Fast**National University of Computer & Emerging Sciences, Karachi  
Fall 2019 CS-Department  
Mid-Term Exam  
22nd -Oct-2019, 11:10 am – 12:20 pm**

|  |  |  |
| --- | --- | --- |
| **Course Code: CL-203** | **Course Name: Database Systems Lab** | |
| **Instructor Name: Ammara Yaseen** | | |
| **Student Roll No:** | | **Section: E** |

**Instructions:**

* Return the question paper.
* In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper.
* You are required to submit your exam in a single text file (Questions must be in order), name your file with your student id i-e (17k-XXXX).
* Each question carry 5 marks.

**Time:** 70 minutes. **Max Marks:** 50

**Q1**-

1. Find the names of gardeners who planted 20-30 seeds.

Select name from gardener where gardenerid in

(select gardenerfk from planted group by gardenerfk having sum(seeds)>=20 and sum(seeds)<=40);

1. Display the name of the gardener who planted 4th highest seeds.

Select g.name from gardener g, planted p where g.GARDENERID=p.GARDENERFK and p.SEEDS in(select max(SEEDS) from planted where SEEDS < (select max(SEEDS) from planted where SEEDS < (select max(SEEDS) from planted where SEEDS < (select max(SEEDS) from planted)) ));

**Q2**- Calculate the total weight of all the corns that were picked from the garden.

Select sum(picked.weight) as corn\_weight from plant,picked where plantid=plantfk and name='Corn';

**Q3**- Write an SQL query to display plant name along with the name who has picked the produced plant (take plant name as user input).

Select p.name, g.name from plant p, planted pl, gardener g where p.plantid = pl.plantfk and g.gardenerid = pl.gardenerfk and p.name='&name';

**Q4**- List the names of the plants planted with the largest number of seeds.

Select name from plant where plantid in

(Select plantfk from planted group by plantfk having sum (seeds) in

(Select max (sum(seeds)) from planted group by plantfk));

**Q5**- Display the name of the plant who has been picked the most in terms of amount and weight it has produced.

select name from plant where plantid in

(select plantfk from picked group by plantfk having sum(amount\*weight) in

(select max(sum(amount\*weight)) from picked group by plantfk));

**Q6-** Write a valid SQL statement that would produce a result set like the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Name** | **Date** | **Amount** |
| Tim | Radish | 2015-07-07 | 23 |
| Tim | Corn | 2015-08-28 | 10 |

select gardener.name,plant.name,to\_char(picked.date\_,'YYYY-MM-DD'),picked.amount from plant,gardener,picked,planted

where plantid=picked.plantfk and gardenerid=picked.gardenerfk and plantid=planted.plantfk and gardenerid=planted.gardenerfk

and gardener.name='Tim' ORDER BY AMOUNT DESC;

**Q7-** Name the gardeners who planted and picked the same plant as Tim.

select gardenerid , name from gardener where gardenerid in

(select gardenerfk from planted where plantfk in

(select plantfk from planted where gardenerfk = (select gardenerid from gardener where name like 'Tim'))

and gardenerfk <> (select gardenerid from gardener where name like 'Tim'));

**Q8-**

1. Insert one record in all tables of the given schema, take user input while inserting record in the Gardener table.

Insert into location values (100, 'east', 0.28, 0.80);

Insert into gardener values (&id, '&name', &age);

Insert into plant values (60, 'Carrot', .26, .82, 0 .08);

Insert into planted values (0, 1, 2, '02-nov-2019', 5);

Insert into picked values (0, 2, 1,'03-nov-2019', 50, 2.65);

1. Write query(s) to change plantid from 4 to 5.

UPDATE plant SET plantid=5 WHERE plantid=4;

Table PLANT is parent table for the PLANTED and Picked table so record from parent table can’t be updated.

If we want to UPDATE records we should delay the constraints checking.

1. Add a new field fertilizer in plant table so that it can store fertilizer necessary for particular plant.

ALTER TABLE PLANT

ADD fertilizer VARCHAR2 (25);

**Q9-** Write DDL for the given ERD:

CREATE TABLE Customers (

customer\_id int NOT NULL,

customer\_full\_name varchar(100) ,

customer\_address varchar(100) ,

customer\_city varchar(100) ,

customer\_state varchar(100) ,

customer\_zip varchar(100) ,

customer\_phone number(11) ,

PRIMARY KEY (customer\_id)

);

CREATE TABLE orders(

invoice\_number number(6) NOT NULL ,

customer\_id int ,

invoice\_date date,

shipping\_charge number(9),

PRIMARY KEY (invoice\_number),

CONSTRAINT cus\_orfk\_1 FOREIGN KEY (customer\_id) REFERENCES customers (customer\_id)

);

CREATE TABLE products (

product\_id number(6) NOT NULL,

product\_name varchar(6),

purchase\_price int,

units\_in\_stock int,

units\_on\_order int,

reorder\_level varchar2(25),

discontinued int,

retail\_price int,

PRIMARY KEY (product\_id)

);

CREATE TABLE order\_items (

invoice\_number number(6),

product\_id number(6),

unit\_price number(5) NOT NULL,

order\_quantity int NOT NULL,

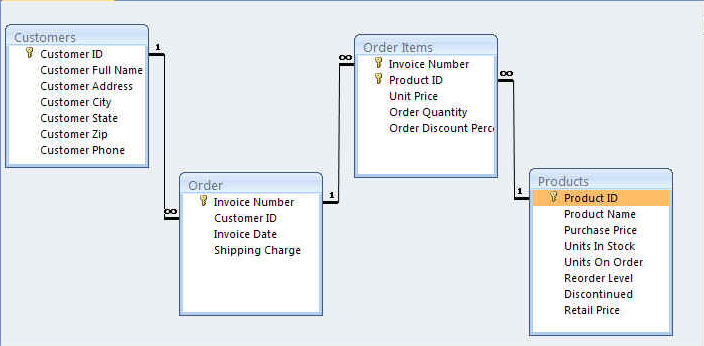
order\_discount int NOT NULL ,

PRIMARY KEY (invoice\_number,product\_id),

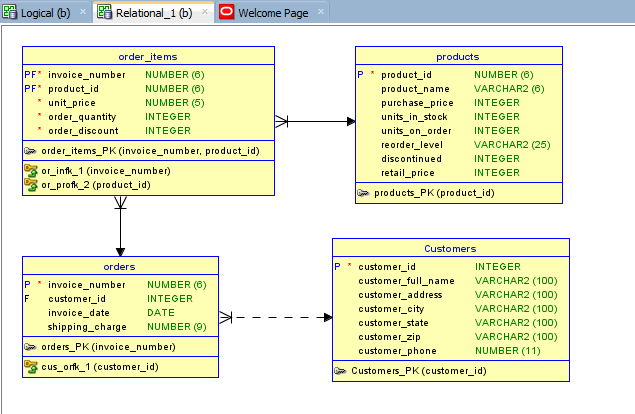
CONSTRAINT or\_infk\_1 FOREIGN KEY (invoice\_number) REFERENCES orders (invoice\_number),

CONSTRAINT or\_profk\_2 FOREIGN KEY (product\_id) REFERENCES products(product\_id)

);



Generate relational model from your written script.



**Q10-** A Health Insurance Company insures its employees in the following cases:

1. If the employee is married.
2. If the employee is unmarried, male & above 30 years of age.
3. If the employee is unmarried, female & above 25 years of age.

In all other cases the employee is not insured. If the marital status, gender and age of the employee are the inputs, write a PL/SQL program to determine whether the employee is to be insured or not.

SET SERVEROUTPUT ON;

DECLARE

MARITAL\_STATUS VARCHAR2(255);

GENDER VARCHAR2(255);

AGE NUMBER;

BEGIN

MARITAL\_STATUS:='&MARITAL\_STATUS';

GENDER:='&GENDER';

AGE:='&AGE';

IF MARITAL\_STATUS = 'YES' THEN

DBMS\_OUTPUT.PUT\_LINE('INSURED');

END IF;

IF GENDER='MALE' AND AGE > 30 THEN

DBMS\_OUTPUT.PUT\_LINE('INSURED');

END IF;

IF GENDER='FEMALE' AND AGE > 25 THEN

DBMS\_OUTPUT.PUT\_LINE('INSURED');

END IF;

DBMS\_OUTPUT.PUT\_LINE(' NOT INSURED');

END;

/

***BEST OF LUCK!***