**B.TECH. (2020-24)**

**Artificial Intelligence**

**open ended problem**

on

**Database Management Systems**

**[CSE201]**

**Logo

Description automatically generated**

Submitted To

**Dr Gaurav Dubey**

Submitted By

**Hitesh**

**A023119820027**

**4CSE11 (AI)**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

NOIDA (U.P)

Open Ended Problem

CASE STUDY - 1

Q.1. A database is being constructed for storing sales information system. A product can be described with a unique product number, product name, selling price, manufacturer name. The product can sale to a particular client and each client have its own unique client number, client name, client addresses, city, pin code, state and total balance to be required to paid. Each client order to buy product from the salesman. In the order, it has unique sales order number, sales order date, client number, salesman number (unique), billed whole payment by the party or not and its delivery date. The salesman has the name, addresses, city, pin code, state, salary of the salesman, delivery date, total quantity ordered, product rate.

ER Diagram

Diagram, engineering drawing

Description automatically generated

* Assumptions taken for construction of ER Diagram

1. There can be some product which is not sold so there is partial participation from the side of entity ‘Product’ to entity ‘Client’.
2. Some clients may not order any product but are still in client list as they may be regular (or potential) clients who may order later so partial participation.
3. An order can only exist if it has been placed by a client to a salesman, thus total participate from the entity ‘Order’.
4. Some salesmen may not get any order to manage, so Partial Participation.
5. Each salesman would have product details for at least one product thus all salesmen would participate, and each product would have its details with some salesman because only then can they be sold, thus indicating total participation from both sides of the relation (has product details).
6. One product can be sale to many clients and one client can also buy many products thus the relation would have many-to-many cardinality.
7. A client may place more than one order while an order can only be placed by one client thus it would hold one-to-many cardinality from client to order.
8. A salesman may manage many orders, but one order can not be managed by more than one salesman which indicates one-to-many cardinality from salesman to order.
9. One salesman can have product details for one or more products and at the same time one single product can have its details with many salesmen. Thus, the relation between Product and Salesman would have many-to-many cardinality.

Relational Mapping

Text, letter

Description automatically generated

Database Implementation in SQL

DDL Command Script to create Tables and Insert data to it

|  |
| --- |
| CREATE TABLE products  (  pname varchar(20),  pno number(10) PRIMARY KEY,  sp real,  mname varchar(20)  );  CREATE TABLE clients  (  cno number(10) PRIMARY KEY,  cname varchar(20),  total\_balance real,  cstate varchar(20),  city varchar(20),  pincode number(10)  );  CREATE TABLE salesman  (  sno number(10) PRIMARY KEY,  sname varchar(20),  city varchar(20),  sstate varchar(20),  pincode number(10),  totalqty real  );  CREATE TABLE orders  (  orderno number(10) PRIMARY KEY,  sodate date,  deldate date,  billedpay real,  cno number(10),  sno number(10),    CHECK (deldate > sodate),  FOREIGN KEY (cno) REFERENCES clients(cno),  FOREIGN KEY (sno) REFERENCES salesman(sno)  );  CREATE TABLE saleto  (  pno number(10),  cno number(10),  FOREIGN KEY (pno) REFERENCES products (pno),  FOREIGN KEY (cno) REFERENCES clients (cno),  PRIMARY KEY (pno,cno)  );  CREATE TABLE productdetails  (  pno number(10),  sno number(10),  productrate real,  FOREIGN KEY (pno) REFERENCES products (pno),  FOREIGN KEY (sno) REFERENCES salesman (sno),  PRIMARY KEY (pno,sno)  );  CREATE TABLE orders2  (  orderno number(10) PRIMARY KEY,  sodate date,  deldate date,  billedpay real,  cno number(10),  sno number(10),    CHECK (deldate > sodate),  CHECK (sno = productdetails.sno),  FOREIGN KEY (cno) REFERENCES clients(cno),  FOREIGN KEY (sno) REFERENCES salesman(sno)  );  CREATE TABLE productdetails2  (  pno number(10),  sno number(10),  productrate real,  FOREIGN KEY (pno) REFERENCES products (pno),  FOREIGN KEY (sno) REFERENCES salesman (sno),    PRIMARY KEY (pno,sno)  );  INSERT INTO products VALUES('slipper',100,500,'paragon');  INSERT INTO products VALUES('Laptop',102,200000,'hp');  INSERT INTO products VALUES('Television',103,3000,'vermillion');  INSERT INTO clients VALUES(200,'jason',50,'kalos','lumious',110011);  INSERT INTO clients VALUES(201,'raph',500,'hoenn','mauville',120022);  INSERT INTO clients values(203,'nathon',60,'alola','hauoli',120011);  INSERT INTO salesman VALUES(300,'merchantA','coal','bituminous',130012,500);  INSERT INTO salesman VALUES(301,'merchantb','capricon','valley',140029,400);  INSERT INTO orders VALUES(401,'04/02/2022','04/20/2022',2000,203,300);  INSERT INTO orders VALUES(400,'04/01/2022','04/28/2022',2000,200,300);  INSERT INTO orders VALUES(402,'04/05/2022','05/02/2022',100,201,301);  INSERT INTO saleto VALUES(100,203);  INSERT INTO saleto VALUES(103,200);  INSERT INTO saleto VALUES(103,201);  INSERT INTO productdetails VALUES(100,300,3000);  INSERT INTO productdetails VALUES(102,300,3420);  INSERT INTO productdetails VALUES(103,301,3400);  INSERT INTO productdetails VALUES(103,300,400);  INSERT INTO productdetails VALUES(102,301,7777); |

DML Commands

Viewing the Created Tables

|  |
| --- |
| select \* from products |

Graphical user interface, application

Description automatically generated

|  |
| --- |
| select \* from clients |

Graphical user interface, application

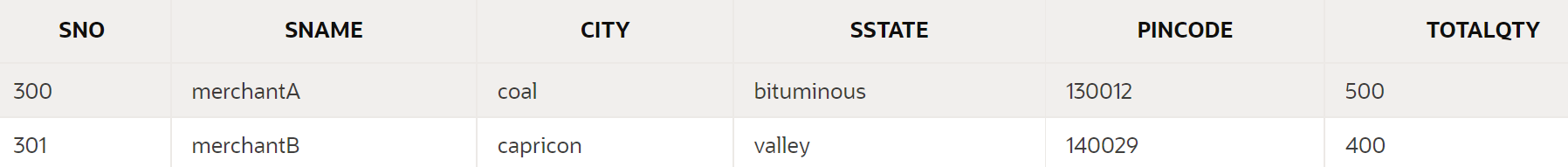
Description automatically generated

|  |
| --- |
| select \* from saleto |

Background pattern

Description automatically generated

|  |
| --- |
| select \* from salesman |



|  |
| --- |
| select \* from orders |

Graphical user interface, application

Description automatically generated

|  |
| --- |
| select \* from productdetails |

Graphical user interface, text, application

Description automatically generated

SQL Queries for Validation

1) Display the names of clients who have an ‘p’ as third letter in their name.

Text

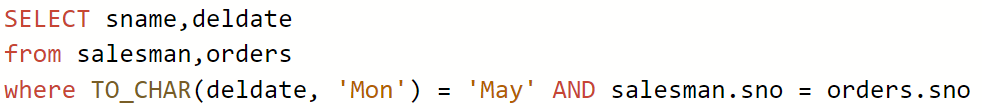
Description automatically generated with medium confidence

Output

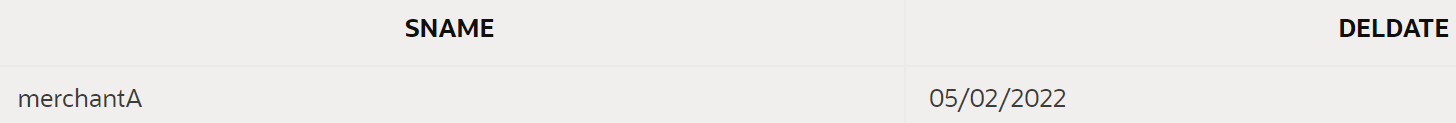
Background pattern

Description automatically generated with medium confidence

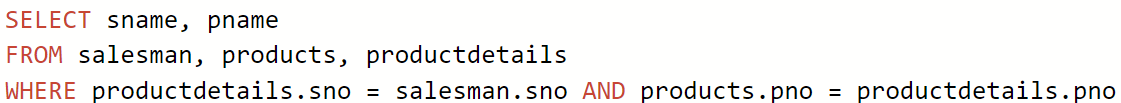
2) Display the salesman names and delivery date for those salesmen who got orders with delivery date for month of May.



Output



3) Display the name of all salesmen along with the product details of the products which they can sell to clients.

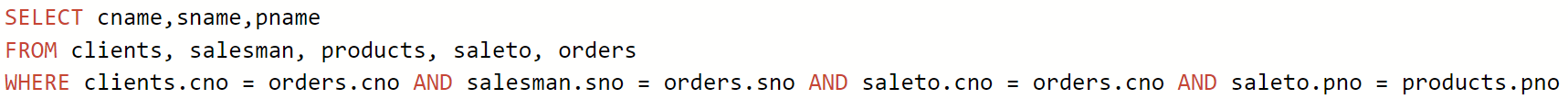


Output

Background pattern

Description automatically generated

4) Show the details for the Orders placed with client name, salesman, product name.

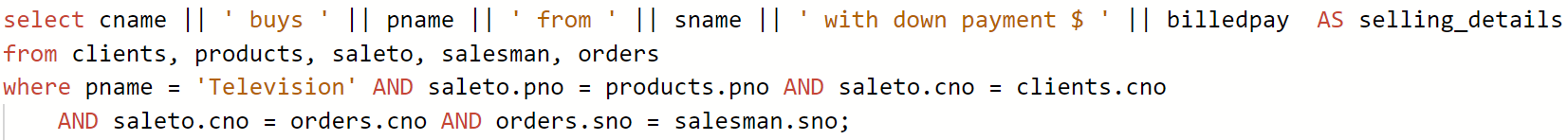


Output

Graphical user interface

Description automatically generated with low confidence

5) Display the names of clients who bought ‘Television’ in the format “jason buys Television from merchantA with down payment $ 2000”



Output

Graphical user interface, text, application

Description automatically generated

A bank database keeps record of the details of customers, accounts, loans and transactions such as deposits or withdraws. Customer record should include customer id, customer name, address, age, contact number, email id etc., accounts details involves account number, account type(fixed account, savings account, monthly account etc), date of creation of the account. Transaction detail keeps information about amount deposited or withdrawn to/from a particular account and the date of transaction. The database should also store record of loans which include loan amount, loan date and the account number to which the loan is granted.