CUSTOMER

create table customer(cust_no varchar(5),cust_name varchar(15),age numeric,phone varchar(10)); a) insert into customer values(1,'A',23,94567); select *from customer; b) alter table customer add d_o_b date; c) create table cust_phone as select cust_name,phone from customer; select *from cust_phone; d) alter table customer drop age; select *from customer; e) ALTER TABLE customer ALTER COLUMN cust_name TYPE varchar(25); select *from customer; f) TRUNCATE customer; select *from customer; g) ALTER TABLE customer RENAME TO cust; select *from cust; h) drop table cust;

SALESMAN AND SALES ORDER

create table sales_man(salesman_no int primary key,s_name varchar(20) not null,place text,phone int unique); create table sales_order(order_no int primary key,order_date date not null,salesman_no int, foreign key(salesman_no) references sales_man(salesman_no) on delete cascade,del_type char(1) check(del_type='P' or del_type='F'),order_status varchar(10) check(order_status='inprocess'or order_status='fullfilled'or order_status='backorder'or order_status='cancelled')); a) insert into sales_man values(1,'A','abc',98456); select * from sales_man; insert into sales_order values(1,'01-01-2000',1,'P','inprocess'); select * from sales_order; b) ALTER TABLE sales_man DROP CONSTRAINT sales_man_pkey cascade; c) ALTER TABLE sales_man DROP CONSTRAINT sales_order_fkey; ALTER TABLE sales_order DROP CONSTRAINT sales_order_del_type_check; d) ALTER TABLE sales_man add primary key(salesman_no); e) ALTER TABLE sales_order add constraint sales_order_del_type_check CHECK (del_type = 'P' OR del_type = 'F'); ALTER TABLE sales_order ADD CONSTRAINT FK_salesman_no FOREIGN KEY (salesman_no) REFERENCES sales_man(salesman_no);

HOSPITAL

create table hospital(doctor_id text,doctor_name varchar(10),department varchar(25),qualification varchar(25),experience text); a) insert into hospital values('D001','miya','cardiologist','mbbs',5); b) select * from hospital; c) select * from hospital where qualification='md'; d) select * from hospital where experience>'5' and qualification!='md'; e) select * from hospital where department='skin'; f) update hospital set experience=5 where doctor_id='D003'; select * from hospital; g) delete from hospital where doctor_id='D005'; select * from hospital;

EMPLOYOEE create table emp(emp_id int,e_name char(10),salary int,department char(10),age int); insert into emp values(001,'ram',360000,'hr',22); select*from emp; a) select count(emp_id) from emp; b) select e_name,age from emp where age=(select max(age)from emp); c) select avg(age)from emp group by department; d) select department,avg(salary)from emp group by department; e) select min(salary) from emp; f) select count(e_name)from emp where department='purchase';; g) select max(salary) from emp where department='sales'; h) select max(salary)-min(salary) difference from emp;

PRODUCT create table product(Product_code int primary key, Product_Name varchar(20),Category varchar(20), Quantity int,Price numeric); insert into product values(1,'colgate','paste',10,100); select * from product; a) select * from product order by product_name desc; b) select product_code,product_name from product where price between 20 and 50; c) select product_name,price from product where category in ('bath soap','paste','washing powder'); d) select * from product where quantity500; e) select product_name from product where product_name like 't%'; f) select product_name from product where category !='paste'; g) insert into product values(10,'radhas','washing powder',600,2000); select product_name from product where product_name like '_a%' and category='washing powder';

EMPLOYEE DATABASE create table works(emp_name varchar(10),company_name varchar(10),salary int); create table company(company_name varchar(10),city varchar(10)); create table manages(emp_name varchar(10),manager_name varchar(10)); insert into employee1 values('ammu','chennai'); select *from employee1; insert into works values('ammu','infosys','11000'); select *from works; insert into company values('infosys','chennai'); select *from company; a) select emp_name from works where company_name='infosys'; b) select e.emp_name,e.city from employee1 e join works w on e.emp_name=w.emp_name where company_name='wipro'; c) select e.emp_name,e.city from employee1 e join works w on e.emp_name=w.emp_name where company_name='infosys' and salary>10000; d) select e.emp_name from employee1 e, works w, company c where e.emp_name =w.emp_name and e.city = c.city and w.company_name = c.company_name; e) select emp_name from works where company_name!='wipro'; f) select company_name from works group by company_name having count (distinct emp_name) >= all (select count (distinct emp_name) from works group by company_name);

AREA OF CIRCLE 3-7 do \ 'declare \ r integer; \ area numeric(5,2); \ pi constant float :=3.14; \begin \ create table area(r integer ,area numeric(5,2)); \ r:=3; \ while r<=7 \ loop \area:=pi*power(r,2); \ insert into area values(r,area); \ r:=r+1; \ end loop; \ end'; select * from area;

FUNCTION PRIME OR NOT

create function primecheck(n integer) returns varchar(10) as' declare \ i integer; \ flag integer:=0; \ begin \ for i in 2..n/2 \ loop \ if mod(n,i)=0 \ then \ flag:=1; \ exit; \ end if; \end loop; \ if flag=0 \then \ return ''prime''; \else \ return ''not prime''; \ end if; \ end;' \ language plpgsql; select primecheck(20);

FIBNOCCI NUMBER UPTO LIMIT

CREATE OR REPLACE FUNCTION fibonacci1(num integer) RETURNS SETOF numeric AS $$ DECLARE \ fib1 numeric := 0; \ fib2 numeric := 1; \ BEGIN \ IF (num <= 0) \ THEN RETURN; \ END IF; \ RETURN NEXT fib1; \ LOOP \ EXIT WHEN num <= 1; \ RETURN NEXT fib2; \ num = num - 1; \SELECT fib2, fib1 + fib2 INTO fib1, fib2; \ END LOOP; \END; \ $$ language plpgsql; \ select fibonacci(12);

EMPLOYEE SALARY AVERAGE

create table emp_sal(empno int,ename varchar(10),edept

varchar(10),salary numeric); select* from emp_sal; \create function avgsal(deptt varchar(10))returns numeric(10,2) as \ 'declare \ avgsal numeric(10,2); \ begin \ avgsal=(select avg(salary) from emp_sal group by edept having edept= deptt); \ return avgsal; \ end;'language plpgsql**;**

EXAM RESULT create table examresult(rollno int,avg_score float,grade char(1)); insert into examresult values(1,90); do ' \ declare \r record; \ gr char(1); \ begin \ for r in select rollno,avg_score,grade from examresult \ loop \ if r.avg_score>=90 then gr:=''A''; \ elseif(r.avg_score>=75) then gr:=''B''; \ elseif(r.avg_score>=60) then gr:=''C''; \ elseif(r.avg_score>=50) then gr:=''D''; \ else gr:=''E''; \ end if; \ update examresult set grade=gr where rollno=r.rollno; \ end loop; \ end'; \ select * from examresult; **STUDENT** create table student1(regno varchar(10), sname varchar(20), sub1 int,sub2 int,sub3 int,sub4 int,sub5 int,total int,avg float); create or replace function proc() returns trigger as 'begin \ new.total=new.sub1+new.sub2 +new.sub3+new.sub4+new.sub 5; \ new.avg=new.total/5.0; \ return new; \ end;' \ language plpgsql; create trigger tr before insert on student1 for each row execute procedure proc(); insert into student1 values('12', 'aa',12,30,40,50,50); select * from student1**;**

PHONE BOOK create table phone_book(pname varchar(10) primary key,mobno integer); create table del_phonebook(pname varchar(10) primary key,mobno integer,modify_date date); insert into phone_book values('aaa',369369369); \ insert into phone_book values('bbb',369369469); \ insert into phone_book values('ccc',469369369); create or replace function proc() returns trigger as \ 'begin \ insert into del_phonebook values (old.pname,old.mobno,current_ date);\ return new;\ end;'language plpgsql;\ create trigger tr after delete on phone_book for each row\ execute procedure proc ();