

CPCS241 – Database I – Spring 2022 – Group Project



DB Design

Group 6:

Student Name	Student Number
Dhay Mohammed Alharbi	1905483
Raghad Sharaz Alzahrani	1906830
Reem Saad Alsubaie	2005404
Jana Faisal Altalhi	2006359
Sara Mohammed Almonjy	2010272

Contents

PART I: Analysis.....	3
1 Problem Definition and Data Requirements.....	3
1.1 Problem Description	3
1.2 Data Requirements	3
1.3 Business Rules.....	5
1.4 Intended Output of the system	6
PART II: DB DEISGN	7
2 ER Diagram Design.....	7
2.1 ER diagram	8
2.2 Design of Business Rules.....	8
3 ER-to-logical schema mapping.....	11
3.1 Mapping of Regular Entity Types	11
3.2 Mapping of Weak Entity Types	12
3.3 Mapping of binary 1-1 relationship types.....	12
3.4 Mapping of binary 1-N relationship types	13
3.5 Mapping of binary M-N relationship types.....	14
3.6 Mapping of multivalued attributes	15
3.7 Mapping of n-ary relationship types.....	15
3.8 Schema Diagram	16
4 Normalization	17
4.1 First Normal Form	17
4.2 Second Normal Form	18
4.3 Third Normal Form.....	19
5 Final DB Schema Diagram	20
PHASE III: IMPLEMENTATION	21
6 Table Creation Script	21
7 Constraints Script.....	34
8 Queries and Transactions	36
APPENDIX.....	42

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

One of the most crucial tasks to concentrate on and pay close attention to is the pharmacy database design project. The pharmacies include several critical data, such as products name, customer and employee information, and so on, all of which must be preserved and retrieved.

A database is helpful in pharmacy because it saves time and effort when entering data and is accessible at any time. It stores data for many years without being damaged or lost, and it connects them all in one place. Employees can also create reports about patients, products, and medicines. In addition, the database allows you to create invoices and print them quickly.

To summarise, the database is required to organize the work in the pharmacy and save time, effort, and access to the necessary information.

1.2 Data Requirements

1- Employee

- EName: the full name of the employee
- EID: a unique ID for each employee
- EAge: the age of the employee
- EGender: the gender of the employee
- EPhone_numbers: the phone number of the employee
- Experience: experience of the employee
- Salary: salary of the employee
- EBirth date: the birth date of the employee

2- Customer

- CName: the full name of the customer
- A unique account of the customer by ID
- CGender: the gender of the customer
- CAge: the age of the customer
- CBirth date: the birth date of the customer
- CAddress: the address of the customer
- CPhone numbers: the phone number of the customer

3- Products

- PName: the full name of the product
- PID: a unique ID for each product
- PType: type of the product (prescribed or non-prescribed)
- PPrice: the price of each product

4- Category

- CaName: the name of the category (medication, vitamins and supplements, hair and skincare)
- CaID: a unique ID for each category (medication, vitamins and supplements, hair and skincare)

5- Orders

- OType: type of the orders (normal order: one week or rush order:3 days)
- ONumber: number of the orders
- OID: a unique ID for each order

6- Department

- DeName: the name of the department
- DeNumber: each department has a unique number
- DeLocation: location for each department

7- Stock

- SID: a unique ID for each stock
- SNumber: number of products in stock
- SItem: the name of the product in stock

8- Dependents

- DName: name of the dependent
- DGender: the gender of the dependent
- DBirth date: the birth date of the dependent
- DRelationship

9- Shipments

- ShID: put ID number for each shipment
- Shipping company name

10-Bill

- Total cost
- Date of the bill
- BID: a unique bill number

1.3 Business Rules

Employee rules:

- The pharmacy employs at least seven employees: one pharmacist, two accountants, two employees service representatives, and two employees who care about the drugstore and organize the medicines.
- Each employee must be at least 21 years old.
- It is preferable if the employee has prior experience, but it is not required.

Shipment rules:

- Every shipment should have an ID, name, company name, and bill.

Product rules:

- The product must have a name and price.
- Each product must be under a specific category.

Stock rules:

- Stock refers to the number of products that are ready to be sold. Each stock item is coded with a unique number to keep track of it.

Category rules:

- There are three categories of pharmacy products:
 - 1- medication
 - 2- vitamins and supplements.
 - 3- hair and skincare
- Each product must be under a specific category.

Customers rules:

- If a person purchases a product from the pharmacy, the person is considered a customer.

Orders rules:

- The customers can make orders. Order includes many Products.

Departments rules:

- There are many departments in the pharmacy.
- Each department has its employees and an employee who manages the department.

Dependents rules:

- Each employee may have many dependents.
- Dependents receive a 10% discount and some specialized health services.

Bill rules:

- Each bill should calculate the total cost for every customer and make a shipment.

1.4 Intended Output of the system

The outputs divide into queries, transactions, and report:

1- Queries:

Search for the customer who has an account by ID

Searching for employees by ID

Search for customer information

Search for products by name or ID

Search for products under a specific category

Searching for shipment by ID

Search for product prices in ascending or descending order

Search for product name in ascending or descending order

2- Transactions

- Insert:

Insert products

Insert customer

Insert shipping company

Insert employee

Insert bill

Insert location

Insert orders

Insert category

- Delete:

Delete products

Delete customer

Delete shipping company

Delete employee

Delete bill

Delete location

Delete orders

Delete category

- Update:

Update price of products

Update Phone numbers for customer and employee

Update address for customer

Update shipped orders

Update the bill

Update the number of branch employees

Update the number of customers

Update salary every year

Update profits calculation

3- Reports:

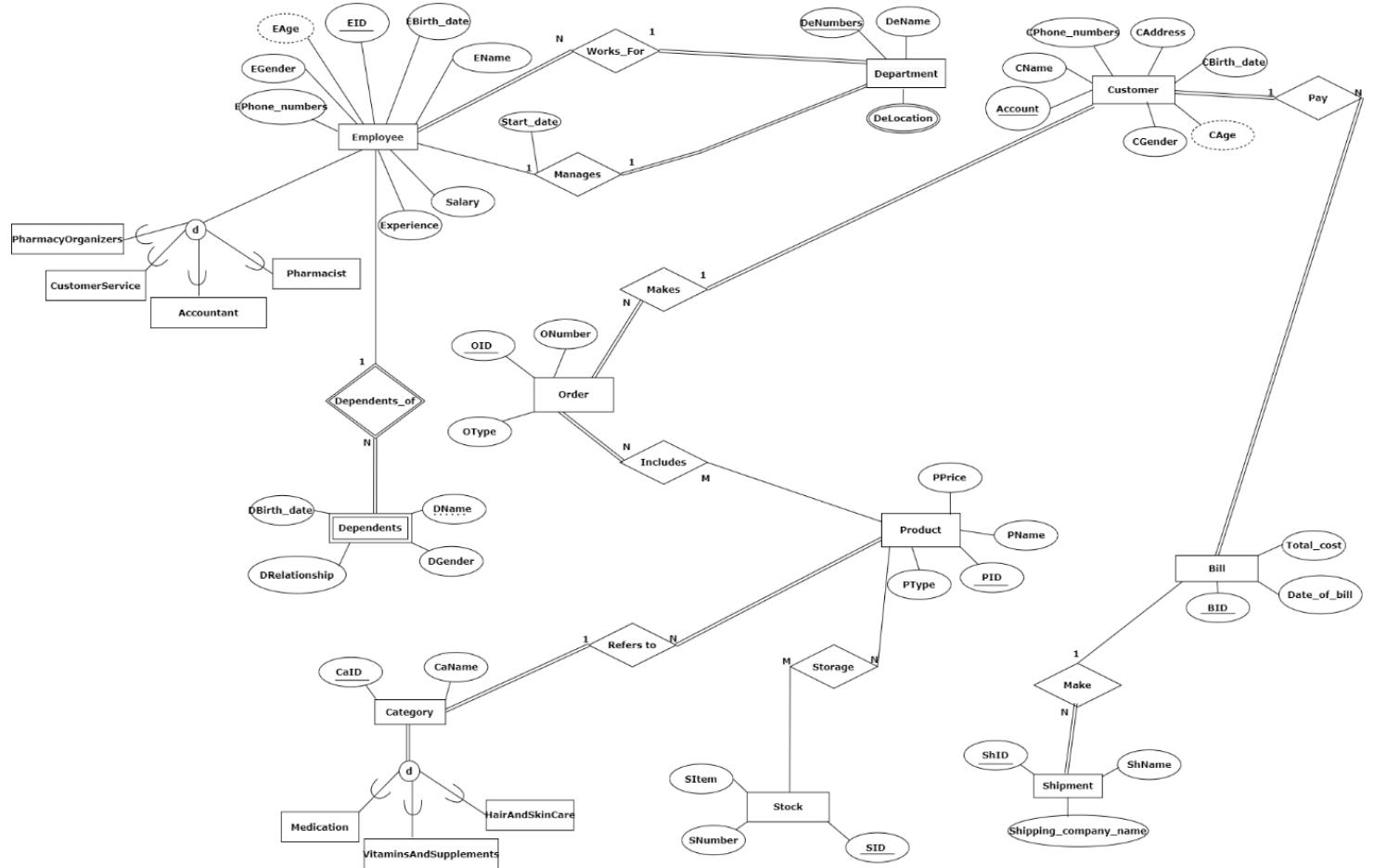
Monthly profit report

Report on the best-selling products

PART II: DB DESIGN

2 ER Diagram Design

2.1 ER diagram



2.2 Design of Business Rules

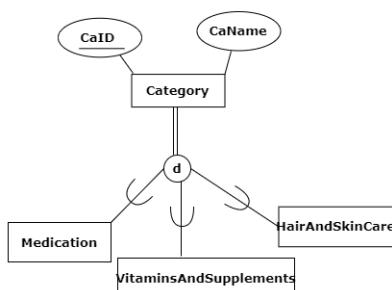
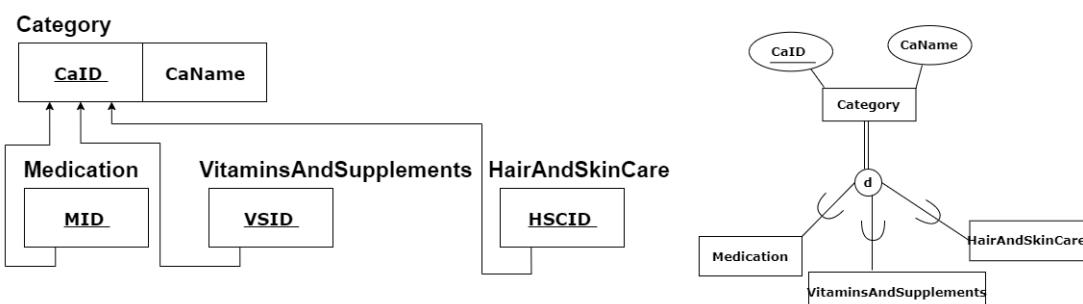
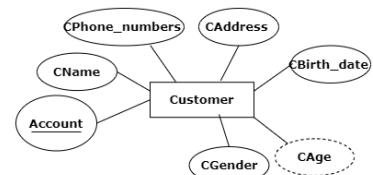
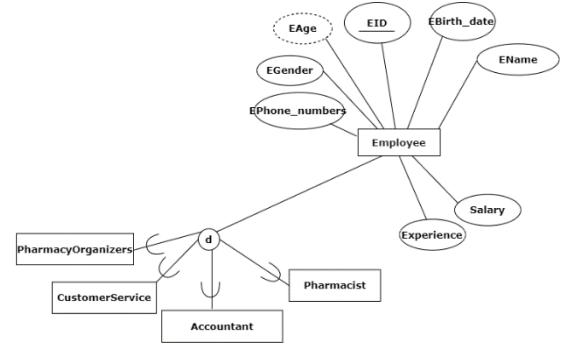
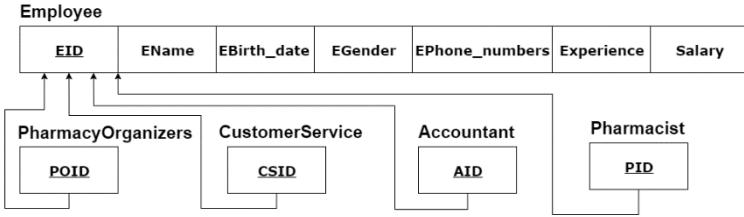
Business Rule	Design Decisions	Justification (if any)
Each product must be in a specific category.	1: N relationship between category and product, and it must be total participation for both parts.	Pharmacy contains many products. Each product belongs to one category.
Each employee may have many dependents.	1: N relationship between employee and dependent, and it must be total participation for Dependent part.	Dependent is a weak entity type. The name of the dependent is the partial key.

There are many departments in the pharmacy and each department has its employees.	1: N relationship between departments and employees, and it must be total participation for both parts.	A department can be related to any number of employees (N), but an employee must only be related to at most one department.
The customers can make orders.	1: N relationship between customer and order, and it must be total participation for both parts.	Each customer can make many orders. Also, a person is considered to be a customer only if they have made an order. The customer entity has total participation in this relationship (all customers have made an order, and we can't have a customer who hasn't made an order).
The pharmacy employs at least seven employees: one pharmacist, two accountants, two employees service representatives, and two employees who care about the drugstore and organize the medicines. Also, there is an employee who manages the departments.	1:1 relationship between the employee (manager) and departments, and it must be total participation for departments part.	An employee can manage at most one department and a department can have at most one manager.
Stock refers to the products that are ready to be sold to the customer.	M: N relationship between stock and product, and its partial participation for both parts.	Many products are stored in many stocks.

Order includes many products.	M: N relationship between order and product. and it must be total participation for the order part.	Order includes any number of products. Each Product can be included in any number of Orders.
Each bill should calculate the total cost for every customer and make a shipment.	M: N relationship between bill and shipment, and it must be total participation for shipment part.	Every shipment should have a bill, but not every bill should make a shipment.
Each bill should calculate the total cost for every customer and make a shipment.	1: N relationship between customer and bill, and it must be total participation for both parts.	Each customer can pay multiple bills. Each bill is paid by one customer.
The experience for the employee is preferred but not essential.	Experience attribute can't be the key attribute.	The experience attribute could be null.
The pharmacy should be divided into the following categories: medication, vitamins & supplements, hair, and skincare.	Superclass/subclass relationship with disjointness and completeness constraints.	Total specialization.
The pharmacy employs at least seven employees: one pharmacist, two accountants, two employees for customer service, and two employees who care about the drugstore and organize the medicines.	Superclass/subclass relationship with disjointness and completeness constraints.	Partial specialization, since not every employee, has a specialization. Each employee has only one specialization, hence it's disjoint.

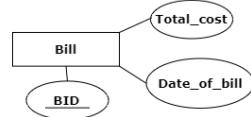
3 ER-to-logical schema mapping

3.1 Mapping of Regular Entity Types



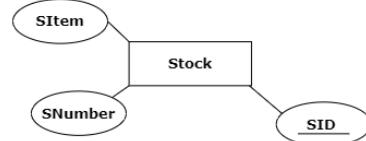
Bill

<u>BID</u>	Date_of_bill	Total_cost
------------	--------------	------------



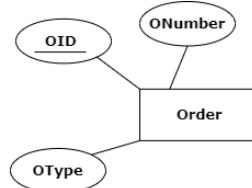
Stock

<u>SID</u>	SItem	SNumber
------------	-------	---------



Order

<u>OID</u>	ONumber	OType
------------	---------	-------



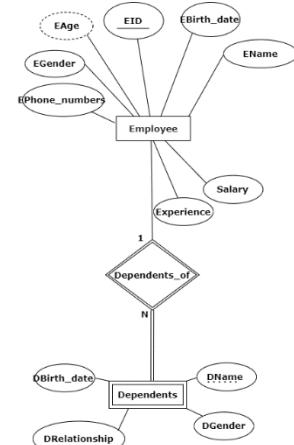
3.2 Mapping of Weak Entity Types

Employee

EID	EName	EBirth_date	EGender	EPhone_numbers	Experience	Salary
-----	-------	-------------	---------	----------------	------------	--------

Dependents

DName_	EID	DBirth_date	DGender	DRelationship
--------	-----	-------------	---------	---------------



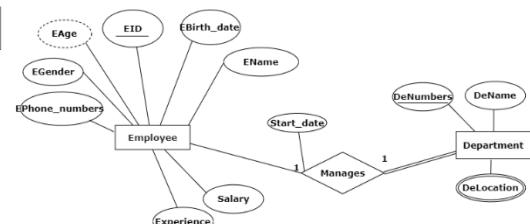
3.3 Mapping of binary 1-1 relationship types

Employee

EID	EName	EBirth_date	EGender	EPhone_numbers	Experience	Salary
-----	-------	-------------	---------	----------------	------------	--------

Department

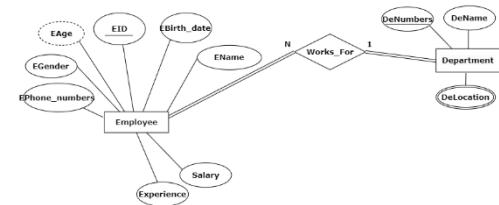
DeNumbers	DeName	Start_date	EID
-----------	--------	------------	-----



3.4 Mapping of binary 1-N relationship types

Employee							
EID	EName	EBirth_date	EGender	EPhone_numbers	Experience	Salary	DeNumbers

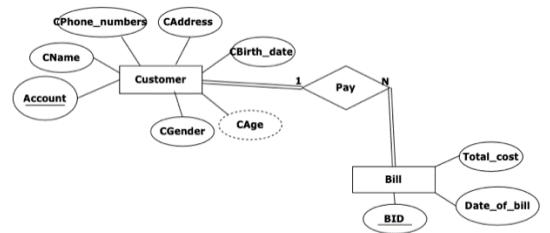
Department		
DeNumbers	DeName	Start_date

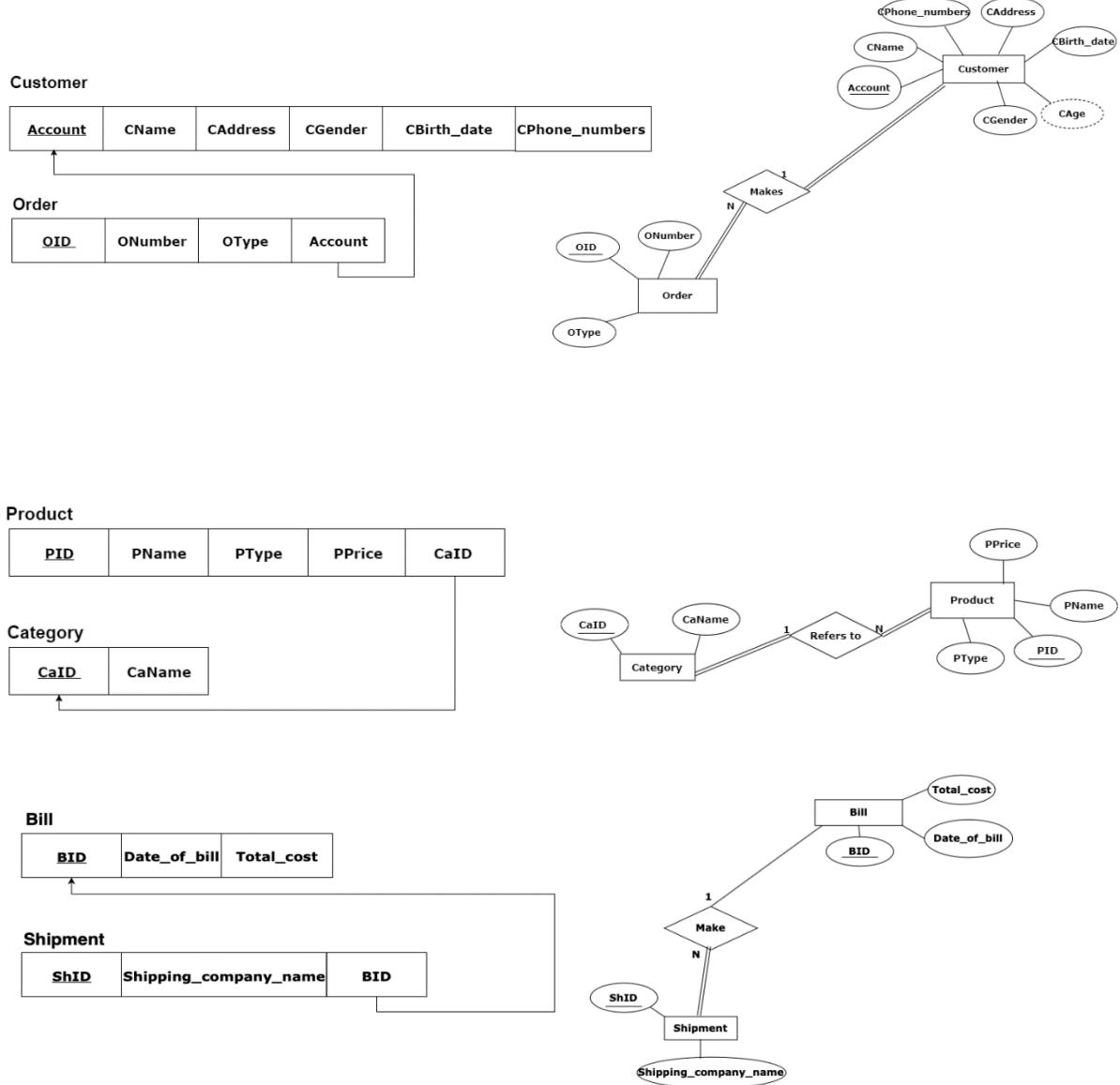


Customer

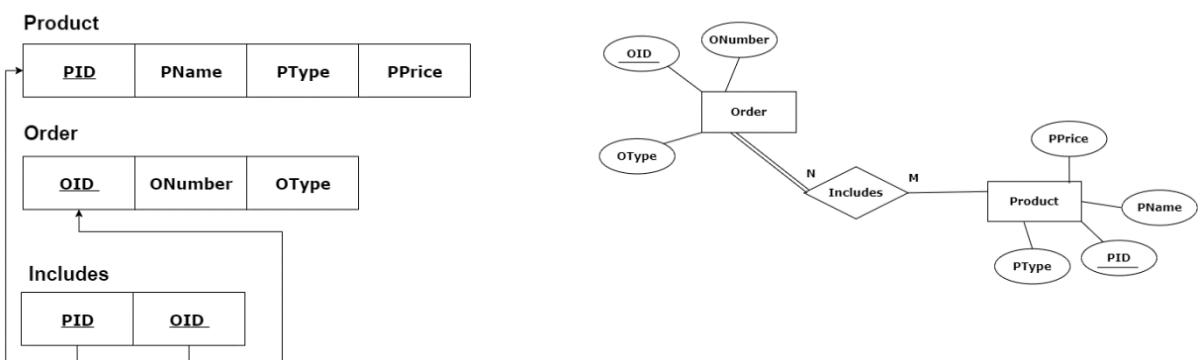
Account	CName	CAddress	CGender	CBirth_date	CPhone_numbers

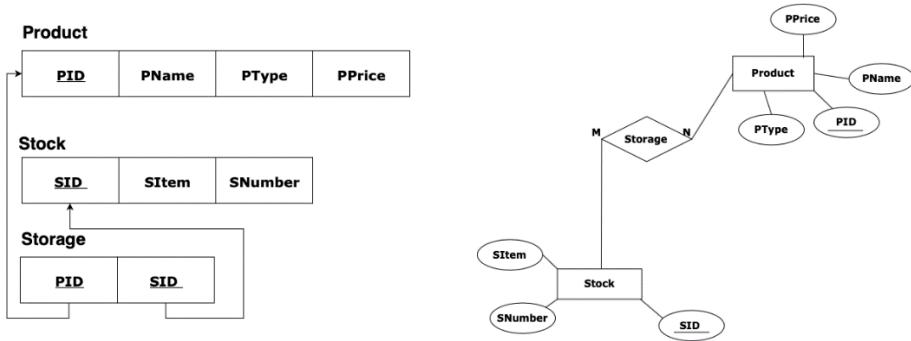
BID	Date_of_bill	Total_cost	Account



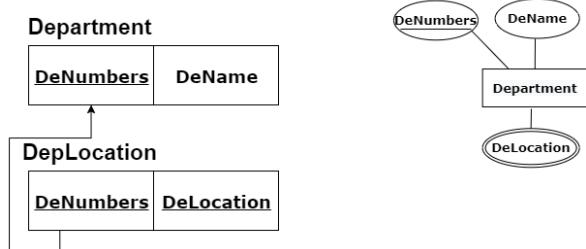


3.5 Mapping of binary M-N relationship types





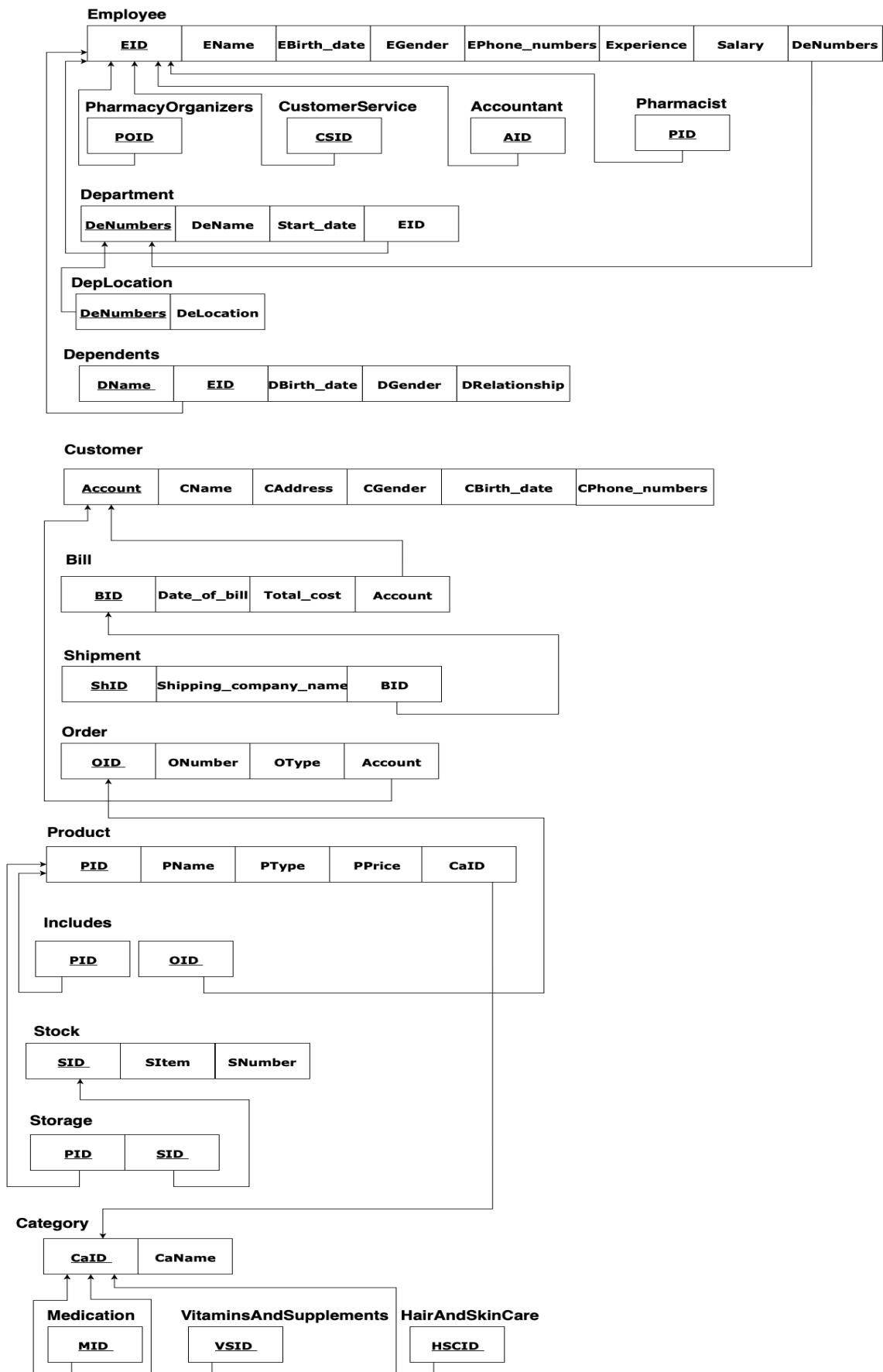
3.6 Mapping of multivalued attributes



3.7 Mapping of n-ary relationship types

None.

3.8 Schema Diagram



4 Normalization

4.1 First Normal Form

Employee

EID	EName	EBirth_date	EGender	EPhone_numbers	Experience	Salary	DeNumbers
-----	-------	-------------	---------	----------------	------------	--------	-----------

PharmacyOrganizers

POID

CustomerService

CSID

Accountant

AID

Pharmacist

PID

Department

DeNumbers	DeName	Start_date	EID
-----------	--------	------------	-----

DepLocation

DeNumbers	DeLocation
-----------	------------

Dependents

DName	EID	DBirth_date	DGender	DRelationship
-------	-----	-------------	---------	---------------

Customer

Account	CName	CAddress	CGender	CBirth_date	CPhone_numbers
---------	-------	----------	---------	-------------	----------------

Bill

BID	Date_of_bill	Total_cost	Account
-----	--------------	------------	---------

Shipment

ShID	Shipping_company_name	BID
------	-----------------------	-----

Order

OID	ONumber	OType	Account
-----	---------	-------	---------

Product

PID	PName	PType	PPrice	CaID
-----	-------	-------	--------	------

Includes

PID	OID
-----	-----

Stock

SID	SItem	SNumber
-----	-------	---------

Storage

PID	SID
-----	-----

Category

CaID	CaName
------	--------

Medication

MID

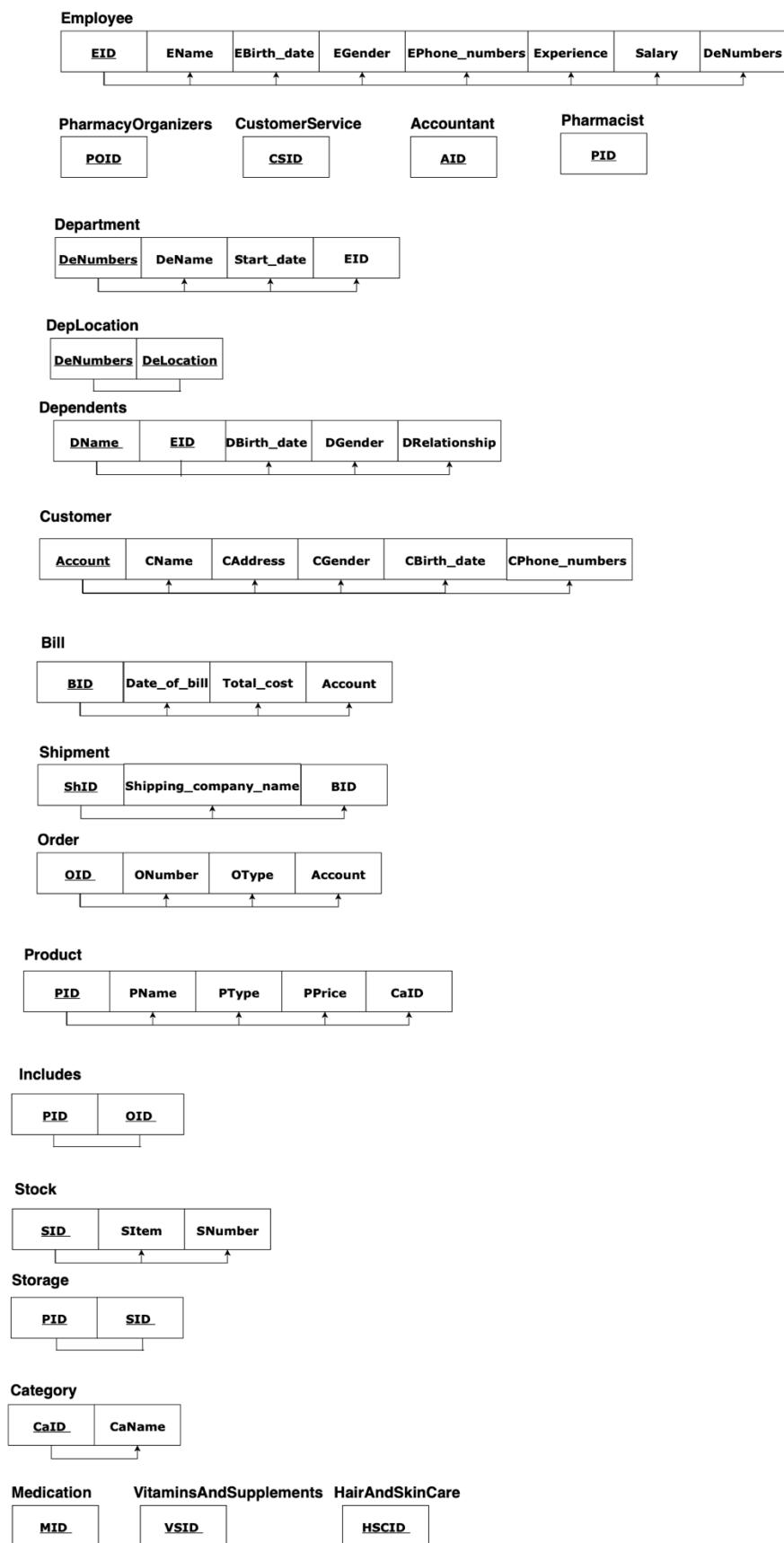
VitaminsAndSupplements

VSID

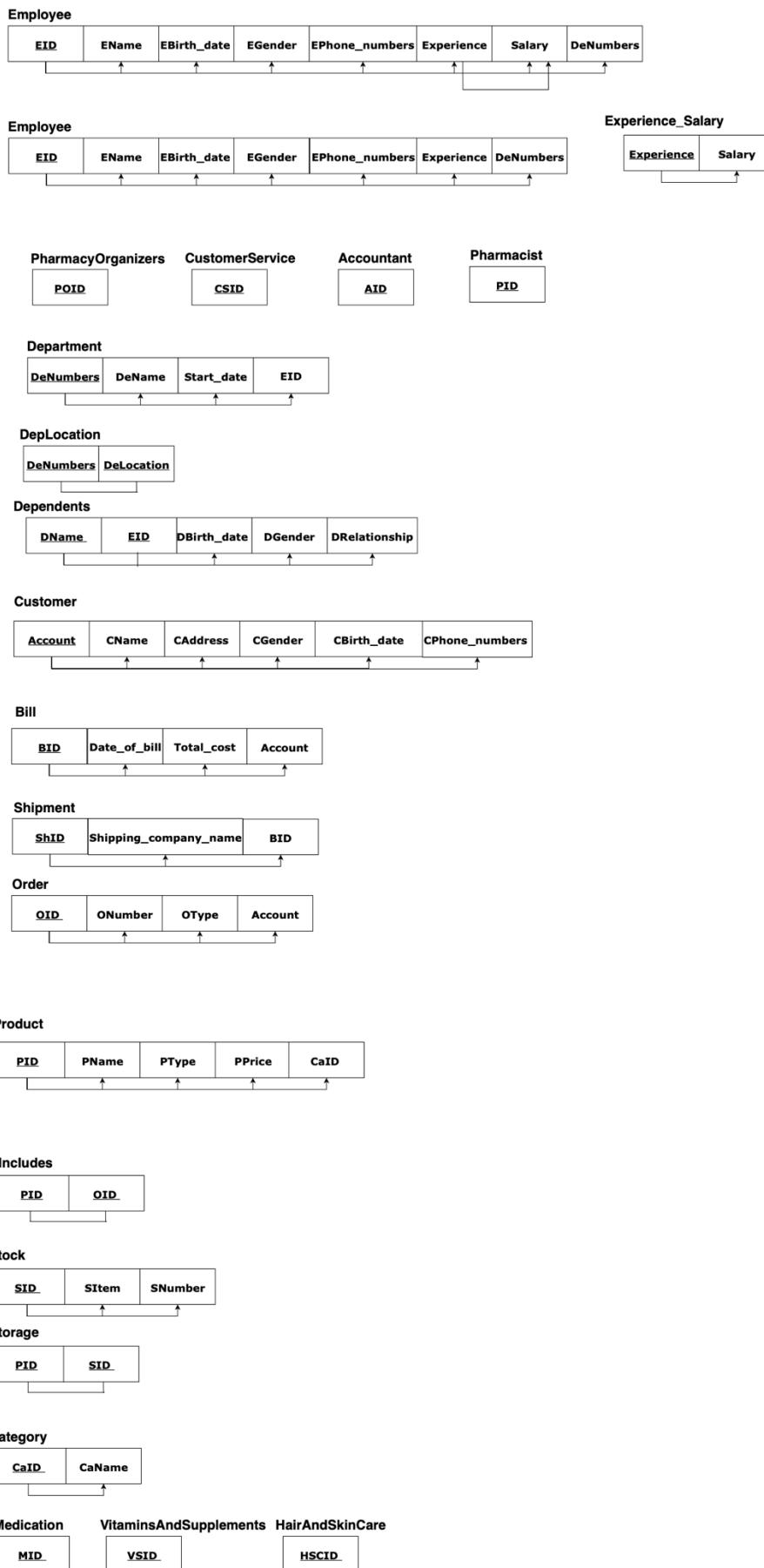
HairAndSkinCare

HSCID

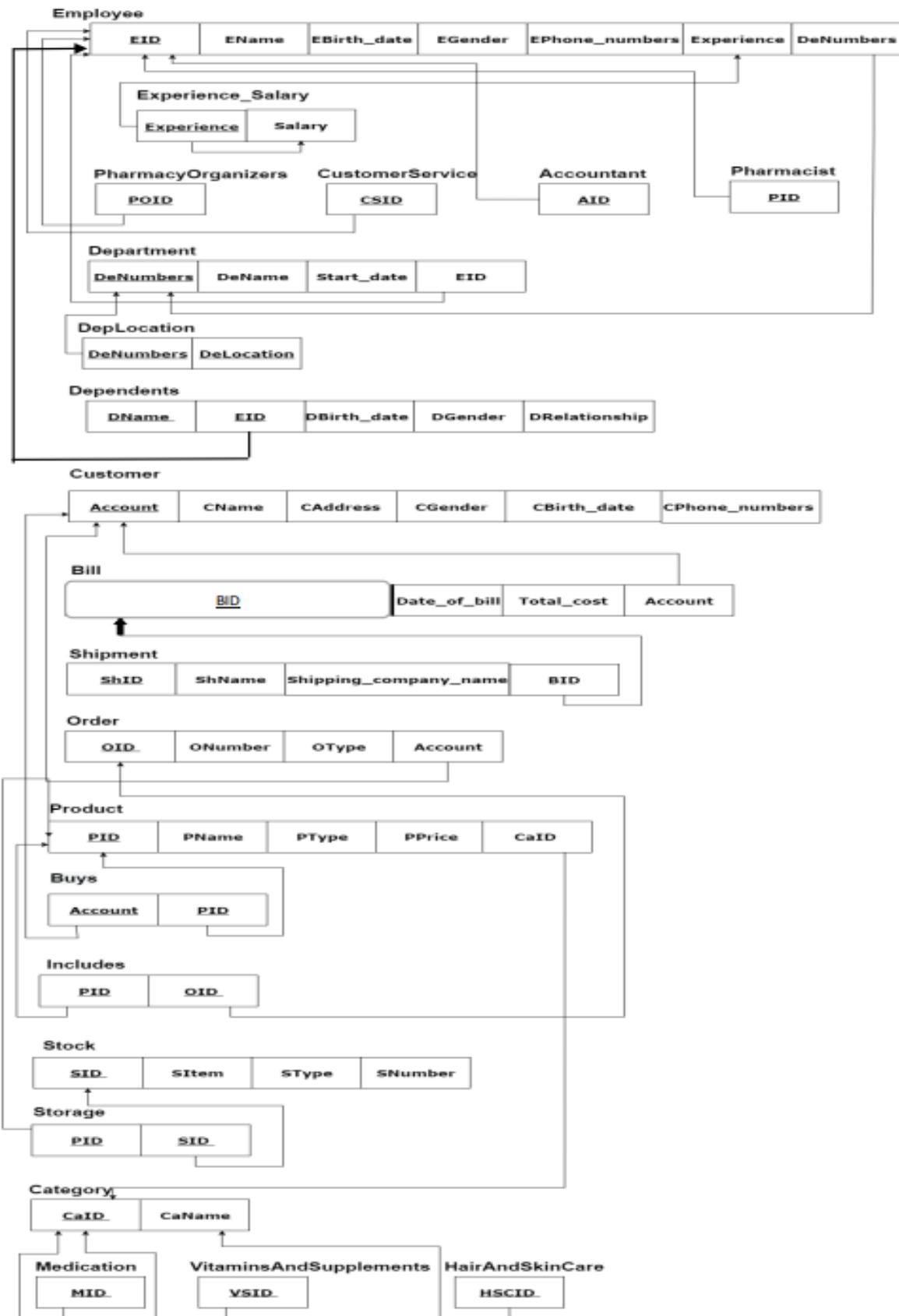
4.2 Second Normal Form



4.3 Third Normal Form



5 Final DB Schema Diagram



PHASE III: IMPLEMENTATION

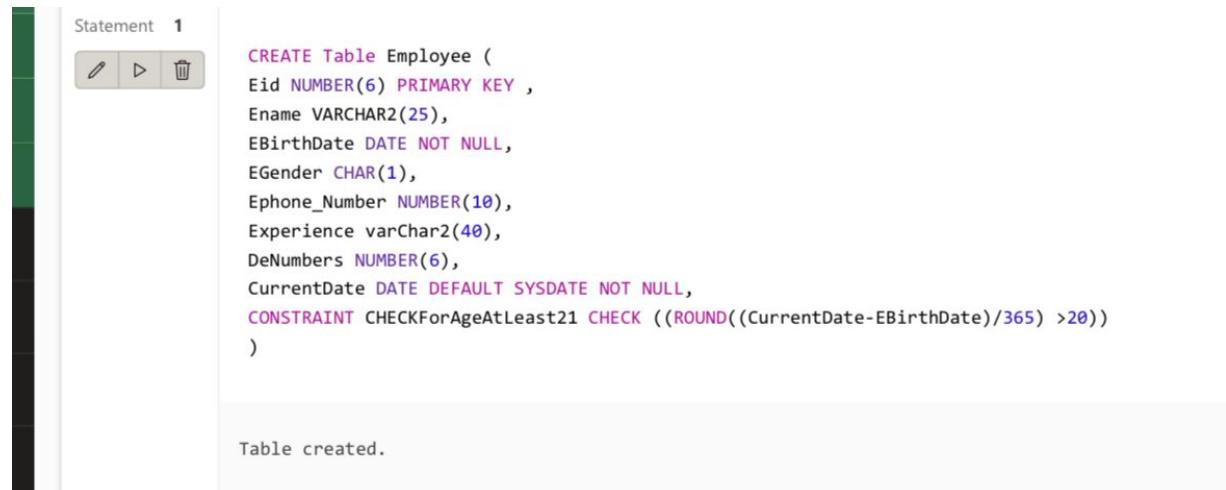
Table Creation Script

6.1 < Employee > TABLE

```
CREATE Table Employee (
    Eid NUMBER(6) PRIMARY KEY ,
    Ename VARCHAR2(25),
    EBirthDate DATE NOT NULL,
    EGender CHAR(1),
    Ephone_Number NUMBER(10),
    Experience varChar2(40),
    DeNumbers NUMBER(6),
    CurrentDate DATE DEFAULT SYSDATE NOT NULL,
    CONSTRAINT CHECKForAgeAtLeast21 CHECK ((ROUND((CurrentDate-EBirthDate)/365) >20))
);
```

Alter table Employee

```
add constraint fk_DeNumbers foreign key (DeNumbers)references department(DeNumbers) ON
DELETE CASCADE;
```



The screenshot shows the Oracle SQL Developer interface with a code editor window. The code in the editor is identical to the one provided above, creating the Employee table with its columns, constraints, and a foreign key constraint named fk_DeNumbers.

```
Statement 1
CREATE Table Employee (
    Eid NUMBER(6) PRIMARY KEY ,
    Ename VARCHAR2(25),
    EBirthDate DATE NOT NULL,
    EGender CHAR(1),
    Ephone_Number NUMBER(10),
    Experience varChar2(40),
    DeNumbers NUMBER(6),
    CurrentDate DATE DEFAULT SYSDATE NOT NULL,
    CONSTRAINT CHECKForAgeAtLeast21 CHECK ((ROUND((CurrentDate-EBirthDate)/365) >20))
)

Table created.
```

6.2 < Experience_Salary > TABLE

```
CREATE TABLE Experience_Salary (
    Experience varChar2(40)PRIMARY KEY,
    salary NUMBER(9)
);
```

Statement 2

```
CREATE TABLE Experience_Salary (
    Experience varChar2(40)PRIMARY KEY,
    salary NUMBER(9)
)
```

Table created.

6.3 < PharmacyOrganizer > TABLE

```
CREATE TABLE PharmacyOrganizer (
    POid NUMBER(6)PRIMARY KEY
);
```

Alter table PharmacyOrganizer

```
add constraint fk_POid foreign key (POid)references Employee(Eid) ON DELETE CASCADE;
```

Statement 3

```
CREATE TABLE PharmacyOrganizer (
    POid NUMBER(6)PRIMARY KEY
)
```

Table created.

6.4 < CustomerService > TABLE

```
CREATE TABLE CustomerService (
    CSid NUMBER(6) PRIMARY KEY
);
```

Alter table CustomerService

```
add constraint fk_CSid foreign key (CSid) references Employee(Eid) ON DELETE CASCADE;
```



The screenshot shows a SQL developer interface with Statement 4. The code entered is:

```
CREATE TABLE CustomerService (
    CSid NUMBER(6) PRIMARY KEY
)
```

The output window below shows the message "Table created."

6.5 < Accountant > TABLE

```
CREATE TABLE Accountant (
    Aid NUMBER(6) PRIMARY KEY
);
```

Alter table Accountant

```
add constraint fk_Aid foreign key (Aid) references Employee(Eid) ON DELETE CASCADE;
```



The screenshot shows a SQL developer interface with Statement 5. The code entered is:

```
CREATE TABLE Accountant (
    Aid NUMBER(6) PRIMARY KEY
)
```

The output window below shows the message "Table created."

6.6 < Pharmacist > TABLE

```
CREATE TABLE Pharmacist (
    Pid NUMBER(6) PRIMARY KEY
);
Alter table Pharmacist
add constraint fk_Pid foreign key (Pid) references Employee(Eid) ON DELETE CASCADE;
```

The screenshot shows the SQL developer interface with Statement 6 selected. The code entered is:

```
CREATE TABLE Pharmacist (
    Pid NUMBER(6) PRIMARY KEY
);
```

The output window below shows the message "Table created."

6.7 < department > TABLE

```
CREATE TABLE department (
    DeNumbers NUMBER(6) PRIMARY KEY,
    DeName VARCHAR2(25),
    StartDate DATE ,
    Eid NUMBER(6) NOT NULL
);
Alter table department
add constraint fk_Eid foreign key (Eid) references Employee(Eid) ON DELETE CASCADE;
```

The screenshot shows the SQL developer interface with Statement 7 selected. The code entered is:

```
CREATE TABLE department (
    DeNumbers NUMBER(6) PRIMARY KEY,
    DeName VARCHAR2(25),
    StartDate DATE ,
    Eid NUMBER(6) NOT NULL
);
```

The output window below shows the message "Table created."

6.8 < Deplocation > TABLE

```
CREATE TABLE Deplocation (
    DeNumbers NUMBER(6),
    DeLocation VARCHAR2(25),
    constraint pk_DeNumberAndDeLocation primary key(DeNumbers,DeLocation)
);
```

Alter table Deplocation

```
add constraint fk_Dep_DeNumbers foreign key (DeNumbers)references department(DeNumbers)
ON DELETE CASCADE;
```

The screenshot shows the Oracle SQL Developer interface. On the left, there's a toolbar with icons for edit, run, and delete. Next to it is a status bar showing 'Statement 8'. The main area contains the SQL code for creating the 'Deplocation' table. Below the code, a message box displays 'Table created.'.

```
Statement 8
CREATE TABLE Deplocation (
    DeNumbers NUMBER(6),
    DeLocation VARCHAR2(25),
    constraint pk_DeNumberAndDeLocation primary key(DeNumbers,DeLocation)
)

Table created.
```

6.9 < Customer > TABLE

```
CREATE TABLE Customer (
    CAccount VARCHAR2(10) PRIMARY KEY,
    CName VARCHAR2(25),
    CAddress NUMBER(4),
    CGender VARCHAR2(1),
    CBirthDate DATE ,
    CPhoneNumbers NUMBER(10)
);
```

Statement 9



```
CREATE TABLE Customer (
    CAccount VARCHAR2(10) PRIMARY KEY,
    CName VARCHAR2(25),
    CAddress NUMBER(4),
    CGender VARCHAR2(1),
    CBirthDate DATE ,
    CPhoneNumbers NUMBER(10)
)
```

Table created.

6.10 < Dependents > TABLE

```
CREATE TABLE Dependents(
    DName VARCHAR2(25) ,
    Eid NUMBER(6),
    DBirthDate DATE ,
    DGender VARCHAR2(9),
    DRelationship VARCHAR2(20),
    constraint pk_DNameAndEid primary key(DName,Eid)
);
```

Alter table Dependents

add constraint fk_Dependent_Eid foreign key (Eid) references Employee(Eid) ON DELETE CASCADE;

Statement 10



```
CREATE TABLE Dependents(
    DName VARCHAR2(25) ,
    Eid NUMBER(6),
    DBirthDate DATE ,
    DGender VARCHAR2(9),
    DRelationship VARCHAR2(20),
    constraint pk_DNameAndEid primary key(DName,Eid)
)
```

Table created.

6.11 < Product > TABLE

```
CREATE TABLE Product (
    Pid Number (6) PRIMARY KEY ,
    PName VARCHAR2(50) NOT NULL,
    PType VARCHAR2(25),
    PPrice Number(4) NOT NULL,
    CAid Number (6) NOT NULL
);
```

Statement 11

```
CREATE TABLE Product (
    Pid Number (6) PRIMARY KEY ,
    PName VARCHAR2(50) NOT NULL,
    PType VARCHAR2(25),
    PPrice Number(4) NOT NULL,
    CAid Number (6) NOT NULL
)
```

Table created.

6.12 < Bill > TABLE

```
CREATE TABLE Bill (
    Bid NUMBER(6) PRIMARY KEY,
    DateOfBill DATE ,
    TotalCost NUMBER(20) NOT NULL,
    BAccount VARCHAR2(10)
);
```

Alter table Bill

```
add constraint fk_Account foreign key (BAccount)references Customer(CAccount) ON DELETE CASCADE;
```

Statement 12



```
CREATE TABLE Bill (
    Bid NUMBER(6) PRIMARY KEY,
    DateOfBill DATE ,
    TotalCost NUMBER(20) NOT NULL,
    BAccount VARCHAR2(10)
)
```

Table created.

6.13< Shipment > TABLE

```
CREATE TABLE Shipment (
    SHid NUMBER (6) NOT NULL PRIMARY KEY ,
    ShippingCompanyName VARCHAR2(100) NOT NULL,
    Bid NUMBER(6) NOT NULL
);
```

Alter table Shipment

```
add constraint fk_Bid foreign key (Bid)references Bill(Bid) ON DELETE CASCADE;
```

Statement 13



```
CREATE TABLE Shipment (
    SHid NUMBER (6) NOT NULL PRIMARY KEY ,
    ShippingCompanyName VARCHAR2(100) NOT NULL,
    Bid NUMBER(6) NOT NULL
)
```

Table created.

6.14 < Orders > TABLE

```
CREATE TABLE Orders (
    ORid NUMBER(6) PRIMARY KEY,
    ONumber NUMBER(2),
    OType VARCHAR2(20),
```

```
OAccount VARCHAR2(10)
```

```
);
```

Alter table Orders

```
add constraint fk_OAccount foreign key (OAccount)references Customer(CAccount) ON DELETE CASCADE;
```

Statement 14



```
CREATE TABLE Orders (
    ORid NUMBER(6) PRIMARY KEY,
    ONumber NUMBER(2),
    OType VARCHAR2(20),
    OAccount VARCHAR2(10)
)
```

Table created.

6.15 < Includes > TABLE

```
CREATE TABLE Includes (
```

```
    ORid NUMBER(6),
```

```
    Pid NUMBER(6),
```

```
constraint pk_PidAndOid primary key(Pid,ORid)
```

```
);
```

Alter table Includes

```
add constraint fk_In_Pid foreign key (Pid)references Product(Pid) ON DELETE CASCADE;
```

Alter table Includes

```
add constraint fk_Oid foreign key (ORid)references Orders(ORid) ON DELETE CASCADE;
```

Statement 15



```
CREATE TABLE Includes (
    ORid NUMBER(6),
    Pid NUMBER(6),

    constraint pk_PidAndOrid primary key(Pid,ORid)
)
```

Table created.

6.16 < Stock > TABLE

```
CREATE TABLE Stock (
    STid NUMBER (6) PRIMARY KEY,
    SItem VARCHAR2(50),
    SNumber NUMBER(4)
);
```

Statement 16



```
CREATE TABLE Stock (
    STid NUMBER (6) PRIMARY KEY,
    SItem VARCHAR2(50),
    SNumber NUMBER(4)
)
```

Table created.

6.17 < Storage1 > TABLE

```
CREATE TABLE Storage1 (
    STid NUMBER(6),
    Pid NUMBER(6),
    constraint pk_PidAndSid primary key(Pid,STid)
);
Alter table Storage1
add constraint fk_st_Pid foreign key (Pid)references Product(Pid) ON DELETE CASCADE;
Alter table Storage1
add constraint fk_STid foreign key (STid)references stock(STid) ON DELETE CASCADE;
```

Statement 17

```
CREATE TABLE Storage1 (
    STid NUMBER(6),
    Pid NUMBER(6),
    constraint pk_PidAndSid primary key(Pid,STid)
)
```

Table created.

6.18 < Category1> TABLE

```
CREATE TABLE Category1 (
    CAid NUMBER (6) PRIMARY KEY,
    CName VARCHAR2(25)
);
```

Statement 18



```
CREATE TABLE Category1 (
    CAid NUMBER (6) PRIMARY KEY,
    CName VARCHAR2(25)
)
```

Table created.

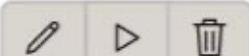
6.19 < Medication > TABLE

```
CREATE TABLE Medication (
    Mid NUMBER(6)PRIMARY KEY
);
```

Alter table Medication

```
add constraint fk_Mid foreign key (Mid)references Category1(CAid) ON DELETE CASCADE;
```

Statement 19



```
CREATE TABLE Medication (
    Mid NUMBER(6)PRIMARY KEY
)
```

Table created.

6.20 < VitaminsAndSupplements > TABLE

```
CREATE TABLE VitaminsAndSupplements (
    VSid NUMBER(6)PRIMARY KEY
);
```

Alter table VitaminsAndSupplements

```
add constraint fk_VSid foreign key (VSid)references Category1(CAid) ON DELETE CASCADE;
```

Statement **20**



```
CREATE TABLE VitaminsAndSupplements (
    VSid NUMBER(6) PRIMARY KEY
)
```

Table created.

6.21 < HairAndSkinCare > TABLE

```
CREATE TABLE HairAndSkinCare(
```

```
    HSCid NUMBER(6) PRIMARY KEY
);
```

```
Alter table HairAndSkinCare
```

```
add constraint fk_HSCid foreign key (HSCid) references Category1(CAid) ON DELETE CASCADE;
```

Statement **21**



```
CREATE TABLE HairAndSkinCare(
    HSCid NUMBER(6) PRIMARY KEY
)
```

Table created.

7 Constraints Script

Business Rule	SQL script	Table
The pharmacy employs at least seven employees: one pharmacist, two accountants, two employees service representatives, and two employees who care about the drugstore and organize the medicines.	<pre>Create the tables CREATE TABLE Experience_Salary (Eperience varchar(5)PRIMARY KEY, salary NUMBER(9)); CREATE TABLE PharmacyOrganizer (POid NUMBER(6)PRIMARY KEY); CREATE TABLE CustomerService (CSid NUMBER(6) PRIMARY KEY); CREATE TABLE Phrmacist (Pid NUMBER(6) PRIMARY KEY);</pre>	Experience_Salary PharmacyOrganizer CustomerService Phrmacist
Each employee must be at least 21 years old	<pre>CONSTRAINT CHECKForAgeAtLeast21 CHECK ((ROUND((CurrentDate- EBirthDate)/365) >21))</pre>	Employee
Every shipment should have an ID, name, company name, and bill.	<pre>CREATE TABLE Shipment (SHid NUMBER (6) NOT NULL PRIMARY KEY , ShippingCompanyName VARCHAR2(100) NOT NULL, Bid NUMBER(6) NOT NULL);</pre>	Shipment
The product must have a name and price.	<pre>CREATE TABLE Product (Pid Number (6) PRIMARY KEY ,</pre>	Product

Each product must be under a specific category.	PName VARCHAR2(25) NOT NULL, PType VARCHAR2(25), PPrice Number(20) NOT NULL, CAid Number (6) NOT NULL);	
<ul style="list-style-type: none"> Each stock item is coded with a unique number to keep track of it. 	SNumber NUMBER(20) UNiQUE	Stock
<ul style="list-style-type: none"> There are three categories of pharmacy products: <ul style="list-style-type: none"> 4- medication 5- vitamins and supplements. 6- hair and skincare 	CREATE TABLE Medication (Mid NUMBER(6)PRIMARY KEY); CREATE TABLE VitaminsAndSupplements (VSid NUMBER(6)PRIMARY KEY); CREATE TABLE HairAndSkinCare(HScid NUMBER(6)PRIMARY KEY);	Medication VitaminsAndSupplements HairAndSkinCare
<ul style="list-style-type: none"> Each bill should calculate the total cost for every customer . 	TotalCost NUMBER(20) NOT NULL,	Bill

8 Queries and Transactions

8.1 < The maximum average of each product type cost >

Query in Natural Language (English):

Calculate the average of each product type cost and then find The maximum one.

SQL Script:

```
select max(pprice_average)
from (select avg(PPRICE) as pprice_average,PTYPE
from Product
group by PTYPE) ;
```

```
1 select max(pprice_average)
2 from (select avg(PPRICE) as pprice_average,PTYPE
3 from Product
4 group by PTYPE) ;
```

Caption of the Output:

MAX(PPRICE_AVERAGE)
85.6

8.2 < The information of bills with a total cost of more than \$100 >

Query in Natural Language (English):

For bills with a total cost of more than \$100, get the bill id and shipping company name.

SQL Script:

```
select Shipment.BID , SHIPPINGCOMPANYNAME
from Shipment
where Shipment.BID in (select Bill.BID
from Bill
where Bill.TOTALCOST>100);
```

```

1 select Shipment.BID , SHIPPINGCOMPANYNAME
2 from Shipment
3 where Shipment.BID in (select Bill.BID
4 from Bill
5 where Bill.TOTALCOST>100);

```

Caption of the Output:

BID	SHIPPINGCOMPANYNAME
159287	aramex
156432	aramex
151284	aramex
152347	aramex
157777	smsa express

8.3 < The employee information that works in a specific department >

Query in Natural Language (English):

Print the Customer service manager id , name of those who work in Customer Service Department.

SQL Script:

Select employee.EID , ENAME

From Employee join CustomerService on Employee.EID = Customerservice.CSID

Where employee.EID in(Select EID

From department

Where DENAME like 'Customer Service%');

```

1 Select employee.EID , ENAME
2 From Employee join CustomerService on Employee.EID = Customerservice.CSID
3 Where employee.EID in(Select EID
4   | From department
5   | Where DENAME like 'Customer Service%');|

```

Caption of the Output:

EID	ENAME
105652	Rahaf

8.4 < The employee information in a specific period of experience >

Query in Natural Language (English):

Select employee whose experience pharmacist for two years and customer service for four years.

SQL Script:

```

SELECT ENAME, EXPERIENCE
FROM employee
WHERE EXISTS (SELECT EXPERIENCE
FROM Experience_Salary
WHERE employee.EXPERIENCE = Experience_Salary.EXPERIENCE
AND EXPERIENCE in ( 'Pharmacist for two years' , 'Customer Service for four years' ));

```

```

SELECT ENAME, EXPERIENCE
FROM employee
WHERE EXISTS (SELECT EXPERIENCE
FROM Experience_Salary
WHERE employee.EXPERIENCE = Experience_Salary.EXPERIENCE
AND EXPERIENCE in ( 'Pharmacist for two years' , 'Customer Service for four years' ));

```

Caption of the Output:

ENAME	EXPERIENCE
Dhay	Pharmacist for two years
Naif	Customer Service for four years

8.5 < The shipments information that have the highest cost during the year >

Query in Natural Language (English):

Print the month and the cost and the shipment ID of the shipments that have the highest cost during the year.

SQL Script:

```
select bill.TOTALCOST, bill.DATEOFBILL, SHIPMENT.SHID  
from bill inner join SHIPMENT on SHIPMENT.BID = bill.BID where bill.TOTALCOST =  
(select max(TOTALCOST) from bill);
```

```
1 select bill.TOTALCOST, bill.DATEOFBILL, SHIPMENT.SHID  
2 from bill inner join SHIPMENT on SHIPMENT.BID = bill.BID  
3 where bill.TOTALCOST = (select max(TOTALCOST)  
4 from bill);|
```

Caption of the Output:

TOTALCOST	DATEOFBILL	SHID
283	01-MAY-22	992995
283	01-MAY-22	992281

8.6 Update Example

Update in Natural Language (English):

Update the salary of those Who has experience of two years by raising the salary by 10%.

SQL Script:

```
update Experience_Salary  
set SALARY=SALARY*1.1  
WHERE Experience_Salary.EXPERIENCE like '%for two years%'
```

Caption of the Output:

Before updating:

```
select EXPERIENCE,SALARY  
from Experience_Salary  
WHERE Experience_Salary.EXPERIENCE like '%for two years%'
```

EXPERIENCE	SALARY
Pharmacist for two years	11000
Pharmacy Organizer for two years	4000
Customer Service for two years	5000

After updating:

EXPERIENCE	SALARY
Pharmacist for two years	12100
Pharmacy Organizer for two years	4400
Customer Service for two years	5500

8.7 Delete Example

Delete in Natural Language (English):

Delete employees whose work in the Customer Service department.

SQL script:

```
DELETE FROM employee  
WHERE DENUMBERS = ( SELECT DENUMBERS  
                      FROM department  
                     WHERE DENAME = 'Customer Service');
```

Caption of the Output:

Before deleting:

```
Select  
EID,ENAME,EBIRTHDATE,EGENDER,EPHONE_NUMBER,EXPERIENCE,DENUMBERS,CURRENTD  
ATE  
  
from employee  
  
where DENUMBERS = ( SELECT DENUMBERS  
                      FROM department  
                     WHERE DENAME = 'Customer Service');
```

EID	ENAME	EBIRTHDATE	EGENDER	EPHONE_NUMBER	EXPERIENCE	DENUMBERS	CURRENTDATE
109233	Naif	01-JAN-99	M	503945422	Customer Service for four years	13044	08-MAY-22
105652	Rahaf	01-SEP-95	F	503029922	Customer Service for five years	13044	08-MAY-22
102384	Abdulaziz	01-FEB-99	M	513029422	Customer Service for two years	13044	08-MAY-22

[Download CSV](#)

3 rows selected.

After deleting:

no data found

APPENDIX

Employee TABLE

```
select *
from Employee
```

EID	ENAME	EBIRTHDATE	EGENDER	EPHONE_NUMBER	EXPERIENCE	DENUMBERS	CURRENTDATE
101942	Dhay	01-SEP-99	F	503029422	Pharmacist for two years	13064	08-MAY-22
109423	Mohammed	10-APR-00	M	564228743	Accountant for one years	14333	08-MAY-22
109599	Nwaf	07-OCT-89	M	558736742	Accountant for seven years	14333	08-MAY-22
102842	Raghad	02-FEB-94	F	503497852	Accountant for three years	14333	08-MAY-22
109322	Anas	03-MAY-97	M	559282468	Pharmacist for five years	13064	08-MAY-22
101940	Waad	01-JUL-98	F	503029752	Pharmacy Organizer for two years	13334	08-MAY-22
108423	Salem	01-FEB-94	M	522029422	Pharmacy Organizer for three years	13334	08-MAY-22
109233	Naif	01-JAN-99	M	503945422	Customer Service for four years	13044	08-MAY-22
105652	Rahaf	01-SEP-95	F	503029922	Customer Service for five years	13044	08-MAY-22
102384	Abdulaziz	01-FEB-99	M	513029422	Customer Service for two years	13044	08-MAY-22

[Download CSV](#)

10 rows selected.

Experience_Salary TABLE

```
select *
from Experience_Salary
```

EXPERIENCE	SALARY
Pharmacist for two years	11000
Pharmacist for five years	12500
Accountant for one years	6500
Accountant for seven years	9500
Accountant for three years	7500
Pharmacy Organizer for two years	4000

Customer Service for five years	6500
Customer Service for two years	5000

[Download CSV](#)

10 rows selected.

PharmacyOrganizer TABLE

```
select *  
from PharmacyOrganizer
```

POID
101940
108423

[Download CSV](#)

2 rows selected.

CustomerService TABLE

```
select *  
from CustomerService
```

CSID
102384
105652
109233

[Download CSV](#)

3 rows selected.

Accountant TABLE

```
select *
from Accountant
```

AID
102842
109423
109599

[Download CSV](#)
3 rows selected.

Pharmacist TABLE

```
select *
from Pharmacist
```

PID
101942
109322

[Download CSV](#)
2 rows selected.

department TABLE

```
select *
from department
```

DENUMBERS	DENAME	STARTDATE	EID
13064	Pharmacist	02-FEB-20	109322
14333	Accountant	07-FEB-17	109599
13334	Pharmacy Organizer	01-NOV-19	108423
13044	Customer Service	10-SEP-18	105652

[Download CSV](#)

4 rows selected.

Deplocation TABLE

```
select *  
from Deplocation
```

DENUMBERS	DELOCATION
13044	Makkah
13044	Jeddah
13044	Riyadh
13064	Jeddah
13064	Makkah
13064	Riyadh
13334	Makkah
13334	Jeddah
13334	Riyadh
14333	Makkah
14333	Jeddah
14333	Riyadh

[Download CSV](#)

12 rows selected.

Customer TABLE

```
select *  
from Customer
```

CACCOUNT	CNAME	CADDRESS	CGENDER	CBIRTHDATE	CPHONENUMBERS
0123RN	Raneem	2429	F	01-JAN-99	564842218
0223RE	Reem	2422	F	01-JUN-98	564899918
0023KK	Khulood	2119	F	01-JAN-94	564841818
0133RN	Nour	3329	F	01-JAN-95	554842218
0333RN	Rama	2229	F	01-JAN-88	564842008

Sale10	Saleh	9999	M	01-JAN-96	556463113
shat22	shatha	9327	F	02-FEB-99	556463111
RR1155	Ranen	303	F	09-NOV-99	556060638

[Download CSV](#)

10 rows selected.

Dependents TABLE

```
select *  
from Dependents
```

DNAME	EID	DBIRTHDATE	DGENDER	DRELATIONSHIP
Ahlam	101942	02-FEB-20	F	Daughter
Ali	101942	10-AUG-18	M	Son
Rowa	109322	08-SEP-01	F	Daughter
Nasser	109322	02-DEC-12	M	Son
Rayan	108423	10-OCT-10	M	Son
Rakan	108423	10-OCT-10	M	Son
Naif	105652	01-JAN-09	M	Son
Worod	105652	09-DEC-06	F	Daughter
Mariam	105652	06-APR-04	F	Daughter
Maram	105652	06-APR-04	F	Daughter

[Download CSV](#)
10 rows selected.

Product TABLE

```
select *  
from Product
```

PID	PNAME	PTYPE	PPRICE	CAID
138372	Ventolin 100 Mcg Evohaler	prescribed	16	668374
132832	Marvelon Tablet 21pcs	prescribed	20	668374
134321	Clexane 4000 IU/0.4ml Pre-Filled Syringe	prescribed	49	668374
139237	Evra Patch 3pcs	prescribed	25	668374
131247	Cipralex 10 mg Tablet 28pcs	prescribed	90	668374
135432	Sanotact Multivitamin 20 Effervescent Tablets	non-prescribed	25	666382
135433	Swiss Image Whitening Night Cream 50 Ml	non-prescribed	55	666000
135439	Bioderma Atoderm Cream 200ml	non-prescribed	65	666000
135440	Vichy Mineral 89 Face Moisturizer 50 Ml	non-prescribed	135	666000
135500	Holista Vitamin C 1000 Mg 150 Chewable Tablets	non-prescribed	148	666382

[Download CSV](#)
10 rows selected.

Bill TABLE

```
select *
from Bill
```

BID	DATEOFBILL	TOTALCOST	BACCOUNT
159287	01-JAN-20	137	012NNN
156432	15-FEB-19	115	Ah0511
155123	23-APR-21	80	Sale10
151284	09-DEC-20	120	shat22
152347	01-MAY-22	283	RR1155
159898	01-MAY-22	85	0123RN
157777	01-MAY-22	283	0333RN
151515	01-DEC-21	65	0133RN
151316	01-JAN-20	55	0023KK
151212	09-MAY-22	25	0223RE

[Download CSV](#)

10 rows selected.

Orders TABLE

```
select *  
from Orders
```

ORID	ONUMBER	OTYPE	OACCOUNT
143493	1	rush order	0123RN
147394	2	rush order	0223RE
142314	3	normal order	0023KK
142236	4	normal order	0133RN
149384	5	normal order	0333RN
142394	6	normal order	012NNN

144123	9	normal order	shat22
149381	10	normal order	RR1155

[Download CSV](#)

10 rows selected.

Includes TABLE

```
select *  
from Includes
```

ORID	PID
142314	131247
143493	132832
147394	134321
142314	135432
142236	135433
149384	135439
149384	135440
142394	135500
143493	138372
142314	139237

[Download CSV](#)

10 rows selected.

Stock TABLE

```
select *  
from Stock
```

STID	SITEM	SNUMBER
223827	Ventolin 100 Mcg Evhohaler	40
228321	Marvelon Tablet 21pcs	71
229283	Clexane 4000 IU/0.4ml Pre-Filled Syringe	29
229217	Evra Patch 3pcs	52
229173	Cipralex 10 mg Tablet 28pcs	37
229191	Sanotact Multivitamin 20 Effervescent Tablets	21
227722	Swiss Image Whitening Night Cream 50 Ml	33
225555	Bioderma Atoderm Cream 200ml	12
221321	Vichy Mineral 89 Face Moisturizer 50 Ml	75
229004	Holista Vitamin C 1000 Mg 150 Chewable Tablets	37

[Download CSV](#)

10 rows selected.

Storage1 TABLE

```
select *  
from Storage1
```

STID	PID
229173	131247
228321	132832
229283	134321
229191	135432
227722	135433
225555	135439
221321	135440
229004	135500
223827	138372
229217	139237

[Download CSV](#)

10 rows selected.

Storage1 TABLE

```
select *  
from Storage1
```

STID	PID
229173	131247
228321	132832
229283	134321
229191	135432
227722	135433
225555	135439
221321	135440
229004	135500
223827	138372
229217	139237

[Download CSV](#)

10 rows selected.

Medication TABLE

Statement 403



```
select *  
from Medication
```

MID
668374

[Download CSV](#)

VitaminsAndSupplements TABLE

Statement 404

```
select *
from VitaminsAndSupplements
```

666382

[Download CSV](#)

HairAndSkinCare TABLE

Statement 405

```
select *
from HairAndSkinCare
```

HSCID
666000

[Download CSV](#)

Shipment TABLE

```
select *
from Shipment
```

SHID	SHIPPINGCOMPANYNAME	BID
991121	aramex	159287
992222	aramex	156432
991737	aramex	155123
994444	aramex	151284
992995	aramex	152347
992189	aramex	159898
992281	smsa express	157777
991888	smsa express	151515
992226	smsa express	151316
998765	ICOMEX	151212

[Download CSV](#)

10 rows selected.