DBMS LAB RECORD

121810311056
JUSTIN CLARKE

Introduction to DBMS:

Database: A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

Structured Query Language(SQL): SQL is a programming language used by nearly all relational databases to query, manipulate, and define data, and to provide access control. SQL was first developed at IBM in the 1970s with Oracle as a major contributor, which led to implementation of the SQL ANSI standard, SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft. Although SQL is still widely used today, new programming languages are beginning to appear.

Database Management System: A database typically requires a comprehensive database software program known as a database management system (DBMS). A DBMS serves as an interface between

the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized. A DBMS also facilitates oversight and control of databases, enabling a variety of administrative operations such as performance monitoring, tuning, and backup and recovery.

Some examples of popular database software or DBMSs include MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database, and dBase.

Data Definition Language (DDL) :

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

Examples of DDL commands:

- **CREATE** – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
- **DROP** – is used to delete objects from the database.
- **ALTER**-is used to alter the structure of the database.
- **TRUNCATE**–is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT** –is used to add comments to the data dictionary.
- **RENAME** –is used to rename an object existing in the database.

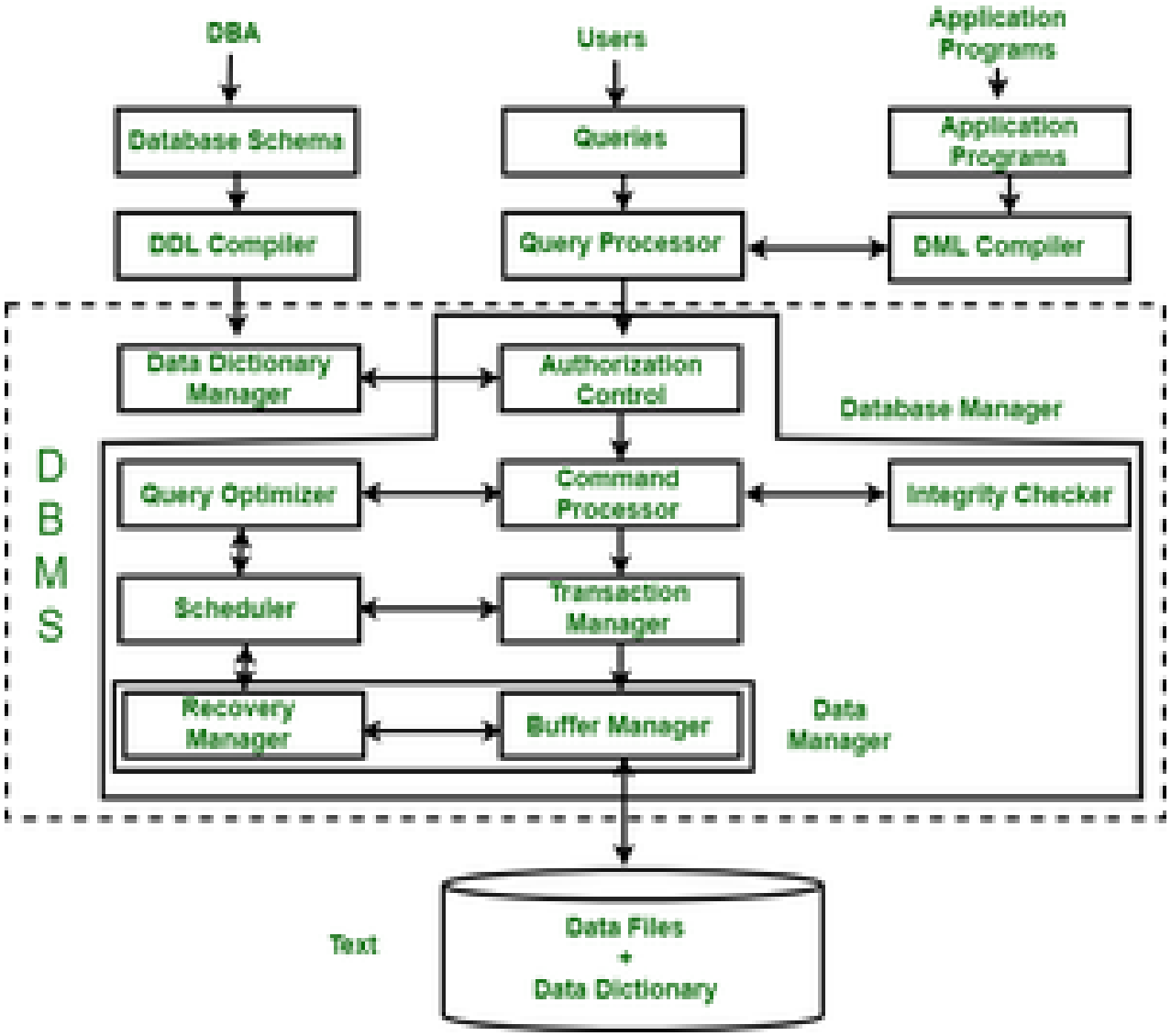
Data Manipulation Language (DML):

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

Examples of DML:

- **INSERT** – is used to insert data into a table.
- **UPDATE** – is used to update existing data within a table.
- **DELETE** – is used to delete records from a database table.

Structure Of DBMS:



Types of DBMS:

Four Types of DBMS systems are:

- Hierarchical database
- Network database
- Relational database
- Object-Oriented database

Hierarchical DBMS:

In a Hierarchical database, model data is organized in a tree-like structure. Data is Stored Hierarchically (top down or bottom up) format. Data is represented using a parent-child relationship. In Hierarchical DBMS parent may have many children, but children have only one parent.

Network Model:

The network database model allows each child to have multiple parents. It helps you to address the need to model more complex relationships like as the orders/parts many-to-many relationship. In this model, entities are organized in a graph which can be accessed through several paths.

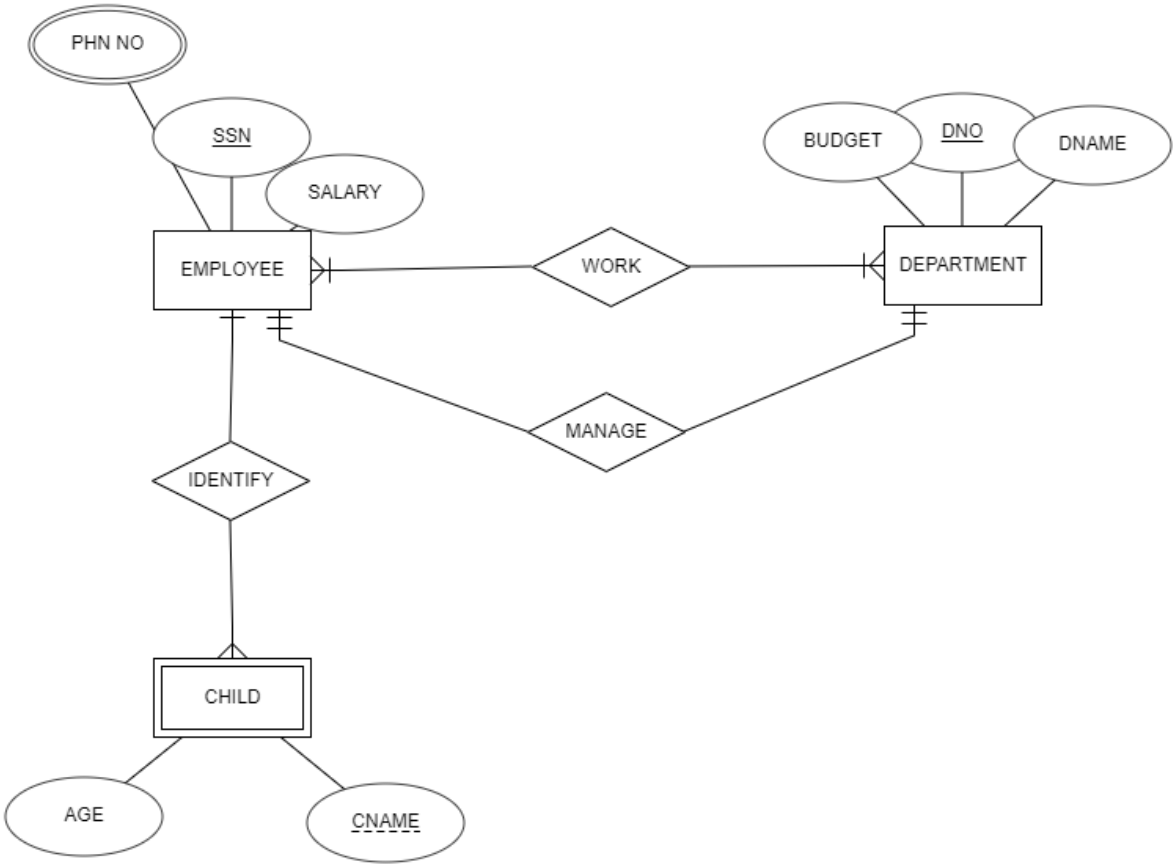
Relational model:

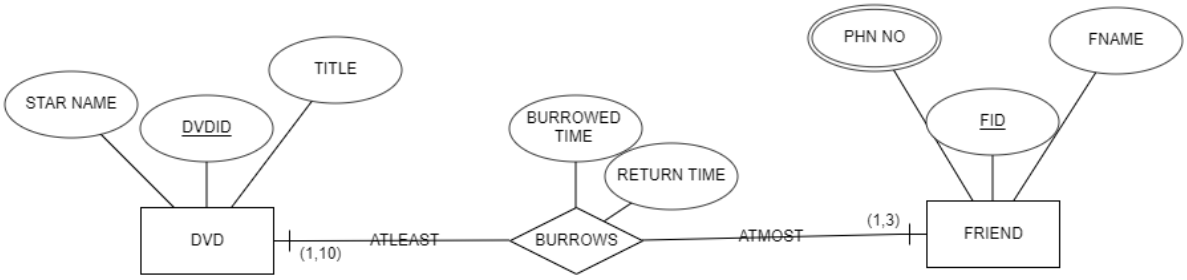
Relational DBMS is the most widely used DBMS model because it is one of the easiest. This model is based on normalizing data in the rows and columns of the tables. Relational model stored in fixed structures and manipulated using SQL.

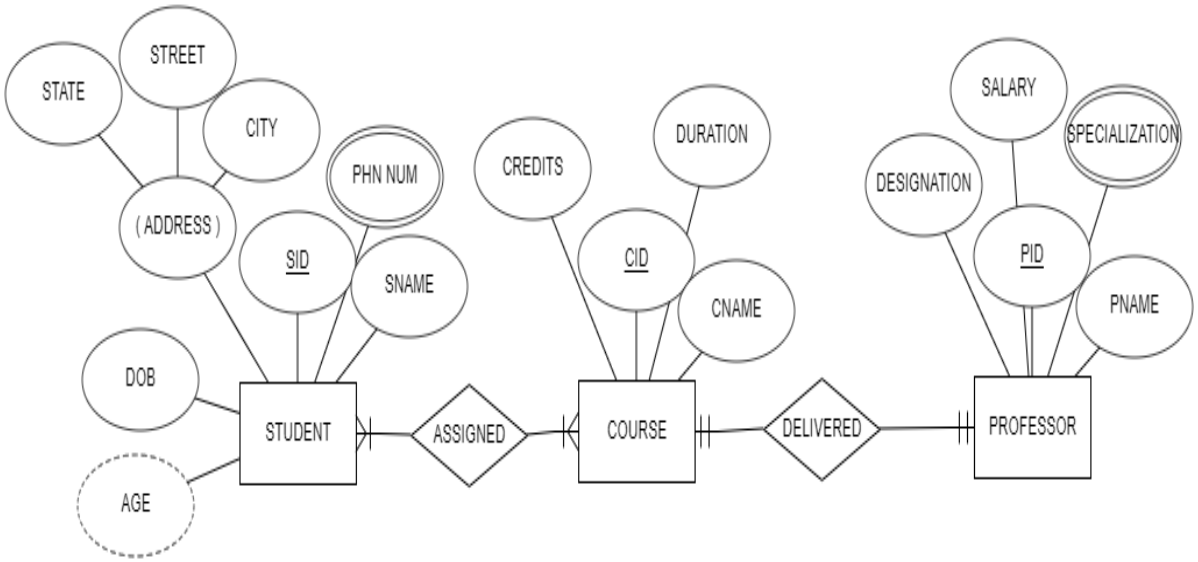
Object-Oriented Model:

In Object-oriented Model data stored in the form of objects. The structure which is called classes which display data within it. It defines a database as a collection of objects which stores both data members values and operations.

ER Diagrams:




PHN NO
FNAME
TITLE
STAR NAME
DVDID
FID
BURROWED TIME
RETURN TIME
DVD
(1,10)
ATLEAST
BURROWS
ATMOST
(1,3)
FRIEND


STREET
STATE
CITY
PHN NUM
(ADDRESS)
SID
SNAME
DOB
AGE
STUDENT
ASSIGNED
CREDITS
DURATION
CID
CNAME
COURSE
DELIVERED
SALARY
SPECIALIZATION
DESIGNATION
PID
PNAME
PROFESSOR

STRING FUNCTIONS:

```
SELECT * FROM DUAL
```

DUMMY
X

Download CSV

```
1  select CONCAT('JUST','IN') from dual
```

CONCAT('JUST','IN')
JUSTIN

Download CSV

```
1  select lpad('Justin',19,'%') from dual
```

LPAD('JUSTIN',19,'%')
%%%%%%%%%%%%%Justin

Download CSV

```
1  select rpad('Justin',19,'#$') from dual
```

RPAD('JUSTIN',19,'#$')
Justin#$#$#$#$#$#$#

Download CSV

```
1  select upper('Justin') from dual
```

UPPER('JUSTIN')
JUSTIN

Download CSV

```
1  select lower('Justin') from dual
```

LOWER('JUSTIN')
justin

Download CSV

```
1  select initcap('justin') from dual
```

INITCAP('JUSTIN')
Justin

Download CSV

```
1  select substr('justin',4,6) from dual
```

SUBSTR('JUSTIN',4,6)
tin

Download CSV

```
1  select instr('justin','t') from dual
```

INSTR('JUSTIN','T')
4

Download CSV

```
1  select replace('justin','in',' do it') from dual
```

REPLACE('JUSTIN','IN','DOIT')
just do it

Download CSV

```
1  select length('justin') from dual
```

LENGTH('JUSTIN')
6

Download CSV

CONVERTION FUNCTIONS:

```
select to_char(sysdate,'dd "th of" month yyyy') from dual
```

TO_CHAR(SYSDATE,'DD"THOF"MONTHYYYY')
11 th of december 2020

Download CSV

```
select to_char(to_date('11-dec-2020'),'dd"th of"month yyyy')from dual
```

TO_CHAR(TO_DATE('11-DEC-2020'),'DD"THOF"MONTHYYYY')
11th ofdecember 2020

Download CSV

NUMERIC FUNCTIONS:

```
select * from dual
```

DUMMY
X

Download CSV

```
select abs(-4)from dual
```

ABS(-4)
4

Download CSV

```
select sin(abs(-22.123456))from dual
```

SIN(ABS(-22.123456))
-.131921750065057917682673121674530876 13

Download CSV

```
select sqrt(5555)from dual
```

SQRT(5555)
74.531872376856332402192344829124325340 14

Download CSV

```
select power(2,20)from dual
```

POWER(2,20)
1048576

Download CSV

```
select exp(3)from dual
```

EXP(3)
20.085536923187667740928529654581717897 06

Download CSV

```
select ln(4)from dual
```

LN(4)
1.386294361119890618834464242916353136 14

Download CSV

```
select mod(2,6)from dual
```

MOD(2,6)
2

Download CSV

```
select ceil(7.5)from dual
```

```
select ceil(7.5)from dual
```

CEIL(7.5)
8

Download CSV

```
select floor(8.2)from dual
```

FLOOR(8.2)
8

Download CSV

```
select * from dual
```

DUMMY
X

Download CSV

DATE FUNCTIONS:

```
select sysdate from dual
```

SYSDATE
11-DEC-20

Download CSV

```
select next_day('17-dec-2020','tue date')from dual
```

NEXT_DAY('17-DEC-2020','TUEDATE')
22-DEC-20

Download CSV

```
select greatest('17-dec-2020','5-feb-2022')from dual
```

GREATEST('17-DEC-2020','5-FEB-2022')
5-feb-2022

Download CSV

```
select trunc((to_date('17-dec-2020')),'month')from dual
```

TRUNC((TO_DATE('17-DEC-2020')),'MONTH')
01-DEC-20

Download CSV

```
select round((to_date('17-dec-2020')),'month')from dual
```

ROUND((TO_DATE('17-DEC-2020')),'MONTH')
01-JAN-21

Download CSV

```
select last_day('17-dec-2020')from dual
```

LAST_DAY('17-DEC-2020')
31-DEC-20

Download CSV

```
select least('17-dec-2020','5-feb-2022')from dual
```

LEAST('17-DEC-2020','5-FEB-2022')
17-dec-2020

Download CSV

DDL and DML Commands:

```sql
CREATE TABLE  "STUDENT"(
    "ST_ID" INT NOT NULL PRIMARY KEY,
    "STUDENT_NAME" VARCHAR(20) NOT NULL,
    "AGE" INT,
    "CGPA" FLOAT NOT NULL,
    "PHNO" VARCHAR(20) NOT NULL
);

INSERT INTO STUDENT VALUES ('121810311056', 'Justin Clarke', '19', '8.1', '3627183');
INSERT INTO STUDENT VALUES ('121810311021', 'Sujan Buddha', '20', '9.7', '7593752');
INSERT INTO STUDENT VALUES ('121810311013', 'Sreeraj Kasa', '20', '10.0', '8549311');
INSERT INTO STUDENT VALUES ('121810311023', 'Tanmay Bandaru', '20', '9.5', '4329833');

SELECT * FROM STUDENT
```

ST_ID	STUDENT_NAME	AGE	CGPA	PHNO
121810311056	Justin Clarke	19	8.1	3627183
121810311021	Sujan Buddha	20	9.7	7593752
121810311013	Sreeraj Kasa	20	10	8549311
121810311023	Tanmay Bandaru	20	9.5	4329833

Download CSV

4 rows selected.

```sql
ALTER TABLE STUDENT DROP COLUMN PHNO

SELECT * FROM STUDENT
```

ST_ID	STUDENT_NAME	AGE	CGPA
121810311056	Justin Clarke	19	8.1
121810311021	Sujan Buddha	20	9.7
121810311013	Sreeraj Kasa	20	10
121810311023	Tanmay Bandaru	20	9.5

Download CSV

4 rows selected.

```sql
ALTER TABLE STUDENT ADD LOGIN VARCHAR(20)

SELECT * FROM STUDENT
```

ST_ID	STUDENT_NAME	AGE	CGPA	LOGIN
121810311056	Justin Clarke	19	8.1	-
121810311021	Sujan Buddha	20	9.7	-
121810311013	Sreeraj Kasa	20	10	-
121810311023	Tanmay Bandaru	20	9.5	-

Download CSV

4 rows selected.

```sql
DROP TABLE STUDENT
```

Table dropped.

```sql
Truncate table student
```

Table truncated.

AGGREGATE and LIKE Commands:

```
1  SELECT * FROM STUDENT
```

ST_ID	STUDENT_NAME	AGE	CGPA	PHNO
56	Justin Clarke	19	8.1	3627183
21	Sujan Buddha	20	9.7	7593752
13	Sreeraj Kasa	20	10	8549311
23	Tanmay Bandaru	20	9.5	4329833

Download CSV

4 rows selected.

```
1  CREATE TABLE "COURSES"(
2      "CID" VARCHAR(20) NOT NULL PRIMARY KEY,
3      "COURSE_NAME" VARCHAR(20) NOT NULL,
4      "CREDITS" INT
5  );
6  INSERT INTO COURSES VALUES ('EID 304', 'EEM', '3');
7  INSERT INTO COURSES VALUES ('EID 301', 'DBMS', '3');
8  INSERT INTO COURSES VALUES ('EID 302', 'FLAT', '3');
9  INSERT INTO COURSES VALUES ('EID 321', 'SE', '4');
10 INSERT INTO COURSES VALUES ('EID 314', 'DAA', '4');
11
12 SELECT * FROM COURSES
```

1 row(s) inserted.

CID	COURSE_NAME	CREDITS
EID 304	EEM	3
EID 301	DBMS	3
EID 302	FLAT	3
EID 321	SE	4
EID 314	DAA	4

Download CSV

5 rows selected.

```sql
CREATE TABLE "RESULTS"(
    "ST_ID" VARCHAR(20) NOT NULL,
    "CID" VARCHAR(20) NOT NULL PRIMARY KEY,
    "MARKS" INT
);
INSERT INTO RESULTS VALUES ('11056','EID 304', '75');
INSERT INTO RESULTS VALUES ('11021','EID 301', '80');
INSERT INTO RESULTS VALUES ('11013','EID 302', '90');
INSERT INTO RESULTS VALUES ('11023','EID 321', '85');
INSERT INTO RESULTS VALUES ('11036','EID 314', '70');

SELECT * FROM RESULTS
```

```
1 row(s) inserted.
```

ST_ID	CID	MARKS
11056	EID 304	75
11021	EID 301	80
11013	EID 302	90
11023	EID 321	85
11036	EID 314	70

Download CSV

5 rows selected.

```sql
SELECT MAX(MARKS) FROM RESULTS
```

MAX(MARKS)
90

Download CSV

```sql
SELECT MIN(MARKS) FROM RESULTS
```

MIN(MARKS)
70

Download CSV

```
1  SELECT SUM(MARKS) FROM RESULTS
```

SUM(MARKS)
400

Download CSV

```
1  SELECT AVG(MARKS) FROM RESULTS
```

AVG(MARKS)
80

Download CSV

```
1  SELECT COUNT(MARKS) FROM RESULTS
```

COUNT(MARKS)
5

Download CSV

```
1  SELECT ST_ID FROM STUDENT
2  WHERE STUDENT_NAME LIKE '_a%'
```

ST_ID
23

Download CSV

```
1  SELECT * FROM STUDENT
2  WHERE STUDENT_NAME LIKE '%u%'
```

ST_ID	STUDENT_NAME	AGE	CGPA	PHNO
56	Justin Clarke	19	8.1	3627183
21	Sujan Buddha	20	9.7	7593752
23	Tanmay Bandaru	20	9.5	4329833

Download CSV

3 rows selected.

VIEWS:

```
1  CREATE VIEW DetailedView AS
2  SELECT STUDENT_NAME, PHNO
3  FROM STUDENT
4  WHERE ST_ID >25
```

View created.

```
1  SELECT * FROM DetailedView
```

STUDENT_NAME	PHNO
Justin Clarke	3627183

Download CSV

```
1  CREATE VIEW StudentNames AS
2  SELECT ST_ID, STUDENT_NAME
3  FROM STUDENT
4  ORDER BY STUDENT_NAME
```

View created.

```
1  SELECT * FROM StudentNames
```

ST_ID	STUDENT_NAME
56	Justin Clarke
13	Sreeraj Kasa
21	Sujan Buddha
23	Tanmay Bandaru

Download CSV

4 rows selected.

```
CREATE VIEW MarksView AS
SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS
FROM StudentDetails, StudentMarks
WHERE StudentDetails.NAME = StudentMarks.NAME
```

View created.

```
1  SELECT * FROM MarksView
```

STUDENT_NAME	PHNO	MARKS
Justin Clarke	3627183	75
Justin Clarke	3627183	80
Justin Clarke	3627183	90
Justin Clarke	3627183	85
Justin Clarke	3627183	70
Sujan Buddha	7593752	75
Sujan Buddha	7593752	80
Sujan Buddha	7593752	90
Sujan Buddha	7593752	85
Sujan Buddha	7593752	70
Sreeraj Kasa	8549311	75
Sreeraj Kasa	8549311	80
Sreeraj Kasa	8549311	90
Sreeraj Kasa	8549311	85

```
1  CREATE VIEW MarksView AS
2  SELECT STUDENT.STUDENT_NAME, STUDENT.PHNO, RESULTS.MARKS
3  FROM STUDENT, RESULTS
```

```
View created.
```

```
1  DROP VIEW MarksView
```

```
View dropped.
```

EMPLOYEE TABLE:

```
create table emp(empno number(5) not null, ename varchar(10),job varchar(9),mgr number(4),
hiredate date,sal number(4),comm number(4),deptno number(6));

insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7369,'smith','clerk',7902,'17-dec-80',800,null,20);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7499,'allen','salesman',7698,'20-feb-81',1600,300,30);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7521,'ward','salesman',7698,'22-feb-81',1250,500,30);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7566,'jones','manager',7839,'02-apr-81',2975,null,20);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7654,'martin','salesman',7698,'28-sep-81',1250,1400,30);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7698,'blake','manager',7839,'01-may-81',2850,null,30);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7782,'clark','manager',7839,'09-jun-81',2450,null,10);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7788,'scott','analyst',7566,'19-apr-87',3000,0,20);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7939,'king','president',null,'17-nov-81',5000,null,10);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7844,'turner','salesman',7698,'08-sep-81',1500,null,30);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7876,'adams','clerk',7788,'23-may-87',1100,null,20);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7900,'james','clerk',7698,'03-dec-81',950,null,30);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7902,'ford','analyst',7566,'03-dec-81',3000,null,20);
insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno) values (7934,'miller','clerk',7782,'23-jan-82',1300,null,10);

select * from emp
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	smith	clerk	7902	17-DEC-80	800	-	20
7499	allen	salesman	7698	20-FEB-81	1600	300	30
7521	ward	salesman	7698	22-FEB-81	1250	500	30
7566	jones	manager	7839	02-APR-81	2975	-	20
7654	martin	salesman	7698	28-SEP-81	1250	1400	30
7698	blake	manager	7839	01-MAY-81	2850	-	30
7782	clark	manager	7839	09-JUN-81	2450	-	10
7788	scott	analyst	7566	19-APR-87	3000	0	20
7939	king	president	-	17-NOV-81	5000	-	10
7844	turner	salesman	7698	08-SEP-81	1500	-	30
7876	adams	clerk	7788	23-MAY-87	1100	-	20
7900	james	clerk	7698	03-DEC-81	950	-	30
7902	ford	analyst	7566	03-DEC-81	3000	-	20
7934	miller	clerk	7782	23-JAN-82	1300	-	10

Download CSV

14 rows selected.

```sql
create table dept(deptno number(6) not null,
dname varchar(20),loc varchar(20));

insert into dept(deptno,dname,loc) values (10,'accounting','new york');
insert into dept(deptno,dname,loc) values (20,'research','dallas');
insert into dept(deptno,dname,loc) values (30,'sales','chicago');
insert into dept(deptno,dname,loc) values (40,'operations','boston');

select * from dept
```

DEPTNO	DNAME	LOC
10	accounting	new york
20	research	dallas
30	sales	chicago
40	operations	boston

Download CSV

4 rows selected.

```sql
create table salgrade(grade number(5),
lowsal number(6),highsal number(6));

insert into salgrade(grade,lowsal,highsal) values(1,700,1200);
insert into salgrade(grade,lowsal,highsal) values(2,1201,1400);
insert into salgrade(grade,lowsal,highsal) values(3,1401,2000);
insert into salgrade(grade,lowsal,highsal) values(4,2001,3000);
insert into salgrade(grade,lowsal,highsal) values(5,3001,9999);

select * from salgrade
```

GRADE	LOWSAL	HIGHSAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Download CSV

5 rows selected.

```
1  select distinct(job) from emp
```

JOB
clerk
salesman
manager
analyst
president

Download CSV

5 rows selected.

```
1  select ename from emp
2  where ename like '%th%' or ename like '%ll%'
```

ENAME
smith
allen
miller

Download CSV

3 rows selected.

```
1  select ename, job, sal from emp where job='manager'
```

ENAME	JOB	SAL
jones	manager	2975
blake	manager	2850
clark	manager	2450

Download CSV

3 rows selected.

```
1  select ename, sal*115/100 increased_sal from emp
```

ENAME	INCREASED_SAL
smith	920
allen	1840
ward	1437.5
jones	3421.25
martin	1437.5
blake	3277.5
clark	2817.5
scott	3450
king	5750
turner	1725
adams	1265
james	1092.5
ford	3450
miller	1495

```
1  select * from emp where hiredate like '%82'
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7934	miller	clerk	7782	23-JAN-82	1300	-	10

Download CSV

```
1  select ename ||' has held the position of '||job||' in department '||deptno||
2  ' since '||hiredate details_of_EMP from emp
```

DETAILS_OF_EMP
smith has held the position of clerk in department 20 since 17-DEC-80
allen has held the position of salesman in department 30 since 20-FEB-81
ward has held the position of salesman in department 30 since 22-FEB-81
jones has held the position of manager in department 20 since 02-APR-81
martin has held the position of salesman in department 30 since 28-SEP-81
blake has held the position of manager in department 30 since 01-MAY-81
clark has held the position of manager in department 10 since 09-JUN-81
scott has held the position of analyst in department 20 since 19-APR-87
king has held the position of president in department 10 since 17-NOV-81
turner has held the position of salesman in department 30 since 08-SEP-81
adams has held the position of clerk in department 20 since 23-MAY-87
james has held the position of clerk in department 30 since 03-DEC-81
ford has held the position of analyst in department 20 since 03-DEC-81
miller has held the position of clerk in department 10 since 23-JAN-82

```
1  select max(sal) from emp e
```

MAX(SAL)
5000

Download CSV

```
1  select min(sal) from emp e
```

MIN(SAL)
800

Download CSV

```
1  select to_char(hiredate, 'Dy') day
2  from emp
```

DAY
Wed
Fri
Sun
Thu
Mon
Fri
Tue
Sun
Tue
Tue
Sat
Thu
Thu
Sat

```
1  select highsal-lowsal as newsal from salgrade
```

NEWSAL
500
199
599
999
6998

Download CSV

5 rows selected.

```
1  select ename ||' - '||job details_of_emp
2  from emp
```

DETAILS_OF_EMP
smith - clerk
allen - salesman
ward - salesman
jones - manager
martin - salesman
blake - manager
clark - manager
scott - analyst
king - president
turner - salesman
adams - clerk
james - clerk
ford - analyst
miller - clerk

```
1  select * from emp where job='manager'
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	jones	manager	7839	02-APR-81	2975	-	20
7698	blake	manager	7839	01-MAY-81	2850	-	30
7782	clark	manager	7839	09-JUN-81	2450	-	10

Download CSV

3 rows selected.

```
1  select ename, hiredate from emp where hiredate like '%81'
```

ENAME	HIREDATE
allen	20-FEB-81
ward	22-FEB-81
jones	02-APR-81
martin	28-SEP-81
blake	01-MAY-81
clark	09-JUN-81
king	17-NOV-81
turner	08-SEP-81
james	03-DEC-81
ford	03-DEC-81

Download CSV

```
1  select d.loc from emp e, dept d
2  where d.deptno=e.deptno and e.ename = 'james'
```

LOC
chicago

Download CSV

```
SELECT ENAME,DNAME,LOC FROM EMP E,DEPT D WHERE D.DEPTNO=E.DEPTNO AND SAL>1500
```

ENAME	DNAME	LOC
ALLEN	SALES	CHICAGO
ALLEN	SALES	CHICAGO
JONES	RESEARCH	DALLAS
BLAKE	SALES	CHICAGO
BLAKE	SALES	CHICAGO
CLARK	ACCOUNTING	NEW YORK
SCOTT	RESEARCH	DALLAS
KING	ACCOUNTING	NEW YORK
FORD	RESEARCH	DALLAS

Download CSV

9 rows selected.

```
SELECT JOB,DNAME FROM EMP E,DEPT D WHERE D.DEPTNO=E.DEPTNO AND  ENAME LIKE 'ALL_N'
```

JOB	DNAME
SALESMAN	SALES
SALESMAN	SALES

Download CSV

2 rows selected.

```
SELECT *FROM EMP WHERE HIREDATE BETWEEN '1-JAN-81' AND '15-APR-85'
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	-	30
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

Download CSV

10 rows selected.

```
SELECT TO_CHAR(HIREDATE,'DAY,DD-MONTH-YEAR')HIREDATE FROM EMP
```

HIREDATE
WEDNESDAY,17-DECEMBER -NINETEEN EIGHTY
FRIDAY ,20-FEBRUARY -NINETEEN EIGHTY-ONE
SUNDAY ,22-FEBRUARY -NINETEEN EIGHTY-ONE
THURSDAY ,02-APRIL -NINETEEN EIGHTY-ONE
FRIDAY ,01-MAY -NINETEEN EIGHTY-ONE
TUESDAY ,09-JUNE -NINETEEN EIGHTY-ONE
SUNDAY ,19-APRIL -NINETEEN EIGHTY-SEVEN
TUESDAY ,17-NOVEMBER -NINETEEN EIGHTY-ONE
TUESDAY ,08-SEPTEMBER-NINETEEN EIGHTY-ONE
SATURDAY ,23-MAY -NINETEEN EIGHTY-SEVEN
THURSDAY ,03-DECEMBER -NINETEEN EIGHTY-ONE
THURSDAY ,03-DECEMBER -NINETEEN EIGHTY-ONE
SATURDAY ,23-JANUARY -NINETEEN EIGHTY-TWO

Download CSV

13 rows selected.

SAILORS TABLE:

```
1  create table sailors(sid number primary key,sname varchar(20),rating number, age real);
2
3  insert into sailors(sid,sname,rating,age)values(22,'Dustin',7,45.0);
4  insert into sailors(sid,sname,rating,age)values(29,'Brutus',1,33.0);
5  insert into sailors(sid,sname,rating,age)values(31,'Lubber',8,55.0);
6  insert into sailors(sid,sname,rating,age)values(32,'Andy',8,25.0);
7  insert into sailors(sid,sname,rating,age)values(58,'Rusty',10,35.0);
8  insert into sailors(sid,sname,rating,age)values(64,'Horatio',7,35.0);
9  insert into sailors(sid,sname,rating,age)values(71,'Zorba',10,16.0);
10 insert into sailors(sid,sname,rating,age)values(74,'Horatio',9,35.0);
11 insert into sailors(sid,sname,rating,age)values(85,'Art',3,25.5);
12 insert into sailors(sid,sname,rating,age)values(95,'Bob',3,63.5);
13
```

SID	SNAME	RATING	AGE
22	Dustin	7	45
29	Brutus	1	33
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35
85	Art	3	25.5
95	Bob	3	63.5

```
16 create table boats(bid number primary key,bname varchar(20),color varchar(20) );
17
18 insert into boats(bid,bname,color)values(101,'interlake','blue');
19 insert into boats(bid,bname,color)values(102,'interlake','red');
20 insert into boats(bid,bname,color)values(103,'clipper','green');
21 insert into boats(bid,bname,color)values(104,'marine','red');
22
```

BID	BNAME	COLOR
101	interlake	blue
102	interlake	red
103	clipper	green
104	marine	red

Download CSV

4 rows selected.

SQL Worksheet

```
25  create table reserves(sid number,bid number,day date, primary key(sid,bid),
26  foreign key(sid)references sailors,foreign key(bid)references boats);
27
28  insert into reserves(sid,bid,day)values(22,101,TO_DATE('1998-10-10','YYYY-MM-DD'));
29  insert into reserves(sid,bid,day)values(22,102,TO_DATE('1998-10-10','YYYY-MM-DD'));
30  insert into reserves(sid,bid,day)values(22,103,TO_DATE('1998-10-8','YYYY-MM-DD'));
31  insert into reserves(sid,bid,day)values(22,104,TO_DATE('1998-10-7','YYYY-MM-DD'));
32  insert into reserves(sid,bid,day)values(31,102,TO_DATE('1998-11-10','YYYY-MM-DD'));
33  insert into reserves(sid,bid,day)values(31,103,TO_DATE('1998-11-6','YYYY-MM-DD'));
34  insert into reserves(sid,bid,day)values(31,104,TO_DATE('1998-11-12','YYYY-MM-DD'));
35  insert into reserves(sid,bid,day)values(64,101,TO_DATE('1998-9-5','YYYY-MM-DD'));
36  insert into reserves(sid,bid,day)values(64,102,TO_DATE('1998-9-8','YYYY-MM-DD'));
37  insert into reserves(sid,bid,day)values(74,103,TO_DATE('1998-9-8','YYYY-MM-DD'));
```

SID	BID	DAY
22	101	10-OCT-98
22	102	10-OCT-98
22	103	08-OCT-98
22	104	07-OCT-98
31	102	10-NOV-98
31	103	06-NOV-98
31	104	12-NOV-98
64	101	05-SEP-98
64	102	08-SEP-98
74	103	08-SEP-98

```
1  select * from sailors s where s.rating > 7
```

SID	SNAME	RATING	AGE
31	Lubber	8	55
32	Andy	8	25
58	Rusty	10	35
71	Zorba	10	16
74	Horatio	9	35

Download CSV

5 rows selected.

```
1  select S.sname from Sailors S, Reserves R where S.sid = R.sid
```

SNAME
Dustin
Dustin
Dustin
Dustin
Lubber
Lubber
Lubber
Horatio
Horatio
Horatio

Download CSV

```
1  select age from Sailors
2  where sname like 'B_%b'
```

AGE
63.5

Download CSV

```
1  select s1.sname from Sailors S1, Reserves R1, Boats B1
2  where S1.sid = R1.sid AND B1.color='red'
3  union
4  select s2.sname from Sailors S2, Reserves R2, Boats B2
5  where S2.sid = R2.sid AND B2.color='green'
```

SNAME
Dustin
Horatio
Lubber

Download CSV

3 rows selected.

```
1  select s1.sname from Sailors S1, Reserves R1, Boats B1
2  where S1.sid = R1.sid AND B1.color='red'
3  intersect
4  select s2.sname from Sailors S2, Reserves R2, Boats B2
5  where S2.sid = R2.sid AND B2.color='green'
```

SNAME
Dustin
Horatio
Lubber

Download CSV

3 rows selected.

```
1  select S.sid from Sailors S where S.rating = 10
2  UNION
3  select R.sid from Reserves R where R.bid = 104
```

SID
22
31
58
71

Download CSV

4 rows selected.

```
1  alter table sailors
2  add credits number
```

Table altered.

```
1  select * from sailors
```

SID	SNAME	RATING	AGE	CREDITS
22	Dustin	7	45	-
29	Brutus	1	33	-
31	Lubber	8	55	-
32	Andy	8	25	-
58	Rusty	10	35	-
64	Horatio	7	35	-
71	Zorba	10	16	-
74	Horatio	9	35	-
85	Art	3	25.5	-
95	Bob	3	63.5	-

Download CSV

NESTED AND CORELATED QUERIES:

```
1  select sname from sailors s
2  where s.sid in (select r.sid from reserves r
3                  where r.bid = 103)
```

SNAME
Dustin
Lubber
Horatio

Download CSV

3 rows selected.

```
1  select sname from sailors s
2  where s.sid not in (select r.sid from reserves r
3                      where r.bid = 103)
```

SNAME
Brutus
Andy
Rusty
Horatio
Zorba
Art
Bob

Download CSV

7 rows selected.

```
1  select sname from sailors s
2  where s.sid in (select r.sid from reserves r
3  where r.bid in (select b.bid from boats b where b.color='red'))
```

SNAME
Dustin
Lubber
Horatio

Download CSV

3 rows selected.

```
1  select sname from sailors s
2  where s.sid in (select r.sid from reserves r
3  where r.bid not in (select b.bid from boats b where b.color='red'))
```

SNAME
Dustin
Lubber
Horatio
Horatio

Download CSV

4 rows selected.

```
1  select sname from sailors s
2  where s.sid not in (select r.sid from reserves r
3  where r.bid not in (select b.bid from boats b where b.color='red'))
```

SNAME
Brutus
Andy
Rusty
Zorba
Art
Bob

Download CSV

6 rows selected.

```
1  select sname from sailors s
2  where exists(select * from Reserves R
3  where R.bid=103 and S.sid=R.sid)
```

SNAME
Dustin
Lubber
Horatio

Download CSV

3 rows selected.

```
1  select sname from sailors s
2  where not exists(select * from Reserves R
3  where R.bid=103 and S.sid=R.sid)
```

SNAME
Brutus
Andy
Rusty
Horatio
Zorba
Art
Bob

Download CSV

7 rows selected.

```
1  select s.sid from sailors s
2  where s.rating > any(select s2.rating
3  from sailors s2 where s2.sname='Horatio')
```

SID
58
71
74
31
32

Download CSV

5 rows selected.

```
1  select s.sid from sailors s
2  where s.rating >= all(select s2.rating
3  from sailors s2 where s2.sname='Horatio')
```

SID
74
71
58

Download CSV

3 rows selected.

```
1  select s.sname from sailors s
2  where s.age > (select max(s1.age)
3  from sailors s1 where s1.rating = 10)
```

SNAME
Dustin
Lubber
Bob

Download CSV

3 rows selected.

```
1  select s.rating, min(s.age) from sailors s
2  where s.age >= 18 group by
3  s.rating having count (*)>1
```

RATING	MIN(S.AGE)
7	35
8	25
3	25.5

Download CSV

3 rows selected.

```
1  select s2.rating from sailors s2
2  where s2.sname = 'Horatio'
```

RATING
7
9

Download CSV

2 rows selected.

TRIGGERS:

```
1  create trigger sail_trigg after insert on sailors
2  begin
3  insert into logg values(sysdate, '1 row inserted','sailors');
4  dbms_output.put_line('CONGRATULATIONS');
5  end;
```

```
Trigger created.
```

```
1  insert into sailors values(21,'Ram',7)
2
```

```
1 row(s) inserted.
CONGRATULATIONS
```

```
1  select * from logg
```

ACCESS_DATE	OPERATION_VALUE	TABLE_NAME
15-DEC-20	1 row inserted	sailors

Download CSV

```
1  create trigger sail_trigg2 after update on sailors
2  begin
3  insert into logg values(sysdate,'1 row updated', 'sailors');
4  dbms_output.put_line('Success!');
5  end;
6
```

```
Trigger created.
```

```
1  update sailors set sname='Uday' where sid=10
```

```
1 row(s) updated.
Success!
```

```
1  select * from logg
```

ACCESS_DATE	OPERATION_VALUE	TABLE_NAME
15-DEC-20	1 row inserted	sailors
15-DEC-20	1 row inserted	sailors
15-DEC-20	1 row updated	sailors

Download CSV

3 rows selected.

```
1  create trigger sail_trigg3 after update on sailors
2  begin
3  dbms_output.put_line('Success!');
4  insert into logg values(sysdate, '1 row successfully update','sailors');
5  end;
```

```
Trigger created.
```

```
1  update sailors set rating =3 where sid =21
```

```
1 row(s) updated.
Success!
Success!
```

```
1  select * from logg
```

ACCESS_DATE	OPERATION_VALUE	TABLE_NAME
15-DEC-20	1 row inserted	sailors
15-DEC-20	1 row inserted	sailors
15-DEC-20	1 row updated	sailors
15-DEC-20	1 row successfully update	sailors
15-DEC-20	1 row updated	sailors
15-DEC-20	1 row successfully update	sailors
15-DEC-20	1 row updated	sailors

Download CSV

7 rows selected.

```
create trigger sail_trigg5 after update on sailors
begin
dbms_output.put_line ('A row deleted!!!');
insert into logg values(sysdate, '1 row successfully deleted ',' CLASS ');
end;
```

Trigger created.

```
delete from sailors where sid=20
```

1 row(s) deleted.

```
select *from logg
```

ACCESS_DATE	OPERATION_VALUE	TABLE_NAME
30-NOV-20	1 row inserted	sailors
30-NOV-20	1 row inserted	sailors
30-NOV-20	1 row updated	sailors
30-NOV-20	1 row successfully updated	sailors
30-NOV-20	1 row updated	sailors
30-NOV-20	1 row Updated	sailors
30-NOV-20	1 row successfully updated	sailors
30-NOV-20	1 row updated	sailors

Download CSV

8 rows selected.

```
1  create trigger sail_trigg4 after update on sailors for each row
2  begin
3  dbms_output.put_line('Sucess!!!');
4  insert into logg values (sysdate, '1 row updates','sailors');
5  end;
```

Trigger created.

```
1  update sailors set rating =10 where sid=21
```

```
1 row(s) updated.
Sucess!!!
Success!
Success!
```

ACCESS_DATE	OPERATION_VALUE	TABLE_NAME
15-DEC-20	1 row updates	sailors
15-DEC-20	1 row successfully update	sailors
15-DEC-20	1 row updated	sailors
15-DEC-20	1 row inserted	sailors
15-DEC-20	1 row inserted	sailors
15-DEC-20	1 row updated	sailors
15-DEC-20	1 row successfully update	sailors
15-DEC-20	1 row updated	sailors
15-DEC-20	1 row successfully update	sailors
15-DEC-20	1 row updated	sailors
15-DEC-20	1 row successfully update	sailors
15-DEC-20	1 row updated	sailors

Download CSV

12 rows selected.

PL/SQL PROGRAMS:

```
Begin
   for i in reverse 1..10 loop
      dbms_output.put_line(i);
    end loop;
End;
```

```
Statement processed.
10
9
8
7
6
5
4
3
2
1
```

```
Declare
   f1 number:=0;
   f2 number:=1;
   f3 number;
Begin
   dbms_output.put_line(f1);
   dbms_output.put_line(f2);
   for i in 1..9 loop
     f3:=f1+f2;
     f1:=f2;
     f2:=f3;
     dbms_output.put_line(f3);
    end loop;
End;
```

```
Statement processed.
0
1
1
2
3
5
8
13
21
34
55
```

```
Declare
   s number:=0;
Begin
   for i in 1..10 loop
      s:=s+i;

    end loop;
    dbms_output.put_line(s);
End;
```

```
Statement processed.
55
```

```
Declare
   n number:=534;
   r number:=0;
Begin
   select
   reverse(to_char(n))
   into r from dual;
   dbms_output.put_line(r);
End;
```

```
Statement processed.
435
```

```
Declare
n number(3):=0;
Begin
while n <=100 loop
  n:=n+2;
  dbms_output.put_line(n);
end loop;
End;
```

```
Statement processed.
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
```

```
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
```

```
98
100
102
```

```
DECLARE
i NUMBER(2):=9;
j NUMBER(2);
BEGIN
    FOR j IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE(i || ' * ' || j || '= ' || i*j);
END LOOP;
END;
```

```
Statement processed.
9 * 1= 9
9 * 2= 18
9 * 3= 27
9 * 4= 36
9 * 5= 45
9 * 6= 54
9 * 7= 63
9 * 8= 72
9 * 9= 81
9 * 10= 90
```

```
DECLARE
  x NUMBER;
  n NUMBER;
  i NUMBER;

  --function for finding sum
  FUNCTION Findmax(n IN NUMBER)
    RETURN NUMBER
  IS
    sums NUMBER := 0;
  BEGIN

    --for loop for n times iteration
    FOR i IN 1..n
    LOOP
      sums := sums + i*(i+1)/2;
    END LOOP;
    RETURN sums;
  END;
  BEGIN

    --driver code
    n := 4;
    x := findmax(n);
    dbms_output.Put_line('Sum: '
    || x);
  END;
```

```
Statement processed.
Sum: 20
```

```
Declare
    n number;
    temp number;
    r number;
Begin
    n := 123456;
    temp:= 0;
    while n <> 0 loop
        r := MOD(n, 10);
        temp:=temp + r;
        n := Trunc(n / 10);
    end loop;
    dbms_output.Put_line('sum of digits = ' || temp);
End;
```

```
Statement processed.
sum of digits = 21
```

```
Declare
  num Number(2) := 30;
Begin
  if (num > 0) then
     DBMS_OUTPUT.PUT_LINE('positive');
  else
     DBMS_OUTPUT.PUT_LINE('Negative');
  end if;
End;
```

```
Statement processed.
positive
```

```
Declare
        n number;
        i number;
        flag number;
Begin
        i:=2;
        flag:=1;
        n:=5;
        for i in 2..n/2
        loop
                if mod(n,i)=0
                then
                        flag:=0;
                        exit;
                end if;
        end loop;
        if flag=1
        then
                dbms_output.put_line('prime');
        else
                dbms_output.put_line('not prime');
        end if;
End;
```

```
Statement processed.
prime
```

```
Declare
    a number;
    b number;
Begin
    a:=5;
    b:=10;
    dbms_output.put_line('before swapping:');
    dbms_output.put_line('a='||a||' b='||b);
    a:=a+b;
    b:=a-b;
    a:=a-b;
    dbms_output.put_line('after swapping:');
    dbms_output.put_line('a='||a||' b='||b);
End;
```

```
Statement processed.
before swapping:
a=5 b=10
after swapping:
a=10 b=5
```

```
Declare
  n1 number;
  n2 number;
  n3 number;
Begin
  n1:=7;
  n2:=8;
  n3:=n1+n2;
   dbms_output.put_line('The addition of'||n1|| 'and' ||n2||'='||n3);
  n3:=n1-n2;
   dbms_output.put_line('The subtraction of'||n1|| 'and' ||n2||'='||n3);
  n3:=n1*n2;
   dbms_output.put_line('The multiplication of'||n1|| 'and' ||n2||'='||n3);
  n3:=n1/n2;
   dbms_output.put_line('The division of'||n1|| 'and' ||n2||'='||n3);
End;
```

```
Statement processed.
The addition of7and8=15
The subtraction of7and8=-1
The multiplication of7and8=56
The division of7and8=.875
```

FUNCTIONS AND PROCEDURES:

```
create or replace function fn_cube(p number)
return number is
y number;
Begin
    y:=p*p*p;
     return y;
End;
```

Function created.

```
Declare
   a number;
Begin
    dbms_output.put_line('the cube of number is:'||fn_cube(8));
End;
```

Statement processed.
the cube of number is:512

```
create or replace procedure greater(a in number,b in number) as
Begin
    if a>b then
      dbms_output.put_line(a||' is greater than '||b);
    else
      dbms_output.put_line(b||' is greater than '||a);
    End if;
End;
```

Procedure created.

```
Declare
    p number;
    q number;
Begin
    p:=15;
    q:=5;
    greater(p,q);
End;
```

Statement processed.
15 is greater than 5