GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY DWARKA,

NEW DELHI-110078

UNIVERSITY SCHOOL OF MANAGEMENT STUDIES

MBA ANALYTICS

Year 2023-2025

DATABASE MANAGEMENT SYSTEM

(MBA (A) 209)

PRACTICAL FILE

SUBMITTED BY:
NAME: Nashreen Shaikh
ENROLLMENT NO: 02516619823

SUBMITTED TO:
Prof. Sanjay Dhingra

INTRODUCTION

A Database Management System (DBMS) is a software application that interacts with end-users, other applications, and the database itself to capture and analyze data. It provides a structured way to store, manage, and retrieve data efficiently.

Key Features of a DBMS:

- Data Definition Language (DDL): Defines the structure of a database, including tables, columns, data types, and relationships between them.
- Data Manipulation Language (DML): Allows users to insert, update, delete, and retrieve data from the database.
- Data Query Language (DQL): Enables users to query the database to extract specific information.
- Data Control Language (DCL): Grants and revokes access privileges to different users.

Advantages of Using a DBMS:

- Data Integrity: Ensures data accuracy and consistency through validation rules and constraints.
- Data Security: Protects sensitive data from unauthorized access through user authentication and authorization mechanisms.
- Data Independence: Separates data from the applications that use it, allowing for flexible changes without affecting the other.
- Efficient Data Access: Provides optimized techniques for storing and retrieving data, improving performance.
- Data Sharing: Enables multiple users to access and modify the same data concurrently.
- Backup and Recovery: Offers mechanisms to create backups and restore data in case of failures or disasters.

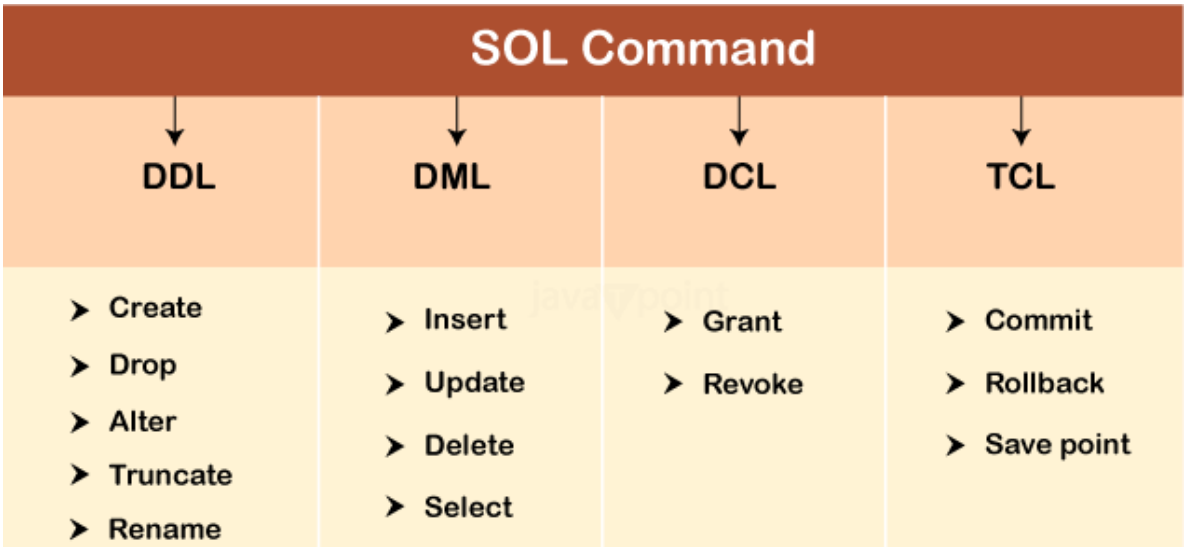
Disadvantages of Using a DBMS:

- Complexity: Requires specialized knowledge and skills to design, implement, and manage databases.
- Cost: Can be expensive, especially for large-scale databases, due to software licensing, hardware infrastructure, and maintenance costs.
- Performance Overhead: Can introduce overhead in terms of processing queries and updating data, especially for complex operations.

SQL (Structured Query Language)

SQL is a powerful language used to interact with relational databases. It allows you to:

- Define Data: Create, modify, and delete database structures like tables, columns, and indexes.
- Manipulate Data: Insert, update, delete, and retrieve data from tables.
- Query Data: Retrieve specific information from the database using various query techniques.
- Control Data Access: Grant and revoke permissions to different users.



SQL QUERIES:

Table Creation: DDL Commands

```
CREATE TABLE dept(
    deptno INT,
    dname VARCHAR(14),
    loc VARCHAR(13),
       constraint pk_dept primary key (deptno)
);
```

```sql
CREATE TABLE emp(
    empno INT,
    ename VARCHAR(10),
    job VARCHAR(9),
    mgr INT,
    hiredate DATE,
    sal DECIMAL(7,2),
    comm DECIMAL(7,2),
    deptno INT,
   constraint pk_emp primary key (empno),
   constraint fk_deptno foreign key (deptno) references dept (deptno)
);
```

Data Insertion: DML Commands

```sql
INSERT INTO dept(deptno, dname, loc) VALUES
(10, 'ACCOUNTING', 'NEW YORK'),
(20, 'RESEARCH', 'DALLAS'),
(30, 'SALES', 'CHICAGO'),
(40, 'OPERATIONS', 'BOSTON');
```

```sql
INSERT INTO emp VALUES
(7839, 'KANE', 'PRESIDENT', null, '1981-11-17', 5000, null, 10),
(7698, 'STEFAN', 'MANAGER', 7839,'1981-05-01', 2850, null, 30),
(7782, 'CLARK', 'MANAGER', 7839, '1981-06-09', 2450, null, 10),
(7566, 'JONES', 'MANAGER', 7839, '1981-04-02',2975, null, 20),
(7788, 'SCOTT', 'ANALYST', 7566, '1981-04-19',3000, null, 20),
(7902, 'DAMON', 'ANALYST', 7566, '1981-03-12', 3000, null, 20),
(7369, 'SMITH', 'CLERK', 7902, '1980-12-17', 800, null, 20),
(7499, 'ALLEN', 'SALESMAN', 7698, '1981-02-20', 1600, 300, 30),
(7521, 'SCOTT', 'SALESMAN', 7698, '1981-02-22',1250, 500, 30),
(7654, 'MARTIN', 'SALESMAN', 7698, '1981-09-28',1250, 1400, 30),
(7844, 'TRAVER', 'SALESMAN', 7698, '1981-09-08',1500, 0, 30),
(7876, 'ADAMS', 'CLERK', 7788, '1987-05-23',1100, null, 20),
(7900, 'JAMES', 'CLERK', 7698, '1981-03-12',950, null, 30),
(7934, 'MILLER', 'CLERK', 7782, '1982-01-23',1300, null, 10)
```

EMP TABLE

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	SCOTT	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7698	STEFAN	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7788	SCOTT	ANALYST	7566	1981-04-19	3000.00	NULL	20
7839	KANE	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7844	TRAVER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-03-12	950.00	NULL	30
7902	DAMON	ANALYST	7566	1981-03-12	3000.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ques 1. Display all the employees who are managers and clerks:

```
41  SELECT * FROM emp WHERE job IN ('MANAGER', 'CLERK');
42
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7698	STEFAN	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-03-12	950.00	NULL	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ques 2. Display the details of the employees who work in dept 20 as well as those in dept 30:

```
SELECT * FROM emp
WHERE deptno IN (20, 30);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	SCOTT	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7698	STEFAN	MANAGER	7839	1981-05-01	2850.00	NULL	30
7788	SCOTT	ANALYST	7566	1981-04-19	3000.00	NULL	20

Ques 3. Display the employees hired in year 1981

```
SELECT * FROM emp
WHERE YEAR(hiredate) = 1981;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	SCOTT	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
7698	STEFAN	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
7788	SCOTT	ANALYST	7566	1981-04-19	3000.00	NULL	20

Ques 4. Display the name of the employee having minimum salary

```
SELECT ename FROM emp
WHERE sal = (SELECT MIN(sal) FROM emp);
```

Result Grid | Filter Rows: | Export:

ename
SMITH

Ques 5. Display names of employees whose name starts with ‘D’ and ends with ‘N’

```
SELECT ename FROM emp
WHERE ename LIKE 'D%N';
```

Result Grid | Filter Rows:

ename
DAMON

Ques 6. Display the details of all the employees who are hired in the year 1981 and 1982

```
SELECT * FROM emp
WHERE YEAR(hiredate) IN (1981, 1982);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7788	SCOTT	ANALYST	7566	1981-04-19	3000.00	NULL	20
7839	KANE	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7844	TRAVER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
7900	JAMES	CLERK	7698	1981-03-12	950.00	NULL	30
7902	DAMON	ANALYST	7566	1981-03-12	3000.00	NULL	20
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10

Ques 7. Display all the employees who are clerks and have a salary less than 1500

```
SELECT * FROM emp
WHERE job = 'CLERK' AND sal < 1500;
```

Result Grid | Filter Rows: | Edit: | Export/Imp

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20
7900	JAMES	CLERK	7698	1981-03-12	950.00	NULL	30
7934	MILLER	CLERK	7782	1982-01-23	1300.00	NULL	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ques 8. Display those employees who are managers and salesman having salary more than 1500

```
SELECT * FROM emp
WHERE job IN ('MANAGER', 'SALESMAN') AND sal > 1500;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7698	STEFAN	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	CLARK	MANAGER	7839	1981-06-09	2450.00	NULL	10
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ques 9. Display the name of the employee who is having 2nd highest salary

```
SELECT ename FROM emp
WHERE sal = (SELECT MAX(sal) FROM emp WHERE sal < (SELECT MAX(sal) FROM emp));
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename
SCOTT
DAMON

Ques 10. Display details of the employees whose name starts with 'M' and ends with 'N'

```
SELECT * FROM emp
WHERE ename LIKE 'M%N';
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ques 11. Display the details of the employees who don't work in the department 10 and 30

```
SELECT * FROM emp
WHERE deptno NOT IN (10, 30);
```

Result Grid | Filter Rows: | Edit: | Export/Import:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7788	SCOTT	ANALYST	7566	1981-04-19	3000.00	NULL	20
7876	ADAMS	CLERK	7788	1987-05-23	1100.00	NULL	20
7902	DAMON	ANALYST	7566	1981-03-12	3000.00	NULL	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Ques 12. Display average salary of each jobtype and display those jobtype whose average salary is less then 2000

```
SELECT job, AVG(sal) AS avg_salary
FROM emp
GROUP BY job
HAVING AVG(sal) < 2000;
```

Result Grid | Filter Rows: | Export: | W

job	avg_salary
CLERK	1037.500000
SALESMAN	1400.000000

Ques 13. Display the total salary pf all employees who are working in department 10 and 20

```
SELECT SUM(sal) AS total_salary
FROM emp
WHERE deptno IN (10, 20);
```

Result Grid | Filter Rows: | Export: | Wr

total_salary
19625.00

Ques 14. Find the no. of employees working in each department no.

```
SELECT deptno, COUNT(*) AS num_employees
```

Result Grid | Filter Rows: | Export: | Wrap C

deptno	num_employees
10	3
20	5
30	6

Ques 15. Find the average salary of employees working in department 10 and 20

```
SELECT AVG(sal) AS avg_salary
FROM emp
WHERE deptno IN (10, 20);
```

Result Grid | Filter Rows: | Export:

avg_salary
2453.125000

Ques 16. Find the no. of jobtypes working in each deptno

```
SELECT deptno, job
FROM emp
GROUP BY deptno, job;
```

Result Grid | Filter Rows: | Expo

deptno	job
20	CLERK
30	SALESMAN
20	MANAGER
30	MANAGER
10	MANAGER
20	ANALYST
10	PRESIDENT
30	CLERK
10	CLERK

Ques 17. Find the total salary of each jobtype and display those jobtype whose total salary is greater than 7000

```
SELECT job, SUM(sal) AS total_salary
FROM emp
GROUP BY job
HAVING SUM(sal) > 7000;
```

Result Grid | Filter Rows: | Export: | W

job	total_salary
MANAGER	8275.00

Ques 18. Find the no. of days of employment for the employees 7499 and 7566

```
SELECT ename, empno, DATEDIFF(CURDATE(), hiredate) AS days_of_employment
FROM emp
WHERE empno IN (7499, 7566);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	empno	days_of_employment
ALLEN	7499	15968
JONES	7566	15927

Ques 19. Find 90 days probation date add of joining date of employees

```
SELECT ename, hiredate, DATE_ADD(hiredate, INTERVAL 90 DAY) AS probation_end_date
FROM emp;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	hiredate	probation_end_date
JONES	1981-04-02	1981-07-01
MARTIN	1981-09-28	1981-12-27
STEFAN	1981-05-01	1981-07-30
CLARK	1981-06-09	1981-09-07
SCOTT	1981-04-19	1981-07-18
KANE	1981-11-17	1982-02-15
TRAVER	1981-09-08	1981-12-07
ADAMS	1987-05-23	1987-08-21
JAMES	1981-03-12	1981-06-10
DAMON	1981-03-12	1981-06-10
MILLER	1982-01-23	1982-04-23

Ques 20. Find 6 months of probation date after the joining date of employees who are hired in 1981

```
SELECT ename, hiredate, DATE_ADD(hiredate, INTERVAL 6 MONTH) AS probation_end_date
FROM emp
WHERE YEAR(hiredate) = 1981;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	hiredate	probation_end_date
ALLEN	1981-02-20	1981-08-20
SCOTT	1981-02-22	1981-08-22
JONES	1981-04-02	1981-10-02
MARTIN	1981-09-28	1982-03-28
STEFAN	1981-05-01	1981-11-01
CLARK	1981-06-09	1981-12-09
SCOTT	1981-04-19	1981-10-19
KANE	1981-11-17	1982-05-17
TRAVER	1981-09-08	1982-03-08
JAMES	1981-03-12	1981-09-12
DAMON	1981-03-12	1981-09-12

Ques 21. Display the hiredate in the format '14th of March, 1981' for the employee who is having the highest salary

```
SELECT ename, DATE_FORMAT(hiredate, '%D of %M, %Y') AS formatted_hiredate
FROM emp
WHERE sal = (SELECT MAX(sal) FROM emp);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	formatted_hiredate
KANE	17th of November, 1981

Ques 22. Display the date in the format '1st April of 1981, 05:35:15' for all the employees.

```
SELECT ename, DATE_FORMAT(hiredate, '%D %M of %Y, %H:%i:%s') AS formatted_hiredate
FROM emp;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	formatted_hiredate
SMITH	17th December of 1980, 00:00:00
ALLEN	20th February of 1981, 00:00:00
SCOTT	22nd February of 1981, 00:00:00
JONES	2nd April of 1981, 00:00:00
MARTIN	28th September of 1981, 00:00:00
STEFAN	1st May of 1981, 00:00:00
CLARK	9th June of 1981, 00:00:00
SCOTT	19th April of 1981, 00:00:00
KANE	17th November of 1981, 00:00:00
TRAVER	8th September of 1981, 00:00:00
ADAMS	23rd May of 1987, 00:00:00

Ques 23. Decrease the salary by 50% for employee having the highest salary

```
SELECT ename, sal, sal * 0.5 AS 'Salary 50%'
FROM emp
WHERE sal = (SELECT MAX(sal) FROM emp);
```

Result Grid | Filter Rows: | Export: | Wrap

ename	sal	Salary 50%
KANE	5000.00	2500.000

Ques 24. Give 10% salary hike to all employees who were hired in the year 1981

```
118 •  select ename, sal, sal*0.1+sal as "Salary With 10% Hike" from emp;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	sal	Salary With 10% Hike
SMITH	800.00	880.000
ALLEN	1600.00	1760.000
SCOTT	1250.00	1375.000
JONES	2975.00	3272.500
MARTIN	1250.00	1375.000
STEFAN	2850.00	3135.000
CLARK	2450.00	2695.000
SCOTT	3000.00	3300.000
KANE	5000.00	5500.000
TRAVER	1500.00	1650.000
ADAMS	1100.00	1210.000
JAMES	950.00	1045.000
DAMON	3000.00	3300.000

EMP179 TABLE

Ques 25. Create a duplicate table of emp with name as emp179

```
128 •  CREATE TABLE emp179 AS
129    SELECT * FROM emp;
130 •  select* from emp179;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	SMITH	CLERK	7902	1980-12-17	800.00	NULL	20
7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
7521	SCOTT	SALESMAN	7698	1981-02-22	1250.00	500.00	30
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30

Ques 26. Add a constraint CHK on salary column to ensure that all salary entered is more than 700

```
135 •  ALTER TABLE emp179
136    ADD CONSTRAINT chk_salary CHECK (sal > 700);
```

Ques 27. Disable the constraint CHK

```
ALTER TABLE emp179
DISABLE CONSTRAINT chk_salary;
```

Ques 28. Drop the constraint CHK from emp1 table

```
ALTER TABLE emp179
DROP CONSTRAINT chk_salary;
```

Ques 29. Drop the column address from emp1 table

```
ALTER TABLE emp179
DROP COLUMN address;
```

BORROWER AND DEPOSITOR COMMAND

```
SELECT* FROM DEPOSITOR;
```

Result Grid Filter Rows:

CUST_ID	ACC_NO
1	101
2	301
5	501
6	701

```
SELECT* FROM BORROWER;
```

Result Grid Filter Rows:

CUST_ID	LOAN_NO
1	101
4	301
8	501
6	701

Ques 30. Display customers having account in bank taken loan or both

```
SELECT CUST_ID, ACC_NO, NULL AS LOAN_NO
FROM DEPOSITOR
UNION
SELECT CUST_ID, NULL AS ACC_NO, LOAN_NO
FROM BORROWER;
```

Result Grid | Filter Rows: | Export:

CUST_ID	ACC_NO	LOAN_NO
1	101	NULL
2	301	NULL
5	501	NULL
6	701	NULL
1	NULL	101
4	NULL	301
8	NULL	501
6	NULL	701

Ques 31. Display customers having account as well as taken loan

```
SELECT D.CUST_ID, D.ACC_NO, B.LOAN_NO
FROM DEPOSITOR D
INNER JOIN BORROWER B
ON D.CUST_ID = B.CUST_ID;
```

Result Grid | Filter Rows: | Export:

CUST_ID	ACC_NO	LOAN_NO
1	101	101
6	701	701

Ques 32. Display the customer having only account in bank and no loan

```
SELECT D.CUST_ID, D.ACC_NO
FROM DEPOSITOR D
LEFT JOIN BORROWER B
ON D.CUST_ID = B.CUST_ID
WHERE B.CUST_ID IS NULL;
```

Result Grid | Filter Rows: | Export:

CUST_ID	ACC_NO
2	301
5	501

Ques 33. Display the customer having only loan no. account

```
SELECT B.CUST_ID, B.LOAN_NO
FROM BORROWER B
LEFT JOIN DEPOSITOR D
ON B.CUST_ID = D.CUST_ID
WHERE D.CUST_ID IS NULL;
```

Result Grid | Filter Rows: | Exp

CUST_ID	LOAN_NO
4	301
8	501

JOIN FUNCTION

Ques 34. Display empno, ename, job, dname, location for all the employee JOHNS AND SMITH

```
SELECT emp.empno, emp.ename, emp.job, dept.dname, dept.loc
FROM emp
JOIN dept ON emp.deptno = dept.deptno
WHERE emp.ename IN ('JOHNS', 'SMITH');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

empno	ename	job	dname	loc
7369	SMITH	CLERK	RESEARCH	DALLAS

Ques 35. Display ename, salary, location of the people who hired first

```
SELECT emp.ename, emp.sal, dept.loc
FROM emp
JOIN dept ON emp.deptno = dept.deptno
WHERE emp.hiredate = (SELECT MIN(hiredate) FROM emp);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ename	sal	loc
SMITH	800.00	DALLAS

Ques 36. Display empno, job, salary, department name for all the employees working in department 10 and 20

```
SELECT emp.empno, emp.job, emp.sal, dept.dname
FROM emp
JOIN dept ON emp.deptno = dept.deptno
WHERE emp.deptno IN (10, 20);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

empno	job	sal	dname
7782	MANAGER	2450.00	ACCOUNTING
7839	PRESIDENT	5000.00	ACCOUNTING
7934	CLERK	1300.00	ACCOUNTING
7369	CLERK	800.00	RESEARCH
7566	MANAGER	2975.00	RESEARCH
7788	ANALYST	3000.00	RESEARCH
7876	CLERK	1100.00	RESEARCH
7902	ANALYST	3000.00	RESEARCH

Ques 37. Display empno, ename, job, deptno, dname, location for all the employees who were hired in year 1981 and 1982.

```
SELECT emp.empno, emp.ename, emp.job, emp.deptno, dept.dname, dept.loc
FROM emp
JOIN dept ON emp.deptno = dept.deptno
WHERE YEAR(emp.hiredate) IN (1981, 1982);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

empno	ename	job	deptno	dname	loc
7934	MILLER	CLERK	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	20	RESEARCH	DALLAS
7902	DAMON	ANALYST	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	30	SALES	CHICAGO
7521	SCOTT	SALESMAN	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	30	SALES	CHICAGO
7698	STEFAN	MANAGER	30	SALES	CHICAGO
7844	TRAVER	SALESMAN	30	SALES	CHICAGO
7900	JAMES	CLERK	30	SALES	CHICAGO

Decode SQL Queries

Ques 38. Change the dept no in the emp table to dept name values.

Worksheet | Query Builder

```
SELECT ENAME, DECODE(DEPTNO,10,'ACCOUNTING',20,
'RESEARCH',30,'SALES'
,40,'OPERATIONS',DEPTNO) AS DNAME FROM EMP;
```

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.002 seconds

	ENAME	DNAME
1	KING	ACCOUNTING
2	BLAKE	SALES
3	CLARK	ACCOUNTING
4	JONES	RESEARCH
5	SCOTT	RESEARCH
6	FORD	RESEARCH
7	SMITH	RESEARCH
8	ALLEN	SALES
9	WARD	SALES
10	MARTIN	SALES
11	TURNER	SALES
12	ADAMS	RESEARCH
13	JAMES	SALES
14	MILLER	ACCOUNTING

Ques 39. Change the salaries of the employees as per the following

- If sal=1200 then increase by 5%
- If sal=1300 then increase by 10%
- If sal=1400 then increase by 15%
- If sal=1500 then increase by 20%
- If sal=1600 then increase by 25%
- If sal=1700 then increase by 30%
- If sal=1800 then increase by 35%
- If sal=1900 then increase by 40%
- If sal=2000 then increase by 45%
- Give 20% increase to unmatched

Worksheet | Query Builder

```
UPDATE EMP
SET SAL = SAL * DECODE(SAL,
                       1200, 1.05,  -- Increase by 5%
                       1300, 1.10,  -- Increase by 10%
                       1400, 1.15,  -- Increase by 15%
                       1500, 1.20,  -- Increase by 20%
                       1600, 1.25,  -- Increase by 25%
                       1700, 1.30,  -- Increase by 30%
                       1800, 1.35,  -- Increase by 35%
                       1900, 1.40,  -- Increase by 40%
                       2000, 1.45,  -- Increase by 45%
                       1.20);       -- Increase by 20% for unmatched
```

Script Output | Query Result

Task completed in 0.026 seconds

```
14 rows updated.
```

Translate SQL Queries

Ques 40. In the emp table change the names of all the employees so that every singlecharacter in the should be replay with next character. (Translate)

Worksheet | Query Builder

```
SELECT ename,TRANSLATE(ename,
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz',
'BCDEFGHIJKLMNOPQRSTUVWXYZAbcdefghijklmnopqrstuvwxyza')
AS transformed_ename
FROM emp;
```

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.005 seconds

	ENAME	TRANSFORMED_ENAME
1	KING	LJOH
2	BLAKE	CMBLF
3	CLARK	DMBSL
4	JONES	KPOFT
5	SCOTT	TDPUU
6	FORD	GPSE
7	SMITH	TNJUI
8	ALLEN	BMMFO
9	WARD	XBSE
10	MARTIN	NBSUJO
11	TURNER	UVSOFS
12	ADAMS	BEBNT
13	JAMES	KBNFT
14	MILLER	NJMMFS

String Functions SQL Queries

Ques 41. Display all the names of the employees whose length of name is more than 5.

Worksheet | Query Builder

```
SELECT ENAME
FROM EMP
WHERE LENGTH(ENAME) > 5;
```

Script Output | Query Result

SQL | All Rows Fetched: 3 in

	ENAME
1	MARTIN
2	TURNER
3	MILLER

Q49. Find the 2nd occurrence of 'A' in the job column of emp.

Worksheet | Query Builder

```
SELECT job,
       INSTR(job, 'A', 1, 2) AS second_occurrence_of_A
FROM emp
WHERE INSTR(job, 'A', 1, 2) > 0;
```

Script Output | Query Result

SQL | All Rows Fetched: 9 in 0.003 seconds

	JOB	SECOND_OCCURRENCE_OF_A
1	MANAGER	4
2	MANAGER	4
3	MANAGER	4
4	ANALYST	3
5	ANALYST	3
6	SALESMAN	7
7	SALESMAN	7
8	SALESMAN	7
9	SALESMAN	7

Ques 42. Display the name of the employees padded with * on the right-hand side with totallength of the name is 25.

Worksheet | Query Builder

```
SELECT ename,
RPAD(ename, 25, '*') AS padded_name
FROM emp;
```

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.003 seconds

	ENAME	PADDED_NAME
1	KING	KING**********
2	BLAKE	BLAKE**********
3	CLARK	CLARK**********
4	JONES	JONES**********
5	SCOTT	SCOTT**********
6	FORD	FORD**********
7	SMITH	SMITH**********
8	ALLEN	ALLEN**********
9	WARD	WARD**********
10	MARTIN	MARTIN**********
11	TURNER	TURNER**********
12	ADAMS	ADAMS**********
13	JAMES	JAMES**********
14	MILLER	MILLER**********

Ques 43. Trim the 1st character from all the name of the employees

Worksheet | Query Builder

```
SELECT ename,
       SUBSTR(ename, 2) AS trimmed_name
FROM emp;
```

Script Output | Query Result

SQL | All Rows Fetched: 14 in 0.002 seconds

	ENAME	TRIMMED_NAME
1	KING	ING
2	BLAKE	LAKE
3	CLARK	LARK
4	JONES	ONES
5	SCOTT	COTT
6	FORD	ORD
7	SMITH	MITH
8	ALLEN	LLEN
9	WARD	ARD
10	MARTIN	ARTIN
11	TURNER	URNER
12	ADAMS	DAMS
13	JAMES	AMES
14	MILLER	ILLER

PL/SQL

Ques 44. Write a PL/SQL Code to calculate area of a circle.

Worksheet | Query Builder

```
set serveroutput on;

create table a (
area number(14,2),
radius number(2));

DECLARE
pi constant number(14,2):=3.14;
radius number(2);
area number(14,2);

BEGIN
radius:=3;
area:=pi*power(radius,2);
INSERT INTO a VALUES (area,radius);
end;
```

Script Output | Task completed in 0.036 seconds

```
Table A created.


PL/SQL procedure successfully completed.
```

Ques 45. Increase the salary of the emp 7369 by 10% if his salary is greater or equal to 2000otherwise increase his salary by 5%.

Worksheet | Query Builder

```
set serveroutput on;

DECLARE
a emp.sal%type;

BEGIN
select sal into a from emp where empno=7369;
if a>2000 then
update emp set sal=sal*1.1 where empno=7369;
else
update emp set sal=sal*1.05 where empno=7369;
end if;
dbms_output.put_line('Salary of 7369 is updated');
end;
```

Script Output

Task completed in 0.031 seconds

```
Salary of 7369 is updated


PL/SQL procedure successfully completed.
```

Ques 46. Write PL/SQL Loop to calculate area of a circle

Worksheet | Query Builder

```
set serveroutput on;

DECLARE
pi constant number(9,7):=3.14;
radius number(5);
area number(5);

BEGIN
radius :=3;
loop
area:=pi*power(radius,2);
dbms_output.put_line('Area is '||area);
radius:=radius+1;
exit when area>100;
end loop;
end;
```

Script Output | Task completed in 0.027 seconds

```
Area is 28
Area is 50
Area is 79
Area is 113


PL/SQL procedure successfully completed.
```

Ques 47. Write PL/SQL using for loop to calculate area of a circle

Worksheet | Query Builder

```
set serveroutput on;

DECLARE
pi constant number(9,7):=3.14;
radius number(5);
area number(5);

BEGIN
for radius in 1..7
loop
area:=pi*power(radius,2);
dbms_output.put_line('Area is '||area);
end loop;
end;
```

Script Output

Task completed in 0.035 seconds

```
Area is 28
Area is 50
Area is 79
Area is 113
Area is 154


PL/SQL procedure successfully completed.
```

Ques 48. Write PL/SQL code to calculate area of circle using cursor.

Worksheet | Query Builder

```
set serveroutput on;

DECLARE
pi constant number(9,7):=3.14;
cursor rad_cursor is
select * from radius_table;
area number(5);
rv rad_cursor%rowtype;

BEGIN
open rad_cursor;
loop
fetch rad_cursor into rv;
exit when rad_cursor%notfound;
area:=pi*power(rv.radius,2);
dbms_output.put_line('Area is '||area);
end loop;
close rad_cursor;
end;
```

Script Output | Task completed in 0.036 seconds

```
Area is 28
Area is 50
Area is 79
Area is 113
Area is 154


PL/SQL procedure successfully completed.
```

Ques 49. Calculate the perimeter of a rectangle, values of length and breadth are given inthe table Para(L,B).

Worksheet | Query Builder

```
set serveroutput on;

DECLARE
cursor par_cursor is
select * from para;
perimeter number(5);
rv par_cursor%rowtype;

BEGIN
for rv in par_cursor
loop
perimeter:= 2*(rv.L+rv.B);
dbms_output.put_line('Perimeter is '|| perimeter);
end loop;
end;
```

Script Output | Task completed in 0.034 seconds

```
Perimeter is 14
Perimeter is 18
Perimeter is 22
Perimeter is 26


PL/SQL procedure successfully completed.
```

Ques 50. In the emp table monitor the salary of the emp through trigger, if change is morethan 10% in the salary than that record of the employee should be shifted to emp9 table, which is having same structure as emp table.

Worksheet | Query Builder

```
create table emp9 as select * from emp where 1=2;

set serveroutput on;

create or replace trigger salm
before update on emp
for each row

BEGIN
if (:new.sal/:old.sal)>1.1 then
INSERT INTO emp9 VALUES (:old.empno, :old.ename, :old.job,
:old.mgr, :old.hiredate, :old.sal, :old.comm, :old.deptno);
dbms_output.put_line('The change in salary of more than 10% is found');
end if;
end;
```

Script Output | Task completed in 0.154 seconds

```
Table EMP9 created.


Trigger SALM compiled
```

Ques 51. Create a trigger to update inventory after sales is done. Also create a trigger tocheck if inventory is lower than economic order quantity.

Worksheet | Query Builder

```
set SERVEROUTPUT on;

create or replace trigger adjust
after insert on sales
for each row

BEGIN
update inventory set qty=qty-:new.sqty where pid=:new.spid;
dbms_output.put_line('inventory has been reduced');
end;
```

Script Output

Task completed in 0.03 seconds

```
Trigger ADJUST compiled
```

```
set SERVEROUTPUT on;

create or replace trigger chk_eoq
after update on inventory
for each row

BEGIN
if :new.eoq>=:new.qty then
dbms_output.put_line('order more qty for the product '|| :new.pid);
end if;
end;
```

Script Output

Task completed in 0.043 seconds

```
Trigger CHK_EOQ compiled
```

Ques 52. Create a procedure to increase salary of the employee by the given amount.

Worksheet | Query Builder

```
set SERVEROUTPUT on;

create or replace procedure increase(eid number, amt number)
is
vsal number;
sal_missing exception;

BEGIN
select sal into vsal from emp where empno=eid;
if vsal is null then
raise sal_missing;
else
update emp set sal=sal+amt where empno=eid;
end if;
exception
when no_data_found then
dbms_output.put_line (eid||' no such number');
when sal_missing then
dbms_output.put_line (eid||' has null salary');
end;
```

Script Output | Task completed in 0.05 seconds

```
Procedure INCREASE compiled
```

Ques 53. Create a function which will accept empno as a parameter and the salary of the empno will be increased as per the following.

- If salary is less than 1500 then increment should be 5% of the salary.
- If salary is between 1500-2000 then increment should be 10% of the salary, otherwise increment should be 15% of the salary.

Worksheet | Query Builder

```
set SERVEROUTPUT on;

create or replace function hike(eid number)
return number is
inc emp.sal%type;
vsal emp.sal%type;

BEGIN
select sal into vsal from emp where empno=eid;
if vsal<=1500 then
inc:=vsal*0.05;
elsif vsal>1500 and vsal <=2000 then
inc:=0.1*vsal;
else
inc:=0.15*vsal;
end if;
return(inc);
end hike;
```

Script Output | Task completed in 0.029 seconds

```
Function HIKE compiled
```

```
DECLARE
sal_inc number(7,2);

BEGIN
sal_inc:= hike(7369);
dbms_output.put_line('Increment is '||sal_inc);
end;
```

Script Output | Task completed in 0.024 seconds

```
Increment is 52.92


PL/SQL procedure successfully completed.
```