1. Create a table customer (cust\_novarchar(5), cust\_namevarchar(15), age number, phone varchar(10) )
2. insert 5 records and display it
3. add new field d\_birth with date datatype
4. create another table cust\_phone with fields cust\_name and phone from customer table
5. remove the field age
6. change the size of the cust\_name to 25
7. delete all the records from the table
8. rename the table cutomer to cust
9. drop the table

create table customer(custno int,custname varchar2(20),phone int,age int);

1. insert 5 records and display it

insert into customer(custno,custname,phone,age) values(&custno,'&custname', '&phone',&age);

1. add new field d\_birth with date datatype

alter table customer add dbirth date

1. create another table cust\_phone with fields cust\_name and phone from customer table

create table custphone as(select custname,phone from customer);

1. remove the field age

alter table customer drop column age;

1. change the size of the cust\_name to 25

ALTER TABLE customer

MODIFY(custname VARCHAR2(25));

1. delete all the records from the table

delete from customer;

1. rename the table cutomer to cust

ALTER TABLE customer RENAME TO cust

1. drop the table

Drop table cust

1. Create a table sales\_man ( salesman\_no primary key, s\_name not null, place, phone unique)

Create table sales\_order(order\_no primary key

order\_date not null

salesman\_no foreign key references salesman\_no in sales\_man

del\_type values should be either P or F (check constraints)

order\_status values should be 'Inprocess','Fullfilled','Backorder', 'Cancelled' (check constraints) )

1. Insert few records in both tables
2. Delete primary key from sales\_man table
3. Delete Foreign key and Check constraints from sales\_order table
4. Add primary key in sales\_man using ALTER TABLE
5. Add foreign key and CHECK constraints in sales\_order table using ALTER TABLE

Create table salesman(salesno int not null,

Sname varchar2(10) not null,

place varchar2(20),

phone int unique,

constraint salesman\_pk primary key(salesno));

Create table salesorder(orderno int primary key,

Orderdate date not null,

Salesno int,

Deltype char(1) check(deltype in(‘p’,’f’)),

Orderstatus varchar2(20) check(Orderstatus in(‘inprocess’,’fullfilled’,’backorder’,’cancelled’)),

Constraint fk\_salesorder FOREIGN KEY (Salesno) REFERENCES salesman(Salesno));

Alter table salesorder drop constraint fk\_salesorder;

1. Insert few records in both tables

Insert into salesorder values(&orderno,&orderdate,&salesno,’&deltype’,’&orderstatus’);

Insert into salesorder values(10, ’28-oct-1984’,12,’f’,’fullfilled’);

Insert into salesorder values(11, ’30-oct-2016’,13,’f’,’fullfilled’);

Insert into salesorder values(10, ’28-oct-1984’,13,’p’,’cancelled’);

Delete primary key from sales\_man table

Alter table salesman drop primary key;

Delete Foreign key and Check constraints from sales\_order table

Alter table salesorder drop constraint fk\_salesorder;

Add primary key in sales\_man using ALTER TABLE

Alter table salesman add primary key (salesno);

Add foreign key and CHECK constraints in sales\_order table using ALTER TABLE

Alter table salesorder add FOREIGN KEY (Salesno) REFERENCES salesman(Salesno);

1. Create a table Hospital with the fields (doctorid,doctorname,department,qualification,experience).

Write the queries to perform the following.

1. Insert 5 records
2. Display the details of Doctors
3. Display the details of doctors who have the qualification ‘MD’
4. Display all doctors who have more than 5 years experience but do not have the qualification ‘MD’
5. Display the doctors in ‘Skin’ department
6. update the experience of doctor with doctored=’D003’ to 5
7. Delete the doctor with DoctorID=’D005’

1. Create the following tables

Bank\_customer (accno primary key, cust\_name,place)

Deposit (accno foreignkey, deposit\_no, damount)

Loan (accno foreign key loan\_no, Lamount)

Write the following queries

1. Display the details of the customers

select \* from bankcustomer;

1. Display the customers along with deposit amount who have only deposit with the bank

select custname,accno from bankcustomer where accno in (select accno from deposit where accno not in(select accno from loan));

1. Display the customers along with loan amount who have only loan with the bank

Select bankcustomer.custname,loan.lamount from bankcustomer inner join loan on

bankcustomer.accno=loan.accno ;

1. Display the customers they have both loan and deposit with the bank

Select deposit.accno,loan.lamount from deposit inner join loan on

deposit.accno=loan.accno ;

1. Display the customer who have neither a loan nor a deposit with the bank

Create table bankcustomer(accno int primary key,

custname varchar2(10) not null,place varchar2(20));

insert into bankcustomer(accno,custname,place) values(&accno,'&custname', '&place');

create table loan(accno int primary key,

loanno int,

lamount number,

FOREIGN KEY (accno) REFERENCES bankcustomer(accno)

);

insert into loan(accno,loanno,lamount) values(&accno,&loanno, &lamount);

Create table deposit(accno int,

Depono int primary key,

Depoamount int,

FOREIGN KEY (accno) REFERENCES bankcustomer(accno));

insert into deposit(accno,depono,depoamount) values(&accno,&depono, &depoamount);

1. Create a table employee with fields (EmpID, EName, Salary, Department, and Age). Insert some records. Write SQL queries using aggregate functions and group by clause

Create table employee(empid int primary key,

Ename varchar2(20) not null,

Salary int,dept varchar2(20),age int);

Insert into employee values(101,’raju’,12000,’sales’,28);

Insert into employee values(102,’rada’,30000,’sales’,28);

Insert into employee values(103,’sanni’,35000,’purchase’,28);

Insert into employee values(104,’sanju’,40000,’hr’,28);

Insert into employee values(106,’raju’,30000,’sales’,29);

Insert into employee values(107,’rada’,28000,’sales’,35);

Insert into employee values(108,’sanni’,30000,’purchase’,42);

Insert into employee values(109,’sanju’,37000,’hr’,56);

1. Display the total number of employees.

Select count(\*) from employee;

1. Display the name and age of the oldest employee of each department.

Select dept,max(age) from employee group by dept;

Select age, ename,dept from employee where age in( select max(age) from employee group by dept);

1. Display the average age of employees of each department

Select dept,avg(age) from employee group by dept;

1. Display departments and the average salaries

Select dept,avg(salary) from employee group by dept;

1. Display the lowest salary in employee table

Select min(salary) from employee;

1. Display the number of employees working in purchase department

Select count(empid) from employee where dept=’purchase’;

1. Display the highest salary in sales department;

Select max(salary) from employee where dept=’sales’;

1. Display the difference between highest and lowest salary

Select max(salary)-min(salary) from employee;

1. Create a table product with the fields (Product\_code primary key, Product\_Name, Category, Quantity, Price).

Insert some records Write the queries to perform the following.

* 1. Display the records in the descending order of Product\_Name
  2. Display Product\_Code, Product\_Name with price between 20 and 50
  3. Display the details of products which belongs to the categories of ‘bath soap’, ‘paste’, or ‘washing powder’
  4. Display the products whose Quantity less than 100 or greater than 500
  5. Display the products whose names starts with 's'
  6. Display the products which not belongs to the category 'paste'
  7. Display the products whose second letter is 'u' and belongs to the Category 'washing powder'

Create table product(procode int primary key,

Pronamevarchar2(20) not null,

Catogery varchar2(20) not null,

Quantity int ,

Price int);

Insert into product values(&procode,’&proname’,’&catogery’,&quantity,&price);

1. Select \* from product order by proname desc;
2. Select procode,proname from product where price between 20 and 50;
3. Select proname,price from product where catogery in(‘soap’,’pen’);
4. select \* from product where quantity<100 or quantity>500;
5. select \* from product where catogery like 's%';
6. select \* from product where catogery='paste';
7. select \* from product where catogery like '\_e%' and catogery='pen';
   * + 1. Consider the employee database given below. Give an expression in SQL for each of the following queries:

EMPLOYEE (Employee-Name, City)

Create table employee1(empname varchar2(20),city varchar2(20));

Insert into employee1 values(‘raju’,’ernakulam’);

Insert into employee1 values(‘rani’,’calicut’);

Insert into employee1 values(‘ram’,’malappuram’);

Insert into employee1 values(‘ramu’,’malappuram’);

WORKS (Employee-Name, Company-Name, Salary)

Create table works (empname varchar2(20),cmpname varchar2(20),salary int);

Insert into works values(‘raju’,’infosis’,25000);

Insert into works values(‘rani’, ’infosis’,35000);

Insert into works values(‘ram’,’wipro’,25000);

Insert into works values(‘ramu’, ’wipro’,20000);

COMPANY (Company-Name, City)

Create table company(cmpname varchar2(20),city varchar2(20));

Insert into company values(‘infosys’,’ernakulam’);

Insert into company values(‘tcs’,’calicut’);

Insert into company values(‘wipro’,’bangloor’);

Insert into company values(‘wipro’,’ernakulam’);

MANAGES (Employee-Name, Manager-Name)

Create table manages(empname varchar2(20),managname varchar2(20))

Insert into manages values(‘raju’,’govind’);

Insert into manages values(‘rani’,’govind’);

Insert into manages values(‘ram’,’goudam’);

Insert into manages values(‘ramu’,’goutham’);

Select E1.empname From Employee as E1, Employee as E2, Manages as M Where E1.empname=M.empname and E2.empname=M.managname and E1.stree=E2.street and E1.city=E2.city

Select empname.employee1, empname.works , from employee1,works where empname.employee1=empname.works ;

1. Find the names of all employees who work in Infosys

select empname from works where cmpname='infosis';

1. Find the names and cities of residence of all employees who works in Wipro

Select \* from employee1 where empname in (select empname from works where cmpname='wipro');

1. Find the names, and cities of all employees whowork in Infosys and earn more than Rs. 10,000.

Select \* from employee1 where empname in (select empname from works where cmpname='infosis' and salary>10000);

1. Find the employees who live in the same cities as the companies for which they work.

Select employee1.empname,works.cmpname,company.city from employee1,works,company where works.cmpname=company.cmpname and employee1.city=company.city;

Select employee1.empname from employee1,works,company where employee1.city=company.city and works.cmpname=company.cmpname;

1. Find all employees who do not work inWipro Corporation.

select empname from works where cmpname!='wipro';

1. Find the company that has the most employees.

Select cmpname from works group by cmpname having count(distinct empname)>=all(select count(distinct empname) from works group by cmpname);

* + - 1. Create table supplier(supcode,sname,city)

Create table product (pcode,pname)

Create table supl\_product(supcode,pcode,qty)

1. Get all pairs of supplier numbers such that the two suppliers are located in the samecity.
2. Get supplier names for suppliers who supply product P2.
3. Get product numbers supplied by more than one supplier.
4. Get supplier numbers for suppliers who are located in the same city as supplier S1.
5. Get supplier names for suppliers who supply part P1.
6. Get the number of Suppliers, who are supplying at least one product.
7. For each product supplied, get the pcode. and the total quantity supplied for that part.
   * + 1. Prepare a salary report of the employees showing the details such as:

EmpNo, Name, Basic Pay, DA, Gross Salary, PF, Net Salary, Annual Salary and Tax

For this purpose, create a table named SALARIES having the following structure.

|  |  |  |
| --- | --- | --- |
| Field Name | Type | Width |
| EmpNo | Character | 10 |
| Name | Character | 20 |
| Basic | Numeric | 6 |

Enter the records of at least 10 employees. Use the following information for calculating the details for the report:

DA is fixed as the 40% of the basic pay.

PF is fixed as 10% of the basic pay.

Gross Salary is (Basic Pay + DA).

Net Salary is (Gross Salary – PF)

Annual Salary is (12 \* Net Salary)

Tax is calculated using the following rules:

If annual salary is less than 100000, No Tax

If annual salary is greater than 100000 but less than or equal to 150000, then the tax is 10% of the excess over 100000.

If annual salary is greater than 150000 but less than or equal to 250000, then the tax is 20% of the excess over 150000.

If annual salary is greater than 250000, then the tax is 30% of the excess over 250000.

* + - 1. Create table exam\_result(rollno, avg\_score, Grade) insert 10 records. Assign null values to the field grade. Write Program block to update the grade field by using the following condition.

avg\_score between 90 and 100 - A

avg\_score 75 -89 - B

avg\_score 60- 74 - C

avg\_score 50 -59 - D

avg\_score below 50 - E

* + - 1. Write a program code to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding value of calculated area in an empty table named areas with field’s radius and area.

Create the table AREAS as:

**CREATE TABLE**AREAS (RADIUS **NUMBER(5),**AREA NUMBER(14,2));

**DECLARE**

pi constant number(4,2) := 3.14 ;

radius number(5);

area number(14,2);

**BEGIN**

radius := 3;

**WHILE**RADIUS <= 7

**LOOP**

area :=pi \* power(radius,2);

**INSERT INTO**areas **VALUES**(radius, area);

radius := radius + 1;

**END LOOP;**

**END;**

* + - 1. Write a program block to calculate the electricity bill by accepting cust\_no and units\_consumed.

SQL> create table electricity(cons\_id varchar(4) primary key, c\_name varchar(20), rent number(2) che

ck (rent=20), unit number(6));

Table created.

SQL> insert into electricity values ('E001','deepika',20,35);

1 row created.

SQL> insert into electricity values ('E002','varna',20,61);

1 row created.

SQL> alter table electricity add (total number (6,2));

Table altered.

declare

v\_total electricity.total%type;

cursor c is select \* from electricity;

begin

for I in c loop

if (i.unit <=40) then

v\_total:=i.rent;

elsif (i.unit<=80) then

v\_total:= i.rent+(i.unit-40)\*.40;

else

v\_total:= i.rent+(40\*.40)+(i.unit-80)\*1.40;

end if;

update electricity set total=v\_total where cons\_id = i.cons\_id;

end loop;

commit;

end;

/

* + - 1. Create a procedure toprint Fibonacci number up to a limit, limit is passed as an argument

declare

first number:=0;

second number:=1;

third number;

n number:=&n;

i number;

begin

dbms\_output.put\_line('Fibonacci series is:');

dbms\_output.put\_line(first);

dbms\_output.put\_line(second);

for i in 2..n

loop

third:=first+second;

first:=second;

second:=third;

dbms\_output.put\_line(third);

end loop;

end;

/

* + - 1. Create a pl/ sql function to check whether a given number is prime or not

declare

n number;

i number;

flag number;

begin

i:=2;

flag:=1;

n:=&n;

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;

* + - 1. create a table stud\_mark(regno, sname ,avg\_mark)

Insert few records

Write a procedure to display number of students got Distinction, first-class,second class , third class or failed (90-100 distinction, 75-89firstclass60-74 second class 50-59 Third class below 50 failed)

* + - 1. create a table emp\_salary(empno,enamedept,salary)

Write a function to return the average salary of a particular department by accepting departmentname as argument.

Create table empsalary(empno int primary key,ename varchar2(20),dept varchar2(20),salary int);

Insert into empsalary values(101,’ram’,’purchase’,25000);

Insert into empsalary values(102,’raj’,’sales’,20000);

Insert into empsalary values(104,’raju’,’sales’,25000);

Insert into empsalary values(103,’ramu’,’purchase’,18000);

Create or replace function averagesalary(d\_pt in varchar2)

return number is

dummyavg number;

begin

select avg(salary) into dummyavg from empsalary where dept=d\_pt;

return dummyavg;

end;

declare

dummyavg number;

begin

dummyavg:= averagesalary(‘sales’);

dbms\_output.put\_line(dummyavg);

end;

* + - 1. create a table Student(regno, sname, sub1,sub2,sub3,sub4,sub5,mark\_total,avg\_mark)

create table studnt(rollno number(3),name varchar(2),m1 number(3),m2 number(3),m3 number(3),m4 number(3),m5 number(3),tot number(3),avrg number(3));

Create a BEFORE INSERT trigger to calculate total mark and average mark and update the corresponding columns.

create or replace trigger tr1 before insert on studnt

for each row

begin

:new.tot:=:new.m1+:new.m2+:new.m3+:new.m4+:new.m5;

:new.avrg:=:new.tot/5;

end;

insert into studnt values(101,'SM',67,89,99,67,76,null,null);

insert into studnt values(102,'SM',60,80,90,60,70,null,null);

* + - 1. create table phonebook(pname,mobno)

Create a Trigger to insert the old records from the table phonebook to del\_phonebook (pname, mobno, modfy\_date) whenever a record is deleted or updated in the phonebook table.

Create table phonebook

(

pname varchar2(20),

mob number

);

Insert into phonebook values(‘kathu’,3343544);

Create table phonebookdel1

(

pname varchar2(20),

mob number,

mdate date

);

Create or replace trigger insertphonebook

after delete on phonebook

for each row

begin

insert into phonebookdel1 values ( :old.pname,:old.mob,sysdate );

end;

19. Create a table shop with fields item\_id,item\_name,price, quantity. Write a pl/sql proceudre 'sales' to update the quantity by accepting item\_id and quanity as argument. Write pl/sql block to invoke the procedure

SQL> create table shop(itemid int,itemname varchar2(20),price int,quantity int);

Table created.

SQL>

insert into shop(itemname,price) values ('pen',100);

SQL>

create or replace procedure updatequantity

(itemsid in integer,quantitye in integer)

is

begin

update shop

set quantity=quantitye

where itemid=itemsid;

end;

/

SQL>EXECUTE updatequantity(10,60);

PL/SQL procedure successfully completed.