PUNJAB UNIVERSITY COLLEGE OF COMPUTING AND INFORMATION TECHNOLOGY

SE Fall 21

Project Report

Project Title

Database of Supermarket POS

GROUP MEMBERS:

1. **BSEF21M033** – Imran Javed
2. **BSEF21M036** – Muaz Saleem
3. **BSEF21M047** – Farman Ali

Instructor Name:

Dr. Asif Sohail

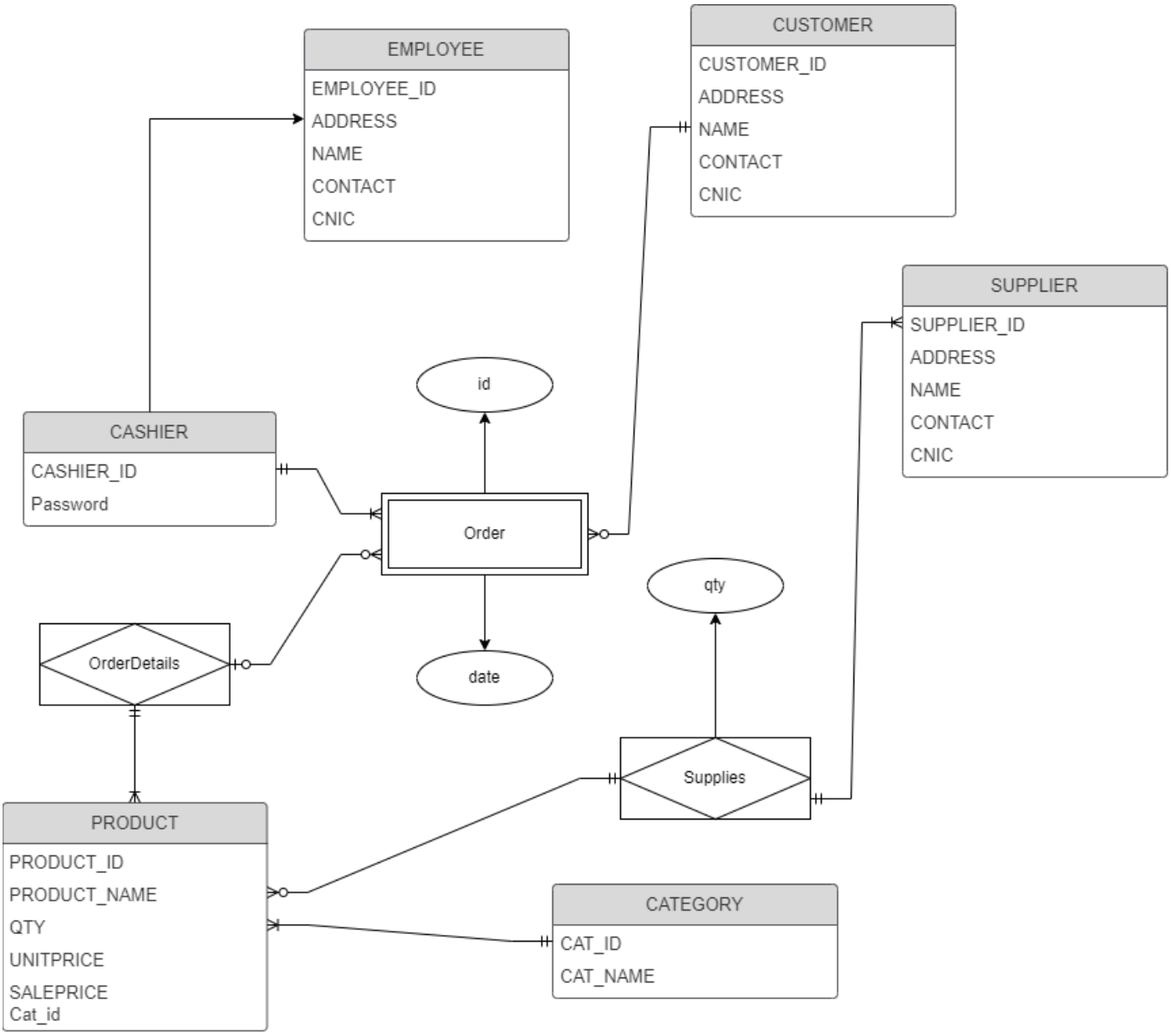
1. Introduction to the working of the system

Efficient management of a supermarket is vital for ensuring smooth operations, customer satisfaction, and profitability. In the modern era, technological advancements have significantly transformed the retail landscape, yet many supermarkets grapple with outdated management systems that hinder their ability to meet evolving consumer demands and streamline internal processes.

Attributes of the system

Supplier_id, Supplier_Name, Supplier_Address, Supplier_Contact, Supplier_CNIC, Category_id, Category_name, Product_id, Product_name, Product_description, Quantity, UnitPrice, SalePrice, inDate, Customer_id, customer_name, Customer_Address, Customer_Contact, customer_CNIC, order_no, Total_Bill, orderDate, Order_Qty, status, Total_Price, order_price, Cashier_id, Cashier _Name, Cashier _Address, Cashier _Contact, Cashier _CNIC

2. ERD of the system



3. Construction of the Relational Schema by using both bottom-up approach and top-down approach.

TOP-DOWN APPROACH:

Strong Relations:

Employee (Employee_id(pk), Name, Address, Contact, CNIC)

Supplier (Supplier_id(pk), Name, Address, Contact, CNIC)

Category (Cat_id(pk), Cat_name)

Product (Product_id(pk), Product_name, Quantity, UnitPrice, SalePrice,Cat_id)

Customer (Customer_id (pk), Name, Address, Contact, CNIC)

Cashier (cashier_id (pk, fk), password)

Weak Relations:

Order relation is dependent on cashier, customer and product

Creating a new relation order

Include all simple attributes as attributes of this relation

Include PK of identifying relation as FK to the new relation (cashier_id, customer_id)

Here we not direactly rely on order and Product relation because they have M: N relationship.

The PK of new relation is (PK of identifying relation + Partial identifier of weak entity)

Before creating associative entity

Order (order_no(pk), Customer_id (fk, pk), cashier_id (fk, pk), Product_id (fk, pk), Quantity, discount, Total_Price, price, Total_Bill, orderDate, status)

After creating associative entity

Order (order_no(pk), Customer_id(fk), cashier_id(fk), Total_Bill, orderDate, status)

Associative Relations:

Supplier and Product: One supplier can supply multiple products, and each product can be supplied by multiple suppliers. Product and supplier have many to many relationship

Product and Order: One order can be applied on multiple products, and each product can be placed in many orders. Product and Order have many to many relationship

Heance,

Creating new relations StockEntry as associative between Supplier and Product and OrderDetails as associative between Order and Product

Include PK of the participating entities along with the attributes of the associative entity in the new relation to construct table while transforming.

StocksEnrtry (product_id (fk, pk)), quantity, inDate, Supplier_id (fk, pk))
Order (order_no(pk), Customer_id(fk), cashier_id(fk), Total_Bill, orderDate, status)
OrderDetail (order_no (fk, pk), Product_id (fk, pk), Quantity, discount, Total_Price, price)

Bottom-Up Approach:

Normalization:

Now Evaluate and constructing the system by normalization.

1. **First Normal Form (1NF):**

For a relation to be in first normal form (1NF), there must not exist any multi-valued attribute within the relation. In the given relation there exist multi value attributes so the relation is not in 1NF. Now we have to remove multi value attributes in another relation.

- Product (**Product_id**, Product_name, Product_description, Quantity, UnitPrice, SalePrice, Category_id, Category_name)

- StockEntry (**Supplier_id, Product_id, inDate**, Quantity, Supplier_Name, Supplier_Address, Supplier_Contact, Supplier_CNIC)
  Product_id is foreign key refers to Product

- Order (**Product_id, order_no,** Customer_id, customer_name, Customer_Address, Customer_Contact, customer_CNIC, Cashier_id, Cashier _Name, Cashier _Address, Cashier _Contact, Cashier _CNIC, password, Total_Bill, orderDate, Order_Qty, status, Total_Price, order_price)
  Product_id is foreign key refers to Product

As in the above tables, we can see that there doesn't exist any multi-valued attribute so we can now consider these relations to be in first normal form.

2. **Second Normal Form (2NF):**

A table is in 2nd Normal Form (2NF) if it is in 1st Normal Form (1NF) and there must not be any partial dependencies, means that for each non-key attribute, its value must be dependent only on the entire candidate key and not on a part of it.

In above StockEntry relation Supplier_id which is part of composite primary key seperately identifying Supplier_Name, Supplier_Address, Supplier_Contact and Supplier_CNIC so we create sepearte relation Supplier and make Supplier_id fk in above relation and similarly for Order Relation.

- Product (**Product_id**, Product_name, Product_description, Quantity, UnitPrice, SalePrice, Category_id, Category_name)

- StockEntry (**Supplier_id, Product_id, inDate**, Quantity)
  Product_id and Supplier_id are foreign keys refers to Product and Supplier respectively

- Supplier (**Supplier_id,** Supplier_Name, Supplier_Address, Supplier_Contact, Supplier_CNIC)

- Order (**order_no,** Customer_id, customer_name, Customer_Address, Customer_Contact, customer_CNIC, Cashier_id, Cashier _Name, Cashier _Address, Cashier _Contact, Cashier _CNIC, password, Total_Bill, orderDate, status)

- OrderDetail (**Product_id, order_no,** Order_Qty, Total_Price, order_price)
  Product_id and Order_no are foreign keys refers to Product and Order respectively

As in the above tables, we can see that there doesn't exist any multi-valued attribute so we can now consider these relations to be in first normal form.

3. Third Normal Form (3NF):

For a table to be in third normal form, it must be in 2nd normal form and there must not exist any transitive functional dependency in the relation which states that:
Non-prime attributes → non-prime attributes OR A →B and B →C

In Product relation *Product_id →Category_id and Category_id →Category_name*
In Order relation *Order_no →Customer_id and Customer_id →Customer_name, address, cnic, etc.*
Also Order_no →Cashier_id and Cashier_id →Cashier_name, address, cnic, etc.
Hence seperating these entities.

- Product (**Product_id**, Product_name, Product_description, Quantity, UnitPrice, SalePrice, Category_id)
  Category_id is foreign key refers to Category

- Category (**Category_id**, Category_name)

- StockEntry (**Supplier_id, Product_id, inDate**, Quantity)
  Product_id and Supplier_id are foreign keys refers to Product and Supplier respectively

- Supplier (**Supplier_id,** Supplier_Name, Supplier_Address, Supplier_Contact, Supplier_CNIC)

- Order (**order_no,** Customer_id, Cashier_id, Total_Bill, orderDate, status)
  Customer_id, Cashier_id are foreign keys refers to Customer, Cashier

- Customer (**Customer_id**, customer_name, Customer_Address, Customer_Contact, customer_CNIC)
- Cashier (**Cashier_id**, Cashier _Name, Cashier _Address, Cashier _Contact, Cashier _CNIC, password)

- OrderDetail (**Product_id, order_no,** Order_Qty, Total_Price, order_price)
  Product_id and Order_no are foreign keys refers to Product and Order respectively

All relational schemas within the system witness that there doesn't exist any such relation within this system, so all the tables are already in 3NF.

4. Description of the relations in the following format:

Employee:

Attribute	Data Type	Size	Constraints
Employee_id	number	10	Primary key
Name	Varchar2	50	Not null
Address	Varchar2	250	
Contact	Varchar2	20	Not null
CNIC	Varchar2	15	

Supplier:

Attribute	Data Type	Size	Constraints
supplier_id	number	10	Primary key
Name	Varchar2	50	Not null
Address	Varchar2	250	
Contact	Varchar2	20	Not null
CNIC	Varchar2	15	

Category:

Attribute	Data Type	Size	Constraints
cat_id	number	10	Primary key
Cat_Name	Varchar2	30	Not null

Product:

Attribute	Data Type	Size	Constraints
Product_id	number	10	Primary key
Name	Varchar2	50	Not null
description	Varchar2	150	
quantity	number	10	Quantity >=0
unitPrice	number	(10,2)	
salePrice	Number	(10,2)	Not null
Cat_id	Number	10	Foreign Key

StockEntry:

Attribute	Data Type	Size	Constraints
Product_id	number	10	Primary Key, Foreign Key

Supplier_id	number	10	Primary Key, Foreign Key
quantity	int		Quantity >=0
inDate	date		Default sysdate,Primary Key

Order:

Attribute	Data Type	Size	Constraints
Order_no	Number	10	Primary Key
customer_id	Number	10	Foreign Key
Cashier_id	Number	10	Foreign Key
orderDate	Date		sysdate
Total_bill	Number	(10,2)	Totalbill >= 0
Status	Varchar2	15	Can be Prending, Paid, Cancelled

OrderDetail:

Attribute	Data Type	Size	Constraints
order_no	Number	10	Foreign Key, Primary Key
Product_id	Number	10	Foreign Key, Primary key
Quantity	Number	10	Quantity >=0 & <=stockQty
Price	Number	(10,2)	Price>=0
discount	Number	(10,2)	discountPrice >= 0
TotalPrice	number	(10,2)	totalPrice >= 0

Customer:

Attribute	Data Type	Size	Constraints
Customer_id	Number	10	Primary Key
Name	Varchar2	50	Not null
Address	Varchar2	250	
Contact	Varchar2	20	Not null
CNIC	Varchar2	15	

Cashier:

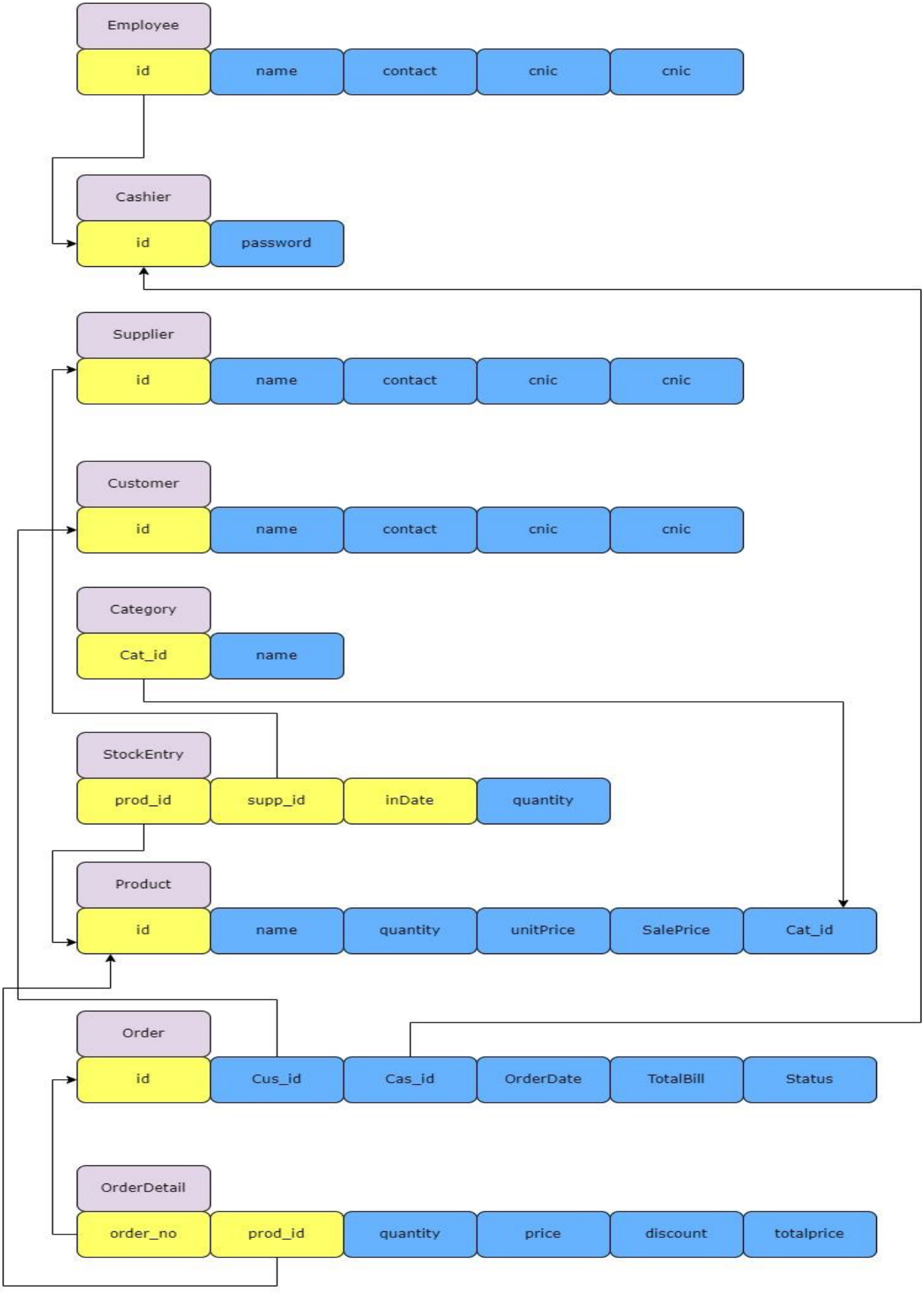
Attribute	Data Type	Size	Constraints
Cashier_id	Number	10	Primary Key, Foreign key
Password	varchar2	25	Not null

5. CREATE TABLE statements for all the relations of your system.

```
create table Employee (
   employee_id number(10) primary key,
   name varchar2(50) not null,
   address varchar(250),
   contact varchar(20) not null,
   cnic varchar2(15)
);

create table supplier (
   supplier_id number(10) primary key,
   name varchar2(50) not null,
   address varchar(250),
   contact varchar(20) not null,
   cnic varchar2(15)
);

create table category (
   cat_id number(10) primary key,
   cat_name varchar(30) not null
);

CREATE TABLE Product (
   Product_id NUMBER(10) PRIMARY KEY,
   Product_name VARCHAR2(50) not null,
   Product_desc   VARCHAR2(150),
   Quantity number(10) not null,
   UnitPrice number(10, 2),
   SalePrice number(10, 2) not null,
   Cat_id number(10),
   Supplier_id number not null,
   FOREIGN KEY (Cat_id) REFERENCES Category(Cat_id),
   FOREIGN KEY (Supplier_id) REFERENCES Supplier(Supplier_id)
);

create table stocksEntry (
   product_id number(10) not null,
   quantity int not null,
   indate date default sysdate,
   supplier_id  number(10) not null,
   FOREIGN KEY (product_id) REFERENCES Product(Product_id),
   FOREIGN KEY (Supplier_id) REFERENCES Supplier(Supplier_id),
   PRIMARY KEY (product_id,Supplier_id,indate));

create table cashier (
   cashier_id number(10) primary key,
   foreign key (cashier_id) references Employee(employee_id),
   password varchar(25) not null
);
```

```
create table Orders (
   order_no number(10) primary key,
   customer_id number(10),
   cashier_id number(10),
   total_bill number(10, 2),
   orderdate date default sysdate,
   status varchar2(15),
   constraint order_cusID_fk foreign key (customer_id) references customer(customer_id),
   constraint order_casID_fk foreign key (cashier_id) references cashier(cashier_id),
   constraint order_status_ck check(status IN('Pending','Paid','Cancelled')),
   constraint order_totalbill_ck check(total_bill >= 0)
);


create table orderdetail (
   order_no number(10) not null,
   product_id number(10) not null,
   quantity number(10) not null,
   discount number(10,2),
   total_price number(10,2),
   price number(10,2),
   foreign key (order_no) references orders(order_no),
   foreign key (product_id) references product(product_id),
   primary key (order_no, product_id),
   constraint trans_qty_ck check(quantity >= 0),
   constraint trans_price_ck check(price >= 0),
   constraint trans_disc_ck check(discount >=0 AND discount < 0.5*price)
);

create table customer (
   customer_id number (10) primary key,
   name varchar2(50) not null,
   address varchar(250),
   contact varchar(20) not null,
   cnic varchar2(15)
);
```

6. Relational data model showing the association among different relations of the relational schema.



7. SELECT statement for at least five common reports to be generated by the system.

Reterive Products having low stock
select product_id, product_name, quantity from product where quantity < 10;

Reterive Sales by Each Category of Products
select c.cat_name as category, sum (od. total_price) as total_sales
from orderdetail od join product p on od. product_id = p.product_id join category c on p.cat_id = c.cat_id
group by c.cat_name;

List of Order with total bill only
select order_no as order_number, total_bill as total_bill, orderdate as order_date, status as order_status from order;

List of Suppliers along with details of product
select s.name as supplier_name, p.product_name from supplier s
join product p on s.supplier_id = p.supplier_id;

list all those employees who are not cashier
select * from employee where employee_id not in (select cashier_id from cashier)

8. Design of at least two VIEWS, that you feel are the most important.

create view productdetails as
select p. product_id, p. product_name, p. product_desc, p. quantity, p. unitprice, p. saleprice, c. cat_id, c.cat_name, s. supplier_id, u.name as supplier_name, u. address as supplier_address, u. contact as supplier_contact
from product p inner join **category** c on p.cat_id = c.cat_id inner join **supplier** s p. supplier_id = s. supplier_id inner join user u on s. supplier_id = u.user_id;

create view orderinfo as
select o. order_no, o. customer_id, c. name as customer_name, c. address as customer_address, c. contactnumber as customer_contact, o. cashier_id, u.name as cashier_name, u. address as cashier_address, u. contactnumber as cashier_contact, o. total_bill, o. orderdate, o. status
from **order** o inner join **customer** c on o. customer_id = c. customer_id inner join **cashier** cs on o. cashier_id = cs. cashier_id inner join user u on cs. cashier_id = u.user_id;