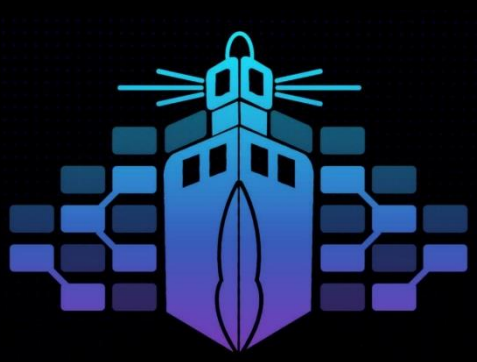


CPCS-241

Data Base Design Project (25%)

Sofas Selling Company

CS1 - Dr. Mohammad Al-Tuwaijri

Group Members:

- Mahmued Alardawi - 2135209

- Submission Date: 10/6/2023

Content

PART I: Analysis 4

1 Problem Definition and Data Requirements 4

1.1 Problem Description 4

1.4 Intended Output of the system 7

PART II: DB DEISGN 8

2 ER Diagram Design 8

2.1 ER diagram 8

2.2 Design of Business Rules 9

3 ER-to-logical schema mapping 10

3.1 Mapping of Regular Entity Types 10

3.2 Mapping of Weak Entity Types 13

3.3 Mapping of binary 1-1 relationship types 13

3.4 Mapping of binary 1-N relationship types 14

3.5 Mapping of binary M-N relationship types 18

3.6 Mapping of multivalued attributes 19

3.7 Mapping of n-array relationship types 19

3.8 Schema Diagram 20

4 Normalization 21

4.1 First Normal Form 21

4.2 Second Normal Form 21

4.3 Third Normal Form 21

5 Final DB Schema Diagram 22

PART III: IMPLEMENTATION 23

6 Table Creation Script 23

6.1 <SALESMAN> TABLE 23

6.2 <PHONE> TABLE 24

6.3 <BRANCH> TABLE 24

6.4 <BRAND> TABLE 24

6.5 <SOFA> TABLE 25

6.6 <CUSTOMER> TABLE 25
6.7 <PAYMENT> TABLE 26
7 Constraints Script 27
8 Queries 28
8.1 <SALESMEN REPORT> 28
8.2 <SALESMEN REPORT FOR 'X' SALESMAN> 29
8.3 <SOFA AVAILABILITY> 30
8.4 <AVERAGE SALARY FOR SALESMAN> 30
8.5 <NUMBER OF SOFAS IN EACH BRANCH> 30
8.6 <DISPALY PAYMENTS HIGHER THAN 'X' PRICE> 30
8.7 <DISPALY PAYMENTS HIGHER THAN 'X' PRICE> 31
Appendix 31

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

Today every company that wants to succeed in the business world needs to have a database to manage and administrate the change that occurs in their products, employees, and everything related to the company. Literally having a database will make the growth and the development of the company way easier. This project is an actual database that I used in a furniture company.

A Sofas Selling Company has multiple assets, Employees, Sofas, Customers, and more. To make the workflow easier having a database is a must.

1.2 Data Requirements

1. Salesman

- National ID
- Name
- Birth date
- Gender
- Employment type
- Salary
- Address
- Supervisor ID
- Branch ID

2. Branch

- Branch ID
- Branch address
- Manager ID

3. Sofa

- Sofa ID
- Sofa type
- Status
- Price
- Brand ID
- Brand name
- Branch ID

4. Costumer

- Costumer ID
- Name
- phone
- Address

5. Payment

- Payment ID
- Amount
- Amount paid
- Remaining to customer
- Payment method
- Payment date
- Delivery date
- Salesman ID
- Customer ID

1.3 Business Rules

1. Salesmen’s rules

- Each salesman has a unique national ID
- Each salesman works in a one branch only
- Each salesman has one supervisor
- Each salesman must be full-time or part-time
- The age of the salesman must be above 18

2. Branch’s rules

- Each branch has a unique ID
- Each branch has at least one manager
- At least 1 salesman work in a branch

3. Sofa’s rules

- Each sofa has a unique ID
- Each sofa must show its availability to costumers
- The price must be written and showed to customers

4. Costumer’s rules

- Each costumer has a unique ID
- Each costumer must specify their preferred delivery address
- Each costumer must have a phone number

5. Payment’s rules

- Each payment has a unique ID
- Each payment has a costumer ID
- Each payment has a salesman ID
- Each payment must specify the delivery date

1.4 Intended Output of the system

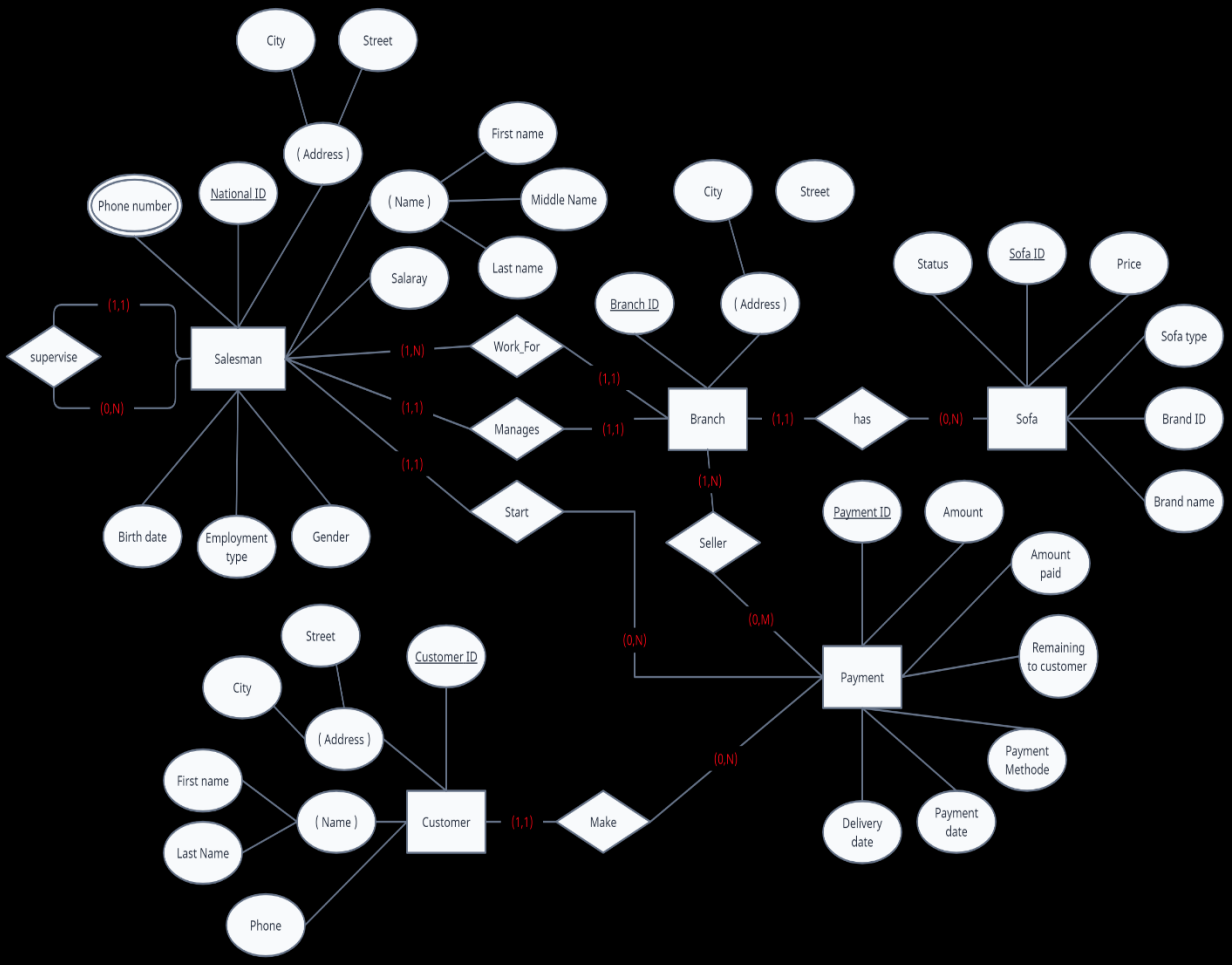
Output and queries:

- Access salesman data
- Access costumer data
- Update all data attributes
- Track available sofas in each branch
- Calculating profit
- Display the payment and delivery date to costumers

PART II: DB DEISGN

2 ER Diagram Design

2.1 ER diagram



2.2 Design of Business Rules

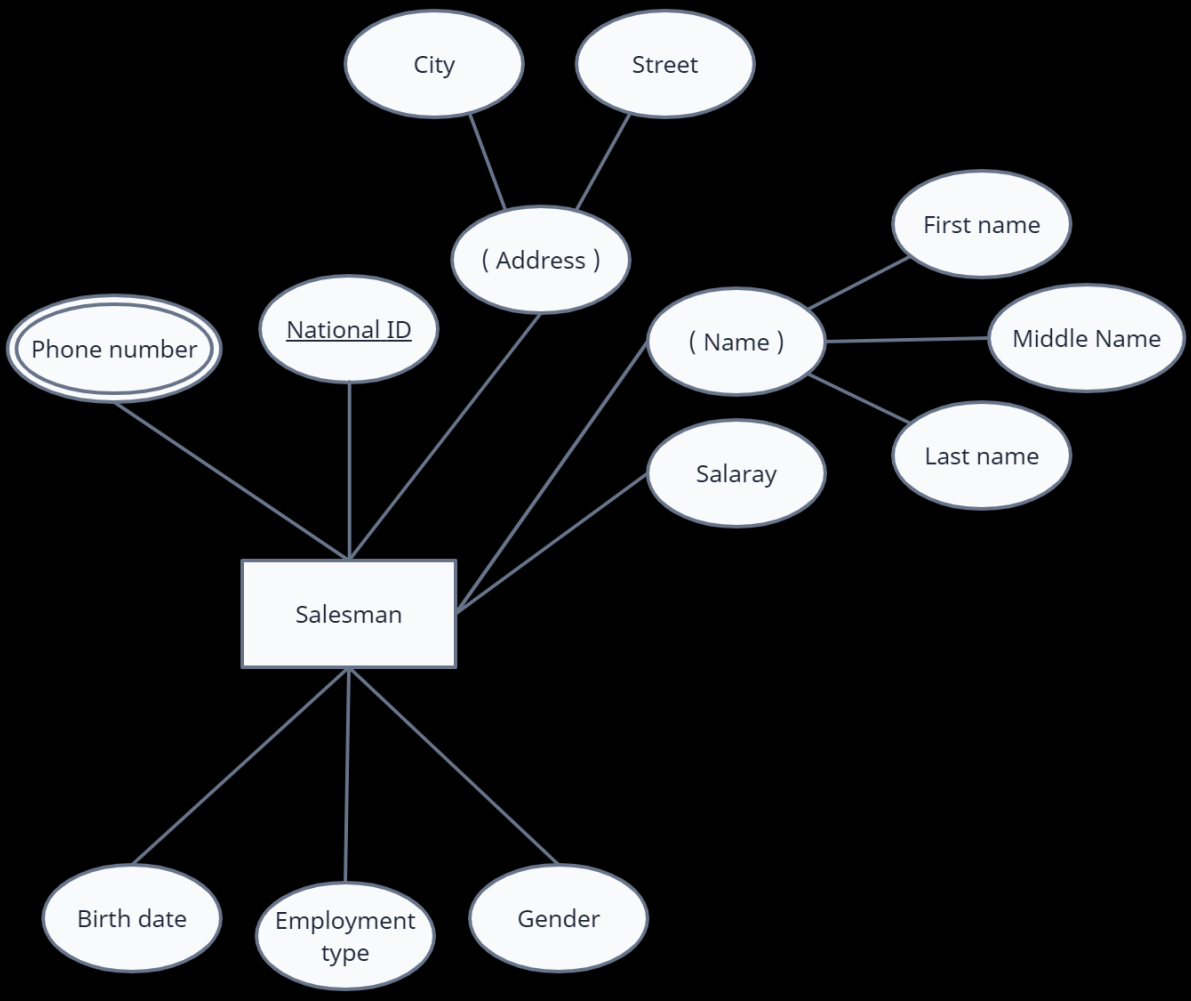
Business Rule	Design Decisions	Justification (if any)
Branch has a manager.	1:1 Binary relationship between salesman and branch.	Each branch has only one manager.
Each salesman has one supervisor.	N:1 Binary relationship between salesman and salesman.	A supervisor can have many salesmen and each salesman must have a supervisor.
salesman works in one branch only. Branch has at least 1 salesman working in it.	N:1 Binary relationship between salesman and branch.	To work for a branch, you must be in the building at your working hours. you can't be in two different branches. The branch can have many salesmen not only 1.
A branch contains multiple sofas.	N:1 Binary relationship between sofa and branch.	Each branch must contain at least one sofa.
Every payment must be started by a salesman.	N:1 Binary relationship between salesman and payment.	A salesman can make many payments and they are made by him for the customer.
Every payment must contain a customer.	N:1 Binary relationship between customer and payment.	A customer can have many payments.
Payments happens in branch.	N:M Binary relationship between branch and payment.	Multiple payments can happen in multiple branches.

3 ER-to-logical schema mapping

3.1 Mapping of Regular Entity Types

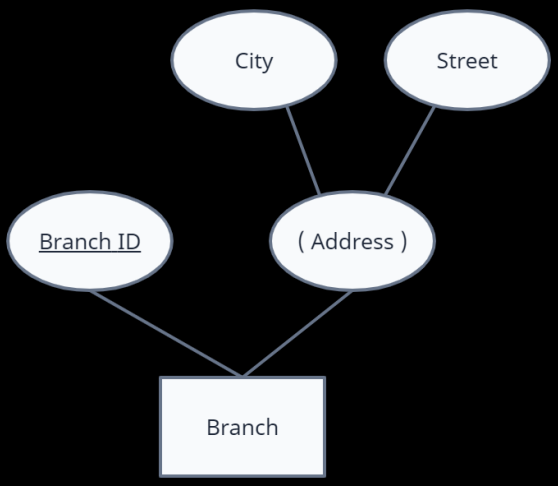
1. Salesman

Salesman									
National ID	FName	MName	LName	BDate	Gender	Emp_Type	Salary	City	Street



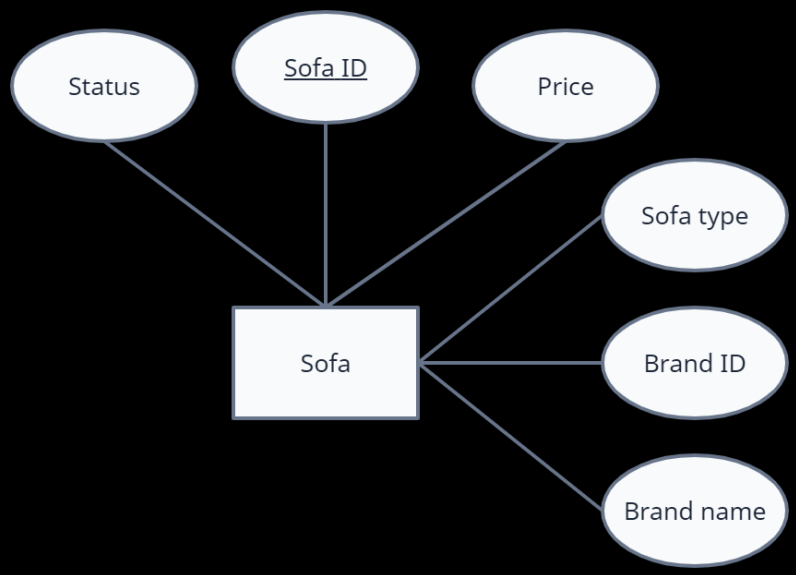
2. Branch

Branch		
Branch ID	City	Street



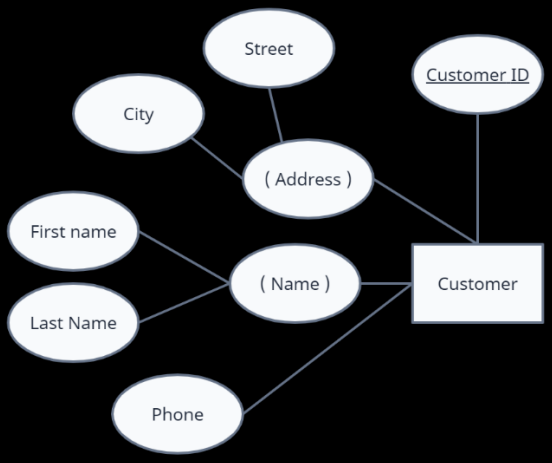
3. Sofa

Sofa					
Sofa ID	Sofa_Type	Status	Price	Brand_ID	Brand_Name



4. Customer

Customer					
Customer ID	FName	LName	Phone	City	Street



5. Payment

Payment						
Payment ID	Amount	Amount_Paid	RemToCus	PMethode	Payment_Date	Delivery _Date

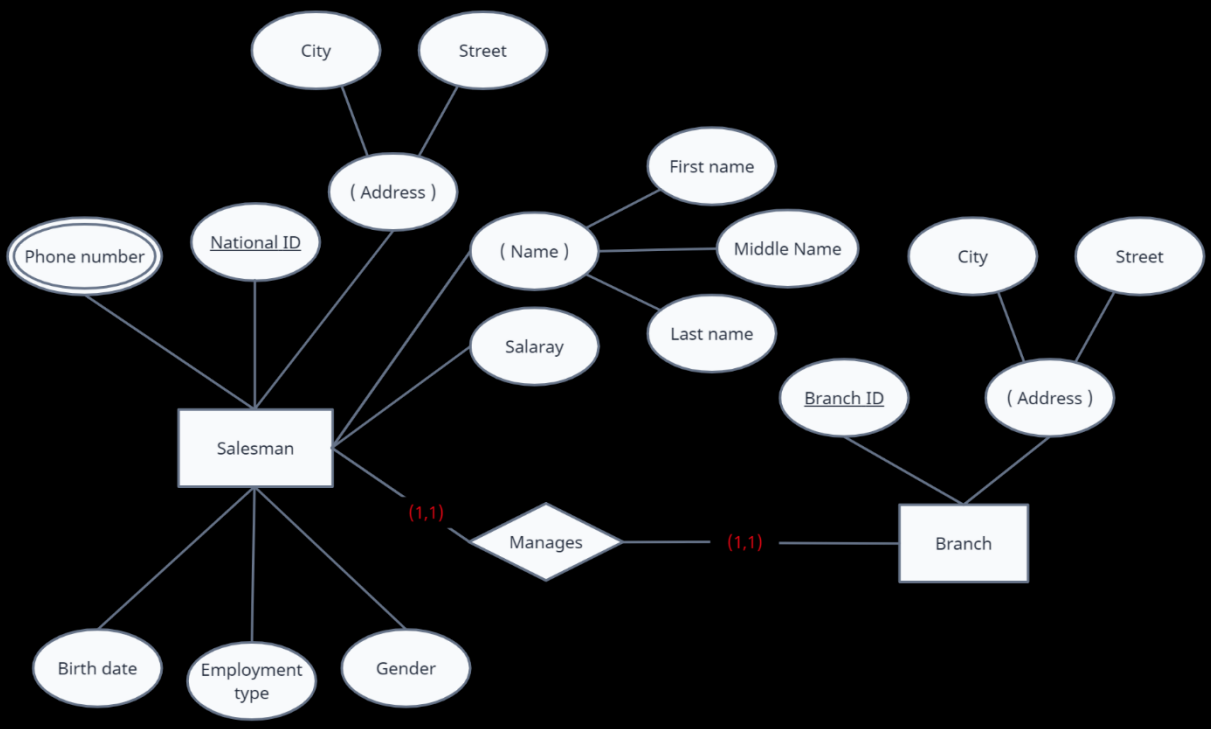
3.2 Mapping of Weak Entity Types

N/A

3.3 Mapping of binary 1-1 relationship types

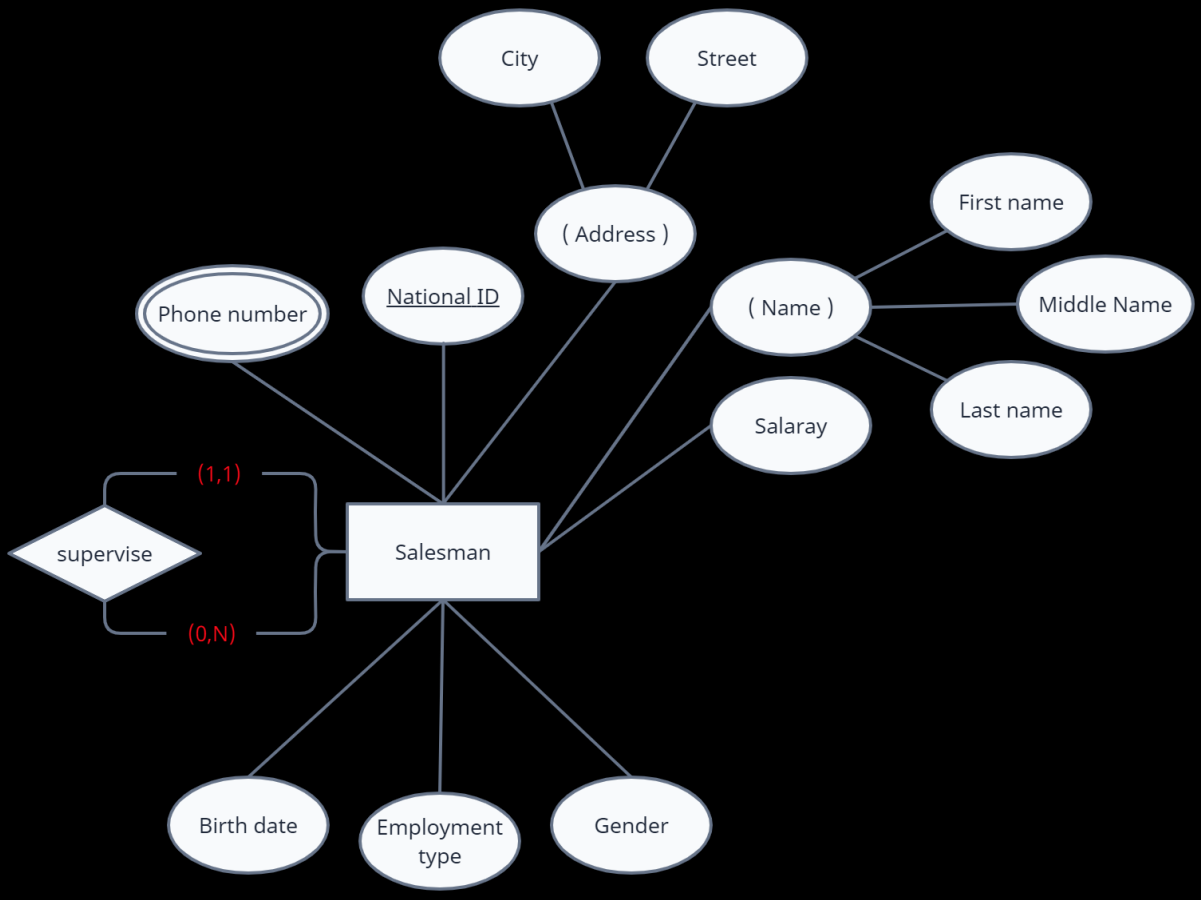
Salesman									
National ID	FName	MName	LName	BDate	Gender	Emp_Type	Salary	City	Street

Branch			
Branch ID	City	Street	Mangar_ID



3.4 Mapping of binary 1-N relationship types

Salesman										
National ID	FName	MName	LName	BDate	Gender	Emp_Type	Salary	City	Street	Sup_ID

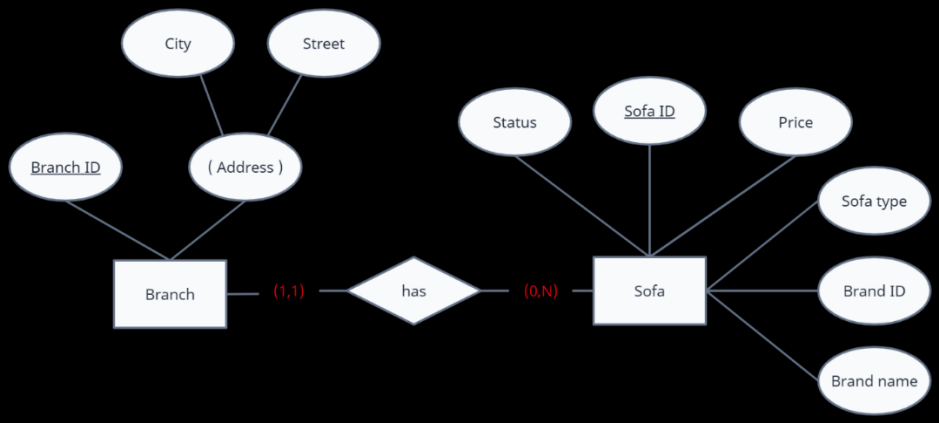


Salesman										
National_ID	FName	MName	LName	BDate	Gender	Emp_Type	Salary	City	Street	Sup_ID

Branch			
Branch ID	City	Street	Mangar_ID

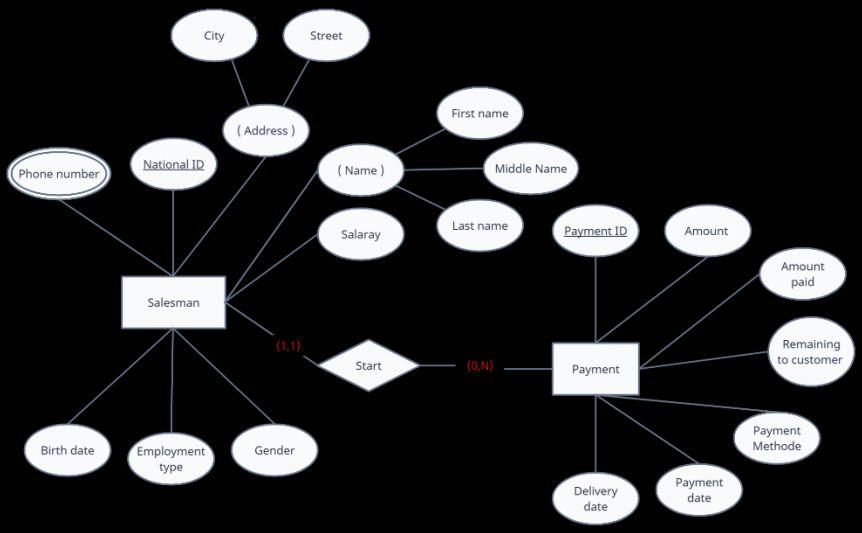
Branch			
Branch ID	City	Street	Mangar_ID

Sofa						
Sofa ID	Sofa_Type	Status	Price	Brand_ID	Brand_Name	Branch_ID



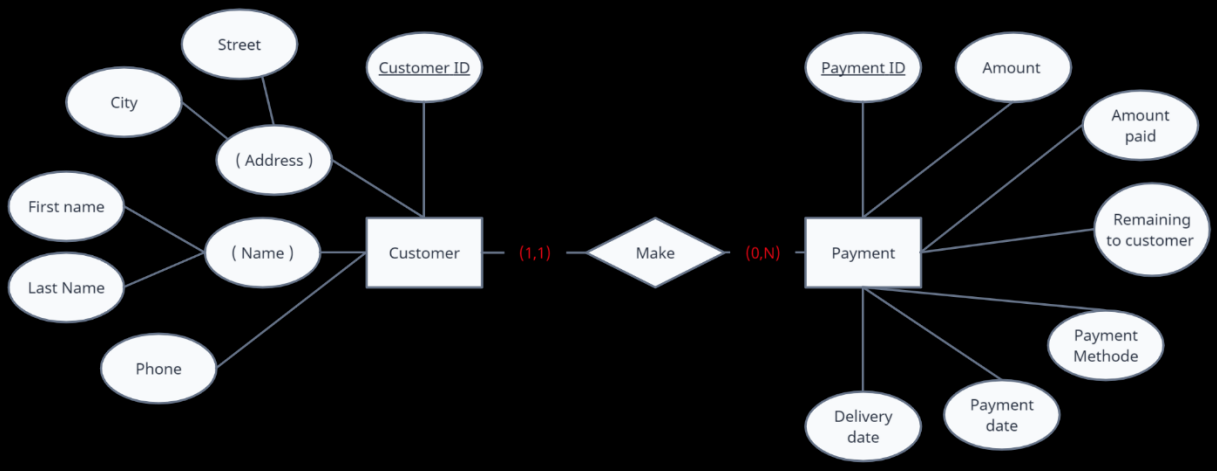
Salesman										
National_ID	FName	MName	LName	BDate	Gender	Emp_Type	Salary	City	Street	Sup_ID

Payment							
Payment ID	Amount	Amount_Paid	RemToCus	PMethode	Payment_Date	Delivery _Date	Salesman_ID



Customer					
Customer ID	FName	LName	Phone	City	Street

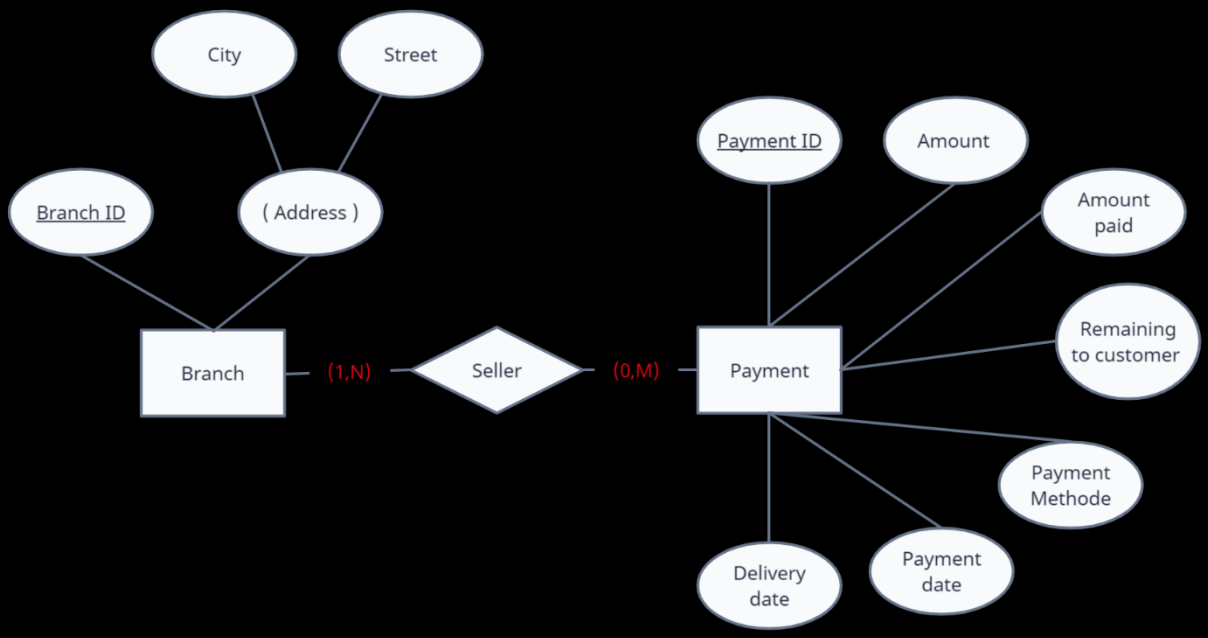
Payment								
Payment ID	Amount	Amount_Paid	RemToCus	PMethode	Payment_Date	Delivery _Date	Salesman_ID	Cus_ID



3.5 Mapping of binary M-N relationship types

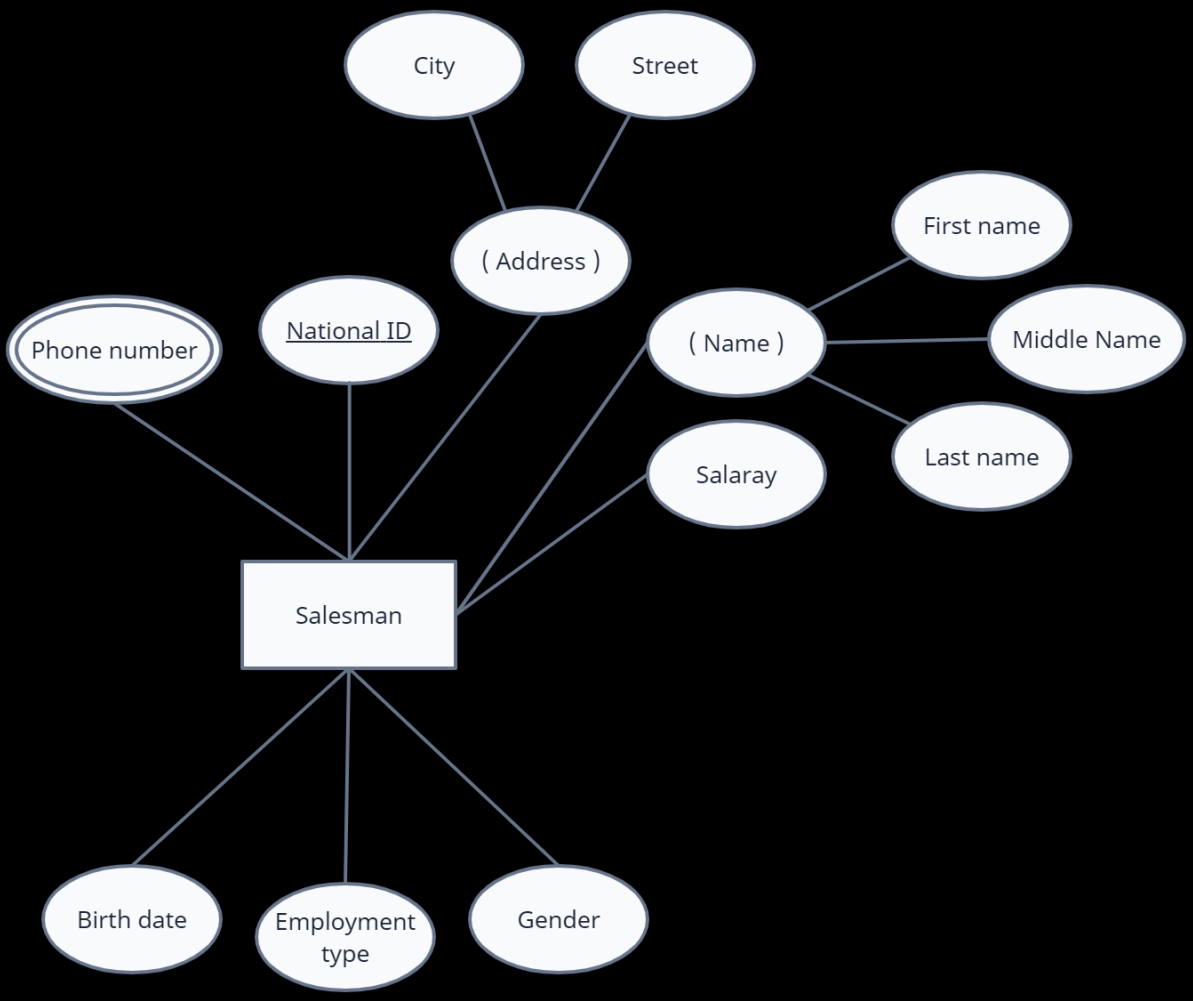
Branch			
Branch ID	City	Street	Mangar_ID

Payment									
Payment ID	Amount	Amount_Paid	RemToCus	PMethode	Payment_Date	Delivery_Date	Salesman_ID	Cus_ID	Branch_ID



3.6 Mapping of multivalued attributes

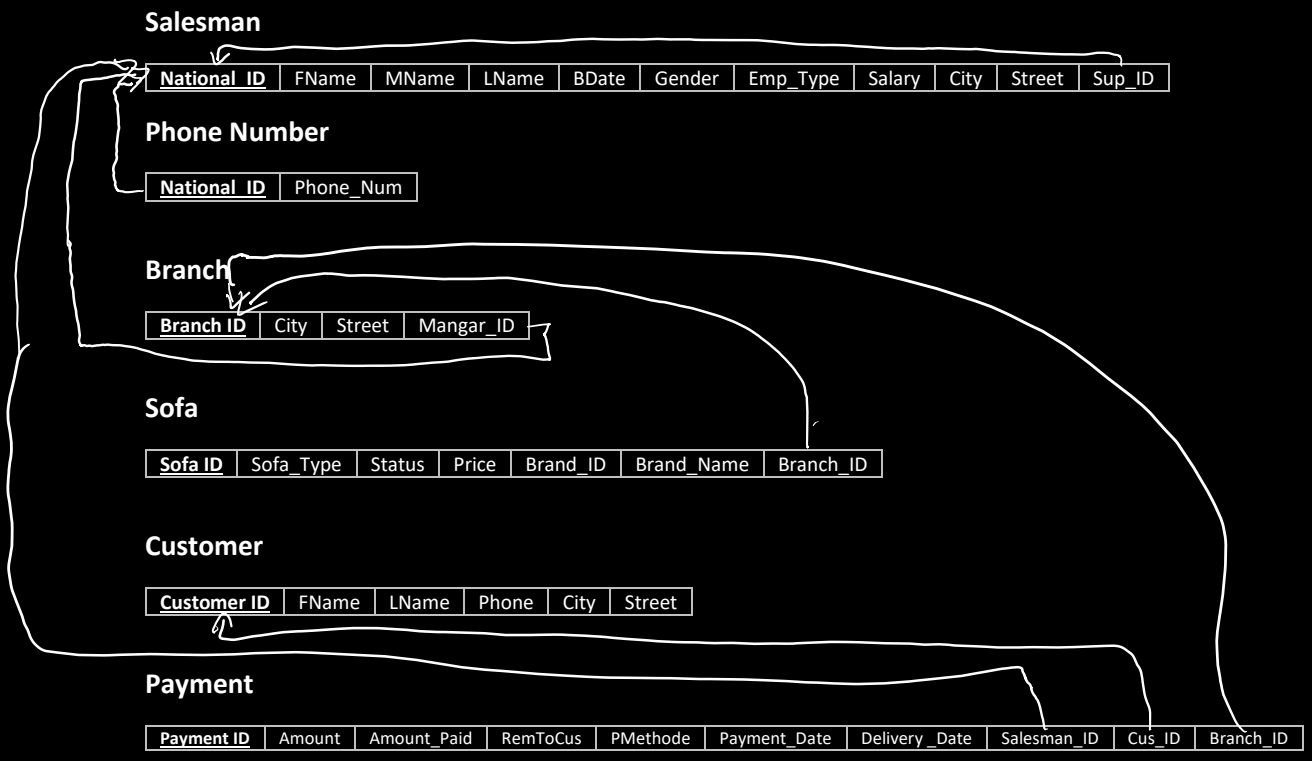
Phone Number	
National_ID	Phone



3.7 Mapping of n-array relationship types

N/A

3.8 Schema Diagram



4 Normalization

4.1 First Normal Form

N/A

4.2 Second Normal Form

Before normalization

Sofa						
Sofa ID	Sofa_Type	Status	Price	Brand_ID	Brand_Name	Branch_ID

After normalization

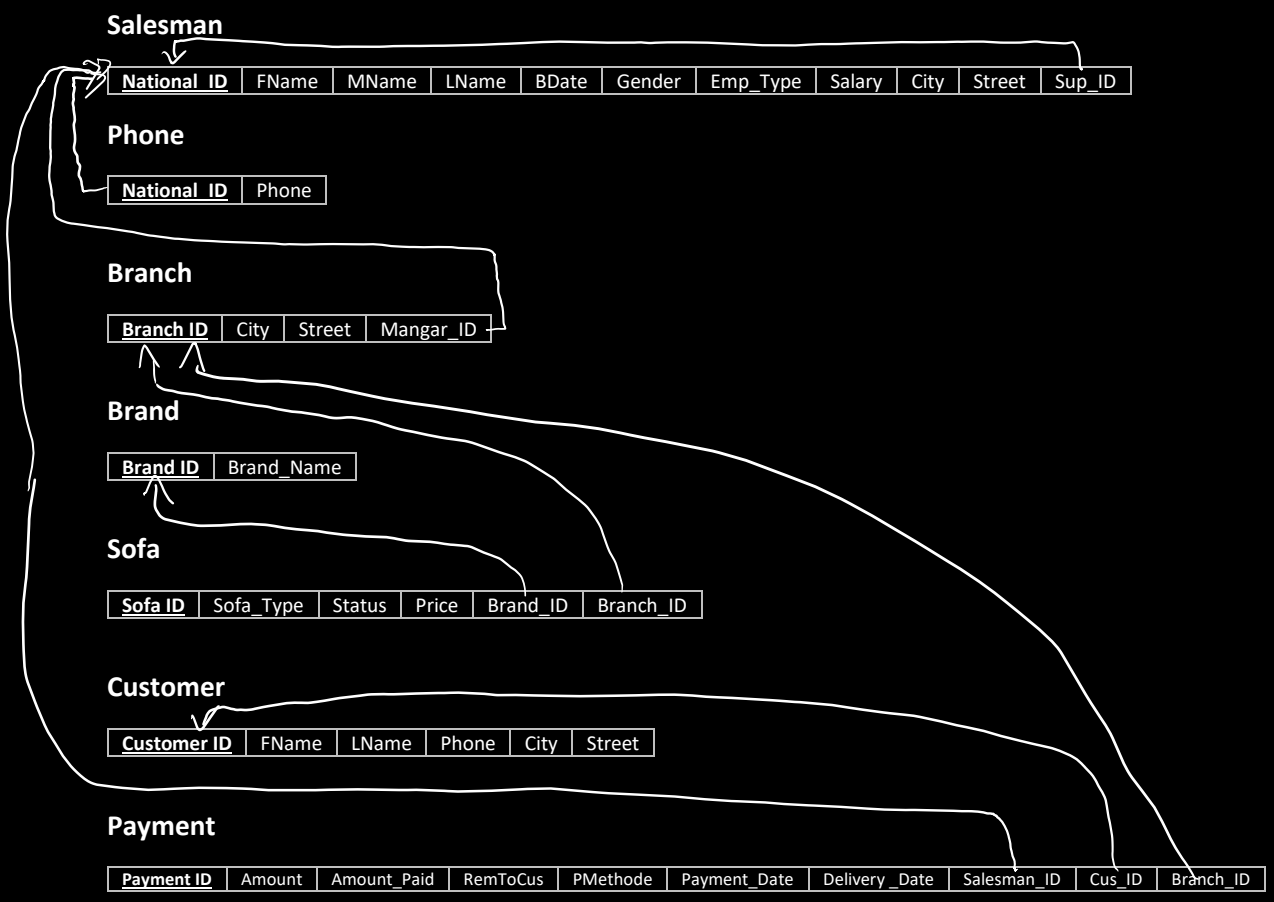
	Sofa				
Sofa ID	Sofa_Type	Status	Price	Brand_ID	Branch_ID

Brand	
Brand_ID	Brand_Name

4.3 Third Normal Form

N/A

5 Final DB Schema Diagram



PART III: IMPLEMENTATION

6 Table Creation Script

6.1 <SALESMAN> TABLE

```
CREATE TABLE SALESMAN (
NATONAL_ID VARCHAR2(10) PRIMARY KEY,
FNAME VARCHAR2(40) NOT NULL,
MNAME VARCHAR2(40),
LNAME VARCHAR2(40) NOT NULL,
BDATE DATE,
GENDER VARCHAR2(20),
EMPLOYMET_TYPE VARCHAR2(20) NOT NULL,
SALARY NUMBER,
CITY VARCHAR2(20),
STREET VARCHAR2(20),
SUP_ID VARCHAR2(10),
BRANCH_ID INT,
SDATE DATE DEFAULT SYSDATE,
CHECK (TRUNC(MONTHS_BETWEEN(SDATE, BDATE)) > 17),
FOREIGN KEY (SUP_ID) REFERENCES SALESMAN (NATONAL_ID) ON DELETE CASCADE
);

ALTER TABLE SALESMAN ADD
FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH (BRANCH_ID) ON DELETE CASCADE;
```

6.2 <PHONE> TABLE

```
CREATE TABLE PHONE (
NATONAL_ID VARCHAR2(10) PRIMARY KEY,
PHONE VARCHAR2(10) UNIQUE,
FOREIGN KEY (NATONAL_ID) REFERENCES SALESMAN (NATONAL_ID) ON DELETE CASCADE
);
```

6.3 <BRANCH> TABLE

```
CREATE TABLE BRANCH (
BRANCH_ID INT PRIMARY KEY,
CITY VARCHAR2(20),
STREET VARCHAR2(20),
MANAGER_ID VARCHAR2(10),
FOREIGN KEY (MANAGER_ID) REFERENCES SALESMAN (NATONAL_ID) ON DELETE CASCADE
);
```

6.4 <BRAND> TABLE

```
CREATE TABLE BRAND (
BRAND_ID INT PRIMARY KEY,
BRAND_NAME VARCHAR2(30) UNIQUE
);
```

6.5 <SOFA> TABLE

```
CREATE TABLE SOFA (
SOFA_ID VARCHAR2(10) PRIMARY KEY,
SOFA_TYPE VARCHAR2(30),
STATUS VARCHAR2(20),
PRICE NUMBER,
BRAND_ID INT,
BRANCH_ID INT,
FOREIGN KEY (BRAND_ID) REFERENCES BRAND (BRAND_ID) ON DELETE CASCADE,
FOREIGN KEY (BRANCH_ID) REFERENCES BRANCH (BRANCH_ID) ON DELETE CASCADE
);
```

6.6 <CUSTOMER> TABLE

```
CUSTUMER_ID INT PRIMARY KEY,
FNAME VARCHAR2(40) NOT NULL,
LNAME VARCHAR(40),
PHONE VARCHAR2(10) NOT NULL,
CITY VARCHAR2(20),
STREET VARCHAR2(20)
);
```

6.7 <PAYMENT> TABLE

```
CREATE TABLE PAYMENT (
PAYMENT_ID INT PRIMARY KEY,
AMOUNT NUMBER NOT NULL,
AMOUNT_PAID NUMBER,
REM_TO_CUS NUMBER,
PAYMENT_METHODE VARCHAR2(30) NOT NULL,
PAYMENT_DATE DATE NOT NULL,
DELIVERY_DATE DATE,
SALESMAN_ID VARCHAR2(10),
CUSTOMER_ID INT,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (NATONAL_ID) ON DELETE CASCADE,
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER (CUSTUMER_ID) ON DELETE CASCADE
);
```

7 Constraints Script

Business Rules	SQL Script	Table
Salesman has primary key as national ID	NATONAL_ID VARCHAR2(10) PRIMARY KEY	SALESMAN
Salesman must have first name	FNAME VARCHAR2(40) NOT NULL	SALESMAN
Salesman must have last name	LNAME VARCHAR2(40) NOT NULL	SALESMAN
Salesman must have employment type to detriment monthly sales target	EMPLOYMET_TYPE VARCHAR2(20) NOT NULL	SALESMAN
Salesman must be above 17 years	CHECK (TRUNC(MONTHS_BETWEEN(SDATE, BDATE)) > 17)	SALESMAN
Phone has primary key as national ID	NATONAL_ID VARCHAR2(10) PRIMARY KEY	PHONE
Phone for each national ID is unique	PHONE VARCHAR2(10) UNIQUE	PHONE
Branch has primary key as branch ID	BRANCH_ID INT PRIMARY KEY	BRANCH
Brand has primary key as brand ID	BRAND_NAME VARCHAR2(30) UNIQUE	BRAND
Sofa has primary key as sofa ID	SOFA_ID VARCHAR2(10) PRIMARY KEY	SOFA
Customer has primary key as customer ID	CUSTUMER_ID INT PRIMARY KEY	CUSTOMER
Customer must have first name	FNAME VARCHAR2(40) NOT NULL	CUSTOMER
Customer must have phone	PHONE VARCHAR2(10) NOT NULL	Customer
Payment has primary key as payment ID	PAYMENT_ID INT PRIMARY KEY	PAYMENT
Payment must have amount	AMOUNT NUMBER NOT NULL	PAYMENT
Payment method must be specified for each payment	PAYMENT_METHODE VARCHAR2(30) NOT NULL	PAYMENT
Payment date must be written with each payment	PAYMENT_DATE DATE NOT NULL,	PAYMENT

8 Queries

8.1 <SALESMEN REPORT>

Display the Salesmen information and check the employment type if its full_time

the monthly target is 250k, and if part_time the monthly target is 150k.

If the target is reached the salesman will get a 500 bonus.

```
SELECT S.SALESMAN_ID, S.FNAME, S.LNAME, S.EMLOYMENT_TYPE, S.SALARY,
SUM(P.AMOUNT) AS TOTAL_SALES,
        CASE
                WHEN S.EMPLOYMENT_TYPE = 'FULL-TIME' AND TOTAL_SALES > 250000
                THEN 'TARGET REACHED (BONUS)'
                WHEN S.EMPLOYMENT_TYPE = 'PART-TIME' AND TOTAL_SALES > 150000
                THEN 'TARGET REACHED (BONUS)'
                ELSE 'TARGET NOT REACHED (NO BONUS)'
        END AS BOUNS
FROM SALESMAN AS S, PAYMENT AS P
WHERE S.SALESMAN_ID = P.SALESMAN_ID
ORDER BY TOTAL;
```

8.2 <SALESMEN REPORT FOR 'X' SALESMAN>

Display the Salesman information and check the employment type if its full_time

the monthly target is 250k, and if part_time the the monthly target is 150k.

If the target is reach the salesman will get a 500 bonus.

```
SELECT S.SALESMAN_ID, S.FNAME, S.LNAME, S.EMLOYMENT_TYPE, S.SALARY,
SUM(P.AMOUNT) AS TOTAL_SALES,
        CASE
                WHEN S.EMPLOYMENT_TYPE = 'FULL-TIME' AND TOTAL_SALES > 250000
                THEN 'TARGET REACHED (BONUS)'
                WHEN S.EMPLOYMENT_TYPE = 'PART-TIME' AND TOTAL_SALES > 150000
                THEN 'TARGET REACHED (BONUS)'
                ELSE 'TARGET NOT REACHED (NO BONUS)'
        END AS BOUNS
FROM SALESMAN AS S, PAYMENT AS P
WHERE S.SALESMAN_ID = 'X' AND P.SALESMAN_ID = 'X';
```

8.3 <SOFA AVAILABILITY>

```
SELECT S.STATUS, S.SOFA_ID, S.SOFA_TYPE, S.BRANCH_ID, B.BRAND_ID, B.BRAND_NAME
FROM SOFA AS S, BRAND AS B
WHERE S.STATUS = 'AVAILABLE' AND S.BRAND_ID = B.BRAND_ID
ORDER BY S.STATUS;
```

8.4 <AVERAGE SALARY FOR SALESMAN>

```
SELECT AVG(SALARY) AS AVERAGE_SALARY, COUNT(*) AS NUMBER_OF_SALESMAN
FROM SALESMAN;
```

8.5 <NUMBER OF SOFAS IN EACH BRANCH>

```
SELECT BRANCH_ID, COUNT(*) AS NUMBER_OF_SOFAS
FROM SOFAS
ORDER BY BRANCH_ID;
```

8.6 <DISPALY PAYMENTS HIGHER THAN 'X' PRICE>

```
SELECT P.PAYMENT_ID, P.AMOUNT, P.CUSTOMER_ID, C.FNAME, C.LNAME, C.PHONE,
P.PAYMENT_DATE
FROM P.PAYMENT
INNER JOIN CUSTOMER AS C ON P.CUSTOMER_ID = C.CUSTOMER_ID AND P.AMOUNT > 'X'
ORDER BY AMOUNT;
```

8.7 <DISPALY PAYMENTS IN ‘X’ DATE>

```
SELECT *

FROM PAYMENTS

WHERE PAYMENT_DATE BETWEEN DATE'X-01-01' AND DATE'X-12-31';
```

Appendix

Sorry Doctor couldn’t do it I’m out of time.