| **Index** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Name of Experiments** | **Page**  **No.** | **Date of**  **Experiment** | **Date of**  **Supervision** | **Remarks** |
|  | **Unit-1** |  |  |  |  |
| 1 | Create Tables | 5 |  |  |  |
|  | **Unit-2** |  |  |  |  |
| 2 | Basic Queries-1 | 8 |  |  |  |
| 3 | Basic Queries-2 | 9 |  |  |  |
| 4 | Formatting Commands | 9 |  |  |  |
| 5 | Joins, Order by, Group by | 11 |  |  |  |
|  | **Unit-3** |  |  |  |  |
| 6 | Use of Functions | 15 |  |  |  |
| 7 | Mixed Queries | 18 |  |  |  |
|  | **Unit-4** |  |  |  |  |
| 8 | Hello World Script | 22 |  |  |  |
| 9 | Add two numbers | 22 |  |  |  |
| 10 | Find area of circle | 22 |  |  |  |
| 11 | Find area of rectangle | 22 |  |  |  |
| 12 | Find maximum value | 23 |  |  |  |
| 13 | Check for Odd/Even value | 23 |  |  |  |
| 14 | Use of CASE statement | 23 |  |  |  |
| 15 | Find factorial (using for) | 24 |  |  |  |
| 16 | Find factorial (using while) | 24 |  |  |  |
| 17 | Use of select into statement | 25 |  |  |  |
| 18 | Use of %rowtype | 25 |  |  |  |
| 19 | Use of cursor | 26 |  |  |  |
| 20 | Cursor with for loop | 26 |  |  |  |
| 21 | Exception handling | 26 |  |  |  |
| 22 | Creating Procedure | 27 |  |  |  |
| 23 | Creating Function | 27 |  |  |  |

**Unit – 1 : DBMS Overview, SQL, SQL\*Plus**

**Create the following tables:**

**Table Name** : DEPT (Use to store information about departments of company)

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Attributes** |
| deptno | Numeric | 2 | Primary Key. |
| Dname | Varchar | 14 | Not Null |
| Loc | Varchar | 13 |  |

create table dept(

  deptno     number(2,0),

  dname      varchar2(14),

  loc        varchar2(13),

  constraint pk\_dept primary key (deptno)

);

**Data:**

|  |  |  |
| --- | --- | --- |
| **deptno** | **Dname** | **Loc** |
| 10 | Accounting | New York |
| 20 | Research | Dallas |
| 30 | Sales | Chicago |
| 40 | Operations | Boston |

**insert into DEPT (DEPTNO, DNAME, LOC) values(10, 'ACCOUNTING', 'NEW YORK');**

**insert into dept values(20, 'RESEARCH', 'DALLAS');**

**insert into dept values(30, 'SALES', 'CHICAGO');**

**insert into dept values(40, 'OPERATIONS', 'BOSTON');**

**Table Name**: EMP   (Use to store information about Employee of company)

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Size** | **Attributes** |
| Empno | Numeric | 4 | Primary Key |
| Ename | Varchar | 10 | Not Null |
| Job | Varchar | 9 |  |
| Mgr | Number | 4 |  |
| Hiredate | Date |  |  |
| Sal | Numeric | 7,2 |  |
| Comm. | Numeric | 7,2 |  |
| Deptno | Numeric | 2 | Not Null / / Foreign key references deptno of Dept Table |

create table emp(

  empno    number(4,0),

  ename    varchar2(10),

  job      varchar2(9),

  mgr      number(4,0),

  hiredate date,

  sal      number(7,2),

  comm     number(7,2),

  deptno   number(2,0),

  constraint pk\_emp primary key (empno),

  constraint fk\_deptno foreign key (deptno) references dept (deptno)

);

**Data :**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **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 |  | 20 |
| 7839 | 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 |

insert into emp values(7839, 'KING', 'PRESIDENT', null, to\_date('17-11-1981','dd-mm-yyyy'), 5000, null, 10);

insert into emp values(7698, 'BLAKE', 'MANAGER', 7839, to\_date('1-5-1981','dd-mm-yyyy'), 2850, null, 30);

insert into emp values(7782, 'CLARK', 'MANAGER', 7839, to\_date('9-6-1981','dd-mm-yyyy'), 2450, null, 10);

insert into emp values(7566, 'JONES', 'MANAGER', 7839, to\_date('2-4-1981','dd-mm-yyyy'), 2975, null, 20);

insert into emp values(7788, 'SCOTT', 'ANALYST', 7566, to\_date('13-JUL-87','dd-mm-rr') - 85, 3000, null, 20);

insert into emp values(7902, 'FORD', 'ANALYST', 7566, to\_date('3-12-1981','dd-mm-yyyy'), 3000, null, 20);

insert into emp values(7369, 'SMITH', 'CLERK', 7902, to\_date('17-12-1980','dd-mm-yyyy'), 800, null, 20);

insert into emp values(7499, 'ALLEN', 'SALESMAN', 7698, to\_date('20-2-1981','dd-mm-yyyy'), 1600, 300, 30);

insert into emp values(7521, 'WARD', 'SALESMAN', 7698, to\_date('22-2-1981','dd-mm-yyyy'), 1250, 500, 30);

insert into emp values(7654, 'MARTIN', 'SALESMAN', 7698, to\_date('28-9-1981','dd-mm-yyyy'), 1250, 1400, 30);

insert into emp values(7844, 'TURNER', 'SALESMAN', 7698, to\_date('8-9-1981','dd-mm-yyyy'), 1500, 0, 30);

insert into emp values(7876, 'ADAMS', 'CLERK', 7788, to\_date('13-JUL-87', 'dd-mm-rr') - 51,1100, null, 20);

insert into emp values(7900, 'JAMES', 'CLERK', 7698, to\_date('3-12-1981','dd-mm-yyyy'), 950, null, 30);

insert into emp values(7934, 'MILLER', 'CLERK', 7782, to\_date('23-1-1982','dd-mm-yyyy'), 1300, null, 10);

**Unit – 2**

**Managing Tables and Data, Data Control and Transaction Control Command**

**Basic Query - 1**

1. LIST ALL ROWS OF THE TABLE EMP.

Select \* From Emp;

1. DISPLAY THE STRUCTURE OF THE TABLE EMP.

Describe Emp;

1. LIST ALL TABLES.

Select \* From Tab;

1. LIST ALL EMPLOYEES NUMBER FROM EMP.

Select empno from emp;

1. LIST ALL EMPLOYEES NAME AND DESIGNATION FROM EMP.

Select ename, job from emp;

1. LIST ALL COMMISSION FROM EMP.

Select empno, comm from emp;

1. LIST ALL MGR FROM EMP.

Select empno, mgr from emp;

1. LIST ALL DEPARTMENTS NUMBER FROM EMP.

Select deptno from dept;

1. LIST ALL EMPLOYEES NUMBER, EMPLOYEES NAME, JOBS, MGR FROM EMP.

Select empno, ename, job, mgr from emp;

1. LIST ALL EMPLOYEES NAME, MGR FROM EMP.

Select ename, mgr from emp;

1. LIST ALL JOB, COMMISSION FROM EMP.

Select job, comm from emp;

1. LIST ALL EMPLOYEES NAME, DEPARTMENTS NUMBER FROM EMP.

Select ename, deptno from emp;

1. LIST ALL DEPARTMENTS NUMRER, EMPLOYEES NAME, SALARIES, JOBS FROM EMP.

Select deptno, ename, sal, job from emp;

1. LIST ALL DEPARTMENTS NO AND NAME FROM DEPT.

Select deptno, dname from dept;

1. LIST ALL Locatoin FROM DEPARTMENT.

Select loc from dept;

1. LIST ALL Department name and Location FROM DEPT.

Select dname, loc from dept;

**Basic Queries – 2**

1. LIST ALL EMPLOYEES WHO ARE MANAGERS.

Select \* from emp where job=’MANAGER’;

1. LIST ALL EMPLOYEES WHO ARE PRESIDENTS.

Select \* from emp where job=’PRESIDENT’;

1. LIST ALL EMPLOYEES WHO ARE EITHER MANAGER OR CLERK.

Select \* from emp where job=’MANAGER’ or job=’CLERK’;

Or

Select \* from emp where job in(‘MANAGER’,’CLERK’);

1. LIST ALL EMPLOYEES WHO ARE MANAGERS OF DEPARTMENT NUMBER 10.

        Select \* from emp where job='Manager' and deptno=10;

1. LIST ALL EMPLOYEES WHO ARE WORKNG IN DEPARTMENT NUMBER 10.

Select \* from emp where deptno=10;

1. LIST ALL EMPLOYEES WHO ARE NOT BELONGING TO DEPARTMENT NUMBER 20, 30, 40 AND ARE NOT MANAGER.

Select \* from emp where deptno in(20,30,40) and job<>’MANAGER’;

Or

Select \* from emp where deptno in(20,30,40) and job not in(’MANAGER’);

1. LIST ALL EMPLOYEES WHO ARE EITHER MANAGER OF DEPARTMENT NUMBER 10 OR ARE ANALYST.

select \* from emp where (job='Manager' and deptno=10) or job='Analyst';

1. LIST ALL EMPLOYEES WHO ARE NOT PRESIDENT, ANALYST OR SALESMAN.

Select \* from emp where job not in(‘PRESIDENT’ ,’ANALYST’, ’SALESMAN’);

1. LIST ALL EMPLOYEES WHO WORK IN DEPARTMENT NUMRER 10 AND THEIR JOB IS CLERK.

Select \* from emp where job='Clerk' and deptno=10;

**Formatting Commands**

1. CHANGE THE COLUMN HEADING ENAME INTO YOUR NAME AND DISPLAY IT.

Select ename "Malay" from emp;

1. CHANGE THE COLUMN HEADING SAL INTO SALARIES.

Select sal "SALARIES" from emp;

1. CHANGE THE COLUMN HEADING ENAME INTO 'EMPLOYEE NAME' AND JOB INTO DESIGNATION.

Select emp "EMPLOYEE NAME",job "Designation" from emp;

1. LIST ALL 'MANAGER' AND ADD 100 RUPEES IN EACH SALARY.

Select empno, ename, job, sal, sal+100 from emp where job=’MANAGER’;

1. LIST ALL EMFLOYEE AND ADD 1000 RUPEES AND SUBTRACT 4500 RUPEES IN EACH SALARY.

Select empno, ename, job, sal, (sal+1000)-4500 from emp;

1. IF EMPLOYEE NAME IS SCOTT THEN ADD 100 RUPEES IN HIS SALARY.

Select empno, ename, sal, (sal+100) from emp where ename=’SCOTT’;

1. LIST ALL EMPLOYEE AFTER ADDING 500 RUPEES, SUBTRACT 300 RUPEES AND DIVIDED BY 100 RUPEES FROM EACH SALARY.

Select empno, ename, sal, ((sal+500)-300)/100 from emp;

1. LIST ALL EMPLOYEES AND FIND FIVE PERCENT FROM EACH SALARY.

Select empno, ename, sal, (sal\*5)/100 from emp;

1. FIND EIGHT PERCENT OF SALESMEN SALARY.

Select empno, ename, sal, (sal\*8)/100 from emp where job=’SALESMEN’;

1. FIND THE TWO MONTH SALARY OF EACH MANAGER.

Select empno, ename, sal, (sal\*2) from emp;

1. FIND THE DAILY WAGES OF EACH EMPLOYEE.

Select empno, ename, sal, (sal/30) from emp;

1. DISPLAY THE COLUMN HEADING ANNSAL INSTEAD OF SAL\*12.

Select empno, ename, sal, (sal\*12) “ANNSAL” from emp;

1. COMBINE EMPNO AND ENAME. THEN CHANGE THE COLUMN HEADING AS EMPLOYEE.

Select empno||' '||ename "Employee" from emp;

1. JOIN ENAME AND JOB. THEN CHANGE THE COLUMN HEADING AS COMP\_NAME.

select ename||' '||job "Comp\_Name" from emp;

1. LIST ALL RECORDS AS FOLLOWS. SMITH IS CLERK AND HE IS WORKING IN DEPT NO 20 SINCE 12-MAR- 1981.

select ename||' is '||job||' and he is working in Dept No '||deptno||' Since '||hiredate from emp;

1. FIND ANNUAL SALARY AND ADD COMM OF ALL EMPLOYEES.

select ename,sal,(sal\*12)+nvl(comm,0) from emp;

1. LIST ALL DISTINCT DEPARTMENT NUMBER FROM EMP TABLE.

select distinct deptno from emp;

1. DISPLAY DISTINCT VALUE OF DEPARTMENT NUMBER AND JOBS.

select distinct deptno, job from emp;

1. LIST ALL EMPLOYEES WHO ARE NOT ELIGIBLE FOR RECEIVING COMMISSION AND ARE 'MANAGER'.

select \* from emp where comm is null and job='Manager';

1. LIST ALL EMPLOYEES WHO ARE ELIGIBLE OF RECEIVING COMMISSION.

select \* from emp where comm is not null;

1. LIST ALL DISTINCT JOBS FROM THE TABLE EMP.

select distinct job from emp;

1. HOW MANY JOBS DO WE HAVE.

select count(distinct job) from emp;

**Joins, Subquery, Order by, Group by**

1. LIST HIREDATE IN DESCENDING ORDER.

select \* from emp order by hiredate desc;

1. LIST HIREDATE IN ASCENDINO ORDER.

select \* from emp order by hiredate asc;

1. FIND ALL DEPARTMENT NAMES WITH THEIR DEPARTMENT NUMBERS WHILE NUMBERS GREATER THEN 20.

select deptno, dname from dept where deptno>20;

1. FIND THOSE EMPLOYEES WHOSE COMMISSION IS GREATER THEN THEIR SALARIES.

select \* from emp where comm>sal;

1. FIND THOSE EMPLOYEES WHOSE COMMISSION IS LESS THEN THEIR SALARIES.

select \* from emp where comm<sal;

1. FIND ALL EMPLOYEES THOSE SALARIES ARE BETWEEN 500 AND 1500.

select \* from emp where sal>=500 and sal<=1500;

OR

select \* from emp where sal between 500 and 1500;

1. FIND THOSE EMPLOYEES WHOSE MGR IS 7902, 7566, 7788.

select \* from emp where mgr in (7902, 7566, 7788);

1. TO FIND THOSE EMPLOYEES WHOSE MGR IS BETWEEN 7788 AND 7092.

select \* from emp where mgr between 7788 and 7902;

1. LIST ALL EMPLOYEES WHOSE NAME START WITH ‘S’.

select \* from emp where ename like 'S%';

1. LIST ALL EMPLOYEES HAVING FIRST NAME AS 'SCOTT’.

select  \* from emp where ename='Scott';

1. LIST OF THOSE EMPLOYEES HAVING ONLY 5 CHARACTERS LONG NAME AND JOB AS A MANAGER.

select  \* from emp where job='Manager' and length(ename)=5;

1. LIST ALL EMPLOYEES WHO DOES NOT HAVE A MANAGER (WHOSE MANAGER IS NULL).

select  \* from emp where mgr is null;

1. FIND THOSE EMPLOYEES WHOSE JOB DO NOT START WITH A'.

select  \* from emp where job not like 'A%';

1. LIST OF THOSE EMPLOYEES WHOSE MGR IS NOT NULL.

select \* from emp where mgr is not null;

1. LIST OF THOSE EMPLOYEES WHOSE JOBS ARE 'MANAGER OR CLERK' OF DEPARTMENT NUMBER 10.

select \* from emp where job in('Manager','Clerk') and deptno=10;

1. FIND ALL MANAGERS AND SALESMEN THEIR SALARIES OVER 1500.

select \* from emp where job in('Manager','Salesman') and sal>1500;

1. FIND THE AVERAGE SALARIES OF DIFFERENT JOBS.

select job, avg(sal) from emp group by job;

1. FIND THE AVERACE SALARIES OF ALL JOBS EXCLUDING MANAGER.

select job, avg(sal) from emp group by job having job not in('Manager');

1. FIND THE AVERACE SALARY OF EACH JOB WITHIN DEPARTMENT.

select job,avg(sal),deptno from emp group by job,deptno;

1. FIND THE AVERAGE SALARIES OF EACH DEPARTMENT.

select deptno,avg(sal),deptno from emp group by deptno;

1. FIND AVERAGE AND SUM OF THE SALARIES OF EACH JOB EXCLUDING SALESMEN', CLERK' AND 'MANAGER'.

select job,sum(sal),avg(sal) from emp group by job having job not in('Salesman','Clerk','Manager');

1. Display the location of SMITH.

select loc from emp e , dept d where e.ename = ‘SMITH’ and e.deptno = d.deptno ;

1. List the total information of EMP table along with DNAME and Loc of all the emps Working Under ‘ACCOUNTING’ & ‘RESEARCH’ in the asc Deptno.

select \* from emp e ,dept d where (dname = ‘ACCOUNTING’ or dname=’RESEARCH’) and e.deptno = d.deptno order by e.deptno asc;

OR

select \* from emp e ,dept d where d.dname in (‘ACCOUNTING’,’RESEARCH’) and e.deptno = d.deptno order by e.deptno asc;

1. List the details of the emps whose Salaries more than the employee BLAKE.

select \* from emp where sal > (select sal from emp where ename =‘BLAKE’);

1. List the emps whose Jobs are same as ALLEN.

select \* from emp where job = (select job from emp where ename =‘ALLEN’);

1. List the emps who are senior to King.

select \* from emp where hiredate < ( select hiredate from emp  
where ename = ‘KING’);

**Unit – 3**

**Other ORACLE Database Objects, Concurrency control using lock**

**Use of Functions**

1. FIND THE ABSOLUTE VALUE OF -18.

1. FIND THE ABSOLUTE VALUE OF 30.

1. FIND THE CELING VALUE OF 22.5.

1. FIND THE CEILING VALUE OF 31.2. .

1. FIND THE FLOOR VALUE OF 37.2.

1. DISPLAY YOUR NAME IN LOWERCASE.

1. FIND THE SOUARE OF 6.

1. FIND THE FIRST AND SECOND CHARACTERS OF ENAMS.

1. DISPLAY THE YEAR ONLY FROM THE HIREDATE.

1. FIND THE SOUNDEX OF ALL ENAME.

1. FIND THE SOUNDEX OF THOSE EMPLOYEE WHOSE ENAME IS ALLEN.

1. FIND THE MINIMUM SALARY FROM THE TABLE EMP.

1. FIND THE MAXIMUM SALARY SROM THE TABLE EMP.

1. FIND THE LENGTH OF ALL ENAME.

1. FIND THE LENGTH OF JOB.

1. FIND THE VSIZE OF ENAME.

1. FIND THE VALUE OF 2 THROUGH ASCII VALUE.

1. FIND THE ROWID OF ALL EMPLOYEES.

1. FIND THE USER ID, WHICH ASSIGNED BY ORACLE.

1. FIND THE USER NAME, IN WHICH YOU ARE WORKING.

1. FIND THE VSIZE OF ALL EMPLOYEES WHO ARE WORKING N DEPARTMENT NUMBER 10.

1. FIND THE SUM OF ALL SALARIES.

1. FIND THE GRADE OF ALL EMPLOYEES.

1. FIND THOSE EMPLOYEES WHOSE DEPARTMENT LOCATION IS NEWYORK.

1. FIND THOSE EMPLOYEES WHO ARE WORKING IN ACCOUNTING DEPARTMENT.

1. DEFINE THE VALUL AS ANNUAL SALARY SAL\*12+NVL(COMM,0) AND DISPLAY THE RESULT.

1. DEFINE THE VALUE AS ANNUAL SAL\*12 AND DISPLAY THE RESULT.

1. LIST ALL EMPLOYEES WHO ARE WORKING IN 'SALES' DEPARTMENT.

1. LIST ALL EMPLOYEES WHOSE LOCATION IS 'NEW YORK' AND DEPARTMENT NAME IS ACCOUNTING'.

1. LIST ALL EMPLOYEES WHOSE LOCATION IS 'CHICAGO'.

1. FIND THE MANAGER OF 'SCOTT'.

1. FIND THE MANAGER OF 'BLAKE'.

1. FIND THE SALARIES OF THOSE EMPLOYEES WHO EARN MORE THEN JONES'.

1. LIST THOSE DEPARTMENT WHO DOSE NOT HAVE ANY EMPLOYEES.

1. FIND THE UNION OF DEPARTMENT NUMFER 10 AND DEPARTMENT NUMBER 30.

1. FIND THE INTERSECT OF DEPARTMENT NUMBER 10 AND DEPARTMENT NUMBER 30.

1. FIND THE MINUS OF DEPARTMENT NUMBER 10 AND DEPARTMENT NUM8ER 30.

1. FIND THE MINIMUM SALARIES WITH EMPLOYEE NAME.

1. FIND ALL EMPLOYEES WHO HAVE THE SAME JOB AS BLAKE.

1. FIND THE MINIMUM THREE SALARIES N EACH DEPARTMENT.

1. FIND ALL THE DEPARTMENTS, WHICH HAVE AN-AVERAGE SALARY IS GREATER THEN DEPARTMENT NUMBER 30.

1. FIND ALL EMPLOYEES IN DEPARTMENT NUMBER 10 WHOSE JOB IS THE SAM AS THE EMPLOYEES JOB IN THE SALES DEPARTMENT.

1. FIND ALL EMPLOYEES WHO EARN MORE THAN THE AVERAGE SALARY OF EMPLOYEES IN THEIR OWN DEPARTMENT AND SORT BY DEPARTMENT NO.

1. FIND ALL EMPLOYEES WHO EARN MORE THAN ANY EMPLOYEE IN DEPARTMENT 30.

1. FIND THE JOB WHICH HAS THE HIGHEST AVERAGE SALARY.

1. FIND ALL JOBS WITH EITHER THE SAME OR AS 'JONES' OR SALARY GREATER THAN OR EQUAL TO FORD, ORDER BY JOB AND SALARY.

1. LIST ALL EMPLOYEES AND THEIR MANAGER IN TREE FORMAT.

1. LIST ALL EMPLOYEES AND THEIR MANAGER IN TREE FORMAT WITH LEVEL.

1. FIND THE AVERAGE SALARY AT EACH LEVEL OF THE EMPLOYEE TREE.

**Mixed Queries**

1. DISPLAY THE STRUCTURE OF THE TABLE EMP.

1. DISPLAY SYSTEM DATE (USING SQL STATEMENT ONLY).

1. COUNT ALL EMPLOYEES.

1. DISPLAY THE SUM OF ALL EMPLOYEES SALARIES.

1. HOW MANY MANAGERS DO WE HAVE.

1. HOW MANY DEPARTMENTS DO WE HAVE.

1. LIST AVERAGE SALARY OF EACH DEPARTMENT.

1. LIST AVERAGE SALARY OF EACH JOB.

1. FIND THE MAXIMUM. MINIMUM SAL OF ALL EMPLOYEES.

1. FIND THE DAY OF ALL EMPLOYEES WHEN THEY WERE HIRED.

1. FIND THE NEXT FRIDAY AFTER THE ‘12-MAR-98'.

1. FIND THE LAST DAY OF EACH HIREDATE.

1. LIST ALL EMPLOYEES WTH THEIR SALGRADE.

1. LIST ALL EMPLOYEES WHO WERE HIRED IN YEAR 1983.

1. LIST ALL EMPLOYEES WHO WERE HIRED BETWEEN 30-MAR-81 TO 01-JUN-82.

1. FIND THE INCOME TAX OF ALL EMPLOYEES @5% OF ITS ANNUAL SALARY.

1. DISPLAY ALL RECORDS IN FOLLOWING FORMAT (FOR ALL RECORDS) ON EMPLOYEE

<NAME>WORKS IN DEPARTMENT<NAME> AND APPOINTED ON <DATE>.

1. FIND THE DURATION OF SERV!CE OF ALL EMPLOYEES.

1. COUNT ALL EMPLOYEES WHO WERE HIRED AFTER 22-MAR-81.

1. LIST THE MOST EXPERIENCED EMPLOYEE.

1. DISPLAY THE DEPARTMENTS NUMSER AND THEIR GIVEN NAMES USING DECODE() FUNCTION.

10 ACCOUNTING.

20 RESEARCH..

30 SALES.

1. IN WHICH YEAR DID MOST PEOPLE JOIN THE COMPANY, DISPLAY THE YEAR AND NUMBER OF EMPLOYEE.

1. FIND THE FIRST OCCURRENCE OF THE LETTER 'L’ IN EMPLOYEE NAME AND CHANGE 'L' WITH 'X'.

1. FIND THE EMPLOYEE WHO HAS AT LEAST ONE PERSON FOR REPORTING.

1. FIND ALL EMPLOYEES WHO’S DEPARTMENT IS NOT IN THE DEPT TABLE.

1. LIST EMPLOYEE IN ORDER (OF MANAGER EMPLOYEE TREE FORMAT).

1. DISPLAY THE DATE COLUMN AT THE LEFT SIDE OF THE EMP TABLE.

1. DISPLAY THE TITLE IN CENTER' REPORT OF THE EMPLOYEE TABLE' ON TABLE EMP.

1. DISPLAY 'PAGES AT THE RIGHT SIDE OF THE EMP TABLE.

1. DISPLAY AT THE END ON LEFT SIDE CONFIDENTIAL REPORT.

**Unit – 4**

**Introduction to PL/SQL, Advanced PL/SQL**

**8. Print Hello World**

|  |  |
| --- | --- |
| 1  2  3  4  5 | --Hello World Script  begin  dbms\_output.put\_line('Hello World');  end; |
|  |  |
| Output:  Hello World | |

**9. Add two numbers**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | -- Using Variables  declare  a number:=10;  b number:=20;  begin  dbms\_output.put\_line('Sum: ' || (a+b));  end; |
|  |  |
| Output:  Sum: 30 | |

**10. Find area of circle**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | --Script to find area of circle.  declare  r number:=&Radius;  PI constant number:=3.14;  area number;  begin  area:=PI\*power(r,2);  dbms\_output.put\_line('Area is: ' || area);  end; |
|  |  |
| Output:  Radius: 5  Area is: 78.5 | |

**11. Find area of rectangle**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | --Script to find area of ractangle.  Declare  h number:=&Height;  w number:=&Width;  Begin  dbms\_output.put\_line('Area is ' || (h\*w));  End; |
|  |  |
| Output: | |

**12. Find max value**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | --Script to find max value.  declare  a number:=&A;  b number:=&B;  begin  if a>b then  dbms\_output.put\_line('Max: ' || a);  elsif b>a then  dbms\_output.put\_line('Max: ' || b);  else  dbms\_output.put\_line('Both Equal');  end if;  end; |
|  |  |
| Output: | |

**13. Check for Odd or Even**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | --Script to find odd or even.  declare  a number:=&Num;  r number;  begin  if a=0 then  dbms\_output.put\_line('Zero');  else  r:=mod(a,2);  if r=0 then  dbms\_output.put\_line('Even');  else  dbms\_output.put\_line('Odd');  end if;  end if;  end; |
|  |  |
| Output: | |

**14. Use of CASE statement**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | --Script to use of case statement.  DECLARE    c\_grade CHAR( 1 );    c\_rank  VARCHAR2( 20 );  BEGIN    c\_grade := 'B';    CASE c\_grade    WHEN 'A' THEN      c\_rank := 'Excellent' ;    WHEN 'B' THEN      c\_rank := 'Very Good' ;    WHEN 'C' THEN      c\_rank := 'Good' ;    WHEN 'D' THEN      c\_rank := 'Fair' ;    WHEN 'F' THEN      c\_rank := 'Poor' ;    ELSE      c\_rank := 'No such grade' ;    END CASE;    DBMS\_OUTPUT.PUT\_LINE( c\_rank );  END; |
|  |  |
| Output: | |

**15. Calculate factorial using for loop**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | --Calculate factorial using for loop.  declare  f number:=1;  n number:=&Num;  i number;  begin  for i in 1..n  loop  f:=f\*i;  end loop;  dbms\_output.put\_line('Factorial is '||f);  end; |
|  |  |
| Output: | |

**16. Calculate factorial using while loop**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | --Calculate factorial using while loop.  declare  f number:=1;  n number:=&Num;  i number:=1;  begin  while i<=n  loop  f:=f\*i;  i:=i+1;  end loop;  dbms\_output.put\_line('Factorial is '||f);  end; |
|  |  |
| Output: | |

**17. Use of select into statement**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | --Use if select into statement...  declare  no dept.deptno%type;  dnm dept.dname%type;  l dept.loc%type;  begin  select deptno,dname,loc into no,dnm,l  from dept  where deptno=&no;  dbms\_output.put\_line('No: '||no);  dbms\_output.put\_line('Name: '||dnm);  dbms\_output.put\_line('Loc: '||l);  end; |
|  |  |
| Output: | |

**18. Use of %rowtype**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | --Use of %rowtype attribute...  declare  dp dept%rowtype;  begin  select \* into dp  from dept  where deptno=&no;  dbms\_output.put\_line('No: '||dp.deptno);  dbms\_output.put\_line('Name: '||dp.dname);  dbms\_output.put\_line('Loc: '||dp.loc);  end; |
|  |  |
| Output: | |

**19. Use of cursor**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | --Script to use of cursor...  declare  cursor dept\_data is select \* from dept;  dept\_rec dept%rowtype;  begin  open dept\_data;  loop  fetch dept\_data into dept\_rec;  exit when dept\_data%notfound;  dbms\_output.put\_line('Sr. No.: '||dept\_data%rowcount);  dbms\_output.put\_line('Deptno: '||dept\_rec.deptno);  dbms\_output.put\_line('Dname: '||dept\_rec.dname);  dbms\_output.put\_line('Loc: '||dept\_rec.loc);  dbms\_output.put\_line('----------------');  end loop;  close dept\_data;  end; |
|  |  |
| Output: | |

**20. Use of cursor with for loop**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  10  11  12  13  14 | --Script to use of cursor with for..loop  declare  cursor dept\_data is select \* from dept;  dept\_rec dept%rowtype;  begin  for dept\_rec in dept\_data  loop  dbms\_output.put\_line('Deptno: '||dept\_rec.deptno);  dbms\_output.put\_line('Dname: '||dept\_rec.dname);  dbms\_output.put\_line('Loc: '||dept\_rec.loc);  dbms\_output.put\_line('----------------');  end loop;  end; |
|  |  |
| Output: | |

**21. Exception Handaling**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  10  11  12  13  14  15 | --Script for exception handaling  declare  n dept.deptno%type;  dept\_rec dept%rowtype;  begin  select \* into dept\_rec from dept  where deptno=&n;  dbms\_output.put\_line('Dname: '||dept\_rec.dname);  dbms\_output.put\_line('loc: '||dept\_rec.loc);  exception  when no\_data\_found then    dbms\_output.put\_line('No such Department');  when too\_many\_rows then    dbms\_output.put\_line('Query return multiple rows');  end; |
|  |  |
| Output: | |

**22. Creating procedure**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  10  11 | --Create procedure to test odd-even value...  CREATE OR REPLACE PROCEDURE OddEven (n in NUMBER) is    r number;  BEGIN    r:=mod(n,2);    if r=0 then  dbms\_output.put\_line('Even');    else  dbms\_output.put\_line('Odd');    end if;  END; |
|  |  |
| Output: | |

**23. Creating function**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  10  11  12  13  14 | --Create function to calculate factorial of given no.  create or replace function myfact (n in number) return number  is  f number:=1;  begin  for i in 1..n  loop  f:=f\*i;  end loop;  return f;  end; |
|  |  |
| Output: | |

**Unit – 5**

**Oracle Database Structure and Storage Database, Resource Management and Task Scheduling**

.