|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  |  |  |
|  |  |  |  |
|  |  | | |
|  |  | | |
|  | | | |
| 1. **Overview and Installation of Oracle 10g Software or livesql.oracle.com Account Creation.**   **(Screen shot of Installation or Account Created)** | | | |
| **Link:**  [**https://vsitedumy.sharepoint.com/:u:/g/personal/rohini\_desai\_vsit\_edu\_in2/EUnbzbpfd5ZPtyUcBXjPZ7ABNHMI2IiTK6ToI2Somv9new?e=SdJaoc**](https://vsitedumy.sharepoint.com/:u:/g/personal/rohini_desai_vsit_edu_in2/EUnbzbpfd5ZPtyUcBXjPZ7ABNHMI2IiTK6ToI2Somv9new?e=SdJaoc) | | | |
| 1. **Case Study on Employee Database**   **Write Queries with Output as Screen shots for the following Questions Based on above Table.**   |  |  | | --- | --- | | **Employee** | | | **Attribute** | **Datatype** | | **Empid** | **Number** | | **Empname** | **Varchar2(15)** | | **Manager\_id** | **Number** | | **Dept\_id** | **Number** | | **Salary** | **Number** |  1. **Create the table for Employee Database.** 2. **Describe Employee table with its attributes with its datatypes.** 3. **Add a new column phone number in the Employee table.** 4. **Change the size of Empname field from 15 to 50.** 5. **Remove the column salary from the Employee table** 6. **Change the table name Employee to Emp\_Details.** 7. **Delete all the rows from Emp\_Details table.** 8. **Delete the entire table Emp\_Details from the database.** | | | |
| **CREATE TABLE Empolyee1(**  **EmpID Number,**  **EmpName varchar(215),**  **ManageID Number,**  **DeptID Number,**  **Salary Number**  **);**    **DESC Empolyee1;**    **alter TABLE Empolyee1 add PhoneNo numbe**  **alter TABLE Empolyee1 modify EmpName varchar(250)**  **alter TABLE Empolyee1 drop column Salary**  **rename Empolyee1 to EmpDetails**    **DESC EmpDetails;**    **truncate table EmpDetails**  **drop table EmpDetails** | | | |
| 1. **Case Study on Client\_Master Database**  |  |  |  | | --- | --- | --- | | **Column Name​** | **Data Type​** | **Size​** | | CLIENTNO​ | Char​ | 6​ | | NAME​ | Char​ | 20​ | | ADDRESS​ | Char​ | 20​ | | CITY​ | Char​ | 15​ | | PINCODE​ | Number​ | 6​ | | STATE​ | Char​ | 15​ | | BALDUE​ | Number​ | 10,2​ |   **Write Queries with Output as Screen shots for the following Questions Based on above Table.**   1. **Create the table Client\_Master database.** 2. **Describe the structure of Client\_Master table.** 3. **Add a new column phone number in the Client\_Master table.** 4. **Change the size of Address field from 20 to 70.** 5. **Remove the column Pincode from the Client\_Master table.** 6. **Delete entire table Client\_Master from the database.** | | | |
| **CREATE TABLE cLIENT\_MASTER(**  **ClientNo char,**  **Name char,**  **Address char,**  **City char,**  **pincode number,**  **state char,**  **baldue number**  **);**    **DESC Client\_Master;**    **alter TABLE Client\_Master add PhoneNo numbe**  **alter TABLE Client\_Master modify Address char(70)**  **alter TABLE Client\_Master drop column pincode**    **drop table Client\_Master** | | | |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Q. 1**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ClientNo** | **Cname** | **CCity** | **State** | **BalDue** | | **C001** | **Ivan b** | **Mumbai** | **Maharashtra** | **1500** | | **C002** | **Mamta** | **Pune** | **Maharashtra** | **2500** | | **C003** | **Ram** | **Navi Mumi** | **Maharashtra** | **5000** | | **C004** | **Shyam** | **Banglore** | **Aandhra Pradesh** | **2000** | | **C005** | **Raj** | **Surat** | **Gujrat** | **0** | | **C006** | **Leena** | **Jaipur** | **Rajsthan** | **2600** | | **C007** | **Neena** | **Madras** | **Aandhra Pradesh** | **2600** | | **C008** | **Nita** | **Navi Mumbai** | **Maharashtra** | **3000** |   **Write the Queries for the following questions and paste the output as screenshots based on above Client\_Master table.**     1. **Write create table query for above Client\_master database.** 2. **Write insert query for to insert the above given records into Client\_master table.** 3. **Display information of all clients present in the Client\_Master table.** 4. **Display Clients name and city of all clients.** 5. **Change the city of ClientNo 'C005' to 'Banglore‘** 6. **Display detalis of client no 6.** 7. **Add column Pincode to above table.** 8. **Change size of the State column to 30.** 9. **Delete the details of client having client no as C008.** 10. **Delete the details of clients staying in the city of Maharashtra.** | | | | | **Program Code:**  CREATE TABLE ClientMaster(  ClientNO varchar(100),  Client\_Name varchar(20),  Client\_City varchar(30),  State varchar(40),  BalDue int  );    DESC ClientMaster    insert into ClientMaster values('C001', 'Ivan B', 'Mumbai', 'Maharashtra', 1500)  insert into ClientMaster values('C002', 'Mamta', 'Pune', 'Maharashtra', 2500)  insert into ClientMaster values('C003', 'Ram', 'Navi Mumbai', 'Maharashtra', 5000)  insert into ClientMaster values('C004', 'Shyam', 'Banglore', 'Aandhra Pradesh', 2000)  insert into ClientMaster values('C005', 'Raj', 'Surat', 'Gujrat', 0)  insert into ClientMaster values('C006', 'Leena', 'Jaipur', 'Rajsthan', 2600)  insert into ClientMaster values('C007', 'Neena', 'Madars', 'Aandhra Pradesh', 2600)  insert into ClientMaster values('C008', 'Nita', 'Navi Mumbai', 'Maharashtra', 3000)    select \* from ClientMaster    select Client\_Name, Client\_City from ClientMaster  update ClientMaster set Client\_City='Banglore' where ClientNo='C005'  select 'C006' from ClientMaster  alter TABLE ClientMaster add PinCode Number  alter TABLE ClientMaster modify State varchar(30)  delete from ClientMaster where ClientNo='C008'  delete from ClientMaster where state='Maharashtra'  **Program Output:** | | | | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | empno | ename | Job | salary | Deptid | | 1 | John | Clerk | 1000 | 10 | | 2 | Maria | Manager | 1400 | 20 | | 3 | Lara | President | 2000 | 30 | | 4 | Fedrick | Officer | 1300 | 40 | | 5 | Meera | Clerk | 1000 | 10 | | 6 | Anita | Manager | 1400 | 10 |   **Write the Queries for the following questions and paste the output as screenshots based on above Employee table.**   1. **Write a query to create table Employee.** 2. **Display the structure of the table Employee.** 3. **Write a query to insert all values in Employee table.** 4. **Display details of employee table.** 5. **Write a query to change ‘Job’of an Employee No. 1 to ‘Accountant’.** 6. **List the name, job, and salary of all employees in department 20.** 7. **Display the empno, ename and job of all employees.** 8. **Display the details of employees working in deptid=10.** 9. **Update the salary of John to Rs. 2000.** 10. **Delete the details of the emplyees who are working as Clerk.** | | | | | |  | | --- | |  | |  |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Q. 1**  **Table Name: Client\_Master**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | CLIENTNO | char | 6 |  | Primary Key | | NAME | char | 20 |  | Not Null | | ADDRESS | char | 20 |  |  | | CITY | Char | 15 |  |  | | PINCODE | Number | 8 |  | Unique | | STATE | Char | 15 |  |  | | BALDUE | Number | 10,2 |  | Greater than 0 |   **Write create command for above table and insert 5 rows and paste the output as screenshots based on above Client\_Master table.** | | | | | **INPUT :**  **create table Client\_Master(**  **ClientNo varchar(6) Primary key,**  **Name varchar(20) Not null,**  **Address varchar(20),**  **City varchar(15),**  **pincode number(8) unique,**  **State varchar(15),**  **Baldue Number(10,2) CHECK(baldue>0)**  **);**    **desc Client\_master;**  **OUTPUT :** | | | | | Table Name: Product\_Master   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | PRODUCTNO | Char | 6 |  | Primary Key | | DESCRIPTION | Char | 15 |  | Not Null | | QTYONHAND | Number | 8 |  | Not Null | | REORDERLVL | Number | 8 |  | Not Null | | SELLPRICE | Number | 8,2 |  | Not Null | | COSTPRICE | Number | 8,2 |  | Not Null |   **Write create command for above table and insert 5 rows and paste the output as screenshots based on above Product\_Master table.** | | | | | **INPUT :**  create table product\_master(  productNo varchar(6) Primary key,  description varchar(15) Not null,  Qtyonhand number(8) not null,  reorderlvl number(8) not null,  sellprice number(8,2) not null,  costprice number(8,2) not null);  desc product\_master;  select \* from product\_master;  Insert into product\_master values('c002','medicine',9,15,150,120);  Insert into product\_master values('c009','tulsi',12,19,90,50);  Insert into product\_master values('c013','jeera',7,18,70,30);  Insert into product\_master values('c019','chocolate',6,13,120,80);  Insert into product\_master values('c021','icecream',2,16,60,30);  **OUTPUT :**     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Q. 3**  **Table Name: Salesman\_Master**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | PRODUCTNO | Char | 6 |  | Primary Key/First letter must start with ‘P’ | | DESCRIPTION | Char | 15 |  | Not Null | | QTYONHAND | Number | 8 |  | Not Null | | REORDERLVL | Number | 8 |  | Not Null | | SELLPRICE | Number | 8,2 |  | Not Null, cannot be 0 | | COSTPRICE | Number | 8,2 |  | Not Null, cannot be 0 |   **Write create command for above table and insert 5 rows and paste the output as screenshots based on above salesman\_Master table.** |   **INPUT :**  **create table salesman\_master(**  **productNo varchar(6) Primary key,**  **description varchar(15) Not null,**  **Qtyonhand number(8) not null,**  **reorderlvl number(8) not null,**  **sellprice number(8,2) not null,**  **costprice number(8,2) not null);**  **desc salesman\_master;**  **select \* from product\_master;**  **Insert into product\_master values('c002','medicine',9,15,150,120);**  **Insert into product\_master values('c009','tulsi',12,19,90,50);**  **Insert into product\_master values('c013','jeera',7,18,70,30);**  **Insert into product\_master values('c019','chocolate',6,13,120,80);**  **Insert into product\_master values('c021','icecream',2,16,60,30);**  **OUTPUT :** | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Q. 4**  **Table Name: Salesman\_Order**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | ORDERNO | Char | 6 |  | Primary Key/First Letter must start with ‘O’ | | CLIENTNO | Char | 6 |  | Foreign Key references ClientNo of Client\_Master table | | SALESMANNO | Char | 6 |  | Foreign Key references SalesmanNo of Salesman\_Master table | | DALYTYPE | Char | 1 | F |  | | ORDERSTATUS | Char | 10 |  | Values (‘In Process’,’Fulfilled’,’BackOrder’, ‘Cancelled’) |   **Write create command for above table and insert 5 rows and paste the output as screenshots based on above Salesman\_Order table.** |   INPUT :       |  | | --- | |  | | 1. **Case Study on Help Table insert minimum 10 records.**  |  |  | | --- | --- | | **Topic** | **Varchar** | | **Seq** | **INT** | | **Info** | **Varchar** |   **Write Queries with Output as Screen shots for the following Questions Based on above Table.**  **1) Display details of help table. ​**  **2) Display info from help table. ​**  **3) Display seq and info of help whose topic is “append”. ​**  **4) Display details of help whose topic is “attribute”. ​**  **5) Display details of help whose topic is accept and seq is 5. ​**  **6) Display details of help whose topic is “change” or “clear”. ​**  **7) Display details of help whose topic is “copy” or “define” or “del”. ​**  **8) Display details of help whose seq between 3 to 12. ​**  **9) Display details of help whose seq between 1 to 5. ​**  **10) Display details of help in ascending order. ​**  **11) Display details of help in descending order. ​**  **12) Display details of help table whose info starts with “L”. ​**  **13) Display details of help table whose info ends with “s”. ​**  **14) Display details of help table whose topic is NULL. ​**  **15) Display details of help whose topic is “CLEAR” and seq is not “5”.​** | | **1. select \* from help**    2. select \*info from help    3. select \* from help where topic='APPEND'    4. select \* from help where topic='ATTRIBUTE'    5. select \* from help where topic='ACCEPT'and seq=5;    6. select \* from help where topic IN('CHANGE','CLEAR')    7. select \* from help where topic IN('COPY','DEFINE','DEL')    **8. select \* from help where seq between 3 and 12**    9. select \* from help where seq between 1 and 5    10. select \* from help order by seq asc;    11. select \* from help order by seq desc;    12. select \* from help where info like'L%';    13. select \* from help where info like'%s';    14. select \* from help where info is NULL;    15.select \* from help where topic='CLEAR' and NOT seq=5; | |  | |  |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Qn.1 Create following table Employee and Insert the given records to it.**    **Q1. Write Queries with Output as Screen shots for the following Questions Based on above Table.**  **1.Display minimum salary of employee.**  **2.Display maximum salary of employee.**  **3.Display average salary of employee who is working as Manager.**  **4.Calculates sum of salary for each department.**  **5.Lists the maximum salary drawn under each job category.** | | | | | 1. **Display minimum salary of employee**   **SELECT MIN(salary) as minimium\_salary FROM public.Employee;**     1. **Display maximum salary of employee.**   **SELECT MAX(salary) as maximum\_salary FROM public.Employee;**     1. **Display average salary of employee who is working as Manager.**   **SELECT AVG(salary) as average\_salary\_manager FROM public.Employee WHERE job = 'Manager';**     1. **Calculate sum of salary for each department**   **SELECT job, SUM(salary) FROM public.Employee GROUP BY job;**     1. **Lists the maximum salary drawn under each job category.**   **SELECT job, MAX(salary) FROM public.Employee GROUP BY job;** | | | | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Qn.1 Create following two tables Employee and Department. Insert the given records to it.**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Empno | ename | Job | | Deptno | Sal | | 101 | King | President | | 10 | 20000 | | 102 | Blake | Manager | | 10 | 12000 | | 103 | Clark | Manager | | 10 | 11000 | | 104 | Jones | Manager | | 20 | 10000 | | 105 | Scott | Analyst | | 20 | 5000 | | 106 | Ford | Analyst | | 30 | 6000 | | 107 | Allen | Salesman | | 40 | 950 | | 108 | Turner | Salesman | | 20 | 1100 | | 109 | James | Salesman | | 10 | 1500 | | 110 | Jimmy | President | | 30 | 25000 | | 111 | Menon | Salesman | | 50 | 1200 | | 112 | Killer | Analyst | | 50 | 450 | | Deptno | Dname | | | 10 | Accounts | | | 20 | Research | | | 30 | Sales | | | 40 | Marketing | |   **Write Queries with Output as Screen shots for the following Questions Based on above Table.**   1. **Write a query using a join to determine which employee is working in which department.** 2. **Write query using cross join to join Employee and Department table.** 3. **Display empno, ename, deptno from Employee and dname from Department using left join.** 4. **Display empno, ename, deptno from Employee and dname from Department using right join.** 5. **Display empno, ename, deptno from Employee and dname from Department using full join.** | | | | | **create table emp33(Empno int,ename varchar2(250),job varchar2(250),Deptno int,Sal int)**  **desc emp33**  **insert into emp33 values ('101','King','President',10,20000)**  **insert into emp33 values ('102','Blake','Manager',10,12000)**  **insert into emp33 values ('103','clark','Manager',10,11000)**  **insert into emp33 values ('104','Jones','Manager',20,10000)**  **insert into emp33 values ('105','Scott','Analyst',20,5000)**  **insert into emp33 values ('106','Ford','Analyst',30,6000)**  **insert into emp33 values ('107','Allen','Saleman',40,950)**  **insert into emp33 values('108','Turner','Salesman',20,1100)**  **insert into emp33 values('109','James','Salesman',10,1500)**  **insert into emp33 values('110','Jimmy','President',30,25000)**  **insert into emp33 values('111','Menon','Salesman',50,1200)**  **insert into emp33 values('112','Killer','Analyst',50,450)**  **select \* from emp33**  **create table Dep(Deptno int,Dname varchar2(250))**  **desc Dep**  **insert into Dep values(10,'Accounts')**  **insert into Dep values(20,'Research')**  **insert into Dep values(30,'Sales')**  **insert into Dep values(40,'Marketing')**  **Select \* from Dep**  **select \* from emp33 natural join Dep;**  **select \* from emp33 cross join Dep;**  **select \* from emp33 left outer join Dep on (emp33.Deptno=Dep.Deptno);**  **select \* from emp33 right outer join Dep on (emp33.Deptno=Dep.Deptno);**  **select \* from emp33 full outer join Dep on (emp33.Deptno=Dep.Deptno);**      **1.**  2.  3.  4.  5. | | | | | **Qn.2 Create following table Employee and Insert the given records to it.**  **Write Queries with Output as Screen shots for the following Questions Based on above Table.**   1. **Write query to find the salaries for all the employees who have a higher salary than Dev.** 2. **Write query to find the second highest salaries from all the employees.** 3. **Display details of an employee whose salary is greater than avg salary of employees who work as analyst.** 4. **Display the name of the employees whose salary is greater than the employee with empno 8.** 5. **Display empno, ename and job of the employees having salary less than manager.** | | | | | create table employee( empid number, ename varchar(40), job varchar(40), Hiredate date, salary number );  select \* from student  insert into employee values(1,'Yogesh','AssiManager','12-Jun-18',50000) insert into employee values(2,'Chinmay','Analyst','12-Jan-18',5000) insert into employee values(3,'Dev','Analyst','12-Feb-18',20000) insert into employee values(4,'Saurabh','Salesman','12-apr-18',5000) insert into employee values(5,'Aadarsh','Peon','12-Mar-18',500) insert into employee values(6,'Sahil','Salesman','12-May-18',20000) insert into employee values(7,'Mansoor','Manager','12-Jun-18',50000) insert into employee values(8,'Yugesh','Assistant','12-Jun-18',50000) insert into employee values(9,'Sahiith','Leader','12-Jun-18',450000) insert into employee values(10,'Sagar','Projleader','12-Jun-18',35000) drop table employee  select \* from employee  Q1. select \* from employee where salary>(select salary from employee where ename='Dev')  2. select \* from employee where salary<(select salary from employee where job='Leader')    3.  select \* from employee where salary>(select avg(salary) from employee where job='Analyst')  Inserting image...  4. select ename from employee where salary>(select salary from employee where empid=8)    5.  select empid,ename,job from employee where salary>(select salary from employee where job='Manager') | | | | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Case Study Employee Table**    **Project Table**    **Write Queries with Output as Screen shots for the following Questions Based on above Table.**   1. **Write a query to create a view for those whose age is greater than 30.** 2. **Write a query to create a view for all Employee with columns emp\_name,salary,age.​** 3. **Write a query to create a view for Project with emp\_id is 3.** 4. **Write a query to create a view for Project with department is ‘Development’.** 5. **Write a query to create a view for Employee with salary not equal to 200000.** 6. **Write a query to create a view whose fourth letter of names is ‘s’ in Employee.** 7. **Write a query to create a read only view to display the emp\_name and city of Employee. ​** 8. **Write a query to create a horizontal view on employee.** 9. **Write a query to create a vertical view on employee.** 10. **Write a query to create a Joined view on employee.** 11. **Write a query to create a Group view on employee.** 12. **Write a query to create a Row and Colum Subset view on employee.** | | | | | **create table employee(**  **empid int,**  **emp\_name char (20),**  **city char(20),**  **salary int,**  **age int);**    **insert into employee values(1,'Angelina','Chicago',200000,30);**  **insert into employee values(2,'Robert','Austin',300000,26);**  **insert into employee values(3,'Christian','Denver',100000,42);**  **insert into employee values(4,'Kristen','Washington',500000,29);**  **insert into employee values(5,'Russell','Los Angels',200000,36);**  **insert into employee values(6,'Marry','Canada',600000,48);**    **select\*from employee**    **create table project(**  **project\_no int,**  **emp\_id int,**  **department char(20));**  **insert into project values(101,1,'Testing');**  **insert into project values(102,2,'Development');**  **insert into project values(103,3,'Designing');**  **insert into project values(104,4,'Development');**  **select\*from project**    **create view emp\_view as select\* from employee where age>30**  **create view emp\_view1 as select emp\_name,salary,age from employee**  **create view project\_view as select\*from project where emp\_id=3**  **create view project\_view1 as select\*from project where department='Development'**  **create view emp\_view2 as select\*from employee where salary!=200000**  **create view emp\_view3 as select\*from employee where emp\_name like '\_\_\_s%'**  **create view emp\_view4 as (select emp\_name ,city from employee) with read only**  **create view emp\_view5 as select\*from employee where salary=200000**  **create view emp\_view6 as select emp\_name,salary from employee**  **create view joinview as select\*from employee,project where employee.empid=project.emp\_id**  **create view groupview as (select department from project group by department)**  **create view subsetview as (select emp\_name,salary from employee where salary>300000)**    **select\*from emp\_view**  **select\*from emp\_view1**  **select\*from project\_view**  **select\*from project\_view1**  **select\*from emp\_view2**  **select\*from emp\_view3**  **select\*from emp\_view4**  **select\*from emp\_view5**  **select\*from emp\_view6**  **select\*from joinview**  **select\*from groupview**  **select\*from subsetview**    **drop view emp\_view3**  **drop table employee**  1.  2.  3.  4.  **5.**  6.  7.  8.  9.  10.  11.  12. | | | | | **Q. 2 Create following two tables called Employee and Department.**   |  |  |  |  | | --- | --- | --- | --- | | **Eid** | **ename** | **esal** | **Did** | | **1** | **A** | **20000** | **10** | | **2** | **B** | **30000** | **20** | | **3** | **C** | **40000** | **20** | | **4** | **D** | **50000** | **30** |  |  |  | | --- | --- | | **Did** | **Dname** | | **10** | **P** | | **20** | **Q** | | **30** | **R** | | **40** | **S** |   **Perform the SET operations on the above given table.**  **create table employee4(**  **eid int,**  **ename char(20),**  **esal int,**  **did int);**    **insert into employee4 values(1,'A',20000,10)**  **insert into employee4 values(2,'B',30000,20)**  **insert into employee4 values(3,'C',40000,20)**  **insert into employee4 values(4,'D',50000,30)**  **select\*from employee4**    **create table department(**  **did int,**  **dname char(20));**    **insert into department values(10,'P')**  **insert into department values(20,'Q')**  **insert into department values(30,'R')**  **insert into department values(40,'S')**    **select\*from department**    **1.union**    **2.union All**    **3.intersect**    **4.minus** | | | | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Write PL/SQL program with Output as Screen shots for the following Questions.**   1. Write a PL/SQL block to check whether the number entered by the user is even or odd. 2. Write a PLSQL block to find factorial of a number given by the user. 3. Write a PL/SQL block to check whether the given number is positive or negative or neutral. 4. Write a PL/SQL block to check prime number or not. 5. Write a PLSQL block Performing Arithmetic Operations accepting value from user. 6. Write a PLSQL block find the grade of Student using CASE statement. 7. Write a PLSQL block to print number between 1 to 10 using for, while, loop. | | | | | |  | | --- | | **1. DECLARE**  **n NUMBER := 1634;**  **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 :**    **2. set serveroutput on**  **declare**  **num number := 6;**  **fact number := 1;**  **temp number;**  **begin**  **temp :=num;**  **while( temp>0 )**  **loop**  **fact := fact\*temp;**  **temp := temp-1;**  **end loop;**  **dbms\_output.put\_line('factorial of '|| num || ' is ' || fact);**  **end;**  **OUTPUT:**    **3. set serveroutput on**  **DECLARE**  **num1 NUMBER := &get\_num;**  **BEGIN**  **IF num1 < 0 THEN**  **DBMS\_OUTPUT.PUT\_LINE ('The number '||num1||' is a negative number');**  **ELSIF num1 = 0 THEN**  **DBMS\_OUTPUT.PUT\_LINE ('The number '||num1||' is equal to zero');**  **ELSE**  **DBMS\_OUTPUT.PUT\_LINE ('The number '||num1||' is a positive number');**  **END IF;**  **END;**  **/**  **OUTPUT :**  **4. declare**  **n number;**  **i number;**  **temp number;**  **begin**  **n := 13;**  **i := 2;**  **temp := 1;**  **for i in 2..n/2**  **loop**  **if mod(n, i) = 0**  **then**  **temp := 0;**  **exit;**  **end if;**  **end loop;**    **if temp = 1**  **then**  **dbms\_output.put\_line('true');**  **else**  **dbms\_output.put\_line('false');**  **end if;**  **end;**    **OUTPUT :**    **5. set serveroutput on**  **BEGIN**  **dbms\_output.put\_line( 10 + 5);**  **dbms\_output.put\_line( 10 - 5);**  **dbms\_output.put\_line( 10 \* 5);**  **dbms\_output.put\_line( 10 / 5);**  **dbms\_output.put\_line( 10 \*\* 5);**  **END;**  **/**    **OUTPUT :**    **6. Set serveroutput on**  **declare**  **per number;**  **Begin**  **per:=&per;**  **case per**  **when 8 then DBMS\_OUTPUT.PUT\_LINE('Your grade: A');**  **when 7 then DBMS\_OUTPUT.PUT\_LINE('Your grade: B');**  **when 6 then DBMS\_OUTPUT.PUT\_LINE('Your grade: C');**  **when 5 then DBMS\_OUTPUT.PUT\_LINE('Your grade: D');**  **else**  **DBMS\_OUTPUT.PUT\_LINE('Your grade: F');**  **end case;**  **end;**  **OUTPUT:**  **7. Set serveroutput on**  **Declare**  **I number;**  **Begin**  **I:=1;**  **While (I<=10)**  **loop**  **Dbms\_output.put\_line(I);**  **I:=I+1;**  **End loop;**  **End;**  **OUTPUT :** | | | | | |  | | | | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Write PL/SQL program with Output as Screen shots for the following Questions.**   1. Write a procedure to add two numbers and call that procedure through the PL/SQL block. 2. Write a PL/SQL procedure that compute square of value. 3. Write a PL/SQL function which will compute and return the maximum of two values 4. Write a function to add two numbers and call that function through the PL/SQL block | | | | | **1.** CREATE PROCEDURE ADD\_TWO\_NOS2  IS  A NUMBER(3):=10; B NUMBER(3):=20; C NUMBER(5);  BEGIN  C:=A+B;  DBMS\_OUTPUT.PUT\_LINE('SUM OF ' || A || ' AND ' || B || ' IS ' || C);  END;  /  SET SERVEROUTPUT ON  BEGIN  ADD\_TWO\_NOS2;  END;  /  **OUTPUT :**  **2.** CREATE PROCEDURE SQUARE\_NO  IS  A NUMBER(3):=4;  C NUMBER(5);  BEGIN  C:=A\*A;  DBMS\_OUTPUT.PUT\_LINE('SQUARE OF NUMBER IS:'||C);  END;  /    SET SERVEROUTPUT ON  BEGIN  SQUARE\_NO;  END;  /  **OUTPUT :**  **3.** Create FUNCTION  findMax(x IN number, y IN number) RETURN number IS z number;  BEGIN  IF x > y  THEN z:= x;  ELSE  z:= y;  END IF;  RETURN z;  end;    DECLARE  a number; b number; c number;  BEGIN  a:=&a; b:= &b;  c := findMax(a, b);  dbms\_output.put\_line(' Maximum of (a, b): ' || c);  END;  **OUTPUT :**  **4.** CREATE FUNCTION ADD\_TWO\_NOS1(A IN NUMBER,B IN NUMBER) RETURN NUMBER IS  C NUMBER(3);  BEGIN  C:=A+B; RETURN C;  END;  /    SET SERVEROUTPUT ON  DECLARE  X NUMBER(3); Y NUMBER(3); Z NUMBER(3);  BEGIN  X:=&X; Y:=&Y;  Z:=ADD\_TWO\_NOS1(X,Y);  DBMS\_OUTPUT.PUT\_LINE('VALUE OF Z IS '||Z);  END;  /  **OUTPUT :** | | | | |  | | | | |  | | | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Perform following queries on Banking Database using the given table.**  **Consider the following database:-**  **I) Account (Branch\_name, Account\_number, Balance)**  **ii)Branch(Branch\_name, Branch\_city, Assets)**  **iii)Customer(Cust\_name, Cust\_street, Cust\_city)**  **iv)Depositor(Cust\_name, Account\_number)**  **v)Loan(Branch\_name, Loan\_number, Amount)**  **vi)Borrower(Cust\_name, Loan\_number)**  **Write the queries for the following.**  **1.Find all loan number for loans made at the Mumbai branch with loan amounts greater than 1200.**  **2.Find the names and loan numbers of all customers who have a loan at the Mumbai branch.**  **3.Find the names of all customers whose street address includes the substring ‘Main’.**  **4.Find the average balance of each customer who lives in Harrison and has at least 3 accounts.**  **5.Find all customers who have both an account and a loan at the Mumbai branch.**  **6.Find the names of all branches that have assets greater than that of each branch in Brooklyn.**  **7.Find the branch that has the higher average balance.** | | | | | create table account(  Branch\_name varchar(40),  Account\_number int,  Balance int);  desc account;    **create table Branch(**  **branch\_name varchar(200) primary key,**  **branch\_city varchar(30),**  **Assets number)**  **desc Branch;**    create table Customer(  Cust\_name varchar(30) primary key,  Cust\_street varchar(300),  Cust\_city varchar(30))  desc Customer;    **create table Depositor(**  **Cust\_name char(50),**  **Account\_number int)**  **desc Depositor**    **create table Loan(**  **Branch\_name char(50),**  **Loan\_number int,**  **Amount int)**  **desc Loan**    create table Borrower(  Cust\_name char(50),  Loan\_number int)  desc Borrower     1. **create table account8(**   **Branch\_name varchar(40),**  **Account\_number int,**  **Balance int);**  **desc account8**  **insert into account8 values('Dadar',1,1000)**  **insert into account8 values('wadala',2,9000)**  **insert into account8 values('chembur',2,8000)**  **insert into account8 values('vashi',3,6000)**  **insert into account8 values('panvel',4,5000)**  **insert into account8 values('virar',5,1000)**  **insert into account8 values('nerul',6,11000)**  **insert into account8 values('worli',7,8000)**  **insert into account8 values('parel',8,4000)**  **insert into account8 values('Lower parel',9,5000)**    **select \* from account8**    **2. create table Branch(**  **Branch\_name varchar(40),**  **Account\_number int,**  **Balance int);**  **desc Branch**  **insert into Branch values('Dadar',1,1000)**  **insert into Branch values('wadala',2,9000)**  **insert into Branch values('chembur',2,8000)**  **insert into Branch values('vashi',3,6000)**  **insert into Branch values('panvel',4,5000)**  **insert into Branch values('virar',5,1000)**  **insert into Branch values('nerul',6,11000)**  **insert into Branch values('worli',7,8000)**  **insert into Branch values('parel',8,4000)**  **insert into Branch values('Lower parel',9,5000)**    **select \* from Branch** | | | | | **3.**    **4.**    **5.**    **7.** | | | |  |  |  |  |  | | --- | --- | --- | --- | |  |  | | | |  |  |  |  | |  |  |  |  | |  |  | | | |  |  | | | |  | | | | | **Write PL/SQL program with Output as Screen shots for the following Questions.**   1. Write a PL/SQL block to print the sum of individual digits of a given number and check whether the given number is Armstrong or not. (Use simple/basic loop) 2. Write a PL/SQL block to check whether the given number is positive or negative or neutral. Also find factorial value of the entered number. 3. Write a PL/SQL block that creates procedure to perform Volume of Cylinder by accepting radius and height as input parameters and store the result in an output parameter. The output parameter is to be printed at the time of execution. 4. Write two procedures area\_rect () and perimeter-rect() that accept length and breadth of rectangle as parameters and calculate area and perimeter of the rectangle respectively. 5. Write a PL/SQLblock to find the sum of square of n numbers and the sum of the individual digits of a given number. (Use while loop) 6. Write a PL/SQLblock to find the sum of cube of n numbers and the reverse of a given number. (Use basic/simple loop). | | | | | **1.** declare  n number:=407;  s number:=0;  r number;  len number;  m number;  begin  m:=n;  len:=length(to\_char(n));  while n>0  loop  r:=mod(n,10);  s:=s+power(r,len);  n:=trunc(n/10);  end loop;  if m=s  then  dbms\_output.put\_line('armstrong number');  else  dbms\_output.put\_line('not armstrong number');  end if;  end;  /    **2.** DECLARE  NO NUMBER;  BEGIN NO:=&NO;  IF NO < 0 THEN  DBMS\_OUTPUT.PUT\_LINE('NEGATIVE NUMBER');  ELSIF NO > 0 THEN  DBMS\_OUTPUT.PUT\_LINE('POSITIVE NUMBER');  ELSE  DBMS\_OUTPUT.PUT\_LINE('EQUAL TO ZERO');  END IF;  END;  /    **3. CREATE OR REPLACE PROCEDURE VOL(R IN NUMBER,H IN NUMBER)**  **IS**  **V NUMBER;**  **BEGIN**  **V:=3.14\*R\*R\*H;**  **DBMS\_OUTPUT.PUT\_LINE('VOL OF CYLINDER IS ' || V);**  **END;**  **/**  **DECLARE**  **RADIUS NUMBER; HEIGHT NUMBER;**  **BEGIN**  **RADIUS:=&RADIUS; HEIGHT:=&HEIGHT;**  **VOL(RADIUS,HEIGHT);**  **END;**  **/**    **4.** CREATE OR REPLACE PROCEDURE AR(L IN NUMBER,B IN NUMBER)  IS  A NUMBER;  BEGIN  A:=L\*B;  DBMS\_OUTPUT.PUT\_LINE('AREA OF RECTANGLE IS ' || A);  END;  /  CREATE OR REPLACE PROCEDURE PER(LE IN NUMBER,BR IN NUMBER)  IS  P NUMBER;  BEGIN  P:=2\*LE+2\*BR;  DBMS\_OUTPUT.PUT\_LINE('PERIMETER OF RECTANGLE IS ' || P);  END;  /  DECLARE  LENGTH NUMBER; BREADTH NUMBER;  BEGIN  LENGTH:=&LENGTH; BREADTH:=&BREADTH;  AR(LENGTH,BREADTH);  PER(LENGTH,BREADTH);  END;  /    **5.** declare  n number;   s number:=0;   r number:=0;   m number;   len number;  c number:=1;  begin  n:=&n;  m:=n;   len:=length(to\_char(n));   while n>0   loop   r:=mod(n,10);   s:=s+power(r,2);   n:=trunc(n/10);   end loop;     dbms\_output.put\_line('sum of ' || m || ' = ' || s);    while(c <= m)  loop  r:=r+power(c,2);  c:= c+1;  end loop;    dbms\_output.put\_line('sum of square of numbers upto ' || m || ' = ' || r);   end;  /    **6.** declare  n number;   s number:=0;   r number:=0;   m number;   len number;  c number:=1;  sum\_of\_cube number := 0;  begin  n:=&n;  m:=n;   len:=length(to\_char(n));     loop   r:=mod(n,10);   s:=(s \* 10) + r;   n:=trunc(n/10);   exit when n <= 0;  end loop;     dbms\_output.put\_line('Reverse of ' || m || ' = ' || s);  loop  sum\_of\_cube:=sum\_of\_cube+power(c,3);  c:= c+1;  exit when c > m;  end loop;  dbms\_output.put\_line('sum of cube of numbers upto ' || m || ' = ' || sum\_of\_cube);    end;  / | | | | |  | | | | |  | | | | | | | | | | | | |  | | | | | | | |
|  | | | |