

**A Project Report on
Electronics Shop Management System
“ ESMS ”**

Developed By:



IT152

Solanki Apar

IT154

Soni Neel

IT151

Shinde Rahul

Guided By

Internal Guide:

Prof. Archana N. Vyas



**Department of Information Technology Faculty of Technology,
DharmSinh Desai University College Road, Nadiad-387001**

October-2022

CERTIFICATE

THIS IS TO CERTIFY THAT THE PROJECT ENTITLED
“ELECTRONICS STOCK MANAGEMENT SYSTEM” IS,
A BONAFIDE REPORT OF THE WORK CARRIED OUT BY

- | | |
|------------------------|----------------------------------|
| 1) Solanki Apar | Student ID No: 20ITUOS060 |
| 2) Shinde Rahul | Student ID No: 20ITUOS004 |
| 3) Soni Neel | Student ID No: 20ITUOS018 |

OF DEPARTMENT OF INFORMATION TECHNOLOGY, SEMESTER V,
UNDER THE GUIDANCE AND
SUPERVISION FOR THE SUBJECT DATABASE MANAGEMENT SYSTEM.
THEY WERE INVOLVED IN PROJECT TRAINING DURING THE ACADEMIC
YEAR 2022-2023.



COMMENDATION

We would like to express our heartfelt gratitude to everyone who contributed to the successful completion of our project
"**ESMS**"

The success and ultimate conclusion of this project necessitated a great deal of advice and support from a large number of individuals and we are incredibly fortunate to have received it all along with the project's completion.

We owe a debt of appreciation to **Prof. Archana N. Vyas**, our project guide, who took an interest in our project work and directed us through it till it was completed by giving all of the required assistance for creating a solid Database System.

We'd also want to express our gratitude to all of our speakers. Finally, we express our gratitude to all of our friends and colleagues.

INDEX

I) Certificate

II) Commendation

1) System Overview

1.1) Current System & Objective of System

1.2) Advantage of System

2) E-R Diagram

3) Schema Diagram

4) Data Dictionary

5) Database Implementation

5.1) Create Schema

5.2) Insert Data Values

5.3) Queries DBMS (Based on constructs)

5.4) Queries DBMS (Based On joins and Subqueries)

5.5) PL/SQL Blocks (Views)

5.6) Functions and Trigger

5.7) Cursors

6) Future Enhancement System

7) Bibliography

Source Code = > Electronics_Stock_Management_System

1) SYSTEM OVERVIEW

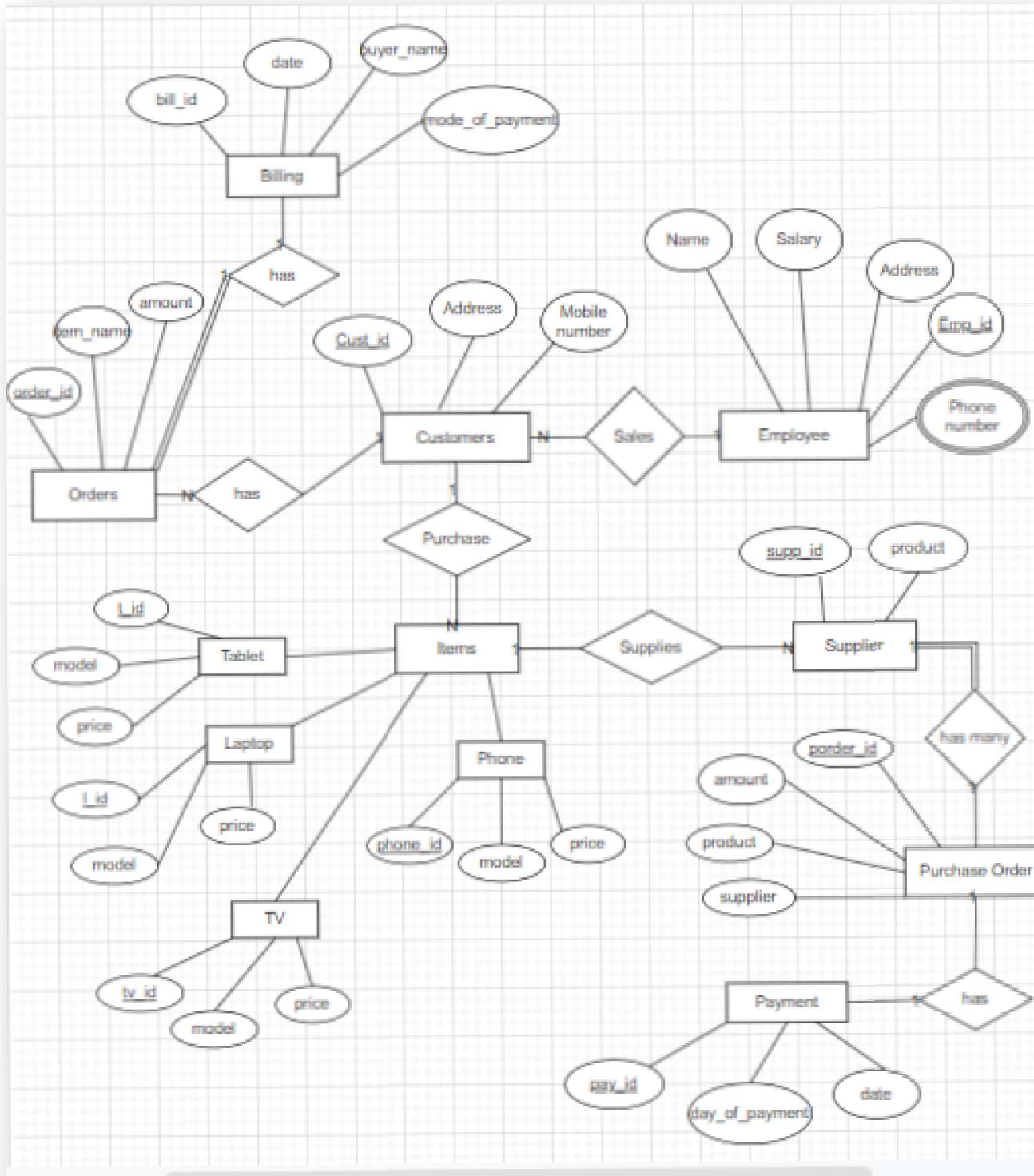
1.1) Current System & Objectives

Our database will be designed for companies or businessmen who are dealing with Clients and suppliers daily. Our major focus will be on structuring this data in such a way that the Person who is the owner of this Application/Database can easily manage their stocks up-to-date & also this help in maintaining whole Money transactions from both side Clients & Suppliers. It will also be done in an effective manner since we do not want our database to become redundant.

1.2) Advantage of System

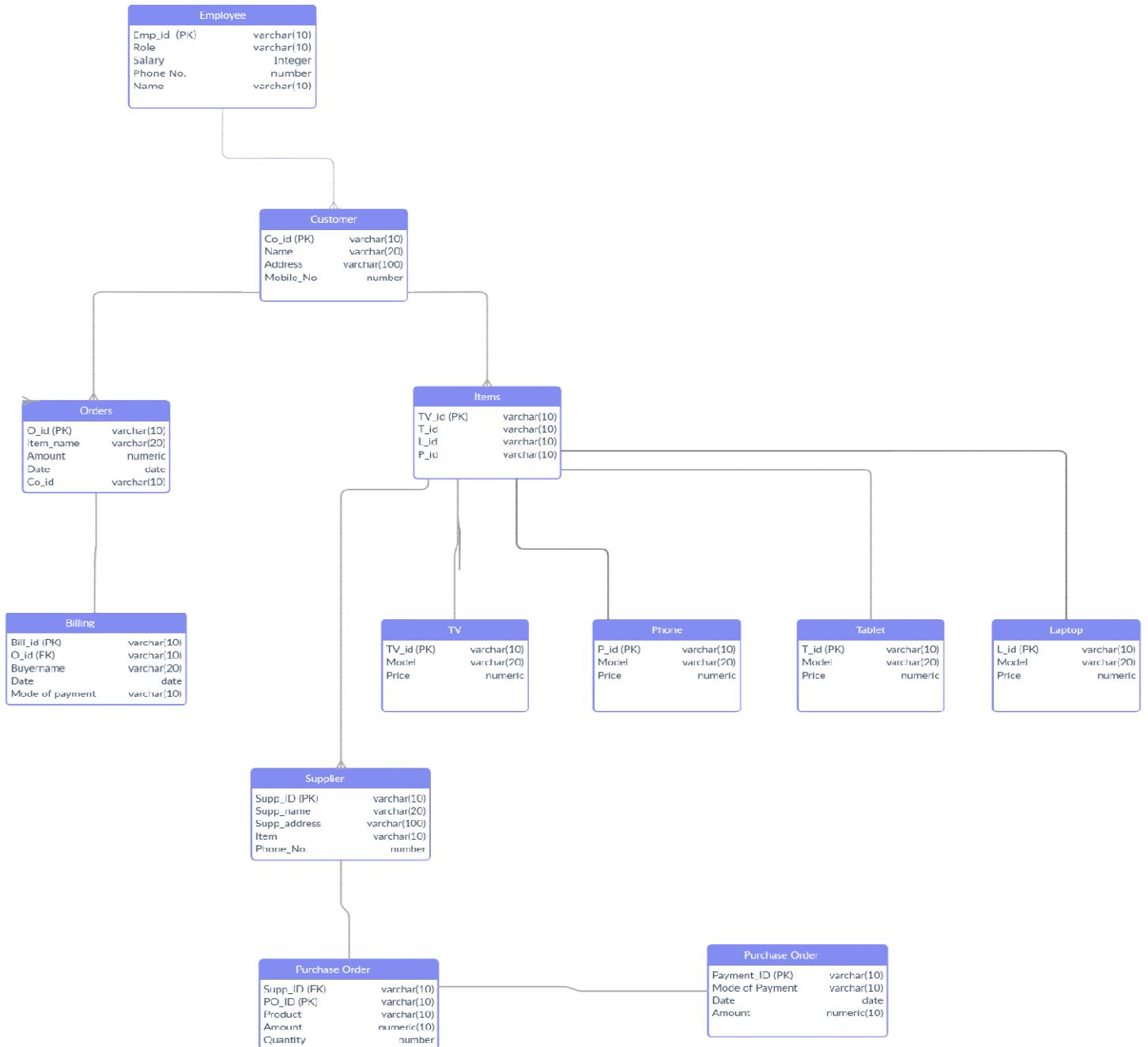
ESMS is Server and Clients Side Application. All users can purchase Electronics systems online like Flip kart - Amazon. With Also Supplier Owner Can purchase Products from the Supplier or Dealer. Both Users and Supplier Details related to Products or Payments everything manages accurately.

2) ER-Entity Relational Diagram



[Link => ER Diagram](#)

3) Relational Schema Diagram



Link => [Relational_Schema](#)

4) Data Dictionary

4.1) Billing

```
project=# \d billing;
          Table "public.billing"
 Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 bill_id     | character varying(10) |
 buyer_name  | character varying(30) |
 mode_of_payment | character varying(10) |
 date_of_purchase | date |
Foreign-key constraints:
 "billing_bill_id_fkey" FOREIGN KEY (bill_id) REFERENCES orders(oid)
```

4.2) Customer

```
project=# \d customer;
          Table "public.customer"
 Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 cid        | character varying(10) |
 c_address  | character varying(50) |
 c_name     | character varying(20) |
 ph_no      | numeric(10,0) |
Indexes:
 "customer_pkey" PRIMARY KEY, btree (cid)
```

4.3) Employee

```
project=# \d employee;
          Table "public.employee"
 Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 emp_id     | character varying(10) |
 role        | character varying(20) |
 phone_no   | numeric(10,0) |
 name        | character varying(20) |
 salary      | numeric(10,0) |
Indexes:
 "employee_pkey" PRIMARY KEY, btree (emp_id)
```

4.4) Items

```
project=# \d items;
           Table "public.items"
  Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
p_id   | character varying(10) |
l_id   | character varying(10) |
t_id   | character varying(10) |
tv_id  | character varying(10) |

Foreign-key constraints:
  "items_l_id_fkey" FOREIGN KEY (l_id) REFERENCES laptop(laptop_id)
  "items_p_id_fkey" FOREIGN KEY (p_id) REFERENCES phone(phone_id)
  "items_t_id_fkey" FOREIGN KEY (t_id) REFERENCES tablet(tablet_id)
  "items_tv_id_fkey" FOREIGN KEY (tv_id) REFERENCES tv(tv_id)
```

4.5) Laptop

```
project=# \d laptop;
           Table "public.laptop"
  Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
laptop_id | character varying(10) |          | not null |
laptop_model | character varying(30) |
laptop_company | character varying(20) |
laptop_price | numeric(10,0) |

Indexes:
  "laptop_pkey" PRIMARY KEY, btree (laptop_id)
Referenced by:
  TABLE "items" CONSTRAINT "items_l_id_fkey" FOREIGN KEY (l_id) REFERENCES laptop(laptop_id)
```

4.6) Orders

```
project=# \d orders;
           Table "public.orders"
  Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
oid    | character varying(10) |          | not null |
item_name | character varying(20) |
amount | numeric(10,0) |

Indexes:
  "orders_pkey" PRIMARY KEY, btree (oid)
Referenced by:
  TABLE "billing" CONSTRAINT "billing_bill_id_fkey" FOREIGN KEY (bill_id) REFERENCES orders(oid)
```

4.7) Payment

```
project=# \d payment;
           Table "public.payment"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
payment_id   | character varying(10) |          | not null |
mode_of_payment | character varying(10) |          |          |
date         | date            |          |          |
amount        | numeric(10,0)    |          |          |
Indexes:
  "payment_pkey" PRIMARY KEY, btree (payment_id)
```

4.8) Phone

```
project=# \d phone;
           Table "public.phone"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
phone_id     | character varying(10) |          | not null |
phone_model  | character varying(30) |          |          |
phone_company | character varying(20) |          |          |
phone_price   | numeric(10,0)    |          |          |
Indexes:
  "phone_pkey" PRIMARY KEY, btree (phone_id)
Referenced by:
  TABLE "items" CONSTRAINT "items_p_id_fkey" FOREIGN KEY (p_id) REFERENCES phone(phone_id)
```

4.9) Purchase_order

```
project=# \d purchase_order;
           Table "public.purchase_order"
  Column      |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
purchase_id   | character varying(10) |          | not null |
s_id          | character varying(10) |          |          |
pur_product   | character varying(20) |          |          |
quantity      | numeric(10,0)    |          |          |
amount        | numeric(10,0)    |          |          |
Indexes:
  "purchase_order_pkey" PRIMARY KEY, btree (purchase_id)
Foreign-key constraints:
  "purchase_order_s_id_fkey" FOREIGN KEY (s_id) REFERENCES supplier(s_id)
```

4.10) Supplier

```

project=# \d supplier;
      Table "public.supplier"
 Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 s_id   | character varying(10) |           | not null |
 supplier_name | character varying(30) |           |
 supplier_address | character varying(20) |           |
 office_no | numeric(10,0) |           |
 product   | character varying(20) |           |
Indexes:
 "supplier_pkey" PRIMARY KEY, btree (s_id)
Referenced by:
 TABLE "purchase_order" CONSTRAINT "purchase_order_s_id_fkey" FOREIGN KEY (s_id) REFERENCES supplier(s_id)

```

4.11) Tablet

```

project=# \d tablet;
      Table "public.tablet"
 Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 tablet_id | character varying(10) |           | not null |
 tablet_model | character varying(30) |           |
 tablet_company | character varying(20) |           |
 tablet_price | numeric(10,0) |           |
Indexes:
 "tablet_pkey" PRIMARY KEY, btree (tablet_id)
Referenced by:
 TABLE "items" CONSTRAINT "items_t_id_fkey" FOREIGN KEY (t_id) REFERENCES tablet(tablet_id)

```

4.12) Tv

```

project=# \d tv;
      Table "public.tv"
 Column |      Type       | Collation | Nullable | Default
-----+-----+-----+-----+-----+
 tv_id   | character varying(10) |           | not null |
 tv_model | character varying(30) |           |
 tv_company | character varying(20) |           |
 tv_price | numeric(10,0) |           |
Indexes:
 "tv_pkey" PRIMARY KEY, btree (tv_id)
Referenced by:
 TABLE "items" CONSTRAINT "items_tv_id_fkey" FOREIGN KEY (tv_id) REFERENCES tv(tv_id)

```

5) Data Implementation

* First Create Database name -> Project *

```
Creat Database Project;
```

5.1) Schema => Total 12 Table

1) Employee

```
CREATE table Employee( Emp_id varchar(10) PRIMARY KEY, role  
varchar(20), phone_no numeric(10),name varchar(20), salary numeric(10));
```

2)Customer

```
CREATE TABLE Customer (CID varchar(10) PRIMARY KEY,c_address  
varchar(50), c_name varchar(20), ph_no numeric(10));
```

3)Orders

```
create table Orders(OID varchar(10) PRIMARY KEY,Item_name varchar(20),  
amount numeric(10));
```

4) Billing

```
Create table Billing( Bill_ID varchar(10) REFERENCES  
Orders(OID),Buyer_name varchar(30),mode_of_payment  
varchar(10),date_of_purchase Date);
```

5) Tv

```
Create table Tv(Tv_id varchar(10) primary key,Tv_model  
varchar(30),Tv_company varchar(20),Tv_price numeric(10));
```

6) Phone

```
Create table Phone( Phone_id varchar(10) primary key, Phone_model  
varchar(30), Phone_company varchar(20), Phone_price numeric(10));
```

7) Laptop

```
Create table Laptop( Laptop_id varchar(10) primary key, Laptop_model  
varchar(30),Laptop_company varchar(20),Laptop_price numeric(10));
```

8) Tablet

```
Create table Tablet(Tablet_id varchar(10) primary key,Tablet_model  
varchar(30),Tablet_company varchar(20),Tablet_price numeric(10));
```

9) Items

```
Create table Items(P_id varchar(10) REFERENCES Phone(Phone_id), L_id  
varchar(10) REFERENCES Laptop(Laptop_id), T_id varchar(10) REFERENCES  
Tablet(Tablet_id), Tv_id varchar(10) REFERENCES Tv(Tv_id) );
```

10) Supplier

```
Create table Supplier(S_id varchar(10) primary key,supplier_name  
varchar(30),supplier_address varchar(20), office_no numeric(10),product  
varchar(20));
```

11) Purchaese_Order

```
Create table Purchase_order(Purchase_id varchar(10) primary key,S_id  
varchar(10) REFERENCES Supplier(S_id), pur_product varchar(20), quantity  
numeric(10), amount numeric(10));
```

12)Payment

```
Create table Payment( payment_id varchar(10) PRIMARY  
KEY,mode_of_payment varchar(10), date DATE,amount numeric(10));
```

5.2) Inserting & Displaying Values

1) Billing

```
project=# select * from billing;
 bill_id | buyer_name | mode_of_payment | date_of_purchase
-----+-----+-----+-----+
 1      | Heet       | Online          | 2022-02-01
 2      | Priyam     | Online          | 2022-04-01
 3      | Nisarg     | Online          | 2022-06-01
 4      | het        | Online          | 2022-08-01
(4 rows)
```

2) Customer

```
project=# select * from customer;
 cid   | c_address | c_name | ph_no
-----+-----+-----+-----+
 C101 | Vadodara | Heet   | 887654321
 C102 | Rajkot    | Priyam | 887654322
 C103 | Surat     | Nisarg | 887654323
 C104 | Ahemdabad | Het    | 887654324
(4 rows)
```

3) Employee

```
project=# select * from employee;
 emp_id | role      | phone_no | name      | salary
-----+-----+-----+-----+-----+
 E101  | General   | 987654321 | MaganBhai | 100000
 E102  | General   | 987654322 | ChhaganBhai | 150000
 E103  | General   | 987654323 | KathanBhai | 100000
 E104  | Manager   | 987654324 | SureshBhai | 200000
(4 rows)
```

4) TV

```
project=# select * from tv;
 tv_id | tv_model | tv_company | tv_price
-----+-----+-----+-----+
 T1   | B       | MI        | 40000
 T2   | B       | SONY      | 100000
 T3   | A       | APPLE     | 200000
(3 rows)
```

5) Tablet

```
project=# select * from tablet;
tablet_id | tablet_model | tablet_company | tablet_price
-----+-----+-----+-----+
 1 | A | Dell | 10000
 2 | B | Dell | 15000
 3 | A | Apple | 80000
 4 | B | Apple | 60000
(4 rows)
```

6) Supplier

```
project=# select * from supplier;
 s_id | supplier_name | supplier_address | office_no | product
-----+-----+-----+-----+-----+
 1020304050 | CompanyA | Rajkot | 1010 | Phone
 1020304051 | CompanyB | Porbandar | 2010 | Laptop
 1020304052 | CompanyC | Nadiad | 4010 | Laptop
 1020304053 | CompanyD | Surat | 3010 | Tablet
(4 rows)
```

7) Purchase_Order

```
project=# select * from purchase_order;
purchase_id | s_id      | pur_product | quantity | amount
-----+-----+-----+-----+-----+
P1          | 1020304050 | Phone       | 5        | 50000
P2          | 1020304051 | Laptop      | 2        | 100000
P3          | 1020304052 | Laptop      | 1        | 50000
P4          | 1020304053 | Tablet      | 1        | 10000
(4 rows)
```

8) Phone

```
project=# select * from phone;
phone_id | phone_model | phone_company | phone_price
-----+-----+-----+-----+
1         | 101        | Samsung      | 10000
2         | 102        | Samsung      | 20000
3         | 201        | Micromax    | 10000
4         | 202        | Micromax    | 15000
5         | 301        | Apple        | 100000
6         | 302        | Apple        | 50000
7         | 401        | MI           | 10000
8         | 402        | MI           | 15000
(8 rows)
```

9) Orders

```
project=# select * from orders;
oid | item_name | amount
-----+-----+-----+
1   | Tv         | 50000
2   | Phone      | 15000
3   | Laptop     | 40000
4   | Tablet     | 10000
(4 rows)
```

10) Payment

```
project=# select * from payment;
payment_id | mode_of_payment |      date      | amount
-----+-----+-----+-----+
 1       | Online          | 2022-02-01  | 50000
 2       | Online          | 2022-04-01  | 100000
 3       | Online          | 2022-06-01  | 50000
 4       | Online          | 2022-08-01  | 10000
(4 rows)
```

11) Laptop

```
project=# select * from laptop;
laptop_id | laptop_model | laptop_company | laptop_price
-----+-----+-----+-----+
 1       | A            | Dell          |      50000
 3       | C            | Dell          |      70000
 4       | A            | HP            |      40000
 5       | A            | Apple         | 200000
(4 rows)
```

5.3) Queries Using Basic DBMS

1) Display the Customers who are **from** vadodara.

```
project=# Select * from customer where c_address='Vadodara';
  cid | c_address | c_name |    ph_no
-----+-----+-----+
 C101 | Vadodara | Heet   | 887654321
(1 row)
```

/*Between & And Operator*/

2)Display Payment recieve **between** 1 February **to** 1 April;

```
project=# Select * from Payment where date between '1-02-2022' and '1-04-2022';
 payment_id | mode_of_payment |      date       | amount
-----+-----+-----+-----+
      1 | Online          | 2022-02-01 | 50000
      2 | Online          | 2022-04-01 | 100000
(2 rows)
```

/*Like Operator*/

3)Display Customer Name start with 'A'

```
project=# Select * from customer where c_name like 'H%';
  cid | c_address | c_name | ph_no
-----+-----+-----+
  C101 | Vadodara | Heet    | 887654321
  C104 | Ahemdabad | Het    | 887654324
(2 rows)
```

/*Order By => ordering in acending */
4)Display Employee Deatials accoring to sort their name

```
project=# Select * from Employee order by name asc;
  emp_id | role     | phone_no | name      | salary
-----+-----+-----+-----+
  E102   | General  | 987654322 | ChhaganBhai | 150000
  E103   | General  | 987654323 | KathanBhai  | 100000
  E101   | General  | 987654321 | MaganBhai   | 100000
  E104   | Manager   | 987654324 | SureshBhai | 200000
(4 rows)
```

/*Aggregate function use & use of group by because we want also display name with aggregate function*/

5)Display count total sales of company in month of august 2022 with Buyer Name

```
project=# Select Buyer_name,count(*) as sales from billing where date_of_purchase between '1-08-2022' and '31-08-2022' group by Buyer_name;
buyer_name | sales
-----+-----
het      |    1
(1 row)
```

/*Simple Aggregate Function also use of alias*/
6)Display count total sales of company in month of august 2022

```
project=# Select count(*) as sales from billing where date_of_purchase between '1-08-2022' and '31-08-2022';
sales
-----
1
(1 row)
```

/*Aggregate Function + Group BY + Order by*/
7)Display count total sales of company in month of august to April 2022 with
Buyer Name with desceding order of name

```
project=# select Buyer_name,count(*) as sales from billing where date_of_purchase between '1-04-2022' and '31-08-2022' group by Buyer_name ORDER BY Buyer_name DESC;
buyer_name | sales
-----+-----
Priyam    |    1
Nisarg    |    1
het       |    1
(3 rows)
```

```
/*Aggregate Function with Inner join*/  
8)Count Supplier name who purchase amount>25000
```

```
project=# Select count(*) as countSupplier from Supplier Inner join Purchase_order on Supplier.S_id=Purchase_order.S_id where amount>25000;  
countsupplier  
-----  
          3  
(1 row)
```

/*Order By*/

9)Display Laptop Detials with their price arranging in aceding order

```
project=# Select * from laptop order by Laptop_price asc;  
laptop_id | laptop_model | laptop_company | laptop_price  
-----+-----+-----+-----  
4        | A           | HP           |      40000  
1        | A           | Dell          |      50000  
3        | C           | Dell          |      70000  
5        | A           | Apple         |    200000  
(4 rows)
```

/*Left join*/

10)Display Detials of Customers who purchase laptop in 2022

```
project=# Select c_name,c_address,ph_no from Customer LEFT JOIN Billing on Customer.CID=Billing.Bill_ID ;  
c_name | c_address | ph_no  
-----+-----+-----  
Het   | Ahemdabad | 887654324  
Heet  | Vadodara  | 887654321  
Priyam | Rajkot    | 887654322  
Nisarg | Surat     | 887654323  
(4 rows)
```

```
/*Full Outer Join*/
```

```
11)Display Details of Customer whose purchase is >2000;
```

```
LINE 1: ..._address,ph_no from Customer where CID in (select bill_id fr...
project=# SELECT a.Buyer_name AS "Name",a.date_of_purchase AS "Date",b.Item_name AS "b.Item",b.amount AS "Amount" FROM    billing a  FULL OUTER JOIN Orders b ON
D = b.OID where amount>2000;
      Name | Date       | b.Item | Amount
-----+-----+-----+-----+
Heet  | 2022-02-01 | Tv     | 50000
Priyam | 2022-04-01 | Phone   | 15000
Nisarg | 2022-06-01 | Laptop  | 40000
het   | 2022-08-01 | Tablet  | 10000
(4 rows)
```

```
/*Cross Join*/
```

```
12)Give Customer Details having billing amount> 50K
```

```
project=# select Distinct c_name,c_address,ph_no from customer cross join orders where amount>'45000';
      c_name | c_address | ph_no
-----+-----+-----+
Heet  | Vadodara | 887654321
Het   | Ahemdabad | 887654324
Nisarg | Surat     | 887654323
Priyam | Rajkot     | 887654322
(4 rows)
```

```
/*Using SubQuery*/
```

```
13)Display Supplier Deatails whose purchase amount > 50000
```

```
project=# Select * from supplier where S_id in (Select S_id from Purchase_order where amount>'50000');
      s_id | supplier_name | supplier_address | office_no | product
-----+-----+-----+-----+-----+
  1020304051 | CompanyB       | Porbandar        |      2010 | Laptop
(1 row)
```

*/*Using SubQuery*/*

14)Display Details of Customer Who purchase Laptop in 2022

```
project=# Select * from customer inner join billing on customer.c_name=billing.Buyer_name where bill_id=(select oid from orders where Item_name='Laptop');
      cid | c_address | c_name | ph_no | bill_id | buyer_name | mode_of_payment | date_of_purchase
-----+-----+-----+-----+-----+-----+-----+-----+
    C103 | Surat     | Nisarg  | 887654323 | 3       | Nisarg     | Online          | 2022-06-01
(1 row)
```

*/*Using SubQuery*/*

15)Display Details of Customer Who purchase Laptop in 2022

```
project=# Select * from customer inner join billing on customer.c_name=billing.Buyer_name where bill_id=(select oid from orders where Item_name='Laptop') and date_of_purchase between '1-01-2022' and '31-12-2022';
      cid | c_address | c_name | ph_no | bill_id | buyer_name | mode_of_payment | date_of_purchase
-----+-----+-----+-----+-----+-----+-----+-----+
    C103 | Surat     | Nisarg  | 887654323 | 3       | Nisarg     | Online          | 2022-06-01
(1 row)
```

*/*Using SubQuery*/*

16)Display Supplier Detials Whose Supplying Tablet/Iphone as we wish

```
project=# Select * from supplier where s_id in (select s_id from Purchase_order where pur_product='Tablet');
      s_id | supplier_name | supplier_address | office_no | product
-----+-----+-----+-----+-----+
 1020304053 | CompanyD      | Surat          |      3010 | Tablet
(1 row)
```

*/*Using SubQuery*/*

17)Give Customer Details having billing amount >1lakh

```
project=# Select Distinct c_name,c_address,ph_no from Customer inner join Billing on  c_name = (Select Buyer_name from Billing where bill_id=(select oid from orders where amount=50000));
      c_name | c_address | ph_no
-----+-----+-----+
 Heet    | Vadodara  | 887654321
(1 row)
```

5.5) PL/SQL Blocks (Views)

1) View

```
project=# select * from billing;
 bill_id | buyer_name | mode_of_payment | date_of_purchase
-----+-----+-----+-----+
 1      | Heet       | Online          | 2022-02-01
 2      | Priyam     | Online          | 2022-04-01
 3      | Nisarg     | Online          | 2022-06-01
 4      | het        | Online          | 2022-08-01
(4 rows)

project=# create view bill_view as
project-# select * from billing where bill_id='1';
CREATE VIEW
project=# select * from bill_view;
 bill_id | buyer_name | mode_of_payment | date_of_purchase
-----+-----+-----+-----+
 1      | Heet       | Online          | 2022-02-01
(1 row)
```

2) RowType

```
project=# select * from employee;
 emp_id | role    | phone_no | name      | salary
-----+-----+-----+-----+-----+
 E101   | General | 987654321 | MaganBhai | 100000
 E102   | General | 987654322 | ChhaganBhai | 150000
 E103   | General | 987654323 | KathanBhai | 100000
 E104   | Manager  | 987654324 | SureshBhai | 200000
(4 rows)

project=# do $$ 
project$$# declare total employee%rowtype;
project$$# begin
project$$# select * from employee
project$$# into total where emp_id='E101';
project$$# raise notice 'Name of Employee is "%" & Salery is "%".',
project$$# total.name,total.salary;
project$$# End
project$$# $$
project-# language plpgsql;
NOTICE: Name of Employee is "MaganBhai" & Salery is "100000".
DO
```

5.6) Functions & Trigger

1) Create order_audit for if any unauthentic person insert/update/delete our most important orders table then all actions sown in order_audit table.

```
*Function*
CREATE OR REPLACE FUNCTION do_order_audit() RETURNS TRIGGER AS
$TriggerNameAsIWant$

BEGIN
    INSERT INTO order_audit(oid_1,Item_name1,entry_date)
    values(NEW.oid,NEW.Item_name,current_timestamp);
    RETURN NEW;
END;
$TriggerNameAsIWant$
LANGUAGE PLPGSQL;

*Trigger*
/*cust_audit trigger name , orders = table name*/
CREATE TRIGGER order_audit_here_trigger_name15
Before INSERT OR UPDATE OR DELETE on Orders
FOR EACH ROW EXECUTE PROCEDURE
do_order_audit();
```

```
project=# select Distinct * from order_audit;
   oid_1 | item_name1 |          entry_date
-----+-----+-----
      13 |    Tv        | 2022-09-30 19:26:14.831536+05:30
(1 row)
```

```

project=# create table order_audit(oid_1 varchar(10) NOT NULL,Item_name1 varchar(10) NOT NULL,entry_date text NOT NULL);
CREATE TABLE
project=# CREATE OR REPLACE FUNCTION do_order_audit() RETURNS TRIGGER AS $TriggerNameAsIWant$
project$#     BEGIN
project$#         INSERT INTO order_audit(oid_1,Item_name1,entry_date) values(NEW.oid,NEW.Item_name,current_timestamp);
project$#         RETURN NEW;
project$#     END;
project$# $TriggerNameAsIWant$
project-# LANGUAGE PLPGSQL;
CREATE FUNCTION
project=# CREATE TRIGGER order_audit_here_trigger_name13
project-# Before  INSERT OR UPDATE OR DELETE on Orders
project-# FOR EACH ROW EXECUTE PROCEDURE
project-# do_order_audit();
ERROR: trigger "order_audit_here_trigger_name13" for relation "orders" already exists
project=# CREATE TRIGGER order_audit_here_trigger_name15
project-# Before  INSERT OR UPDATE OR DELETE on Orders
project-# FOR EACH ROW EXECUTE PROCEDURE
project-# do_order_audit();
CREATE TRIGGER
project-#
project=# Insert into Orders(OID, Item_name, amount)  values (13, 'Tv',10000);
INSERT 0 1
project=# select * from orders;
 oid | item_name | amount
-----+-----+-----
 1  | Tv        | 50000
 2  | Phone     | 15000
 3  | Laptop    | 40000
 4  | Tablet    | 10000
 8  | Lap        | 800
 9  | Laptop    | 8000
10  | Redmi    | 10000
11  | Redmi    | 10000
13  | Tv        | 10000
(9 rows)

```

```

project=# select * from orders;
 oid | item_name | amount
-----+-----+-----
 1  | Tv        | 50000
 2  | Phone     | 15000
 3  | Laptop    | 40000
 4  | Tablet    | 10000
 8  | Lap        | 800
 9  | Laptop    | 8000
10  | Redmi    | 10000
11  | Redmi    | 10000
13  | Tv        | 10000
(9 rows)

```

2) Create Function & Trigger for if supplier add wrong entry like negative value trigger shows.

```
*Trigger*
```

```
CREATE TRIGGER checkage11
Before INSERT OR UPDATE OR DELETE on Purchase_order
FOR EACH ROW EXECUTE PROCEDURE
fun();
```

```
*Function*
```

```
CREATE OR REPLACE FUNCTION fun() RETURNS TRIGGER AS $$
```

```
BEGIN
    IF new.quantity < 0 THEN
        RAISE EXCEPTION 'You Must Order Something!!!';
        /*RETURN NEW;*/
    END IF;
    RETURN New;
END;
```

```
$$
```

```
LANGUAGE PLPGSQL;
```

```

project=# select * from purchase_order;
   purchase_id | s_id      | pur_product | quantity | amount
-----+-----+-----+-----+-----+
    P1 | 1020304050 | Phone       |      5 | 50000
    P2 | 1020304051 | Laptop      |      2 | 100000
    P3 | 1020304052 | Laptop      |      1 | 50000
    P4 | 1020304053 | Tablet      |      1 | 10000
(4 rows)

project=#
project=# CREATE OR REPLACE FUNCTION fun() RETURNS TRIGGER AS $$ 
project$#     BEGIN
project$#         IF new.quantity < 0 THEN
project$#             RAISE EXCEPTION 'You Must Order Something!!!';
project$#             RETURN NEW;
project$#         END IF;
project$#         RETURN NULL;
project$#     END;
project$# $$ 
project-# LANGUAGE PLPGSQL;
CREATE FUNCTION
project=#
project=# CREATE TRIGGER checkage5
project-# After INSERT OR UPDATE OR DELETE on Purchase_order
project-# FOR EACH ROW EXECUTE PROCEDURE
project-# fun();
ERROR: trigger "checkage5" for relation "purchase_order" already exists
project=#
project=# CREATE TRIGGER checkage6
project-# After INSERT OR UPDATE OR DELETE on Purchase_order
project-# FOR EACH ROW EXECUTE PROCEDURE
project-# fun();
CREATE TRIGGER
project=#
project=#
project=# Insert into Purchase_order(Purchase_id,S_id,pur_product,quantity,amount) values ('P5',1020304050,'Phone',-2,50000);
ERROR: You Must Order Something!!!
CONTEXT: PL/pgSQL function fun() line 4 at RAISE

```

5.7) Cursor

Create a [Cursor](#) which traverses through users table where role is General.

```
project=# BEGIN;
BEGIN
project-*# DECLARE mycursor CURSOR FOR
project-*# SELECT * FROM employee WHERE role = 'General';
DECLARE CURSOR
project-*# FETCH NEXT FROM mycursor;
  emp_id | role      | phone_no | name        | salary
-----+-----+-----+-----+-----+
  E101  | General   | 987654321 | MaganBhai  | 100000
(1 row)

project-*# FETCH NEXT FROM mycursor;
  emp_id | role      | phone_no | name        | salary
-----+-----+-----+-----+-----+
  E102  | General   | 987654322 | ChhaganBhai | 150000
(1 row)

project-*# FETCH NEXT FROM mycursor;
  emp_id | role      | phone_no | name        | salary
-----+-----+-----+-----+-----+
  E103  | General   | 987654323 | KathanBhai | 100000
(1 row)

project-*# FETCH PRIOR FROM mycursor;
  emp_id | role      | phone_no | name        | salary
-----+-----+-----+-----+-----+
  E102  | General   | 987654322 | ChhaganBhai | 150000
(1 row)

project-*# CLOSE mycursor;
CLOSE CURSOR
project-*# end;
COMMIT
project=#

```

```
BEGIN;
DECLARE mycursor CURSOR FOR
SELECT * FROM employee WHERE role = 'General';
FETCH NEXT FROM mycursor;
FETCH PRIOR FROM mycursor;
CLOSE mycursor;
end;
```

6) Future Enhancements of this System

- We will design Front-end using React Framework and Develop Backend in NodeJS
 - Methods and user data input will be a lot easy after the implementation of GUI.
 - In the future, we can place the system on the cloud so the maintenance of the data can be reduced.

7) Bibliography

- We created ER-Model on Whimsical and Relational Schema on Creatively

- *ER-MODEL* created using

<https://whimsical.com/>

- *RELATIONAL SCHEMA* created using

<https://app.creately.com/d/yQAR0D8Dgpa/edit>

- For the implementation of this project, we referred to materials shared by

Prof. Archana N.Vyas & the following websites and books:

Book:

Database System Concepts

-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

Websites:

=> https://www.w3schools.com/sql/sql_syntax.asp

=> <https://www.tutorialspoint.com/index.htm>

=> <https://dev.mysql.com/doc/>

=><ms-database-management-system-set-1/>

=><https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/>
