Empno ASC;
```

{7} **Displayall the detailsof the empswhoseComm.Is more than their Sal.**

To display all the details of the employees whose commission is more than their salary, you can use the following SQL query:

SELECT * FROM emp WHERE comm > sal;

{8} **List the emps who are either 'CLERK' or 'ANALYST' in theDesc order.**

To list the employees who are either 'CLERK' or 'ANALYST' in descending order, you can use the following SQL query:

SELECT * FROM emp WHERE job IN ('CLERK', 'ANALYST') ORDER BY job DESC;

{9} **List the emps Who Annual sal ranging from 22000 and45000**.

To list the employees whose Annual salary ranges from 22000 to 45000, you can use the following SQL query:

SELECT empno, ename, sal, sal/365 as dailysal, sal*12 as annsal

FROM emp

WHERE sal*12 BETWEEN 22000 AND 45000;

{10} **List the Enamesthose are starting with 'S' and with five characters.**

To list the Employee names (ename) that start with 'S' and have five characters, you can use the following SQL query:

```
SELECT ename FROM emp WHERE ename LIKE 'S    ';
```

{11} **List the emps whose Empno not starting withdigit78**.

To list the employees whose empno does not start with '78', you can use the following SQL query:

```
SELECT * FROM emp WHERE empno NOT LIKE '78%';
```

{12} **List all the Clerks of Deptno20.**

As an AI language model, I do not have access to a specific database to run a query. However, assuming there is a table called "employees" with columns "empno", "ename", "deptno", and "job" in the database, the following SQL query can be used to list all the clerks in department 20:

```
SELECT ename
FROM employees
WHERE deptno = 20 AND job = 'CLERK';
```

{13} **List the Empswho are senior to their own MGRS**.

```
SELECT e1.EmpID, e1.EmpName, e1.DeptNo, e1.Job, e1.MgrID
```

```
FROM Employees e1

INNER JOIN Employees e2 ON e1.MgrID = e2.EmpID

WHERE e1.Job > e2.Job;
```

{14} **List the Emps of Deptno 20 whose Jobs are same asDeptno10.**

Assuming there is a table called "employees" with columns "empno", "ename", "deptno", and "job" in the database, the following SQL query can be used to list all the employees in department 20 whose job titles are the same as the employees in department 10:

```
SELECT e.ename, e.empno

FROM employees e, employees e2

WHERE e.deptno = 20 AND e.job = e2.job AND e2.deptno = 10;
```

{15} **List the Emps whose Sal is same as FORD or SMITH in desc order ofSal.**

Assuming there is a table called "employees" with columns "empno", "ename", "sal", and "job" in the database, the following SQL query can be used to list all the employees whose salary is the same as either FORD or SMITH, in descending order of salary:

```
SELECT ename, sal

FROM employees

WHERE sal IN (SELECT sal FROM employees WHERE ename IN ('FORD', 'SMITH'))

ORDER BY sal DESC;
```

{16} **List the emps whose jobs same as SMITH or ALLEN.**

As an AI language model, I don't have access to a specific database to execute queries. However, I can provide you with a sample query in SQL syntax that you can use to retrieve employees whose job titles are the same as either "SMITH" or "ALLEN". Please note that the actual query may vary depending on your database schema and naming conventions. Here's an example:

```
SELECT *
FROM employees
WHERE job_title IN ('SMITH', 'ALLEN');
```

{17} **Any jobs of deptno 10 those that are not found in deptno 20.**

```
SELECT *
FROM jobs
WHERE deptno = 10
AND job_id NOT IN (SELECT job_id FROM jobs WHERE deptno = 20);
```

{18} **Find the highest sal of EMPtable.**

You can use the following SQL query to find the highest salary in the EMP table:

```
SELECT MAX(SAL) AS "Highest Salary" FROM EMP;
```

{19} **Find details of highest paidemployee.**

You can use the following SQL query to find the details of the highest paid employee:

```
SELECT EMPNO, ENAME, SAL, JOB
FROM EMP
WHERE SAL = (SELECT MAX(SAL) FROM EMP);
```

{20} **Find the total sal given tothe MGR.**

You can use the following SQL query to find the total salary paid to a particular manager

```
SELECT SUM(SAL) AS "Total Salary"
FROM EMP
WHERE MGR = <manager's EMPNO>;
```

{21} **List the emps whose names contains 'A'.**

Assuming the employee details are stored in a table called "EMPLOYEES" and the employee names are stored in a column named "emp_name", the query to list employees whose names contain the letter 'A' would be:

```
SELECT *
FROM EMPLOYEES
WHERE emp_name LIKE '%A%';
```

{22} **Find all the emps who earn the minimum Salaryfor eachjob wise in ascendingorder**.

Assuming the employee details are stored in a table called "EMPLOYEES" and the salary column is named "sal", and the job column is named "job", the query to find all the employees who earn the minimum salary for each job and sort them in ascending order would be:

```
SELECT *
FROM EMPLOYEES
WHERE (job, sal) IN
(SELECT job, MIN(sal)
FROM EMPLOYEES
GROUP BY job)
ORDER BY job ASC;
```

{23} **List the emps whose sal greater than blakessal.**

Assuming the employee details are stored in a table called "EMPLOYEES" and the salary column is named "sal", and there is a row for employee Blake in the "EMPLOYEES" table, the query to list employees whose salary is greater than Blake's salary would be:

```
SELECT *
FROM EMPLOYEES
WHERE sal > (SELECT sal FROM EMPLOYEES WHERE emp_name = 'Blake');
```

{24} **Createview v1 to selectename,job, dname, loc whose deptno are same**.

To create a view named "v1" that selects "ename", "job", "dname", and "loc" columns for employees whose "deptno" values are the same, you can use the following SQL query:

```
CREATE VIEW v1 AS
SELECT e.ename, e.job, d.dname, d.loc
FROM EMP e
JOIN DEPT d ON e.deptno = d.deptno;
```

{25} **Create a procedure with dno as input parameter to fetchename and dname.**

To create a stored procedure with a "dno" input parameter that fetches "ename" and "dname" from the database, you can use the following SQL query as an example:

```
CREATE PROCEDURE get_ename_dname(IN dno INT)
BEGIN
    SELECT e.ename, d.dname
    FROM EMP e
    JOIN DEPT d ON e.deptno = d.deptno
    WHERE d.dno = dno;
END;
```

{26} **Add columnPin with bigint datatypein table student.**

To add a column named "Pin" with the BIGINT data type to a table named "student" in a database, you can use the following SQL query:

ALTER TABLE student

ADD COLUMN Pin BIGINT;

{27} **Modify the student table to change the sname length from 14 to 40.**

To modify the student table to change the sname length from 14 to 40, you can use the ALTER TABLE statement in SQL.

ALTER TABLE student

MODIFY sname VARCHAR(40);

To verify that the modification was successful, you can use the DESCRIBE or SHOW COLUMNS command to display the schema of the student table:

DESCRIBE student;

This will show the updated definition of the sname column with the new length of 40 characters.

{28} **Create trigger to insert data in emp_logtable whenever any updateof sal in emp table. You can set action as 'New Salary'.**

CREATE TRIGGER trg_emp_sal_update

AFTER UPDATE OF sal ON emp

FOR EACH ROW

BEGIN

```
    INSERT INTO emp_logtable (emp_id, new_salary, action)
    VALUES (NEW.emp_id, NEW.sal, 'New Salary');
```