**Database Systems**

**Report**

**Chapter I: Database Design**

**Enterprise Description**

Family Pharm is a retail pharmaceutical store located in southern Ontario. The store can hold up to a max 50 customers at any given time. It sells a wide variety of drugs, beauty care products, and other personal care products. The drug types consist of: antibiotics, anti-allergy, antiviral, decongestants, antidepressants, antacids, antipyretics, analgesics, antiseptics. laxatives, and can be prescribed and non-prescribed. The beauty supply products consist of: eyeliner, primer, nail polish, nail polish remover, blemish, and many more. Other personal care product types include: deodorants, antiperspirants, toothpaste, shampoos, body wash, soaps, and many more. There are 3 departments in the pharmacy: drugs, beauty products, and health products. The store has 5 main positions: store manager, assistant manager, professional pharmacist, shift supervisor, and store clerk. There is 1 owner, 1 manager and 4 professional pharmacists. There are 5 cashiers currently employed who have all worked for the company for varying years. The pharmacy also has accountants who handle the finance of the company.

Family Pharm provides many different ways of purchasing drugs from the store such as walk-in, call-in, or delivery to certain areas. The drugs can only be refilled if the prescription states it is allowed. For the exchange of a product and refund, the conditions are: 30 days unopened with receipt. No receipt can get you a store credit but no cash refund.

The POS system keeps a sales history for each annum. If a customer is purchasing prescribed medication, a unique identifier is assigned the the customer along with a record of their address, OHIP number, and contact information. Included in the history is the number of times the customer have refilled their prescribed drugs.

**Functional Requirement**

**Record Customer's Information**

When a customer makes a prescription drug purchase, his/her information such as name, telephone number, address, and health card number are recorded. That record is also indexed as customer identification number (CID).

**Record Payment Method**

The pharmacy accepts three type of payment; cash, credit, and certified check. Once the payment is received, the salesperson gives proof-of-payment note to the customer.

**Recording a Sale**

When a sale is completed, the system records customer id, the product purchased, total payment, date of purchase, and type of payment. The system also records the pharmacist id (PID) that sells the medicine for legal purposes. The inventory also changes according to the product(s) and amount purchased.

**Adding/removing Employee's Data**

The system keeps record of all employee data such as employee id, name, date of beginning employment, date of end employment and employee's bank account for the sake any positional changes and tax revenue.

**Recording a refund**

Employee makes the appropriate changes to the inventory on its newly updated from the refund according to the product and its quantity. The employee also makes record in the sale if the he system records customer id, product returned, total refunds, date of purchase, and type of refund (to debit or cash) for legal reasons.

**Inventory Purchase**

Employee makes the appropriate changes to the inventory on its newly updated from the refund according to the product and its quantity. The employee also makes record in the sale if the he system records customer id, product returned, total refunds, date of purchase, and type of refund (to debit or cash) for legal reasons.

**Paying Employees**

All employees are paid on a bi-weekly basis. It is either deposited directly to employees' bank accounts or by cheque.

**Normalization Tables (3.6 NF)**

**Prescription Customer Table:**

Customer ID, Customer Name, Customer Telephone Number, Customer Address, Customer's Health Card Number.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **CNAME** | **CTELE** | **CADDR** | **CHLTHCRD** |
| 001 | Alex | 905-123-4567 | 123 Fake St | 123456789 |
| 002 | Ben | 416-321-1234 | 321 Derp Cr | 323202920 |
| 003 | Carl | 647-222-2222 | 999 Unreal Ave | 203894840 |
| 004 | Eddy | 416-967-1111 | 409 Harris Blvd | 161612663 |
| 005 | Frank | 905-887-3039 | 6270 Yonge St | 500516452 |
| 006 | Gina | 416-202-2020 | 747 Boeing St | 798949494 |

**Prescription Sale Table:**

Sale ID, Customer ID, Payment, Purchased Product, Day of Purchase, Quantity Purchased, Pharmacist ID (EI), Type of Payment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SID** | **CID** | **PYMT** | **PP** | **DOPUR** | **QP** | **EI** | **TOP** |
| 1 | 002 | $20 | 99119 | 2014-3-11 | 1 | 116 | Cash |
| 2 | 004 | $15 | 99121 | 2014-5-5 | 2 | 119 | Cash |
| 3 | 003 | $30 | 99117 | 2014-10-1 | 2 | 118 | Credit |
| 4 | 005 | $27 | 99115 | 2014-2-22 | 1 | 113 | Debit |
| 5 | 006 | $8 | 99119 | 2013-12-25 | 2 | 113 | Credit |
| 6 | 001 | $50 | 99101 | 2013-11-20 | 1 | 118 | Credit |

**Employee Record Table:**

Employee Identification, Employee Name, Date of Employment, Date of Termination, Bank Account information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EI** | **EN** | **DOE** | **DOT** | **BAI** |
| 111 | Mary | 04/03/07 | N/A | 112434072 |
| 112 | Alice | 04/03/07 | N/A | 738642372 |
| 113 | John | 04/03/07 | N/A | 824253543 |
| 114 | David | 04/03/07 | N/A | 786387378 |
| 115 | Bob | 08/12/07 | N/A | 737322433 |
| 116 | Leo | 09/25/07 | N/A | 234535437 |
| 117 | Kevin | 08/18/10 | N/A | 437837343 |
| 118 | Chris | 11/27/10 | N/A | 398378378 |
| 119 | Jane | 04/04/11 | N/A | 378345312 |
| 120 | Mara | 06/11/13 | N/A | 355878383 |
| 121 | Cindy | 03/01/14 | N/A | 890745098 |

**Scheduling Table:**

Employee Identification, Position, Weekly Schedule

|  |  |  |
| --- | --- | --- |
| **EI** | **PSTN** | **WS** |
| 111 | Owner | N/A (Does not work regular hours. Visits when needed.) |
| 112 | Manager | Mon - Fri 9-5 |
| 115 | Assistant Manager | Wed-Sun 11-7 |
| 114 | Shift Supervisor | Mon, Tues 11-7, Sat, Sun 9-3 |
| 120 | Cashier | Mon, Wed, Fri 12-5 |
| 117 | Cashier | Tues, Thurs, Sat, Sun 12-5 |
| 121 | Cashier | Mon, Thurs, Sat 11-4 |
| 116 | Pharmacist | Mon-Fri 9-5 |
| 118 | Pharmacist | Wed-Sun 11-7 |
| 119 | Pharmacist | Mon-Fri 9-5 |
| 113 | Pharmacist | Wed-Sun 11-7 |

**Inventory Table:**

Inventory Name, Product Number, Quantity, Date Acquired

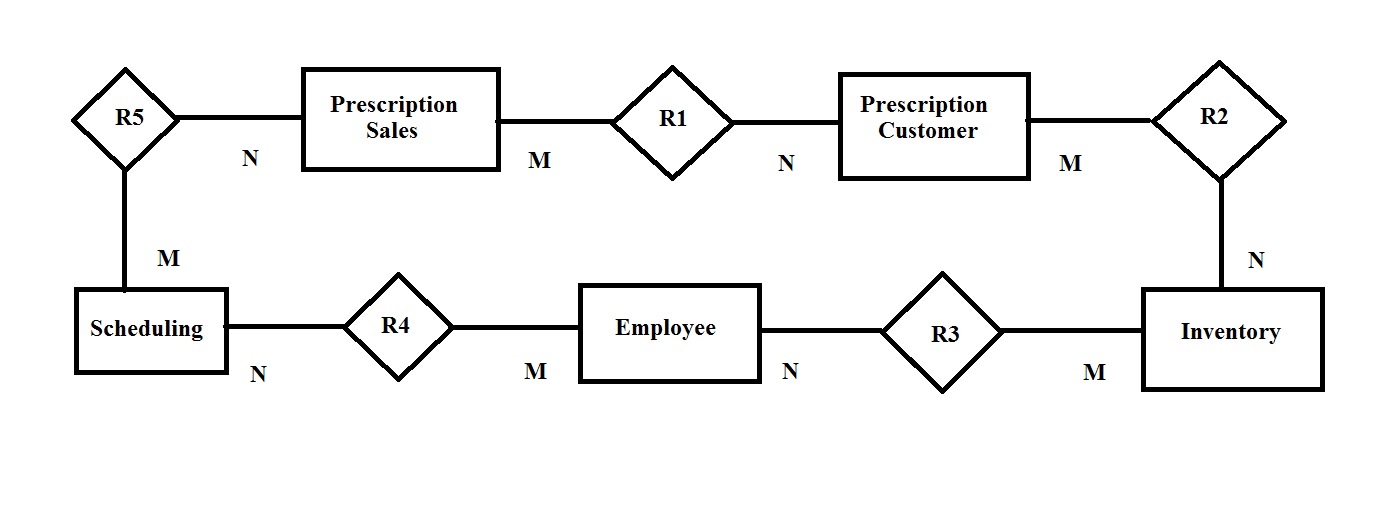
|  |  |  |  |
| --- | --- | --- | --- |
| **IN** | **PNum** | **QTY** | **DA** |
| antibiotics | 99119 | 32 | 15/09/14 |
| anti-allergy drugs | 99117 | 21 | 08/09/14 |
| antiviral drugs | 99121 | 46 | 11/09/14 |
| decongestant | 99101 | 15 | 08/09/14 |
| antiseptics | 99115 | 14 | 15/09/14 |
| deodorants | 99211 | 53 | 22/09/14 |
| toothpaste | 99263 | 47 | 22/09/14 |
| body wash | 99234 | 36 | 22/09/14 |
| soaps | 99267 | 64 | 15/09/14 |
| antiperspirants | 99248 | 30 | 15/09/14 |
| eyeliner | 99346 | 35 | 08/09/14 |
| nail polish | 99378 | 38 | 08/09/14 |
| Face creams | 99368 | 42 | 15/09/14 |
| Foundation | 99311 | 56 | 22/09/14 |
| Concealers | 99305 | 28 | 08/09/14 |

**Transaction Table:**

Transaction ID, Table Name, Operation

|  |  |  |
| --- | --- | --- |
| **TID** | **TN** | **OP** |

**ER Diagram**



|  |  |
| --- | --- |
| **R1** | Prescription customers can conduct more than 1 prescription sales. |
| **R2** | Prescription customers decrease the inventory with each purchase. |
| **R3** | Employees replenishes inventory. |
| **R4** | Employees have scheduling for their working hours. |
| **R5** | Employees scheduling depends on prescription sales and employees must be held accountable for any errors in prescription sales. |

**Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Data Code** | **Data Type** | **Data Definition** |
| **CID** | Integer | Customer ID |
| **CNAME** | Char | Customer Name |
| **CTELE** | BIGINT | Customer Telephone |
| **CADDR** | Char | Customer Address |
| **CHLTHCRD** | BIGINT | Customer’s Health Card Number |
| **PYMT** | Integer | Payment |
| **PP** | Char | Purchased Product |
| **DOPUR** | Char | Day of Purchase |
| **QP** | Integer | Quantity Purchased |
| **PID** | Integer | Pharmacist ID |
| **TOP** | Char | Type of Payment |
| **EI** | Integer | Employee Identification |
| **EN** | Char | Employee Name |
| **DOE** | Char | Date of Employment |
| **DOT** | Char | Date of Termination |
| **BAI** | BIGINT | Bank Account Information |
| **PSTN** | Char | Position |
| **PNum** | Integer | Product Number |
| **WS** | Char | Weekly schedule |
| **IN** | Char | Inventory Name |
| **QTY** | Integer | Quantity |
| **DA** | Char | Date Acquired |
| **SID** | Integer | Sale ID |
| **TID** | Integer | Transaction ID |
| **TN** | Char | Table Name |
| **OP** | Char | Operation (Add, Delete, Modify) |

**Changes to Part 1 and Part 2:**

* PID field in Prescription Sales Table changed to EI to match EI field in Employee Records Table
* Changed R5 in Description in ER diagram
* Fixed duplicate primary key value in second record from 002 to 004 in CID field in Prescription Sales table
* Changed EI as a Char type to Integer type
* Changed BAI as a Char type to BIGINT type
* Added unique key into Prescription Sales table
* Added Transaction Table

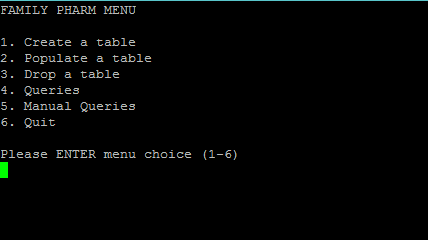
**Chapter II: User Manual**

**Installation**

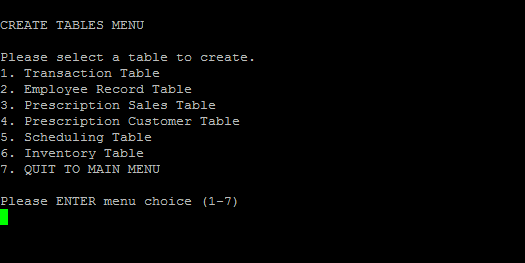
1. Make sure user has all the files: dbui.sh, prescriptionSales.txt, prescriptionCustomer.txt, employeeRecords.txt, inventory.txt, scheduling.txt
2. Make dbui.sh executable using the command “chmod 700 dbui.sh” in the terminal of a Linux operating system.
3. Move all files listed in the step 1 into turing.acs.ryerson.ca by using SSH File Transfer Client.
4. Login to “turing.acs.ryerson.ca”.
5. Run the program by typing “sh dbui.sh”.

**Instructions**

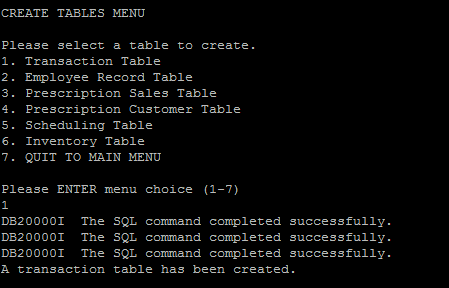
1. After starting the program, the main menu will be displayed.



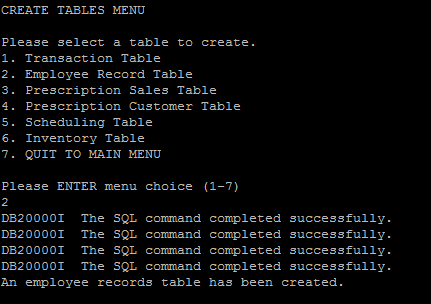
1. The user may choose an option
2. Create a table
3. Populate a table
4. Drop a table
5. Pre-existing queries
6. Manual input queries
7. Quit program
8. **Create a table**



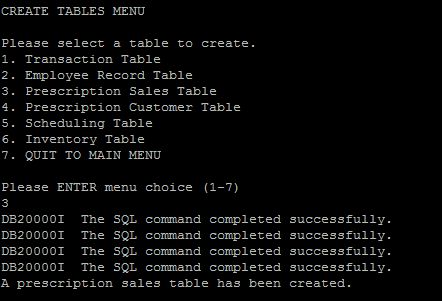
1. Create Transaction Table



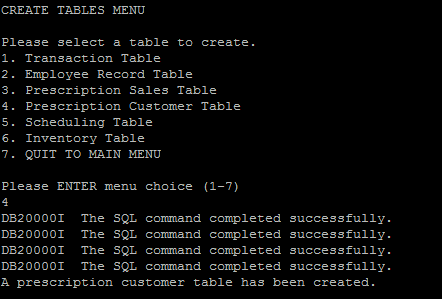
1. Create Employee Record Table



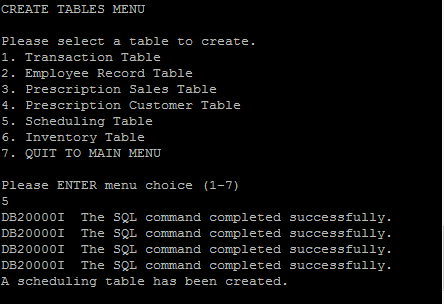
1. Create Prescription Sales Table



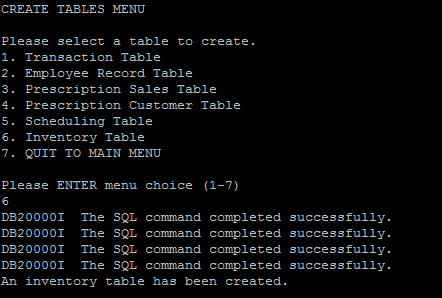
1. Create Prescription Customer Table



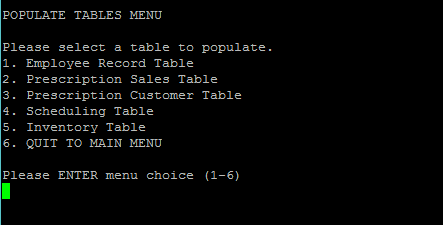
1. Create Scheduling Table



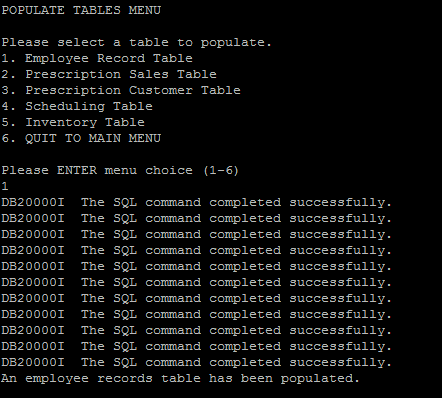
1. Create Inventory Table



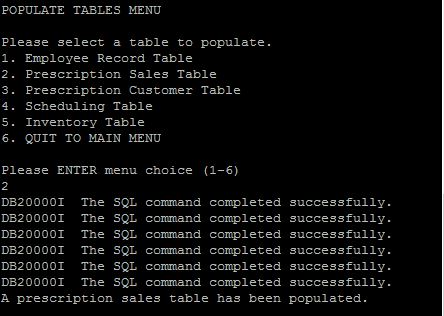
1. **Populate a table**



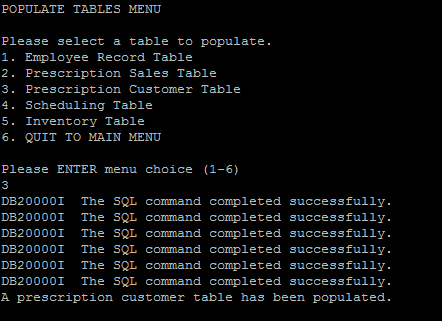
1. Populate Employee Record Table



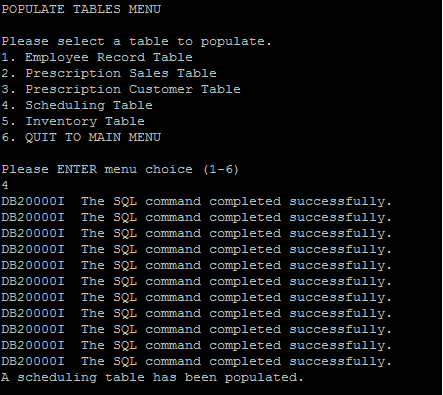
1. Populate Prescription Sales Table



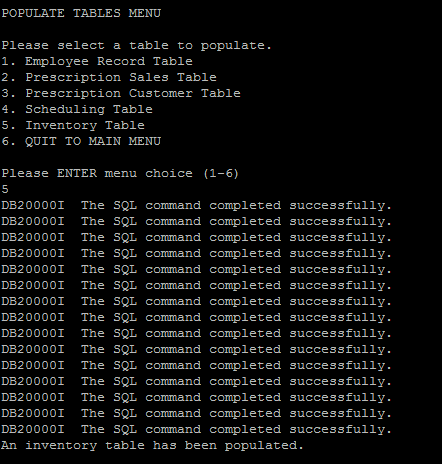
1. Populate Prescription Customer Table



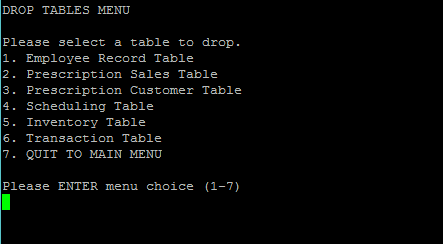
1. Populate Scheduling Table



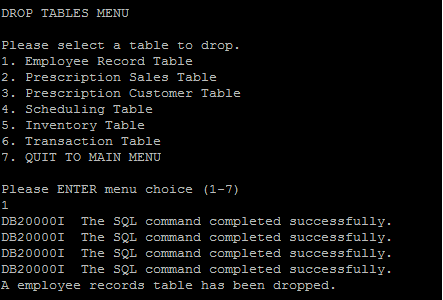
1. Populate Inventory Table



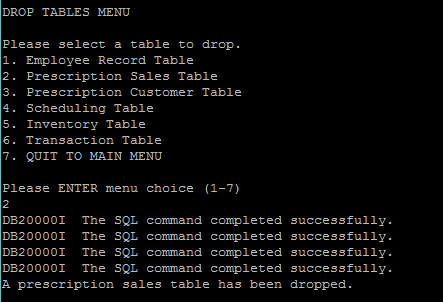
1. **Drop a table**



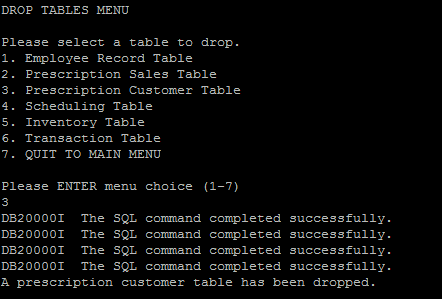
1. Drop Employee Record Table



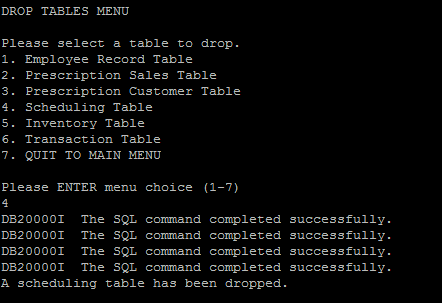
1. Drop Prescription Sales Tables



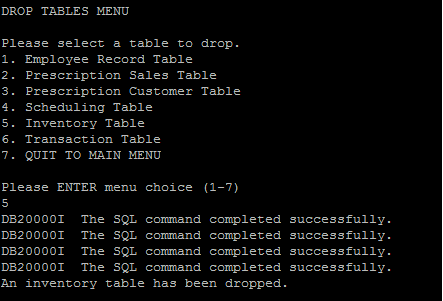
1. Drop Prescription Customer Table



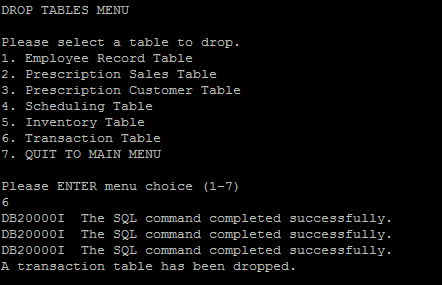
1. Drop Scheduling Table



1. Drop Inventory Table

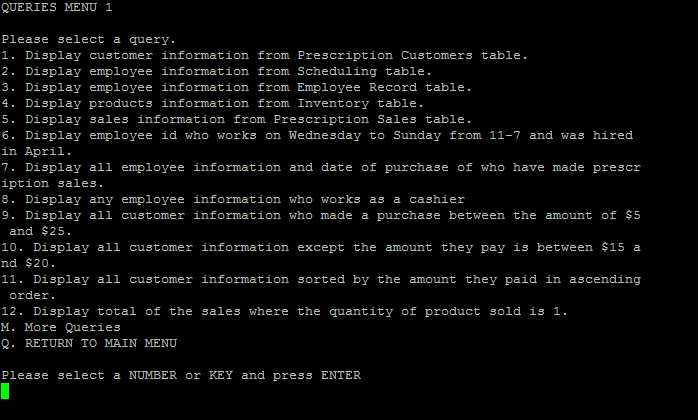


1. Drop Transaction Table

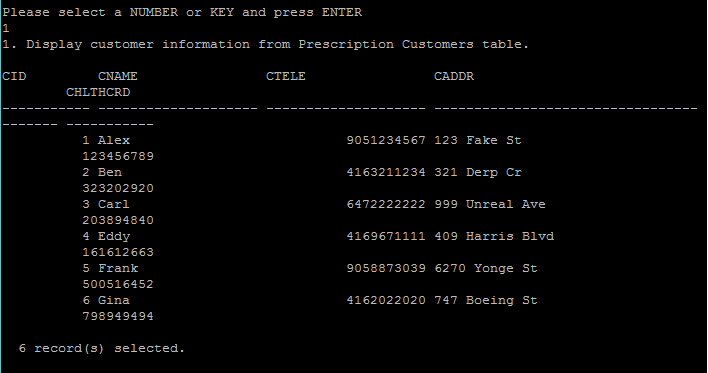


1. **Pre-existing queries**

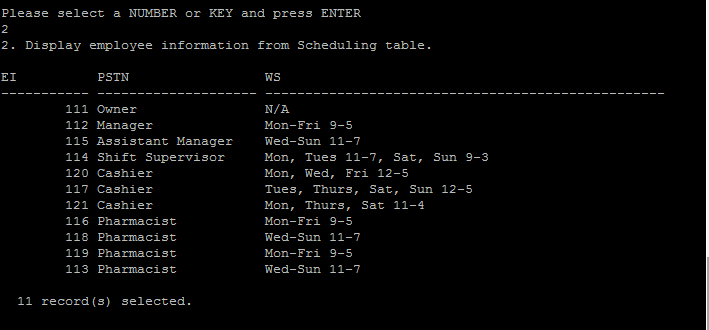
Query Menu 1



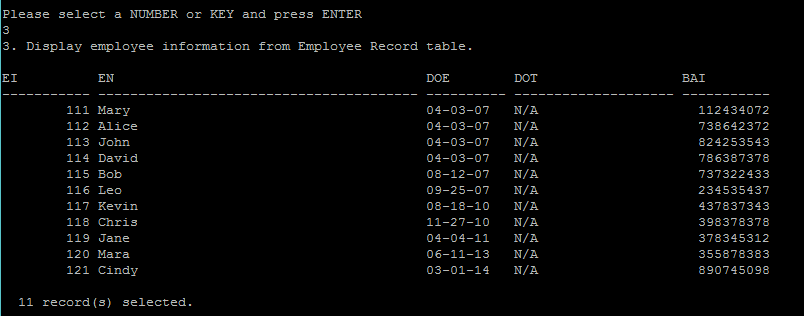
1. Display customer information from Prescription Customers table.



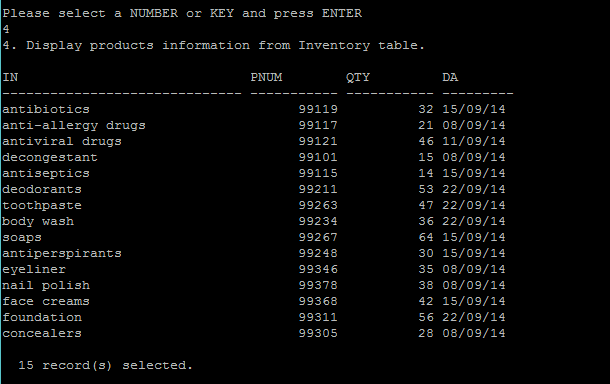
1. Display employee information from Scheduling table.



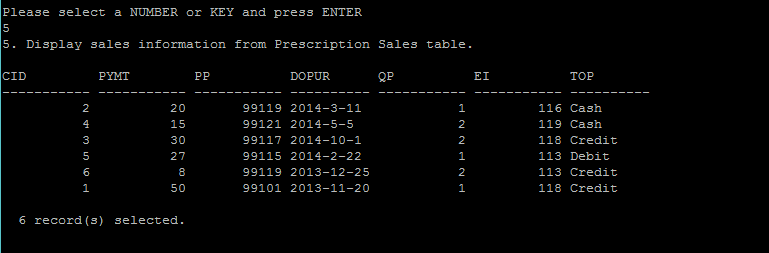
1. Display employee information from Employee Record table.



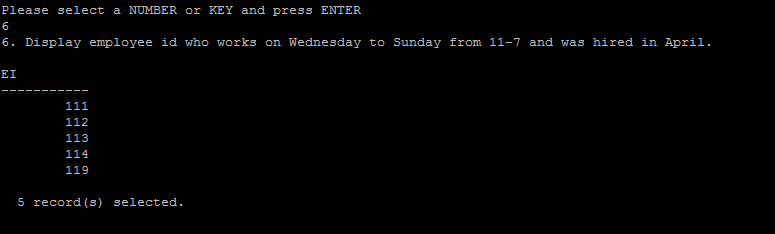
1. Display products information from Inventory table.



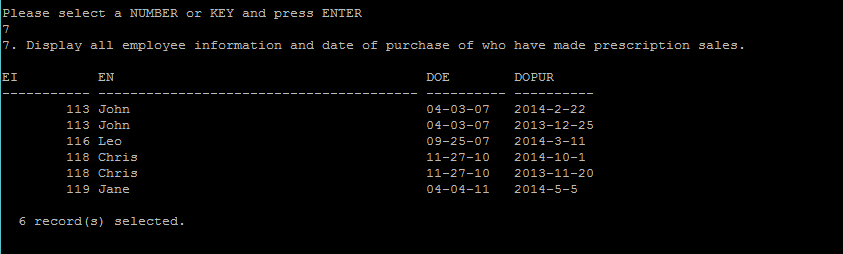
1. Display sales information from Prescription Sales table.



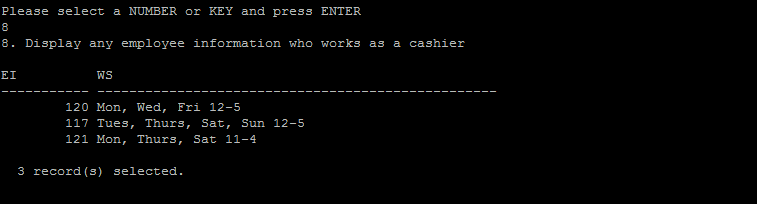
1. Display employee id who works on Wednesday to Sunday from 11-7 and was hired in April.



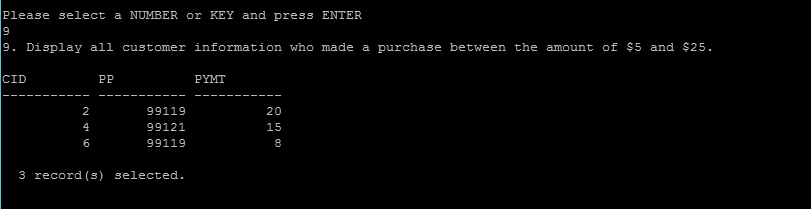
1. Display all employee information and date of purchase of who have made prescription sales.



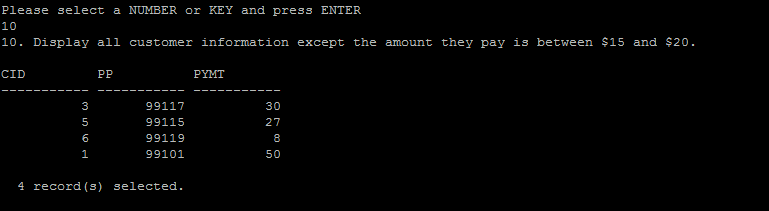
1. Display any employee information who works as a cashier



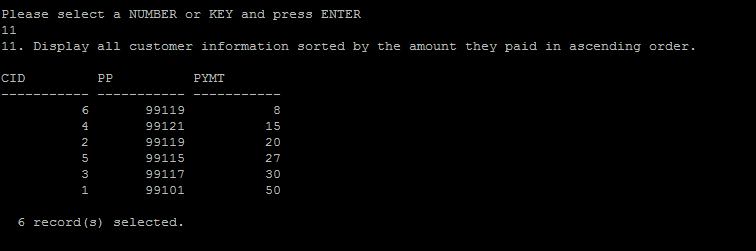
1. Display all customer information who made a purchase between the amount of \$5 and \$25.



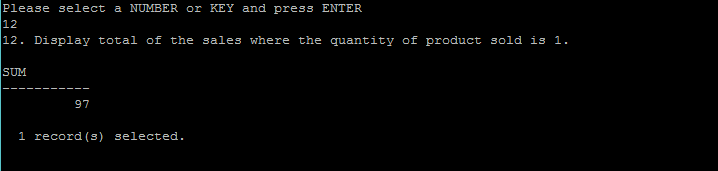
1. Display all customer information except the amount they pay is between \$15 and \$20.



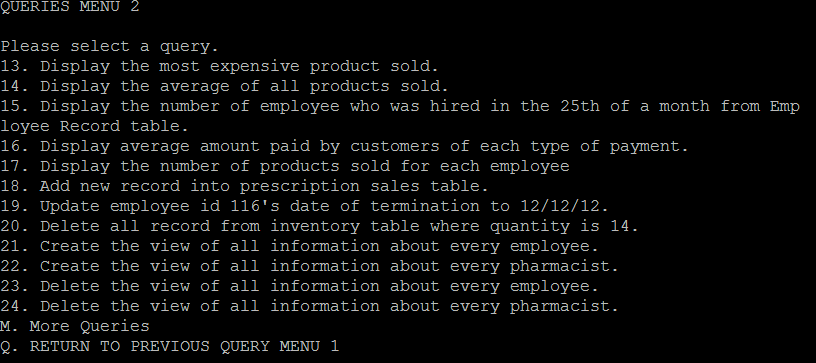
1. Display all customer information sorted by the amount they paid in ascending order.



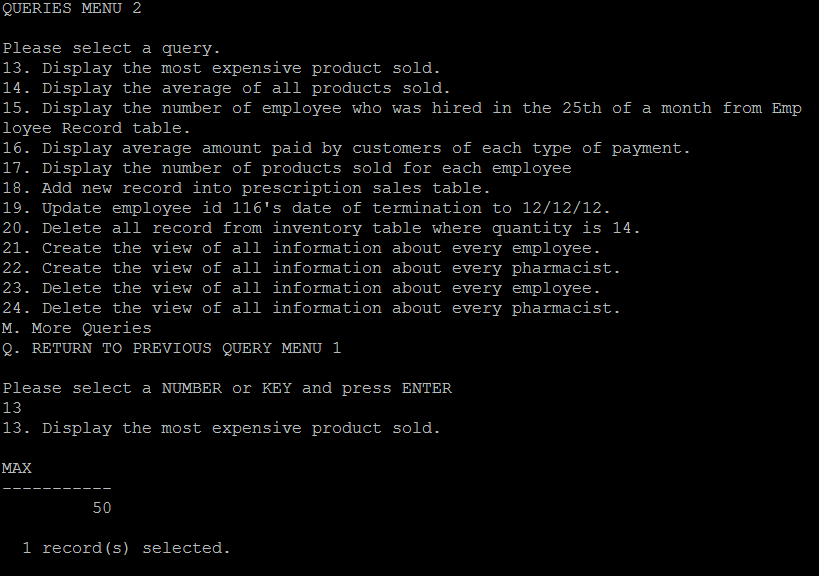
1. Display total of the sales where the quantity of product sold is 1.



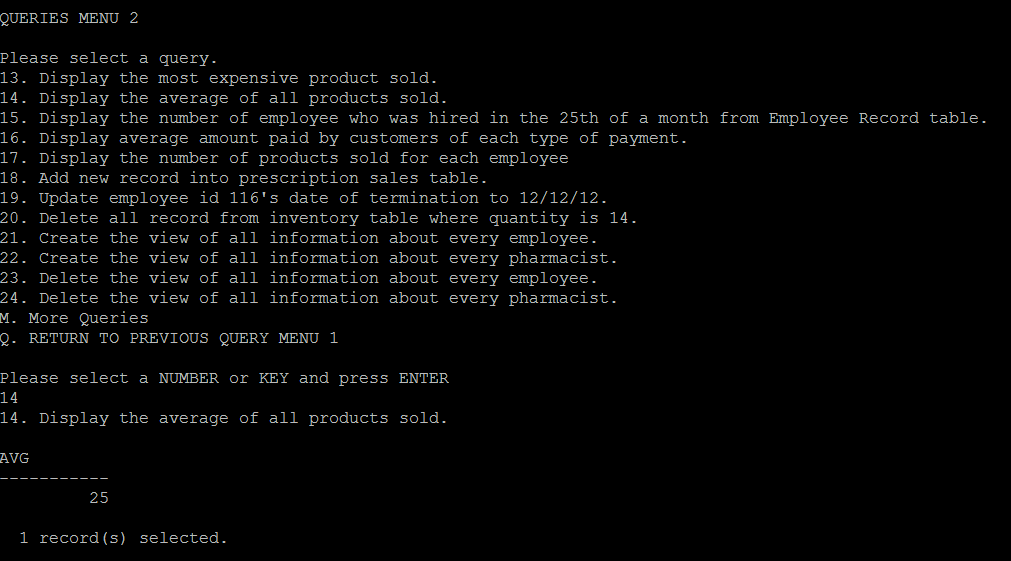
Query Menu 2



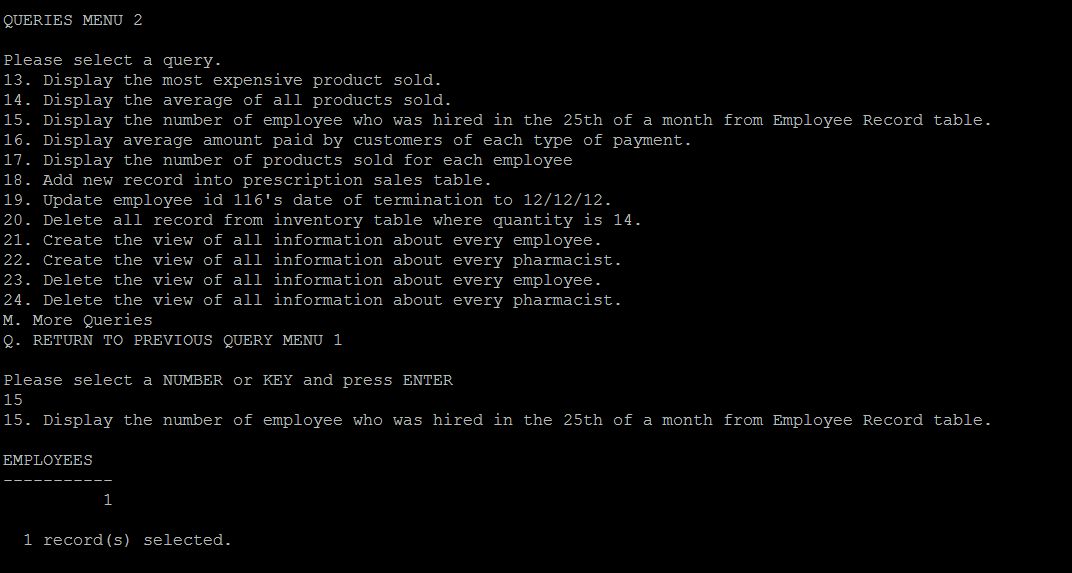
1. Display the most expensive product sold.



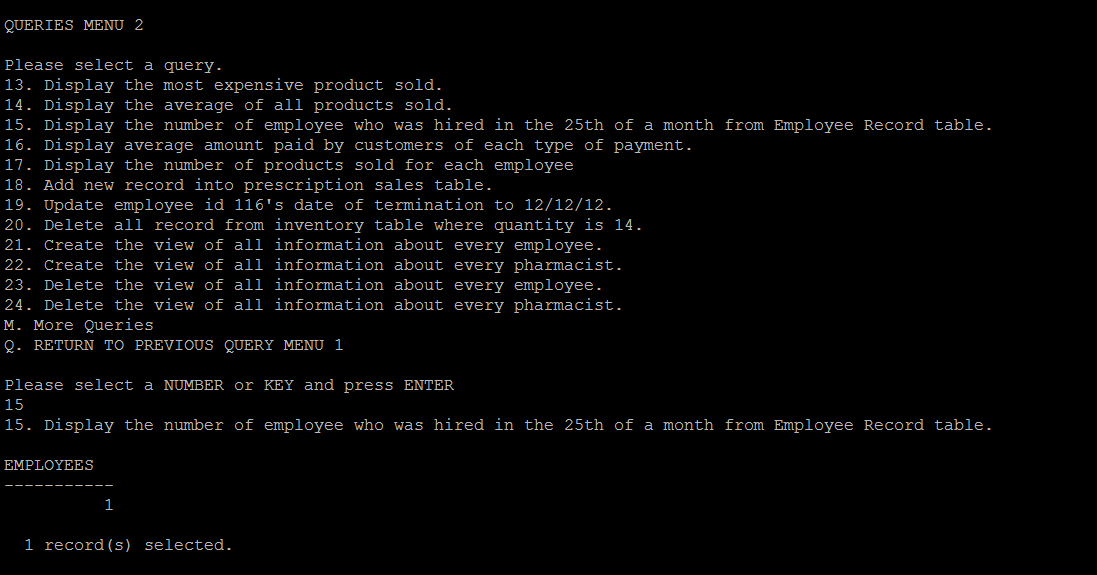
1. Display the average of all products sold.



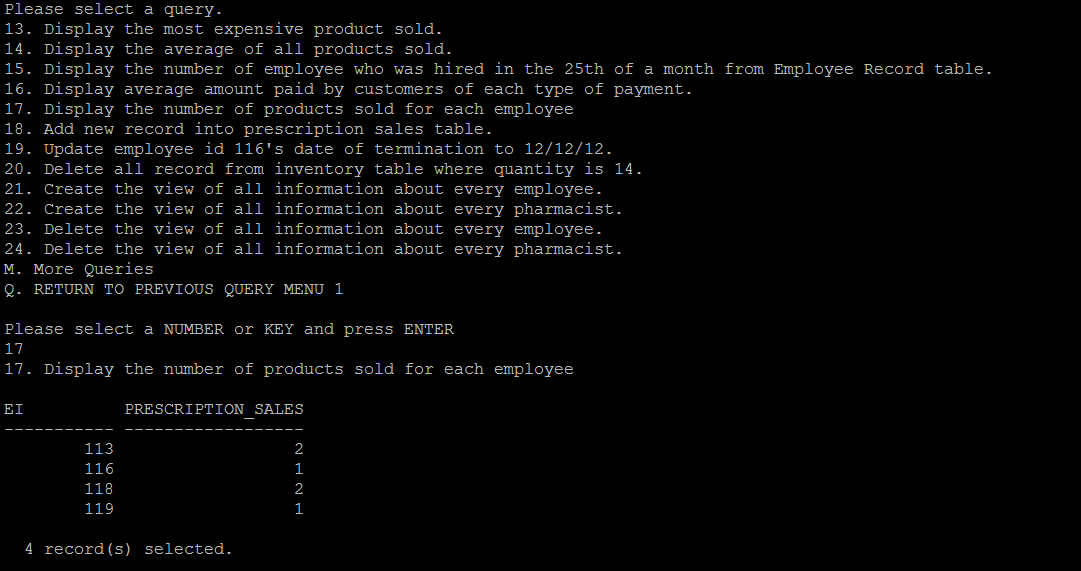
1. Display the number of employee who was hired in the 25th of a month from Employee Record table.



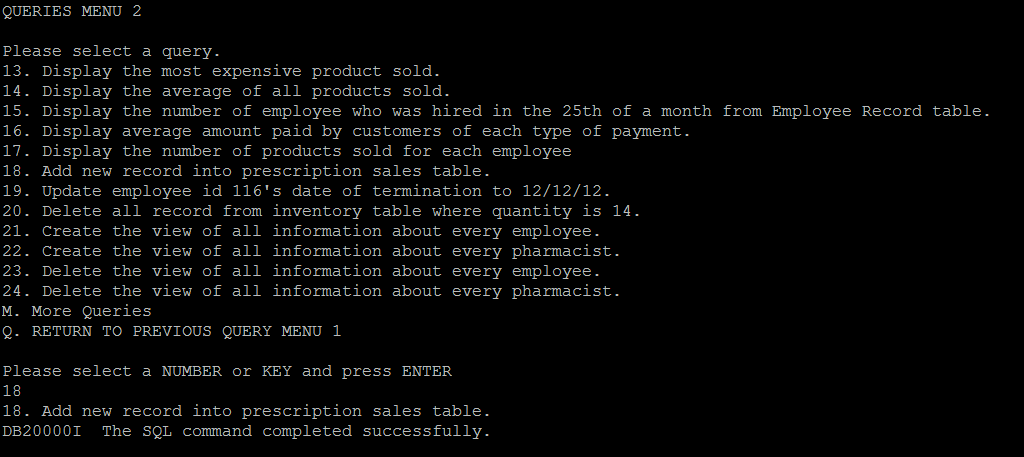
1. Display average amount paid by customers of each type of payment.



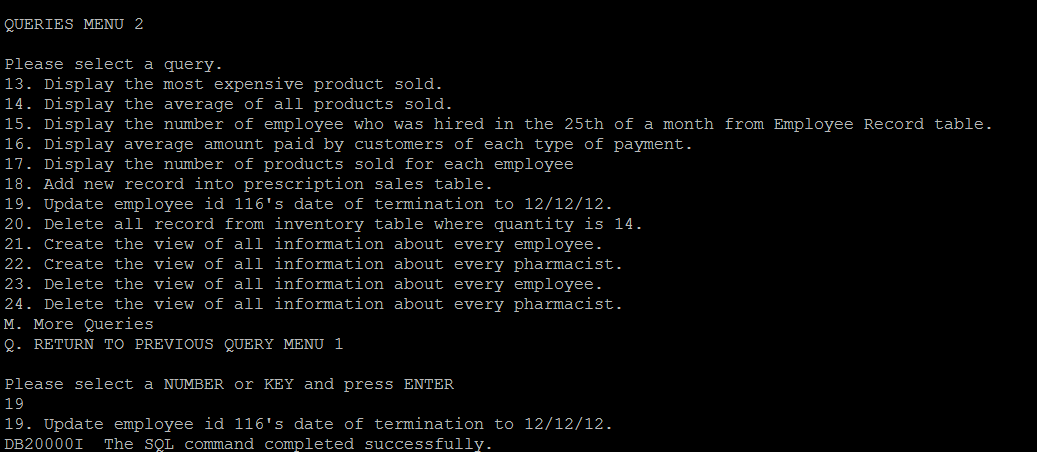
1. Display the number of products sold for each employee



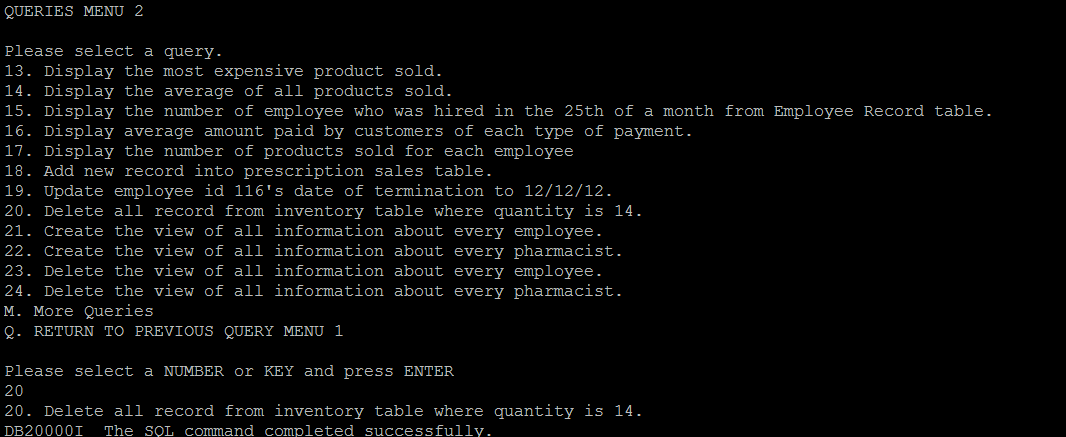
1. Add new record into prescription sales table.



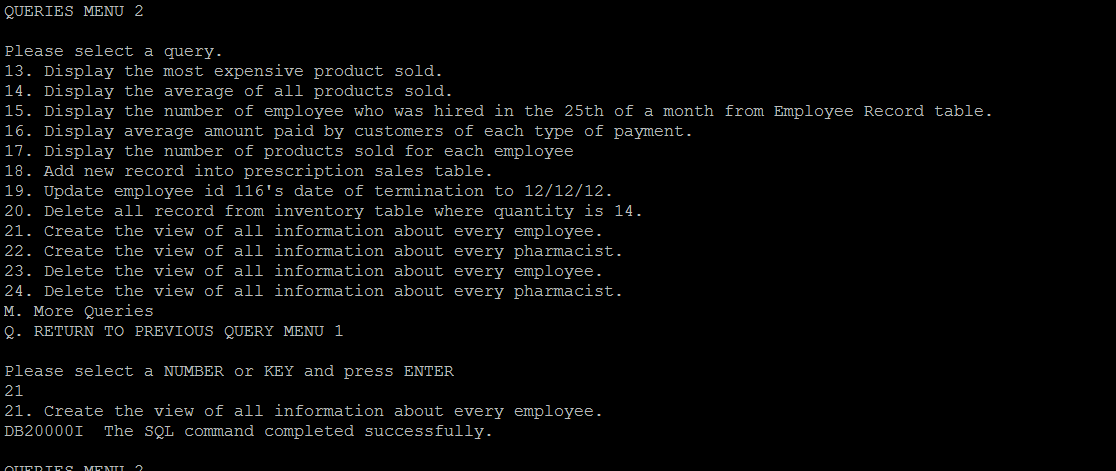
1. Update employee id 116's date of termination to 12/12/12.



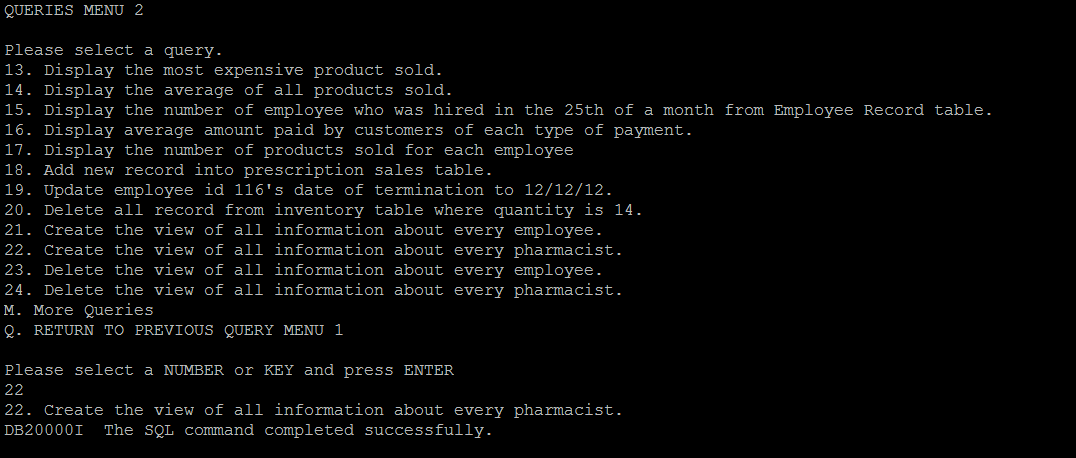
1. Delete all record from inventory table where quantity is 14.



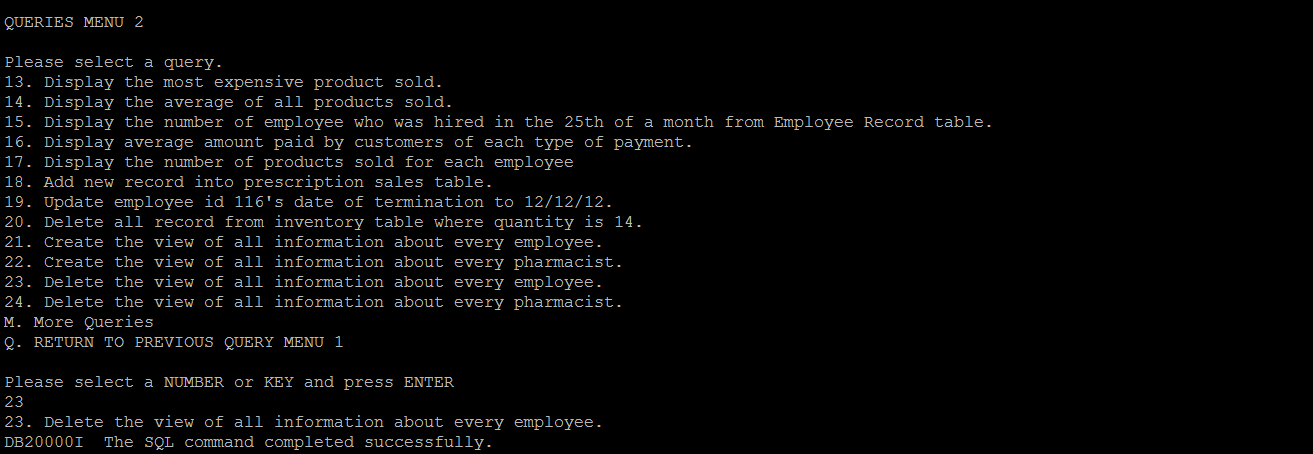
1. Create the view of all information about every employee.



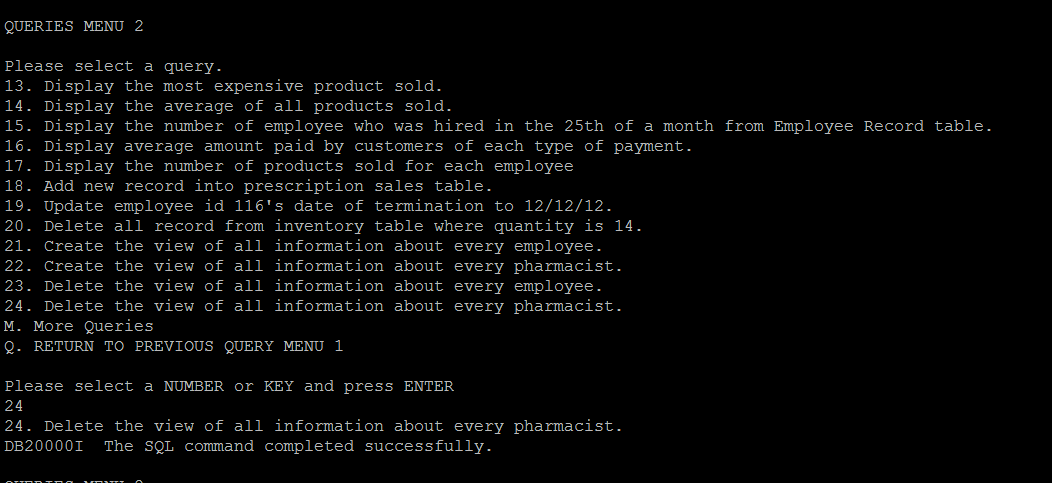
1. Create the view of all information about every pharmacist.



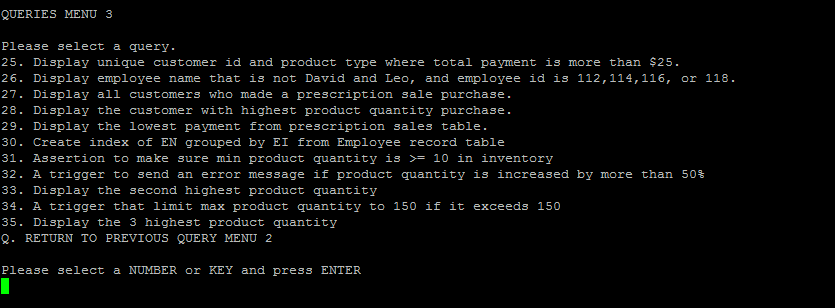
1. Delete the view of all information about every employee.



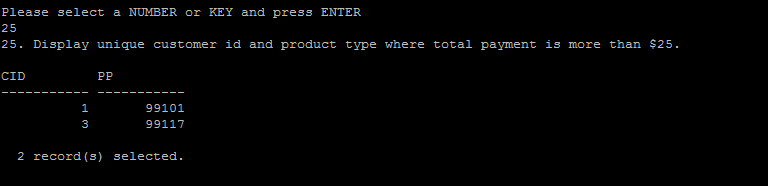
1. Delete the view of all information about every pharmacist.



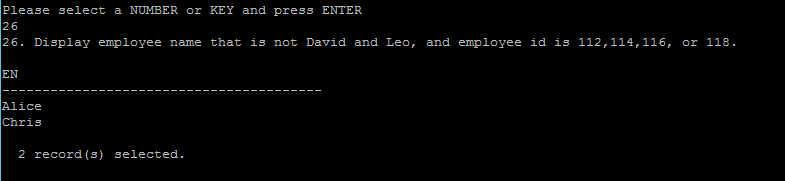
Query Menu 3



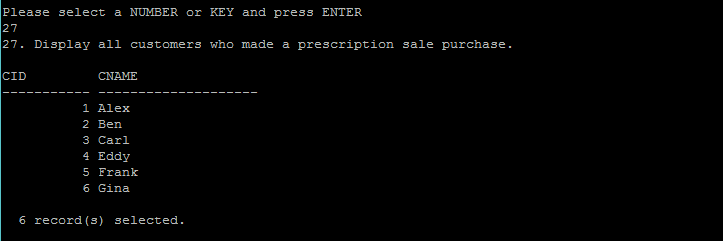
1. Display unique customer id and product type where total payment is more than \$25.



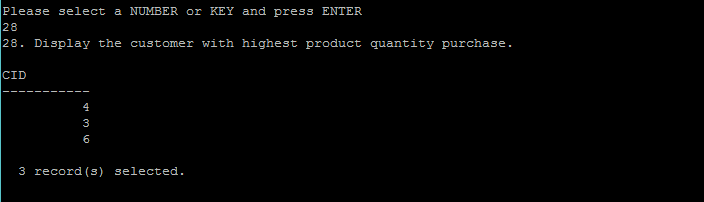
1. Display employee name that is not David and Leo, and employee id is 112,114,116, or 118.



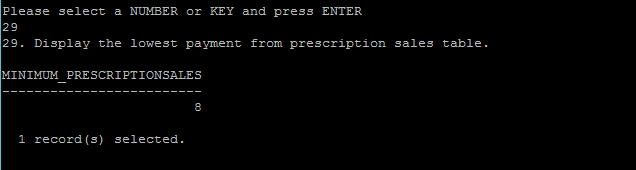
1. Display all customers who made a prescription sale purchase.



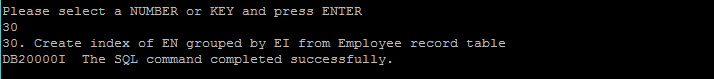
1. Display the customer with highest product quantity purchase.



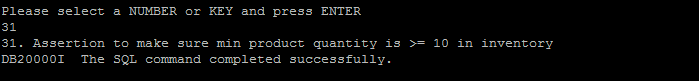
1. Display the lowest payment from prescription sales table.



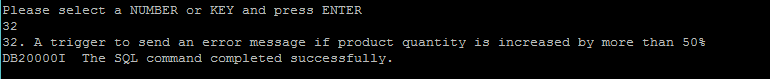
1. Create index of EN grouped by EI from Employee record table



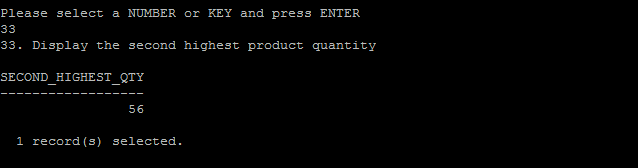
1. Assertion to make sure min product quantity is >= 10 in inventory



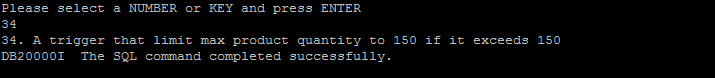
1. A trigger to send an error message if product quantity is increased by more than 50%"Display the second highest product quantity



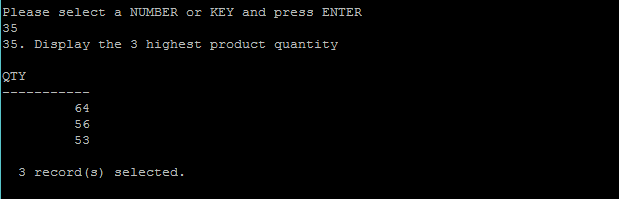
1. A trigger that limit max product quantity to 150 if it exceeds 150



1. Display the 3 highest product quantity

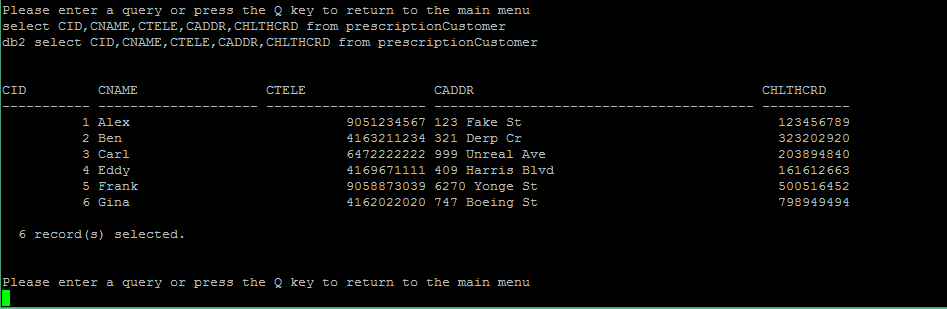


1. Display the 3 highest product quantity

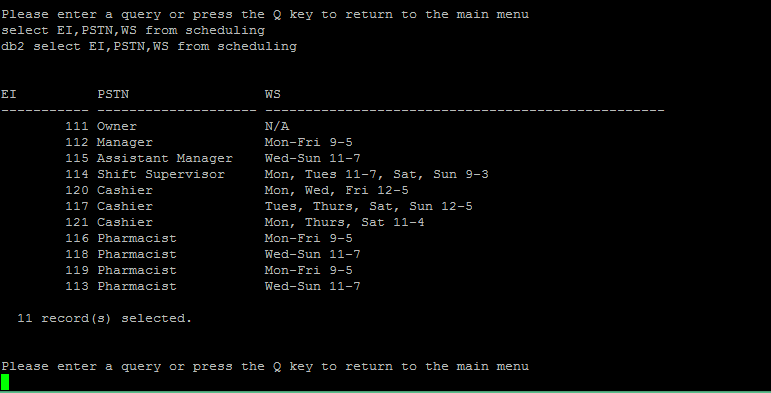


1. **Manual input queries**

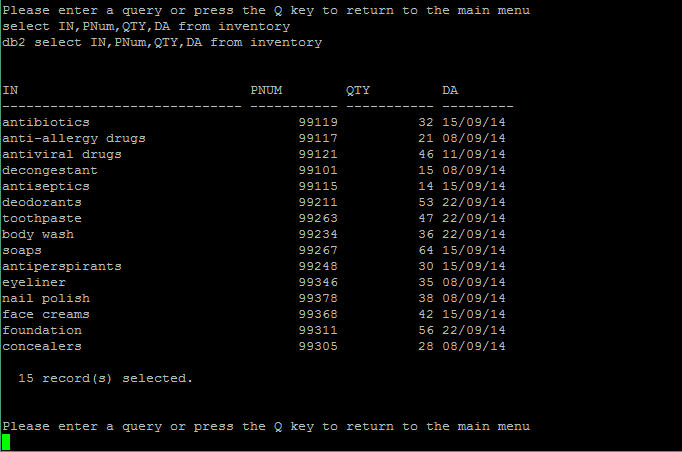
1. Display customer information from Prescription Customers table.



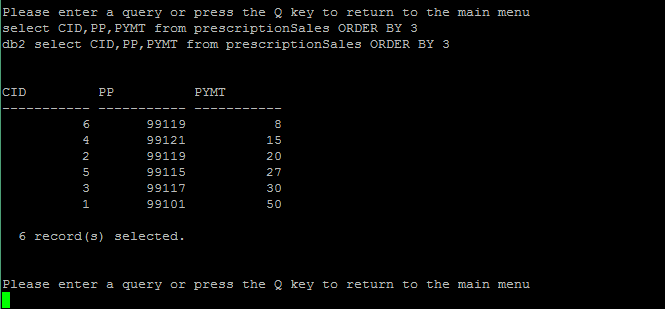
2. Display employee information from Scheduling table.



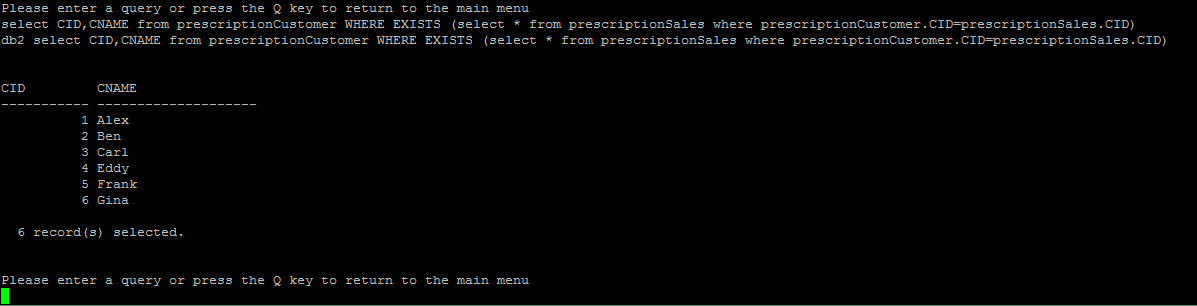
3. Display products information from Inventory table.



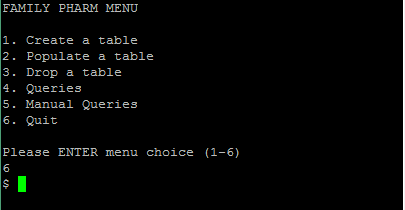
1. Display all customer information sorted by the amount they paid in ascending order



1. Display all customers who made a prescription sale purchase.



1. **Quit program**



**Chapter III: System**

**Source Code**

?#!/bin/sh

. db2init

db2 connect to db2data

TITLE=" FAMILY PHARM MENU "

CREATE\_TABLES\_TITLE=" CREATE TABLES MENU "

POPULATE\_TABLES\_TITLE=" POPULATE TABLES MENU "

DROP\_TABLES\_TITLE=" DROP TABLES MENU "

QUERIES\_TITLE1=" QUERIES MENU 1 "

QUERIES\_TITLE2=" QUERIES MENU 2 "

QUERIES\_TITLE3=" QUERIES MENU 3 "

ENTER\_QUERIES\_TITLE=" QUERIES MENU "

QUERY1="1. Display customer information from Prescription Customers table."

QUERY2="2. Display employee information from Scheduling table."

QUERY3="3. Display employee information from Employee Record table."

QUERY4="4. Display products information from Inventory table."

QUERY5="5. Display sales information from Prescription Sales table."

QUERY6="6. Display employee id who works on Wednesday to Sunday from 11-7 and was hired in April."

QUERY7="7. Display all employee information and date of purchase of who have made prescription sales."

QUERY8="8. Display any employee information who works as a cashier"

QUERY9="9. Display all customer information who made a purchase between the amount of \$5 and \$25."

QUERY10="10. Display all customer information except the amount they pay is between \$15 and \$20."

QUERY11="11. Display all customer information sorted by the amount they paid in ascending order."

QUERY12="12. Display total of the sales where the quantity of product sold is 1."

QUERY13="13. Display the most expensive product sold."

QUERY14="14. Display the average of all products sold."

QUERY15="15. Display the number of employee who was hired in 25th of a month from Employee Record table."

QUERY16="16. Display average amount paid by customers of each type of payment."

QUERY17="17. Display the number of products sold for each employee"

QUERY18="18. Add new record into prescription sales table."

QUERY19="19. Update employee id 116's date of termination to 12/12/12."

QUERY20="20. Delete all record from inventory table where quantity is 14."

QUERY21="21. Create the view of all information about every employee."

QUERY22="22. Create the view of all information about every pharmacist."

QUERY23="23. Delete the view of all information about every employee."

QUERY24="24. Delete the view of all information about every pharmacist."

QUERY25="25. Display unique customer id and product type where total payment is more than \$25."

QUERY26="26. Display employee name that is not David and Leo, and employee id is 112,114,116, or 118."

QUERY27="27. Display all customers who made a prescription sale purchase."

QUERY28="28. Display the customer with highest product quantity purchase."

QUERY29="29. Display the lowest payment from prescription sales table."

QUERY30="30. Create index of EN grouped by EI from Employee record table"

QUERY31="31. Assertion to make sure min product quantity is >= 10 in inventory"

QUERY32="32. A trigger to send an error message if product quantity is increased by more than 50%"

QUERY33="33. Display the second highest product quantity"

QUERY34="34. A trigger that limit max product quantity to 150 if it exceeds 150"

QUERY35="35. Display the 3 highest product quantity"

MENU1="1. Create a table";

MENU2="2. Populate a table";

MENU3="3. Drop a table";

MENU4="4. Queries";

MENU5="5. Manual Queries";

MENU6="6. Quit";

displayMenu(){

clear

echo $TITLE

echo

echo $MENU1

echo $MENU2

echo $MENU3

echo $MENU4

echo $MENU5

echo $MENU6

echo

echo "Please ENTER menu choice (1-6)";

}

prescriptionCustomerCreate(){

db2 " CREATE TABLE prescriptionCustomer(

CID INTEGER NOT NULL,

CNAME CHAR(20) NOT NULL,

CTELE BIGINT NOT NULL,

CADDR CHAR(40) NOT NULL,

CHLTHCRD INTEGER NOT NULL,

PRIMARY KEY(CID)

)"

echo "A prescription customer table has been created."

}

prescriptionCustomerPopulate(){

while read var1 var2 var3 var4 var5

do

db2 "insert into prescriptionCustomer

values($var1,$var2,$var3,$var4,$var5)"

done < prescriptionCustomers.txt

echo "A prescription customer table has been populated."

}

prescriptionCustomerDrop(){

db2 "DROP TABLE prescriptionCustomer"

echo "A prescription customer table has been dropped."

}

prescriptionSalesCreate(){

db2 "CREATE TABLE prescriptionSales (

SID INTEGER NOT NULL,

CID INTEGER NOT NULL,

PYMT INTEGER NOT NULL,

PP INTEGER NOT NULL,

DOPUR CHAR(10) NOT NULL,

QP INTEGER NOT NULL,

EI INTEGER NOT NULL,

TOP CHAR(10) NOT NULL,

PRIMARY KEY(SID),

FOREIGN KEY(EI) references employee\_record

)"

echo "A prescription sales table has been created."

}

prescriptionSalesPopulate(){

while read var1 var2 var3 var4 var5 var6 var7 var8

do

db2 "insert into prescriptionSales

values($var1,$var2,$var3,$var4,$var5,$var6,$var7,$var8)"

done < prescriptionSales.txt

echo "A prescription sales table has been populated."

}

prescriptionSalesDrop(){

db2 "DROP TABLE prescriptionSales"

echo "A prescription sales table has been dropped."

}

employeeRecordCreate(){

db2 "CREATE TABLE employee\_record (

EI INTEGER NOT NULL,

EN CHAR(40) NOT NULL,

DOE CHAR(10) NOT NULL,

DOT CHAR(20) NOT NULL,

BAI INTEGER NOT NULL,

PRIMARY KEY(EI)

)"

echo "An employee records table has been created."

}

employeeRecordPopulate(){

while read var1 var2 var3 var4 var5

do

db2 "insert into employee\_record

values($var1,$var2,$var3,$var4,$var5)"

done < employeeRecord.txt

echo "An employee records table has been populated."

}

employeeRecordDrop(){

db2 "DROP TABLE employee\_record"

echo "A employee records table has been dropped."

}

schedulingCreate(){

db2 "CREATE TABLE scheduling(

EI INTEGER NOT NULL,

PSTN CHAR(20) NOT NULL,

WS CHAR(50) NOT NULL,

PRIMARY KEY(EI)

)"

echo "A scheduling table has been created."

}

schedulingPopulate(){

while read var1 var2 var3

do

db2 "insert into scheduling values($var1,$var2,$var3)"

done < scheduling.txt

echo "A scheduling table has been populated."

}

schedulingDrop(){

db2 "DROP TABLE scheduling"

echo "A scheduling table has been dropped."

}

inventoryCreate(){

db2 "CREATE TABLE inventory (

IN INTEGER NOT NULL,

PNum CHAR(20) NOT NULL,

QTY INTEGER NOT NULL,

DA CHAR(8) NOT NULL.

PRIMARY KEY(IN)

)"

echo "An inventory table has been created."

}

inventoryPopulate(){

while read var1 var2 var3 var4

do

db2 "insert into inventory

values($var1,$var2,$var3,$var4)"

done < inventory.txt

echo "An inventory table has been populated."

}

inventoryDrop(){

db2 "DROP TABLE inventory"

echo "An inventory table has been dropped."

}

transactionCreate(){

db2 "CREATE TABLE transaction (

TID INTEGER NOT NULL,

TN CHAR(30) NOT NULL,

OP CHAR(10) NOT NULL,

PRIMARY KEY(TID)

)"

echo "A transaction table has been created."

}

transactionDrop(){

db2 "DROP TABLE transaction"

echo "A transaction table has been dropped."

}

queriesMenu1(){

clear

while true

do

echo

echo $QUERIES\_TITLE1

echo

echo "Please select a query."

echo $QUERY1

echo $QUERY2

echo $QUERY3

echo $QUERY4

echo $QUERY5

echo $QUERY6

echo $QUERY7

echo $QUERY8

echo $QUERY9

echo $QUERY10

echo $QUERY11

echo $QUERY12

echo "M. More Queries"

echo "Q. RETURN TO MAIN MENU"

echo

echo "Please select a NUMBER or KEY and press ENTER"

read answer

case $answer in

1)echo $QUERY1

db2 "select CID,CNAME,CTELE,CADDR,CHLTHCRD from prescriptionCustomer";;

2)echo $QUERY2

db2 "select EI,PSTN,WS from scheduling";;

3)echo $QUERY3

db2 "select EI,EN,DOE,DOT,BAI from employee\_record";;

4)echo $QUERY4

db2 "select IN,PNum,QTY,DA from inventory";;

5)echo $QUERY5

db2 "select CID,PYMT,PP,DOPUR,QP,EI,TOP from prescriptionSales";;

6)echo $QUERY6

db2 "select EI from employee\_record where DOE LIKE '04%' union select EI from scheduling where WS = 'Wed-Sun%11-7' ";;

7)echo $QUERY7

db2 "select employee\_record.EI,employee\_record.EN,employee\_record.DOE,prescriptionSales.DOPUR from employee\_record,prescriptionSales where employee\_record.EI = prescriptionSales.EI";;

8)echo $QUERY8

db2 "select EI,WS from scheduling where PSTN = 'Cashier'";;

9)echo $QUERY9

db2 "select CID,PP,PYMT from prescriptionSales where PYMT BETWEEN 5 AND 25";;

10)echo $QUERY10

db2 "select CID,PP,PYMT from prescriptionSales where PYMT NOT BETWEEN 15 AND 20";;

11)echo $QUERY11

db2 "select CID,PP,PYMT from prescriptionSales ORDER BY 3";;

12)echo $QUERY12

db2 "select SUM(PYMT) AS sum from precsriptionSales where QP = 1";;

m|M)clear

queriesMenu2;;

q|Q)clear

break;;

\*)echo Invalid Choice;;

esac

done

}

queriesMenu2(){

clear

while true

do

echo

echo $QUERIES\_TITLE2

echo

echo "Please select a query."

echo $QUERY13

echo $QUERY14

echo $QUERY15

echo $QUERY16

echo $QUERY17

echo $QUERY18

echo $QUERY19

echo $QUERY20

echo $QUERY21

echo $QUERY22

echo $QUERY23

echo $QUERY24

echo "M. More Queries"

echo "Q. RETURN TO PREVIOUS QUERY MENU 1"

echo

echo "Please select a NUMBER or KEY and press ENTER"

read answer

case $answer in

13)echo $QUERY13

db2 "select MAX(PYMT) AS max from prescriptionSales";;

14)echo $QUERY14

db2 "select AVG(PYMT) AS avg from prescriptionSales";;

15)echo $QUERY15

db2 "select COUNT(\*) AS employees from employee\_record where DOE LIKE '%25%'";;

16)echo $QUERY16

db2 "select TOP, COUNT(\*) AS count, AVG(PYMT) as averagePYMT from prescriptionSales GROUP BY TOP HAVING count(\*) > 1 order by TOP";;

17)echo $QUERY17

db2 "select EI,count(prescriptionSales.EI) AS prescription\_sales from prescriptionSales,prescriptionCustomer where prescriptionSales.CID=prescriptionCustomer.CID group by prescriptionSales.EI";;

18)echo $QUERY18

db2 "insert into prescriptionSale values(7,001,40.00,99305,'2014-9-15',2,118,'Debit')";;

19)echo $QUERY19

db2 "update employee\_record set DOT='12/12/12' where EI=116";;

20)echo $QUERY20

db2 "delete from inventory where QTY = 14";;

21)echo $QUERY21

db2 "create view allEmployees as (select employee\_record.EI,employee\_record.EN,employee\_record.DOE,employee\_record.DOT,employee\_record.BAI, scheduling.PSTN, scheduling.WS from employee\_record, scheduling where employee\_record.EI = scheduling.EI))";;

22)echo $QUERY22

db2 "create view allPharmacists as (select employee\_record.EI,employee\_record.EN,employee\_record.DOE,employee\_record.DOT,employee\_record.BAI, scheduling.PSTN, scheduling.WS from employee\_record, scheduling where employee\_record.EI = scheduling.EI AND scheduling.PSTN = 'Pharmacist')";;

23)echo $QUERY23

db2 "drop view allEmployees";;

24)echo $QUERY24

db2 "drop view allPharmacists";;

m|M)clear

queriesMenu3;;

q|Q)clear

break;;

\*)echo Invalid Choice;;

esac

done

}

queriesMenu3(){

clear

while true

do

echo

echo $QUERIES\_TITLE3

echo

echo "Please select a query."

echo $QUERY25

echo $QUERY26

echo $QUERY27

echo $QUERY28

echo $QUERY29

echo $QUERY30

echo $QUERY31

echo $QUERY32

echo $QUERY33

echo $QUERY34

echo $QUERY35

echo "Q. RETURN TO PREVIOUS QUERY MENU 2"

echo

echo "Please select a NUMBER or KEY and press ENTER"

read answer

case $answer in

25)echo $QUERY25

db2 "select DISTINCT x.CID,x.PP from prescriptionSales x,prescriptionSales y where x.PYMT\*x.QP > y.PYMT\*y.QP+25";;

26)echo $QUERY26

db2 "select EN from employee\_record where EI IN (112,114,116,118) and EN not in ('David','Leo')";;

27)echo $QUERY27

db2 "select CID,CNAME from prescriptionCustomer WHERE EXISTS (select \* from prescriptionSales where prescriptionCustomer.CID=prescriptionSales.CID)";;

28)echo $QUERY28

db2 "select CID from prescriptionSales x where QP >= ALL (select QP from prescriptionSales y)";;

29)echo $QUERY29

db2 "select MIN(PYMT) as minimum\_prescriptionSales from prescriptionSales";;

30)echo $QUERY30

db2 "create INDEX en\_by\_ei ON employee\_record (EI,EN)";;

31)echo $QUERY31

db2 "alter table inventory add constraint min\_productQuantity check (QTY >= 10)";;

32)echo $QUERY32

db2 "create trigger qty\_increase after update of QTY on inventory referencing new as new\_inventory old as old\_inventory for each row mode db2sql when (new\_inventory.QTY > old\_inventory.QTY\*1.5) begin atomic signal sqlstate '75001' ('Invalid increase to QTY - Exceed 50%'); end";;

33)echo $QUERY33

db2 "select distinct QTY as SECOND\_Highest\_QTY from inventory x where 2 = (select count(distinct QTY) from inventory y where x.QTY <= y.QTY)";;

34)echo $QUERY34

db2 "create trigger max\_QTY after update of QTY on inventory referencing new as n old as o for each row mode db2sql when (n.QTY > 150)begin atomic update inventory set QTY=150 where QTY>150; end";;

35)echo $QUERY35

db2 "select distinct QTY from inventory x where 3 >= (select count(distinct QTY) from inventory y where x.QTY <= y.QTY) order by x.QTY desc";;

q|Q)clear

break;;

\*)echo Invalid Choice;;

esac

done

}

manualQueriesMenu(){

clear

echo $ENTER\_QUERIES\_TITLE

while true

do

echo

echo "Please enter a query or press the Q key to return to the main menu"

read answer

case $answer in

q|Q) clear;break;;

\*) echo db2 "$answer";echo;db2 "$answer";;

esac

done

}

createMenu(){

clear

while true

do

echo

echo $CREATE\_TABLES\_TITLE

echo

echo "Please select a table to create."

echo "1. Employee Record Table"

echo "2. Prescription Sales Table"

echo "3. Prescription Customer Table"

echo "4. Scheduling Table"

echo "5. Inventory Table"

echo "6. Transaction Table"

echo "7. QUIT TO MAIN MENU"

echo

echo "Please ENTER menu choice (1-7)"

read input

case $input in

1)employeeRecordCreate;;

2)prescriptionSalesCreate;;

3)prescriptionCustomerCreate;;

4)schedulingCreate;;

5)inventoryCreate;;

6)transactionCreate;;

7)confirmMenu;;

\*)echo Invalid Input;;

esac

done

}

populateMenu(){

clear

while true

do

echo

echo $POPULATE\_TABLES\_TITLE

echo

echo "Please select a table to populate."

echo "1. Employee Record Table"

echo "2. Prescription Sales Table"

echo "3. Prescription Customer Table"

echo "4. Scheduling Table"

echo "5. Inventory Table"

echo "6. QUIT TO MAIN MENU"

echo

echo "Please ENTER menu choice (1-6)"

read input

case $input in

1)employeeRecordPopulate;;

2)prescriptionSalesPopulate;;

3)prescriptionCustomerPopulate;;

4)schedulingPopulate;;

5)inventoryPopulate;;

6)confirmMenu;;

\*)echo Invalid Input;;

esac

done

}

dropMenu(){

clear

while true

do

echo

echo $DROP\_TABLES\_TITLE

echo

echo "Please select a table to drop."

echo "1. Employee Record Table"

echo "2. Prescription Sales Table"

echo "3. Prescription Customer Table"

echo "4. Scheduling Table"

echo "5. Inventory Table"

echo "6. Transaction Table"

echo "7. QUIT TO MAIN MENU"

echo

echo "Please ENTER menu choice (1-7)"

read input

case $input in

1)employeeRecordDrop;;

2)prescriptionSalesDrop;;

3)prescriptionCustomerDrop;;

4)schedulingDrop;;

5)inventoryDrop;;

6)transactionDrop;;

7)confirmMenu;;

\*)echo Invalid Input;;

esac

done

}

confirmMenu(){

echo

echo "Press any key to return to the main menu."

read answer

break

}

while true

do

displayMenu

read input

case $input in

1) createMenu;;

2) populateMenu;;

3) dropMenu;;

4) queriesMenu1;;

5) manualQueriesMenu;;

6) clear;break;;

\*) echo Invalid Input;;

esac

done

**Tables**

**employeeRecord.txt**

111 'Mary' '04-03-07' 'N/A' 112434072

112 'Alice' '04-03-07' 'N/A' 738642372

113 'John' '04-03-07' 'N/A' 824253543

114 'David' '04-03-07' 'N/A' 786387378

115 'Bob' '08-12-07' 'N/A' 737322433

116 'Leo' '09-25-07' 'N/A' 234535437

117 'Kevin' '08-18-10' 'N/A' 437837343

118 'Chris' '11-27-10' 'N/A' 398378378

119 'Jane' '04-04-11' 'N/A' 378345312

120 'Mara' '06-11-13' 'N/A' 355878383

121 'Cindy' '03-01-14' 'N/A' 890745098

**inventory.txt**

'antibiotics' 99119 32 '15/09/14'

'anti-allergy\ drugs' 99117 21 '08/09/14'

'antiviral\ drugs' 99121 46 '11/09/14'

'decongestant' 99101 15 '08/09/14'

'antiseptics' 99115 14 '15/09/14'

'deodorants' 99211 53 '22/09/14'

'toothpaste' 99263 47 '22/09/14'

'body\ wash' 99234 36 '22/09/14'

'soaps' 99267 64 '15/09/14'

'antiperspirants' 99248 30 '15/09/14'

'eyeliner' 99346 35 '08/09/14'

'nail\ polish' 99378 38 '08/09/14'

'face\ creams' 99368 42 '15/09/14'

'foundation' 99311 56 '22/09/14'

'concealers' 99305 28 '08/09/14'

**prescriptionCustomer.txt**

001 'Alex' 9051234567 '123\ Fake\ St' 123456789  
002 'Ben' 4163211234 '321\ Derp\ Cr' 323202920  
003 'Carl' 6472222222 '999\ Unreal\ Ave' 203894840  
004 'Eddy' 4169671111 '409\ Harris\ Blvd' 161612663  
005 'Frank' 9058873039 '6270\ Yonge\ St' 500516452  
006 'Gina' 4162022020 '747\ Boeing\ St' 798949494

**prescriptionSales.txt**

1 002 20 99119 '2014-3-11' 1 116 'Cash'

2 004 15 99121 '2014-5-5' 2 119 'Cash'

3 003 30 99117 '2014-10-1' 2 118 'Credit'

4 005 27 99115 '2014-2-22' 1 113 'Debit'

5 006 8 99119 '2013-12-25' 2 113 'Credit'

6 001 50 99101 '2013-11-20' 1 118 'Credit'

**scheduling.txt**

111 'Owner' 'N/A'

112 'Manager' 'Mon-Fri\ 9-5'

115 'Assistant\ Manager' 'Wed-Sun\ 11-7'

114 'Shift\ Supervisor' 'Mon,\ Tues\ 11-7,\ Sat,\ Sun\ 9-3'

120 'Cashier' 'Mon,\ Wed,\ Fri\ 12-5'

117 'Cashier' 'Tues,\ Thurs,\ Sat,\ Sun\ 12-5'

121 'Cashier' 'Mon,\ Thurs,\ Sat\ 11-4'

116 'Pharmacist' 'Mon-Fri\ 9-5'

118 'Pharmacist' 'Wed-Sun\ 11-7'

119 'Pharmacist' 'Mon-Fri\ 9-5'

113 'Pharmacist' 'Wed-Sun\ 11-7'