A Project Report On

Horticulture Management System

Made By:

IT-115 Pipaliya Nisarg

IT-123 Salodiya Snehal

IT-125 Sangani Kevin

Guided By
Prof. Shweta Jambukia

Department of Information Technology
Faculty of Technology
DD University

Index:

	Page NO.
Certificate	3
1. Project Overview	4
2. ER and Relational Schema	5
3. Data Dictionary	7
4. Implementation	12
5. Inserted Data	21
6. Join and subquery	26
7. Functions and triggers	31

CERTIFICATE

This is to certify that the project entitled “Horticulture Management system” is a bonafide. report of the work carried out by:

1)Pipaliya Nisarg-21ITUES081
2)Sangani Kevin-21ITUOS067
3)Salodiya Snehal-21CHUOT152

of Department of Information Technology, semester IV, under the guidance and supervision for the subject Database Management System. They were involved in Project training during the academic year 2022-2023.

Prof. Shweta Jambukia
Project Guide, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:15/03/2023

Prof. Vipul Dabhi
Head, Department of Information Technology

1.Project Overview

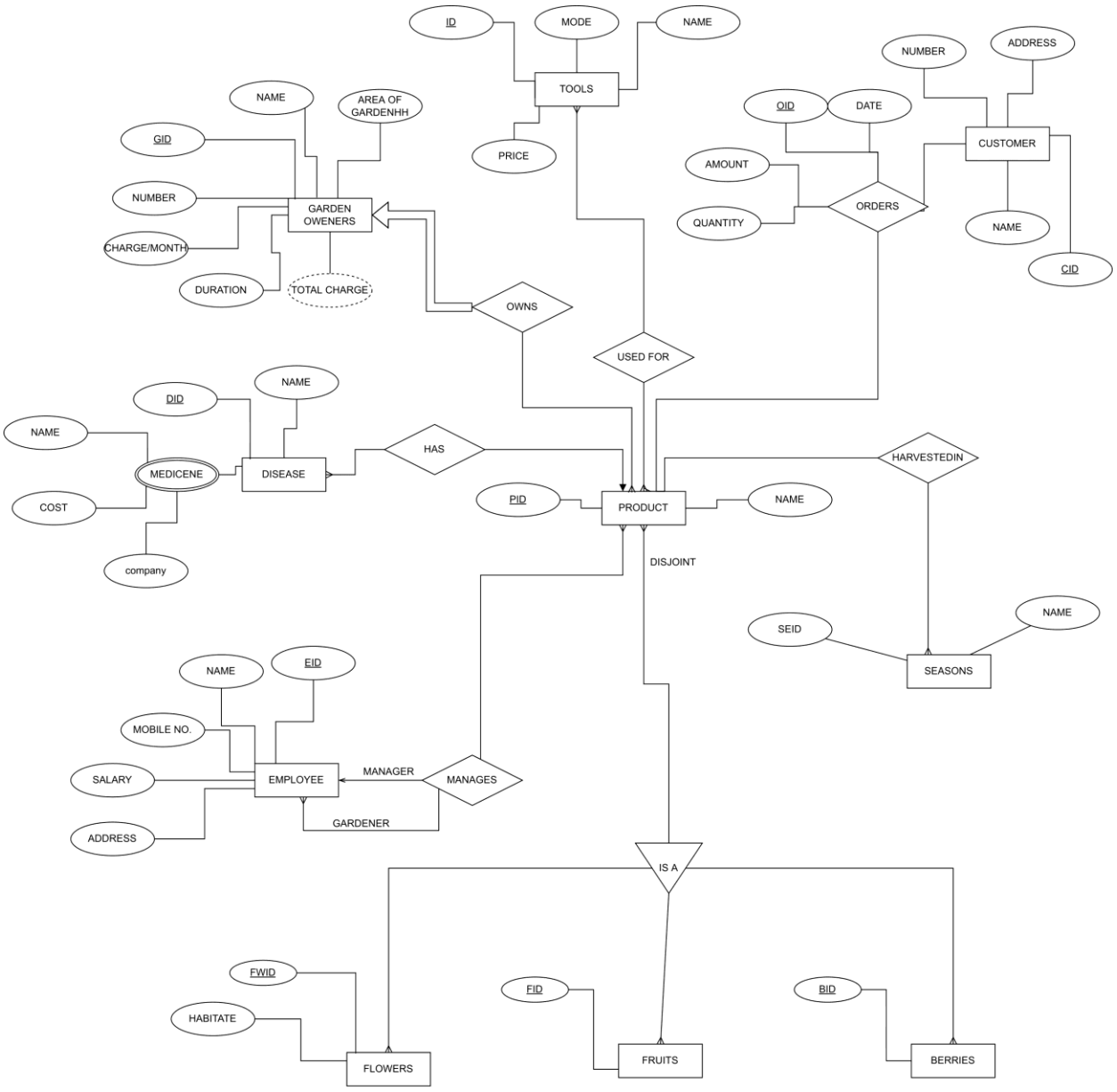
Horticulture management system.

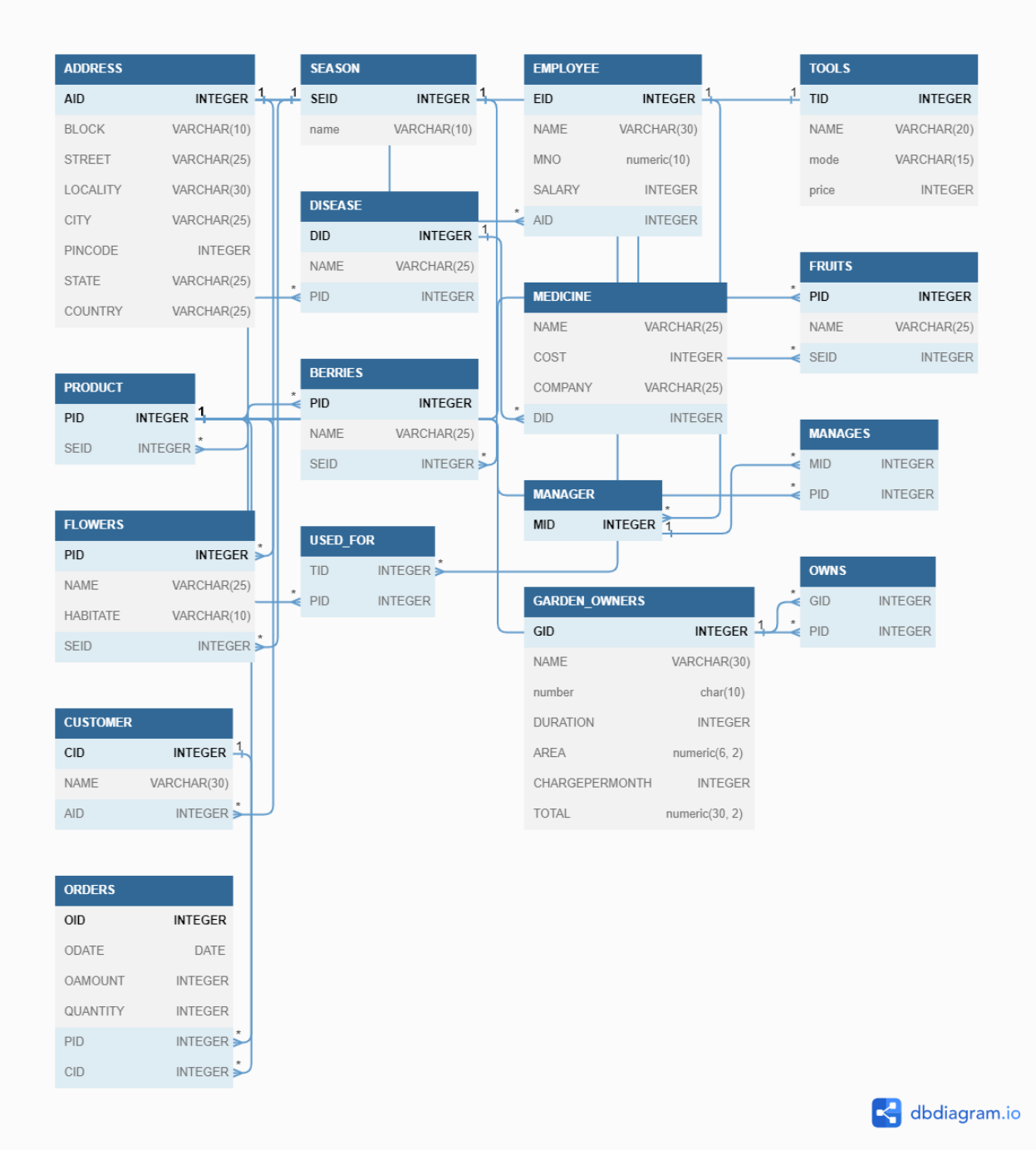
Our goal was to design a management system for a firm which was providing skilled labors, tool, medicines, etc things to the farmers who want a helping hand in managing their farm and in return they get the rent which they have decided.

Here we have only considered Fruits, Flowers and Berries and we can refer the farm/land where any of this three grows as “Garden” so we refer it as garden throughout the whole file.

This system helps in analyzing that which crop/produce was mostly chosen by the farmer or which is the most useful tool or multipurpose tool, which produce gives us maximum profit or which kind of disease is common among plants and around which period.

2.ER-Diagram & Relational Schema.




ADDRESS
AID INTEGER
BLOCK VARCHAR(10)
STREET VARCHAR(25)
LOCALITY VARCHAR(30)
CITY VARCHAR(25)
PINCODE INTEGER
STATE VARCHAR(25)
COUNTRY VARCHAR(25)
SEASON
SEID INTEGER
name VARCHAR(10)
EMPLOYEE
EID INTEGER
NAME VARCHAR(30)
MNO numeric(10)
SALARY INTEGER
AID INTEGER
TOOLS
TID INTEGER
NAME VARCHAR(20)
mode VARCHAR(15)
price INTEGER
DISEASE
DID INTEGER
NAME VARCHAR(25)
PID INTEGER
FRUITS
PID INTEGER
NAME VARCHAR(25)
SEID INTEGER
MEDICINE
NAME VARCHAR(25)
COST INTEGER
COMPANY VARCHAR(25)
DID INTEGER
PRODUCT
PID INTEGER
SEID INTEGER
BERRIES
PID INTEGER
NAME VARCHAR(25)
SEID INTEGER
MANAGES
MID INTEGER
PID INTEGER
MANAGER
MID INTEGER
FLOWERS
PID INTEGER
NAME VARCHAR(25)
HABITATE VARCHAR(10)
SEID INTEGER
USED_FOR
TID INTEGER
PID INTEGER
OWNS
GID INTEGER
PID INTEGER
GARDEN_OWNERS
GID INTEGER
NAME VARCHAR(30)
number char(10)
DURATION INTEGER
AREA numeric(6, 2)
CHARGEPERMONTH INTEGER
TOTAL numeric(30, 2)
CUSTOMER
CID INTEGER
NAME VARCHAR(30)
AID INTEGER
ORDERS
OID INTEGER
ODATE DATE
OAMOUNT INTEGER
QUANTITY INTEGER
PID INTEGER
CID INTEGER
dbdiagram.io

3.Data Dictionary

1.Address:

```
postgres=# \d address;
                    Table "public.address"
  Column  |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 aid      | integer               |           | not null |
 block    | character varying(10) |           | not null |
 street   | character varying(25) |           | not null |
 locality | character varying(30) |           |          |
 city     | character varying(25) |           | not null |
 pincode  | integer               |           | not null |
 state    | character varying(25) |           | not null |
 country  | character varying(25) |           | not null |
Indexes:
    "address_pkey" PRIMARY KEY, btree (aid)
Check constraints:
    "address_pincode_check" CHECK (pincode <= 999999 AND pincode >= 100000)
Referenced by:
    TABLE "customer" CONSTRAINT "customer_aid_fkey" FOREIGN KEY (aid) REFERENCES address(aid)
    TABLE "employee" CONSTRAINT "employee_aid_fkey" FOREIGN KEY (aid) REFERENCES address(aid)
```

2.Customers

```
postgres=# \d customer;
                   Table "public.customer"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 cid    | integer               |           | not null |
 name   | character varying(30) |           | not null |
 aid    | integer               |           | not null |
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cid)
Foreign-key constraints:
    "customer_aid_fkey" FOREIGN KEY (aid) REFERENCES address(aid)
Referenced by:
    TABLE "orders" CONSTRAINT "orders_cid_fkey" FOREIGN KEY (cid) REFERENCES customer(cid)
```

3.Employees

```
postgres=# \d employee;
                   Table "public.employee"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 eid    | integer               |           | not null |
 name   | character varying(30) |           | not null |
 mno    | numeric(10,0)         |           | not null |
 salary | integer               |           | not null |
 aid    | integer               |           |          |
Indexes:
    "employee_pkey" PRIMARY KEY, btree (eid)
Foreign-key constraints:
    "employee_aid_fkey" FOREIGN KEY (aid) REFERENCES address(aid)
Referenced by:
    TABLE "manager" CONSTRAINT "manager_mid_fkey" FOREIGN KEY (mid) REFERENCES employee(eid)
```

4.Manager

```
postgres=# \d manager;
              Table "public.manager"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 mid    | integer |           | not null |
Indexes:
    "manager_pkey" PRIMARY KEY, btree (mid)
Foreign-key constraints:
    "manager_mid_fkey" FOREIGN KEY (mid) REFERENCES employee(eid)
Referenced by:
    TABLE "manages" CONSTRAINT "manages_mid_fkey" FOREIGN KEY (mid) REFERENCES manager(mid)
```

5.Manages

```
postgres=# \d manages;
              Table "public.manages"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 mid    | integer |           | not null |
 pid    | integer |           | not null |
Foreign-key constraints:
    "manages_mid_fkey" FOREIGN KEY (mid) REFERENCES manager(mid)
    "manages_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
```

6.Orders

```
postgres=# \d orders;
                Table "public.orders"
  Column  |  Type   | Collation | Nullable | Default
----------+---------+-----------+----------+---------
 oid      | integer |           | not null |
 odate    | date    |           | not null |
 oamount  | integer |           | not null |
 quantity | integer |           | not null |
 pid      | integer |           | not null |
 cid      | integer |           | not null |
Indexes:
    "orders_pkey" PRIMARY KEY, btree (oid)
Check constraints:
    "orders_oamount_check" CHECK (oamount > 0)
    "orders_quantity_check" CHECK (quantity > 0)
Foreign-key constraints:
    "orders_cid_fkey" FOREIGN KEY (cid) REFERENCES customer(cid)
    "orders_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
```

7.Garden Owners

```
postgres=# \d garden_owners;
                                        Table "public.garden_owners"
     Column     |         Type          | Collation | Nullable |                         Default
----------------+-----------------------+-----------+----------+-----------------------------------------------------------
 gid            | integer               |           | not null |
 name           | character varying(30) |           | not null |
 number         | character(10)         |           | not null |
 duration       | integer               |           | not null |
 area           | numeric(6,2)          |           | not null |
 chargepermonth | integer               |           | not null |
 total          | numeric(30,2)         |           |          | generated always as ((duration * chargepermonth)) stored
Indexes:
    "garden_owners_pkey" PRIMARY KEY, btree (gid)
Check constraints:
    "garden_owners_area_check" CHECK (area > 0::numeric)
    "garden_owners_chargepermonth_check" CHECK (chargepermonth > 0)
    "garden_owners_duration_check" CHECK (duration > 0 AND duration < 13)
    "garden_owners_number_check" CHECK (length(number) = 10)
Referenced by:
    TABLE "owns" CONSTRAINT "owns_gid_fkey" FOREIGN KEY (gid) REFERENCES garden_owners(gid)
```

8.Owns

```
postgres=# \d owns;
                Table "public.owns"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 gid    | integer |           | not null |
 pid    | integer |           | not null |
Foreign-key constraints:
    "owns_gid_fkey" FOREIGN KEY (gid) REFERENCES garden_owners(gid)
    "owns_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
```

9.Medicine

```
postgres=# \d medicine;
                   Table "public.medicine"
 Column  |         Type          | Collation | Nullable | Default
---------+-----------------------+-----------+----------+---------
 name    | character varying(25) |           | not null |
 cost    | integer               |           | not null |
 company | character varying(25) |           | not null |
 did     | integer               |           |          |
Foreign-key constraints:
    "medicine_did_fkey" FOREIGN KEY (did) REFERENCES disease(did)
```

10.Seasons

```
postgres=# \d seasons;
Did not find any relation named "seasons".
postgres=# \d season;
                    Table "public.season"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 seid   | integer               |           | not null |
 name   | character varying(10) |           | not null |
Indexes:
    "season_pkey" PRIMARY KEY, btree (seid)
Referenced by:
    TABLE "berries" CONSTRAINT "berries_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
    TABLE "flowers" CONSTRAINT "flowers_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
    TABLE "fruits_n" CONSTRAINT "fruits_n_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
    TABLE "fruits" CONSTRAINT "fruits_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
    TABLE "product" CONSTRAINT "product_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
```

11.Tools

```
postgres=# \d tools;
                     Table "public.tools"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 tid    | integer               |           | not null |
 name   | character varying(20) |           | not null |
 mode   | character varying(15) |           | not null |
 price  | integer               |           | not null |
Indexes:
    "tools_pkey" PRIMARY KEY, btree (tid)
Referenced by:
    TABLE "used_for" CONSTRAINT "used_for_tid_fkey" FOREIGN KEY (tid) REFERENCES tools(tid)
```

12.Used for

```
postgres=# \d used_for;
              Table "public.used_for"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 tid    | integer |           | not null |
 pid    | integer |           | not null |
Foreign-key constraints:
    "used_for_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    "used_for_tid_fkey" FOREIGN KEY (tid) REFERENCES tools(tid)
```

13.Product

```
postgres=# \d product;
              Table "public.product"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 pid    | integer |           | not null |
 seid   | integer |           |          |
Indexes:
    "product_pkey" PRIMARY KEY, btree (pid)
Foreign-key constraints:
    "product_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
Referenced by:
    TABLE "berries" CONSTRAINT "berries_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "disease" CONSTRAINT "disease_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "flowers" CONSTRAINT "flowers_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "fruits_n" CONSTRAINT "fruits_n_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "fruits" CONSTRAINT "fruits_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "manages" CONSTRAINT "manages_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "orders" CONSTRAINT "orders_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "owns" CONSTRAINT "owns_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    TABLE "used_for" CONSTRAINT "used_for_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
```

14.Fruits

```
postgres=# \d fruits;
                    Table "public.fruits"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 pid    | integer               |           | not null |
 name   | character varying(25) |           | not null |
 seid   | integer               |           |          |
Indexes:
    "fruits_pkey" PRIMARY KEY, btree (pid)
Foreign-key constraints:
    "fruits_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    "fruits_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
```

15.Flowers

```
postgres=# \d flowers;
                    Table "public.flowers"
  Column  |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 pid      | integer               |           | not null |
 name     | character varying(25) |           | not null |
 habitate | character varying(10) |           | not null |
 seid     | integer               |           | not null |
Indexes:
    "flowers_pkey" PRIMARY KEY, btree (pid)
Foreign-key constraints:
    "flowers_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    "flowers_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
```

16.Berries

```
postgres=# \d berries;
                    Table "public.berries"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 pid    | integer               |           | not null |
 name   | character varying(25) |           | not null |
 seid   | integer               |           | not null |
Indexes:
    "berries_pkey" PRIMARY KEY, btree (pid)
Foreign-key constraints:
    "berries_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
    "berries_seid_fkey" FOREIGN KEY (seid) REFERENCES season(seid)
```

17.Disease

```
postgres=# \d disease;
                    Table "public.disease"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 did    | integer               |           | not null |
 name   | character varying(25) |           | not null |
 pid    | integer               |           | not null |
Indexes:
    "disease_pkey" PRIMARY KEY, btree (did)
Foreign-key constraints:
    "disease_pid_fkey" FOREIGN KEY (pid) REFERENCES product(pid)
Referenced by:
    TABLE "medicine" CONSTRAINT "medicine_did_fkey" FOREIGN KEY (did) REFERENCES disease(did)
```

4.Implementation

SCHEMAS:

```
CREATE TABLE ADDRESS(
  AID INTEGER PRIMARY KEY,
  BLOCK VARCHAR(10) NOT NULL,
  STREET VARCHAR(25) NOT NULL,
  LOCALITY VARCHAR(30) ,
  CITY VARCHAR(25) NOT NULL,
  PINCODE INTEGER NOT NULL CHECK(PINCODE <= 999999 AND PINCODE >=100000),
  STATE VARCHAR(25) NOT NULL,
  COUNTRY VARCHAR(25) NOT NULL
);

CREATE TABLE SEASON (
  seid INTEGER PRIMARY KEY,
  name VARCHAR(10) NOT NULL
);
CREATE TABLE EMPLOYEE (
  EID INTEGER PRIMARY KEY,
  NAME VARCHAR(30) NOT NULL,
  MNO numeric(10) NOT NULL,
  SALARY INTEGER NOT NULL,
  AID INTEGER references ADDRESS(AID)
);


CREATE TABLE TOOLS(
  TID INTEGER PRIMARY KEY,
  NAME VARCHAR(20) NOT NULL,
  mode VARCHAR(15) not NULL,
  price INTEGER not NULL
);


CREATE TABLE PRODUCT(
  PID INTEGER PRIMARY KEY,
  SEID INTEGER references SEASON(SEID)
);
```

```
CREATE TABLE DISEASE(
  DID INTEGER PRIMARY KEY,
  NAME VARCHAR(25) NOT NULL,
  PID INTEGER references PRODUCT(PID) NOT NULL);

CREATE TABLE MEDICINE(
  NAME VARCHAR(25) NOT NULL,
  COST INTEGER NOT NULL,
  COMPANY VARCHAR(25) NOT NULL,
  DID INTEGER references DISEASE(DID)
);


CREATE TABLE FRUITS(
  PID INTEGER references PRODUCT(PID) PRIMARY KEY,
  NAME VARCHAR(25) NOT NULL,
  SEID INTEGER references SEASON(SEID)
);

CREATE TABLE FLOWERS(
  PID INTEGER references PRODUCT(PID) PRIMARY KEY,
  NAME VARCHAR(25) NOT NULL,
  HABITATE VARCHAR(10) NOT NULL,
  SEID INTEGER references SEASON(SEID) NOT NULL
);

CREATE TABLE BERRIES(
  PID INTEGER references PRODUCT(PID) PRIMARY KEY,
  NAME VARCHAR(25) NOT NULL,
  SEID INTEGER references SEASON(SEID) NOT NULL
);



CREATE TABLE MANAGER(
  MID INTEGER references EMPLOYEE(EID) PRIMARY KEY
);

CREATE TABLE MANAGES(
  MID INTEGER references MANAGER(MID) NOT NULL,
  PID INTEGER references PRODUCT(PID) NOT NULL
);
```

```
CREATE TABLE CUSTOMER(
  CID INTEGER PRIMARY KEY,
  NAME VARCHAR(30) NOT NULL,
  AID INTEGER references ADDRESS(AID) NOT NULL
);

CREATE TABLE USED_FOR(
  TID INTEGER references TOOLS(TID) NOT NULL,
  PID INTEGER references PRODUCT(PID) NOT NULL
);

CREATE TABLE GARDEN_OWNERS(
  GID INTEGER PRIMARY KEY,
  NAME VARCHAR(30) NOT NULL,
  number char(10) NOT NULL CHECK(LENGTH(number) =10),
  DURATION INTEGER NOT NULL CHECK(DURATION >0  AND DURATION <13),
  AREA numeric(6,2) NOT NULL CHECK (AREA > 0),
  CHARGEPERMONTH INTEGER NOT NULL CHECK(CHARGEPERMONTH > 0),
  TOTAL numeric(30,2) GENERATED ALWAYS AS (DURATION * CHARGEPERMONTH) STORED
);

CREATE TABLE OWNS(
  GID INTEGER references GARDEN_OWNERS(GID) NOT NULL,
  PID INTEGER references PRODUCT(PID) NOT NULL
);

CREATE TABLE ORDERS(
  OID INTEGER PRIMARY KEY,
  ODATE DATE NOT NULL,
  OAMOUNT INTEGER NOT NULL CHECK (OAMOUNT > 0),
  QUANTITY INTEGER NOT NULL CHECK(QUANTITY > 0),
  PID INTEGER references PRODUCT(PID) NOT NULL,
  CID INTEGER references CUSTOMER(CID) NOT NULL
);
```

Data Insertion:

```
INSERT INTO address (aid, block, street, locality, city, pincode, state, country)
VALUES
  (1, 123, 'Main Street', 'XYZ Colony', 'Mumbai', 400000, 'Maharashtra', 'India'),
  (2, 456, 'Park Avenue', 'ABC Nagar', 'Delhi', 110001, 'Delhi', 'India'),
  (3, 789, 'Ocean Drive', 'LMN Society', 'Goa', 403001, 'Goa', 'India'),
  (4, 987, 'River Road', 'PQR Colony', 'Chennai', 600000, 'Tamil Nadu', 'India'),
  (5, 654, 'Mountain View', 'STU Nagar', 'Hyderabad', 500032, 'Telangana', 'India'),
  (6, 321, 'Desert Drive', 'VWX Society', 'Jaipur', 302001, 'Rajasthan', 'India'),
  (7, 741, 'Highway Road', 'YZ Colony', 'Kolkata', 700064, 'West Bengal', 'India'),
  (8, 852, 'Beach Drive', 'AB Society', 'Vishakapatnam', 530001, 'Andhra Pradesh', 'India'),
  (9, 963, 'Meadow Lane', 'CD Nagar', 'Lucknow', 226001, 'Uttar Pradesh', 'India'),
  (10, 147, 'Forest Road', 'EF Colony', 'Bengaluru', 560001, 'Karnataka', 'India'),
  (11, 256, 'Sunset Boulevard', 'GHI Nagar', 'Pune', 411001, 'Maharashtra', 'India'),
  (12, 369, 'Rainbow Road', 'JKL Society', 'Ahmedabad', 380009, 'Gujarat', 'India'),
  (13, 159, 'River Bank', 'MNO Colony', 'Surat', 395007, 'Gujarat', 'India'),
  (14,753, 'Mountain Trail', 'PQR Nagar', 'Nagpur', 440018, 'Maharashtra', 'India'),
  (15, 951, 'Valley View', 'STU Society', 'Indore', 452001, 'Madhya Pradesh', 'India'),
  (16, 864, 'Desert Mirage', 'VWX Colony', 'Bhopal', 462001, 'Madhya Pradesh', 'India'),
  (17, 729, 'Highway Hill', 'YZ Nagar', 'Chandigarh', 160017, 'Chandigarh', 'India'),
  (18, 638, 'Beach Shore', 'AB Society', 'Dehradun', 248001, 'Uttarakhand', 'India'),
  (19, 547, 'Meadow Fields', 'CD Nagar', 'Patna', 800001, 'Bihar', 'India'),
  (20, 456, 'Forest Glade', 'EF Colony', 'Ranchi', 834001, 'Jharkhand', 'India');
```

```
INSERT INTO customer (cid, name, aid)
VALUES
  (101, 'John Doe', 2),
  (102, 'Jane Smith', 1),
  (103, 'Bob Brown', 3),
  (104, 'Alice Smith', 5),
  (105, 'Tom Jones', 4),
  (106, 'Sarah Johnson', 6),
  (107, 'Michael Davis', 7),
  (108, 'Emily Martinez', 8),
  (109, 'William Anderson', 9)
  ;
```

```sql
INSERT INTO season (seid, name)
VALUES
  (1, 'Spring'),
  (2, 'Summer'),
  (3, 'Autumn'),
  (4, 'Winter'),
  (5, 'Monsoon'),
  (6,'nse');

INSERT INTO employee (eid, name, mno, salary, aid)
VALUES
  (201, 'John Doe', '9876543210', 50000, 10),
  (202, 'Jane Doe', '9765432100', 60000, 11),
  (203, 'Jim Smith', '9567432100', 70000, 12),
  (204, 'Emily Brown', '9234567890', 80000, 13),
  (205, 'Michael Johnson', '9632147850', 90000, 14),
  (206, 'Emily Davis', '9087654321', 55000, 15),
  (207, 'William Wilson', '9867123456', 65000, 16),
  (208, 'Daniel Brown', '9561237890', 75000, 17),
  (209, 'David Anderson', '9461237890', 85000, 18),
  (210, 'Richard Wilson', '9741503698', 95000, 19);

INSERT INTO tools (tid, name, mode, price)
VALUES
  (301, 'Lawn Mower', 'Manual', 1000),
  (302, 'Pruning Shears', 'Manual', 100),
  (303, 'Hedge Trimmer', 'Electric', 1500),
  (304, 'Leaf Blower', 'Gas', 2000),
  (305, 'Garden Trowel', 'Manual', 50),
  (306, 'Garden Hoe', 'Manual', 75),
  (307, 'Garden Rake', 'Manual', 80),
  (308, 'Wheelbarrow', 'Manual', 300),
  (309, 'Gardening Gloves', 'Manual', 20),
  (310, 'Watering Can', 'Manual', 25),
  (311, 'Garden Shovel', 'Manual', 75),
  (312, 'Tree Pruner', 'Manual', 200),
  (313, 'Garden Scissors', 'Manual', 60),
  (314, 'Soil Knife', 'Manual', 40),
  (315, 'Garden Spade', 'Manual', 90);
```

```
  INSERT INTO product (pid, seid)
VALUES
  (401, 1),
  (402, 2),
  (403, 3),
  (404, 4),
  (405, 5),
  (406, 1),
  (407, 2),
  (408, 3),
  (409, 4),
  (410, 5),
  (411, 1),
  (412, 2),
  (413, 3),
  (414, 4),
  (415, 5),
  (416, 1),
  (417, 2),
  (418, 3),
  (419, 4),
  (420, 5);

INSERT INTO disease (did, name, pid)
VALUES
  (501, 'Blackspot', 402),
  (502, 'Powdery Mildew', 402),
  (503, 'Rust', 403),
  (504, 'Leaf Spot', 404),
  (505, 'Bacterial Wilt', 405),
  (506, 'Late Blight', 406),
  (507, 'Downy Mildew', 407),
  (508, 'Verticillium Wilt', 408),
  (509, 'Fusarium Wilt', 409),
  (510, 'Clubroot', 410),
  (511, 'Root Rot', 411),
  (512, 'Crown Rot', 412),
  (513, 'Aphid Infestation', 413),
  (514, 'Spider Mite Infestation', 414),
  (515, 'Thrips Infestation', 415),
  (516, 'Whitefly Infestation', 415),
  (517, 'Blossom End Rot', 417),
  (518, 'Fire Blight', 418),
  (519, 'Scab', 419),
  (520, 'Verticillium Wilt',420),
```

```
  (521, 'Damping Off', 411),
  (522, 'Virus Infection', 412);


INSERT INTO medicine (name, cost, company, did)
VALUES
  ('Amoxicillin', 15, 'XYZ Laboratories', 501),
  ('Metronidazole', 20, 'PQR Inc.', 502),
  ('Ciprofloxacin', 25, 'DEF Corp', 503),
  ('Clindamycin', 30, 'GHI Enterprises', 504),
  ('Azithromycin', 35, 'JKL Ltd', 505),
  ('Erythromycin', 40, 'MNO Inc', 506),
  ('Doxycycline', 45, 'STU Enterprises', 507),
  ('Minocycline', 50, 'VWX Pvt Ltd', 508),
  ('Tetracycline', 55, 'YZ Company', 509),
  ('Levofloxacin', 60, 'ABC Limited', 510),
  ('Linezolid', 65, 'DEF Pvt Ltd', 511),
  ('Nitrofurantoin', 70, 'GHI Corp', 512),
  ('Penicillin', 75, 'JKL Inc', 513),
  ('Sulfamethoxazole', 80, 'MNO Laboratories', 514),
  ('Trimethoprim', 85, 'PQR Enterprises', 515),
  ('Cefuroxime', 90, 'STU Corp', 516),
  ('Cefpodoxime', 95, 'VWX Ltd',517),
  ('Cefixime', 100, 'YZ Pvt Ltd',518),
  ('Ceftriaxone', 105, 'ABC Limited', 519),
  ('Ceftazidime', 110, 'DEF Enterprises',520);


INSERT INTO FRUITS (PID, NAME, SEID)VALUES
(401, 'Apple', 4),
(402, 'Banana', 6),
(403, 'Mango', 2),
(404, 'Papaya', 2),
(405, 'Grapes', 4),
(420,'Watermelon',2);


INSERT INTO FLOWERS (PID, NAME,HABITATE, SEID)VALUES
(411,'Rose','Garden',5),
(412,'Marigold','Wild',3),
(413,'Hibiscus','Tropical',4),
(414,'Jasmin','Temperate',2),
(415,'Lotus','Woodland',2);
```

```
INSERT INTO BERRIES (PID,NAME,SEID) VALUES
(406,'Blackberry',5),
(407,'Blueberry',3),
(408,'Craneberry',1),
(409,'Avocado',1),
(410,'Tayberry',4),
(416,'Strawberry',4),
(417,'Raspberry',2),
(418,'Lychee',2),
(419,'abcberry',5);

INSERT INTO MANAGER (MID) VALUES
(210),
(209),
(208);

INSERT INTO MANAGES(MID,PID) VALUES
(210,401),
(210,411),
(208,416),
(209,417),
(210,402),
(209,403),
(210,404),
(208,405),
(209,406),
(209,407),
(208,408),
(209,409),
(210,410),
(208,412),
(208,413),
(210,414),
(210,415),
(209,418),
(209,419),
(208,420);

INSERT INTO USED_FOR(TID,PID) VALUES
(301,411),
(303,412),
(302,413),
(301,414),
(302,401),
(305,402),
```

```
(303,403),
(305,404),
(304,405),
(306,412),
(310,417),
(310,420),
(315,410),
(313,417),
(314,418);


  insert into GARDEN_OWNERS(gid, name, number,DURATION, AREA, CHARGEPERMONTH)
VALUES
  (101, 'Nisarg', 9999999999, 12,50,50000),
  (102, 'Snehal', 9998999999, 10,22,5000),
  (103, 'Akshay', 9999999989, 6,100,100000),
  (104, 'ABC', 9999999999, 12,120,80000),
  (105, 'XYZ', 9999999999, 5,500,125000);

 insert into owns(gid, pid) VALUES
 (101,402),
 (101,405),
 (102,412),
 (103,415),
 (103,404),
 (104,405),
 (105,420);

 insert into ORDERS(oid, odate, oamount, QUANTITY,pid, cid) VALUES
 (201,'2023-02-09', 20000,12, 402, 101),
(202,'2023-02-14', 29000,50, 405, 102),
(203,'2023-02-15', 1000,3, 412, 105),
(204,'2023-02-10', 89000,50, 415, 107),
(205,'2023-02-12', 36000,23, 420, 106),
(206,'2023-01-10', 96000,100,415,109);
```

5.Inserted Data:

1.Address

```
postgres=# select * from address;
 aid | block |      street      |  locality   |     city      | pincode |     state      | country
-----+-------+------------------+-------------+---------------+---------+----------------+---------
   1 | 123   | Main Street      | XYZ Colony  | Mumbai        |  400000 | Maharashtra    | India
   2 | 456   | Park Avenue      | ABC Nagar   | Delhi         |  110001 | Delhi          | India
   3 | 789   | Ocean Drive      | LMN Society | Goa           |  403001 | Goa            | India
   4 | 987   | River Road       | PQR Colony  | Chennai       |  600000 | Tamil Nadu     | India
   5 | 654   | Mountain View    | STU Nagar   | Hyderabad     |  500032 | Telangana      | India
   6 | 321   | Desert Drive     | VWX Society | Jaipur        |  302001 | Rajasthan      | India
   7 | 741   | Highway Road     | YZ Colony   | Kolkata       |  700064 | West Bengal    | India
   8 | 852   | Beach Drive      | AB Society  | Vishakapatnam |  530001 | Andhra Pradesh | India
   9 | 963   | Meadow Lane      | CD Nagar    | Lucknow       |  226001 | Uttar Pradesh  | India
  10 | 147   | Forest Road      | EF Colony   | Bengaluru     |  560001 | Karnataka      | India
  11 | 256   | Sunset Boulevard | GHI Nagar   | Pune          |  411001 | Maharashtra    | India
  12 | 369   | Rainbow Road     | JKL Society | Ahmedabad     |  380009 | Gujarat        | India
  13 | 159   | River Bank       | MNO Colony  | Surat         |  395007 | Gujarat        | India
  14 | 753   | Mountain Trail   | PQR Nagar   | Nagpur        |  440018 | Maharashtra    | India
  15 | 951   | Valley View      | STU Society | Indore        |  452001 | Madhya Pradesh | India
  16 | 864   | Desert Mirage    | VWX Colony  | Bhopal        |  462001 | Madhya Pradesh | India
  17 | 729   | Highway Hill     | YZ Nagar    | Chandigarh    |  160017 | Chandigarh     | India
  18 | 638   | Beach Shore      | AB Society  | Dehradun      |  248001 | Uttarakhand    | India
  19 | 547   | Meadow Fields    | CD Nagar    | Patna         |  800001 | Bihar          | India
  20 | 456   | Forest Glade     | EF Colony   | Ranchi        |  834001 | Jharkhand      | India
(20 rows)
```

2.Products

```
postgres=# select * from product;
 pid | seid
-----+------
 401 |    1
 402 |    2
 403 |    3
 404 |    4
 405 |    5
 406 |    1
 407 |    2
 408 |    3
 409 |    4
 410 |    5
 411 |    1
 412 |    2
 413 |    3
 414 |    4
 415 |    5
 416 |    1
 417 |    2
 418 |    3
 419 |    4
 420 |    5
(20 rows)
```

3.Season

```
postgres=# select * from season;
 seid |  name
------+---------
    1 | Spring
    2 | Summer
    3 | Autumn
    4 | Winter
    5 | Monsoon
    6 | nse
(6 rows)
```

4.Berries

```
postgres=# select * from berries;
 pid |    name    | seid
-----+------------+------
 406 | Blackberry |    5
 407 | Blueberry  |    3
 408 | Craneberry |    1
 409 | Avocado    |    1
 410 | Tayberry   |    4
 416 | Strawberry |    4
 417 | Raspberry  |    2
 418 | Lychee     |    2
 419 | abcberry   |    5
(9 rows)
```

5.Flowers

```
postgres=# select * from flowers;
 pid |   name    | habitate  | seid
-----+-----------+-----------+------
 411 | Rose      | Garden    |    5
 412 | Marigold  | Wild      |    3
 413 | Hibiscus  | Tropical  |    4
 414 | Jasmin    | Temperate |    2
 415 | Lotus     | Woodland  |    2
(5 rows)
```

6.Fruits

```
postgres=# select * from fruits;
 pid |    name    | seid
-----+------------+------
 401 | Apple      |    4
 402 | Banana     |    6
 403 | Mango      |    2
 404 | Papaya     |    2
 405 | Grapes     |    4
 420 | Watermelon |    2
(6 rows)
```

7.Customers

```
postgres=# select * from customer;
 cid |       name       | aid
-----+------------------+-----
 101 | John Doe         |   2
 102 | Jane Smith       |   1
 103 | Bob Brown        |   3
 104 | Alice Smith      |   5
 105 | Tom Jones        |   4
 106 | Sarah Johnson    |   6
 107 | Michael Davis    |   7
 108 | Emily Martinez   |   8
 109 | William Anderson |   9
(9 rows)
```

8.Disease

```
postgres=# select * from disease;
 did |          name           | pid
-----+-------------------------+-----
 501 | Blackspot               | 402
 502 | Powdery Mildew          | 402
 503 | Rust                    | 403
 504 | Leaf Spot               | 404
 505 | Bacterial Wilt          | 405
 506 | Late Blight             | 406
 507 | Downy Mildew            | 407
 508 | Verticillium Wilt       | 408
 509 | Fusarium Wilt           | 409
 510 | Clubroot                | 410
 511 | Root Rot                | 411
 512 | Crown Rot               | 412
 513 | Aphid Infestation       | 413
 514 | Spider Mite Infestation | 414
 515 | Thrips Infestation      | 415
 516 | Whitefly Infestation    | 415
 517 | Blossom End Rot         | 417
 518 | Fire Blight             | 418
 519 | Scab                    | 419
 520 | Verticillium Wilt       | 420
 521 | Damping Off             | 411
 522 | Virus Infection         | 412
(22 rows)
```

9.Employees

```
postgres=# select * from employee;
 eid |      name       |    mno     | salary | aid
-----+-----------------+------------+--------+-----
 201 | John Doe        | 9876543210 |  50000 |  10
 202 | Jane Doe        | 9765432100 |  60000 |  11
 203 | Jim Smith       | 9567432100 |  70000 |  12
 204 | Emily Brown     | 9234567890 |  80000 |  13
 205 | Michael Johnson | 9632147850 |  90000 |  14
 206 | Emily Davis     | 9087654321 |  55000 |  15
 207 | William Wilson  | 9867123456 |  65000 |  16
 208 | Daniel Brown    | 9561237890 |  75000 |  17
 209 | David Anderson  | 9461237890 |  85000 |  18
 210 | Richard Wilson  | 9741503698 |  95000 |  19
(10 rows)
```

10.Garden Owners

```
postgres=# select * from garden_owners;
 gid |  name  |   number   | duration |  area  | chargepermonth |   total
-----+--------+------------+----------+--------+----------------+-----------
 101 | Nisarg | 9999999999 |       12 |  50.00 |          50000 | 600000.00
 102 | Snehal | 9998999999 |       10 |  22.00 |           5000 |  50000.00
 103 | Akshay | 9999999989 |        6 | 100.00 |         100000 | 600000.00
 104 | ABC    | 9999999999 |       12 | 120.00 |          80000 | 960000.00
 105 | XYZ    | 9999999999 |        5 | 500.00 |         125000 | 625000.00
(5 rows)
```

11.Managers

```
postgres=# select * from manager;
 mid
-----
 210
 209
 208
(3 rows)
```

12. Manages

```
postgres=# select * from manages;
 mid | pid
-----+-----
 210 | 401
 210 | 411
 208 | 416
 209 | 417
 210 | 402
 209 | 403
 210 | 404
 208 | 405
 209 | 406
 209 | 407
 208 | 408
 209 | 409
 210 | 410
 208 | 412
 208 | 413
 210 | 414
 210 | 415
 209 | 418
 209 | 419
 208 | 420
(20 rows)
```

13.Medicine

```
postgres=# select * from medicine;
       name       | cost |     company      | did
------------------+------+------------------+-----
 Amoxicillin      |   15 | XYZ Laboratories | 501
 Metronidazole    |   20 | PQR Inc.         | 502
 Ciprofloxacin    |   25 | DEF Corp         | 503
 Clindamycin      |   30 | GHI Enterprises  | 504
 Azithromycin     |   35 | JKL Ltd          | 505
 Erythromycin     |   40 | MNO Inc          | 506
 Doxycycline      |   45 | STU Enterprises  | 507
 Minocycline      |   50 | VWX Pvt Ltd      | 508
 Tetracycline     |   55 | YZ Company       | 509
 Levofloxacin     |   60 | ABC Limited      | 510
 Linezolid        |   65 | DEF Pvt Ltd      | 511
 Nitrofurantoin   |   70 | GHI Corp         | 512
 Penicillin       |   75 | JKL Inc          | 513
 Sulfamethoxazole |   80 | MNO Laboratories | 514
 Trimethoprim     |   85 | PQR Enterprises  | 515
 Cefuroxime       |   90 | STU Corp         | 516
 Cefpodoxime      |   95 | VWX Ltd          | 517
 Cefixime         |  100 | YZ Pvt Ltd       | 518
 Ceftriaxone      |  105 | ABC Limited      | 519
 Ceftazidime      |  110 | DEF Enterprises  | 520
```

14.Orders

```
postgres=# select * from orders;
 oid |   odate    | oamount | quantity | pid | cid
-----+------------+---------+----------+-----+-----
 201 | 2023-02-09 |   20000 |       12 | 402 | 101
 202 | 2023-02-14 |   29000 |       50 | 405 | 102
 203 | 2023-02-15 |    1000 |        3 | 412 | 105
 204 | 2023-02-10 |   89000 |       50 | 415 | 107
 205 | 2023-02-12 |   36000 |       23 | 420 | 106
 206 | 2023-01-10 |   96000 |      100 | 415 | 109
(6 rows)
```

15.Owns

```
postgres=# select * from owns;
 gid | pid
-----+-----
 101 | 402
 101 | 405
 102 | 412
 103 | 415
 103 | 404
 104 | 405
 105 | 420
 105 | 406
 105 | 419
 104 | 418
 101 | 406
(11 rows)
```

16.Tools

```
postgres=# select * from tools;
 tid |       name       |   mode   | price
-----+------------------+----------+-------
 301 | Lawn Mower       | Manual   |  1000
 302 | Pruning Shears   | Manual   |   100
 303 | Hedge Trimmer    | Electric |  1500
 304 | Leaf Blower      | Gas      |  2000
 305 | Garden Trowel    | Manual   |    50
 306 | Garden Hoe       | Manual   |    75
 307 | Garden Rake      | Manual   |    80
 308 | Wheelbarrow      | Manual   |   300
 309 | Gardening Gloves | Manual   |    20
 310 | Watering Can     | Manual   |    25
 311 | Garden Shovel    | Manual   |    75
 312 | Tree Pruner      | Manual   |   200
 313 | Garden Scissors  | Manual   |    60
 314 | Soil Knife       | Manual   |    40
 315 | Garden Spade     | Manual   |    90
(15 rows)
```

17.Used for

```
postgres=# select * from used_for;
 tid | pid
-----+-----
 301 | 411
 303 | 412
 302 | 413
 301 | 414
 302 | 401
 305 | 402
 303 | 403
 305 | 404
 304 | 405
 306 | 412
 310 | 417
 310 | 420
 315 | 410
 313 | 417
 314 | 418
(15 rows)
```

6.QUERIES USING BASIC DBMS CONSTRUCTS JOIN & SUBQUERIES:

1. select t.name from (tools t join used_for u on u.tid=t.tid)join fruits f on f.pid=u.pid;

```
      name
----------------
 Pruning Shears
 Garden Trowel
 Hedge Trimmer
 Garden Trowel
 Leaf Blower
 Watering Can
```

2. select e.name from (employee e join manages m ON m.mid=e.eid) where m.mid in (select m.mid from manages m join fruits f on f.pid=m.pid);

```
      name
---------------
 Richard Wilson
 Richard Wilson
 Daniel Brown
 David Anderson
 Richard Wilson
 David Anderson
 Richard Wilson
```

Daniel Brown

David Anderson

David Anderson

Daniel Brown

David Anderson

Richard Wilson

Daniel Brown

Daniel Brown

Richard Wilson

Richard Wilson

David Anderson

David Anderson

Daniel Brown

3. select c.cid,c.name from customer c where c.name like 'J%';

```
  cid |    name
  -----+------------
   101 | John Doe
   102 | Jane Smith
```

4. select cid,c.name from customer c where c.cid in (select cid from orders where oamount=(select max(oamount) from orders));

```
 cid |       name
-----+------------------
 109 | William Anderson
```

5. select c.name,o.* from customer c join orders o on o.cid=c.cid order by odate limit 1;

```
          name       | oid |   odate    | oamount | quantity | pid | cid
-----------------+-----+------------+---------+----------+-----+-----
 William Anderson | 206 | 2023-01-10 |   96000 |      100 | 415 | 109
```

Queries Given by Madam:

1. select name from garden_owners where gid in(select o.gid from owns o join berries b on o.pid=b.pid);

```
  name
--------
 Nisarg
 ABC
```

XYZ

1. select seid,count(*) from product group by seid order by seid;

seid	count
1	4
2	4
3	4
4	4
5	4

2. select distinct e.name,t.name from (employee e join manages m ON m.mid=e.eid), (tools t join used_for u on u.tid=t.tid)join fruits f on f.pid=u.pid where m.mid in (select m.mid from manages m join fruits f on f.pid=m.pid);

name	name
Richard Wilson	Leaf Blower
Daniel Brown	Hedge Trimmer
Daniel Brown	Watering Can
Richard Wilson	Hedge Trimmer
Richard Wilson	Watering Can

Richard Wilson | Garden Trowel

David Anderson | Hedge Trimmer

David Anderson | Leaf Blower

Daniel Brown | Pruning Shears

Daniel Brown | Garden Trowel

Daniel Brown | Leaf Blower

David Anderson | Pruning Shears

David Anderson | Watering Can

David Anderson | Garden Trowel

Richard Wilson | Pruning Shears

7.Pl/PGSQL queries:

Trigger 1:

Create a trigger to prevent the deletion of a row from the PRODUCT table if there are any corresponding rows in the ORDERS table.

```
CREATE OR REPLACE FUNCTION prevent_deletion() RETURNS trigger AS $$
DECLARE
  orders_count INTEGER;
BEGIN
  SELECT COUNT(*) INTO orders_count
  FROM orders
  WHERE pid = OLD.pid;
  IF orders_count > 0 THEN
    RAISE EXCEPTION 'Cannot delete product with associated orders.';
  END IF;
  RETURN OLD;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER prevent_deletion
BEFORE DELETE ON product
FOR EACH ROW
EXECUTE FUNCTION prevent_deletion();
```

Triger 2:

When new employee is inserted, increase salary of previous employees.

```
CREATE OR REPLACE FUNCTION update_employee_salary()
RETURNS TRIGGER AS $$
BEGIN
  UPDATE employee
  SET salary = salary + (salary * 0.1)
  WHERE eid = NEW.eid;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER increase_employee_salary
AFTER INSERT ON employee
FOR EACH ROW
EXECUTE FUNCTION update_employee_salary();
```

Cursor:

Create a cursor to calculate tax on salary of employees.

```
create or replace procedure calc_tax()
language plpgsql
as $$
declare
rec1 record;
tax integer;
cur1 cursor for select * from employee ;
begin
    open cur1;
     loop
        fetch cur1 into rec1;
        exit when not found;
        tax := rec1.salary*0.3;
```

```
            raise notice 'NAME = %',rec1.name;
            raise NOTICE 'Tax = %',tax;
         end loop;
        close cur1;
    end; $$;
```

Function 1:

Fetch the average salary of employees of one city

```
CREATE OR REPLACE FUNCTION get_avg_salary_by_city(ucity VARCHAR(25))
RETURNS INTEGER AS $$
DECLARE
    avg_salary INTEGER;
BEGIN
    SELECT AVG(SALARY)
    INTO avg_salary
    FROM EMPLOYEE e
    INNER JOIN ADDRESS a ON e.AID = a.AID
    WHERE a.CITY = ucity;

    RETURN COALESCE(avg_salary, 0);
END;
$$ LANGUAGE plpgsql;
```

```
postgres=# select get_avg_salary_by_city('Surat');
 get_avg_salary_by_city
------------------------
                  80000
(1 row)
```

Function 2:

Get the revenue generated a product

```
CREATE OR REPLACE FUNCTION get_garden_owner_revenue(upid INTEGER)
RETURNS numeric(30,2) AS $$
DECLARE
    total numeric(30,2);
BEGIN
    SELECT SUM(OAMOUNT * QUANTITY)
    INTO total
    FROM ORDERS
    WHERE PID = upid;
```

```
    RETURN COALESCE(total, 0);
END;
$$ LANGUAGE plpgsql;
```

```
postgres=# select * from flowers;
 pid |   name    | habitate  | seid
-----+-----------+-----------+------
 411 | Rose      | Garden    |    5
 412 | Marigold  | Wild      |    3
 413 | Hibiscus  | Tropical  |    4
 414 | Jasmin    | Temperate |    2
 415 | Lotus     | Woodland  |    2
(5 rows)


postgres=# select get_garden_owner_revenue(415);
 get_garden_owner_revenue
--------------------------
              14050000.00
(1 row)
```