UNIVERSITY OF COMPUTER STUDIES,

YANGON

Database Project

Bookstore Management System.

(Faculty of Information Science.)

Team Participants-

1.Mg Aung Htet Kyaw (4CS-97)

2.Ma Hnin May Thu Aung (4CS-88)

3.Ma Htet Htet Htoo Ko (4CS-79)

4.Mg Minn Khant Oo (4CS-67)

5.Ma Nwe Ni Oo Wai (4CS-58)

6.Ma Nyein Nyein Thu (4CS-78)

7.Mg Thet Paing Htwe (4CS-57)

8.Ma Yadanar Htet (4CS-110)

-Contents-

Chapter 1:Introduction

1.1:Project Description

1.2:Project Objectives

Chapter 2:Database Management System

2.1:ER Diagram

2.2:Data Dictionary

2.3:Tables

2.4:SQL Statements

2.5:Integirty and Security Constraints

Chapter 3:Conclusion.

1

**Chapter(1)**

**Introduction**

* 1. **Project Description**

The project “Online Book Store System” is a user interactive and delivery based system and it acts as a central database containing various books in stock along with their title, author and cost. The purpose is to change the existing manual system with the computerized system by the help of computer software, fulfilling their requirements, so that their valuable data can be stored for a longer period of time. The required software and hardware are easily available and easy to work with.

It provides the user with a catalog of different books available for purchase in the store. A user visiting the website can see a wide range of books arranged in respective categories. The user may select desired book and view its price. The user may even search for specific books on the website.

2

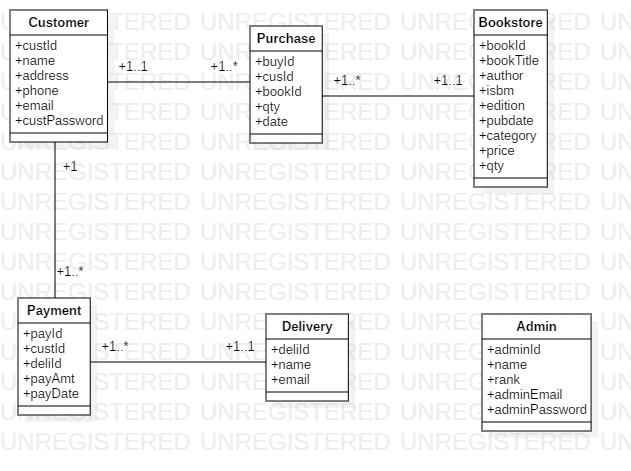
The project describes how to manage for good performance and better services for the clients.

* 1. **Project Objectives**
* To provide error free, secure, reliable and fast management system
* To make products easily available to the customers
* To provide customers to get their book delivered instead of actually going and buying the book
* To save both time and travelling cost of customers
* To reduce salesperson in order to reduce his or her salary
* To reduce cost for space to store book

**Chapter(2)**

**Database Diagram and Schema**

**2.1 E-R Diagram**



**3**

**2.2. Data Dictionary**

**Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraint** |
| adminId | Varchar | 10 | Primary Key |
| Name | Varchar | 30 |  |
| Rank | Varchar | 20 |  |
| adminEmail | Varchar | 30 |  |
| adminPassword | Varchar | 15 |  |

**Customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraint** |
| custId | Varchar | 10 | Primary Key |
| Name | Varchar | 30 |  |
| address | Varchar | 50 |  |
| Phone | Varchar | 15 |  |
| Email | Varchar | 30 |  |
| custPassword | Varchar | 15 |  |

**Bookstore**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraint** |
| bookId | Varchar | 10 | Primary Key |
| bookTitle | Varchar | 80 |  |
| author | Varchar | 30 |  |
| Isbn | Varchar | 30 |  |
| edition | Varchar | 20 |  |
| pubdate | Varchar | 30 |  |
| category | Varchar | 40 |  |
| Price | Int | 10 |  |
| Qty | Int | 10 |  |

**Purchase**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraint** |
| PId | Varchar | 10 | Primary Key |
| custId | Varchar | 10 | Foreign Key |
| bookId | Varchar | 10 | Foreign Key |
| Qty | Varchar | 10 |  |
| Date | Varchar | 30 |  |

**4**

**Delivery**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraint** |
| deliId | Varchar | 10 | Primary Key |
| Name | Varchar | 30 |  |
| Email | Varchar | 30 |  |

**Payment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraint** |
| payId | Varchar | 10 | Primary Key |
| custId | Varchar | 10 | Foreign Key |
| deliId | Varchar | 10 | Foreign Key |
| payAmt | Int | 10 |  |
| payDate | Varchar | 30 |  |

**5**

**2.3 Tables**

mysql>create table Admin(adminId varchar(10) primary key, name varchar(30), rank varchar(20), adminEmail varchar(30), adminPassword varchar(15));

mysql>insert into Admin values('A001','James','Adminstrator','james12@gmail.com','13572@js'),('A002','Ella','Moderator','ella99@gmail.com','weri567');

mysql> select \* From Admin;

+---------+-------+--------------+-------------------+---------------+

| adminId | name | rank | adminEmail | adminPassword |

+---------+-------+--------------+-------------------+---------------+

| A001 | James | Adminstrator | james12@gmail.com | 13572@js |

| A002 | Ella | Moderator | ella99@gmail.com | weri567 |

+---------+-------+--------------+-------------------+---------------+

mysql>create table Customer(custId varchar(10) primary key, name varchar(30), address varchar(50), phone varchar(15), email varchar(30), custPassword varchar(15));

mysql>insert into Customer values('C001','Hnin Cho','Mandalay','09425271825','hnincho2@gmail.com','hnin234%'),('C002','Aung Moe','Yankin','09779723456','aungm@gmail','fghj34'),('C003','Yan Pyae','Bago','09654856743','ypae56@gmail.com','yp$#5'),('C004','Aye Mya','Thuwanna','09796723479','ayemya11@gmail.com','ayam$\*'),('C005','Yu Wai','Mawlamyine','09425213579','yuwai13@gmail.com','yuw1418'),('C006','Myat Khine','Dawei','09778234567','mkine34@gmail.com','mke$%^'),('C007','Phue Nwe','Innwa','0978794927','phuenwe@gmail.com','tyu67'),('C008','Tun Kyaw','Hledan','09456712389','tunk34@gmail.com','45ert$');

6

mysql> select \* From Customer;

+--------+------------+------------+-------------+--------------------+-----------+

| custId | name | address | phone | email | custPword |

+--------+------------+------------+-------------+--------------------+-----------+

| C001 | Hnin Cho | Mandalay | 09425271825 | hnincho2@gmail.com | hnin2 |

| C002 | Aung Moe | Yankin | 09779723456 | aungm@gmail | fghj |

| C003 | Yan Pyae | Bago | 09654856743 | ypae56@gmail.com | yp$#5 |

| C004 | Aye Mya | Thuwanna | 09796723479 | ayemya11@gmail.com | ayam$ |

| C005 | Yu Wai | Mawlamyine | 09425213579 | yuwai13@gmail.com | yuw14 |

| C006 | Myat Khine | Dawei | 09778234567 | mkine34@gmail.com | mke$% |

| C007 | Phue Nwe | Innwa | 0978794927 | phuenwe@gmail.com | tyu67 |

| C008 | Tun Kyaw | Hledan | 09456712389 | tunk34@gmail.com | 45ert |

+--------+------------+------------+-------------+--------------------+-----------+

mysql>create table BookStore(bookId varchar(10) primary key, bookTitle varchar(80), author varchar(30), isbn varchar(30), edition varchar(20), pubdate varchar(30), category varchar(40), price int(10), qty int(10));

mysql>insert into Bookstore values('S001','Enterprise Resource Planning','Ellen Monk Bret Wagner','978-1-111-82039-8','fourth editin','2010-12-5','IT Fiction','13000','170');

mysql>insert into Bookstore values('S002','Traction Startup Guide','Gabriel Weinberg','978-0-976-33969-

**7**

1','seventh edition','2013-01-20','IT Fiction','20000','100');

mysql>insert into Bookstore values('S003','A Man like Him','Journal Kyaw Ma Ma Lay','978-1-501-71935-6','third edition','2017-04-20','Novel','5500','150');

mysql>insert into Bookstore values('S004','Mg KoKo and Mya Nandar','Kyi Aye','978-0-523-43589-8','second edition','2015-05-7','Novel','2500','100');

mysql>insert into Bookstore values('S005','A Kiss Before Dying','Ira Levin','978-0-671-20179-1','first edition','1953-06-18','Horror Fiction','7500','100');

mysql>insert into Bookstore values('S006','Carrie','Stephen King','978-0-385-08695-0','nineth edition','2014-11-14','Horror Fiction','6000','200');

mysql>insert into Bookstore values('S007','Mulan','Xu Ming Lee','978-0-414-07-2','first edition','1992-7-16','Comic Fiction','7000','300');

mysql>insert into Bookstore values('S008','Mickey Mouse:The Greatest Adventures','Walt Disney','978-1-

**8**

68396-122-2','lastest edition','2018-11-21','Comic Fiction','12000','200');

mysql>insert into Bookstore values('S009',' Clean Code','Robert Martins','978-1-62536-112-2','second edition','2008-4-13','IT Fiction ','7000','0');

mysql> select \* From Bookstore;

+--------+--------------------------------------+------------------------+-------------------

| bookId | bookTitle | author | isbn | edition | pubdate | category | price | qty |

+--------+--------------------------------------+------------------------+-------------------+-----------------+------------+----------------+-------+------+

| S001 | Enterprise Resource Planning | Ellen Monk Bret Wagner | 978-1-111-39-8 | fourth editin | 2010-12-5 | IT Fiction | 13000 | 170 |

**9**

| S002 | Traction Startup Guide | Gabriel Weinberg | 978-0-976-69-1 | seventh edition | 2013-01-20 | IT Fiction | 20000 | 100 |

| S003 | A Man like Him | Journal Kyaw Ma Ma Lay | 978-1-501-35-6 | third edition | 2017-04-20 | Novel | 5500 | 150 |

| S004 | Mg KoKo and Mya Nandar | Kyi Aye | 978-0-523-89-8 | second edition | 2015-05-7 | Novel | 2500 | 100 |

| S005 | A Kiss Before Dying | Ira Levin | 978-0-671-79-1 | first edition | 1953-06-18 | Horror Fiction | 7500 | 100 |

| S006 | Carrie | Stephen King | 978-0-385-95-0 | nineth edition | 2014-11-14 | Horror Fiction | 6000 | 200 |

| S007 | Mulan | Xu Ming Lee | 978-0-414-07-2 | first edition | 1992-7-16 | Comic Fiction | 7000 | 300 |

| S008 | Mickey Mouse:The Greatest Adventures | Walt Disney | 978-1-68396-122-2 | lastest edition | 2018-11-21 | Comic Fiction | 12000 | 200 |

**10**

| S009 | Clean Code | Robert Martins | 978-1-62536-112-2 | second edition | 2008-4-13 | IT Fiction | 7000 | 0 |

+--------+--------------------------------------+------------------------+-------------------+-----------------+------------+----------------+-------+------+

mysql>create table Purchase(PId varchar(10) primary key, custId varchar(10), bookId varchar(10), foreign key(custId) references Customer(custId),foreign key(bookId) references Bookstore(bookId),qty int(10), date varchar(30));

mysql>insert into Purchase values('P001','C002','S001','2','2019-02-15');

mysql>insert into Purchase values('P002','C003','S003','15','2019-05-23');

mysql>insert into Purchase values('P003','C003','S006','20','2019-06-18');

mysql>insert into Purchase values('P004','C007','S005','12','2019-06-22');

**11**

mysql>insert into Purchase values('P005','C001','S006','5','2019-09-5');

mysql> select \* From Buy;

+-------+--------+--------+------+------------+

| PId | custId | bookId | qty | date |

+-------+--------+--------+------+------------+

| P001 | C002 | S001 | 2 | 2019-02-15 |

| P002 | C003 | S003 | 15 | 2019-05-23 |

| P003 | C003 | S006 | 20 | 2019-06-18 |

| P004 | C007 | S005 | 12 | 2019-06-22 |

| P005 | C001 | S006 | 5 | 2019-09-5 |

+-------+--------+--------+------+------------+

**12**

mysql>create table Delivery(deliId varchar(10) primary key,name varchar(30), email varchar(30));

mysql>insert into Delivery values('D001','Astrid','astrid1@gmail.com'),('D002','John','john23@gmail.com'),('D003','Ashely','ashely5@gmail.com');

mysql> select \* From Delivery;

+--------+--------+-------------------+

| deliId | name | email |

+--------+--------+-------------------+

| D001 | Astrid | astrid1@gmail.com |

| D002 | John | john23@gmail.com |

| D003 | Ashely | ashely5@gmail.com |

+--------+--------+-------------------+

mysql>create table Payment(payId varchar(10) primary key, custId varchar(10), deliId varchar(10), payAmt int(10), payDate varchar(30), foreign key(custId)

**13**

references Customer(custId), foreign key(deliId) references Delivery(deliId));

mysql>insert into Payment values('P001','C002','D002','40000','2019-02-17');

mysql>insert into Payment values('P002','C003','D001','825000','2019-05-25');

mysql>insert into Payment values('P003','C003','D003','120000','2019-06-20');

mysql>insert into Payment values('P004','C007','D001','90000','2019-06-24');

mysql>insert into Payment values('P005','C001','D002','30000','2019-09-7');

mysql> select \* From Payment;

+-------+--------+--------+--------+------------+

| payId | custId | deliId | payAmt | payDate |

+-------+--------+--------+--------+------------+

**14**

| P001 | C002 | D002 | 40000 | 2019-02-17 |

| P002 | C003 | D001 | 825000 | 2019-05-25 |

| P004 | C007 | D001 | 90000 | 2019-06-24 |

| P005 | C001 | D002 | 30000 | 2019-09-7 |

+-------+--------+--------+--------+------------+

**15**

**Query Statements for Proposed System**

**2.4 SQL Statements**

1.Get total amount of payment in which the payment date.

select SUM(payAmt) from Payment where payDate='2019-02-17';

+-----------------+

| SUM(payAmt) |

+-----------------+

| 40000 |

+------------------+

1 row in set (0.00 sec)

2.Get full detail of customer who give maximum amount of payment.

select Customer.\* from Customer,Payment where Customer.custId=Payment.custId AND

**16**

Payment.payAmt=(select MAX(payAmt) from Payment);

+--------+----------+---------+-------------+------------------+--------------+

| custId | name | address | phone | email | custPassword |

+--------+----------+---------+-------------+------------------+--------------+

| C003 | Yan Pyae | Bago | 09654856743 | ypae56@gmail.com | yp$#5 |

+--------+----------+---------+-------------+------------------+--------------+

1 row in set (0.08 sec)

3.Get full detail of customer who do not purchase bookId 'S001'.

select Customer.\* from Customer,Purchase where Purchase.bookId<>'S001' AND Customer.custId=Purchase.custId;

+--------+----------+----------+-------------+--------------------+--------------+

| custId | name | address | phone | email | custPassword |

**17**

+--------+----------+----------+-------------+--------------------+--------------+

| C003 | Yan Pyae | Bago | 09654856743 | ypae56@gmail.com | yp$#5 |

| C003 | Yan Pyae | Bago | 09654856743 | ypae56@gmail.com | yp$#5 |

| C007 | Phue Nwe | Innwa | 0978794927 | phuenwe@gmail.com | tyu67 |

| C001 | HninCho | Mandalay | 09425271825 | hnincho2@gmail.com | hnin234% |

+--------+----------+----------+-------------+--------------------+--------------+

4 rows in set (0.00 sec)

4.Get book title for bookstore that either qty more than 100 or bought by customer 'C001' or both.

select Distinct BookStore.bookTitle from BookStore,Purchase where BookStore.qty>100 OR Purchase.custId='C001';

+--------------------------------------+

| bookTitle |

+--------------------------------------+

| Enterprise Resource Planning |

| Traction Startup Guide |

**18**

| A Man Like Him |

| Mg Ko Ko and Mya Nandar |

| A Kiss Before Dying |

| Carrie |

| Mulan |

| Mickey Mouse:The Greatest Adventures |

+--------------------------------------+

8 rows in set (0.00 sec)

5. Get customer name for customer who bought 'Novel' books.

select Distinct Customer.name from Customer,Purchase,BookStore where BookStore.category='Novel' AND Purchase.custId=Customer.custId;

+----------+

| name |

+----------+

| HninCho |

| Aung Moe |

**19**

| Yan Pyae |

| Phue Nwe |

+----------+

4 rows in set (0.00 sec)

| S003   | A Man Like Him                        | Journal Kyaw Ma Ma Lay | 978-1-501-71935-6 | third edition    | 2017-04-20 | Novel | 5500 | 150 |

| S004   | Mg Ko Ko and Mya Nandar         | Kyi Aye                          | 978-0-523-43589-8 | second edition| 2015-05-7  | Novel | 2500 | 100 |

| S005   | A Kiss Before Dying                   | lra Levin                        | 978-0-671-2017-1 | first edition     | 1953-06-18 | Horror Fiction| 7500 | 100 |

| S006   | Carrie                                       | Stephen King                 | 978-0-385-08695-0 | nineth edition | 2014-11-14 | Horror Fiction|  6000 | 200 |

| S007   | Mulan                                     | Xu Ming Lee                  | 978-0-414-07-2 | first edition     | 1992-7-16 | Comic Fiction | 7000 | 300 |

| S008   | Mickey Mouse:The Greatest Adventures | Walt Disney                  | 978-1-68396-122-2 | lastest edition | 2018-11-21 | Comic Fiction | 12000 |  200 |

|S009    |The Kite Runner              |Khaled Hosseini              |978-2-423-45-7        |second edition |2009-4-13    |Horror Fiction |7000 |120 |

**20**

+--------+------------------------------------------------+-----------------------------+-------------------------+-----------------+---------------+----------------+-------+------+

**2.4.1 Relational Algebra**

1.Get total amount of payment from payment.

SUMMARIZE Payment PER Payment{payAmt} ADD sum(payAmt) AS totalAmount.

2.Get full detail of customer who give maximum amount of payment.

((SUMMARIZE Payment PER Payment{payAmt,payId} ADD MAX(payAmt) AS MaxAmount){payId})JOIN Customer

3.Get full detail of customer who do not buy book ‘S001’.

(Customer{cusId} MINUS (Purchase WHERE bookId=bookId(‘S001’){cusId})JOIN Customer

**21**

4.Get book title for bookstore that either qty more than 100 or bought by customer ‘C001’ , or both.

((BookStore WHERE qty>100){BookId} UNION ((Customer WHERE cusId=cusId(‘C001’)){cusId}JOIN Purchase )JOIN BookStore){BookTitle}

5.Get customer name for customer who bought ‘Novel’ books.

((BookStore WHERE category=category(‘Novel’)){BookId} JOIN Purchase ){cusId} JOIN Customer ){name}

**22**

**2.4.2 Relational Calculus**

*Tuple Calculus*

1.Get total amount of payment from payment.

SUM(PaymentX.payAmt)AS TP

2.Get full detail of customer who give maximum amount of payment.

CustomerX WHERE EXISTS PaymentX

(CustomerX.custId=PaymentX.custId

AND MAX( PaymentX .payAmt) AS MaxAmt)

3.Get full detail of customer who do not buy book ‘S001’.

CustomerX WHERE NOT EXISTS BuyX(

CustomerX.cusId=BuyX.cusId

AND BuyX.bookId=bookId(‘S001’))

**23**

4.Get book title for bookstore that either qty more than 100 or bought by customer ‘C001’ , or both.

BookstoreX.bookTitle WHERE qty>100 OR EXISTS BuyX(

BookstoreX.bookId=BuyX.bookId

AND BuyX.cusId=cusId(‘C001’))

5.Get customer name for customer who bought ‘Novel’ books.

CustomerX.name WHERE EXITS PurchaseX EXISTS BookStoreX(

CustomerX.cusId=PurchaseX.cusId

AND PurchaseX.bookId=BookStoreX.BookId

AND BookStoreX.catgory=catgory(‘Novel’))

**24**

*Domain Calculus*

1.Get full detail of customer who buy book ‘S001’.

(cusIdX,nameX,addressX,phoneX.emailX,custPasswordX) WHERE

Customer(cusId:cusIdX,address:address,phone:phoneX,email:emailX,custPassword:custPasswordX)

AND Purchase(cusId:cusIdX.bookId:bookId(‘S001’))

2.Get book title for bookstore that either qty more than 100 and bought by customer ‘C001’ .

bookTitleX WHERE Exists qtyX(

BookStore(bookId:bookIdX)AND

Purchase(bookId:bookIdX, qty:qtyX, cusId:cusId(‘C001’),

AND qtyX>100) )

3.Get customer name for customer who bought ‘Novel’ books.

Customer.nameX WHERE EXITS CX EXISTS BX (

**25**

Customer(cusId : CX, name : nameX)AND

Purchase(cusId : CX, bookId : BX)AND

Bookstore(bookId : BX, category :category(‘Novel’))

*View*

Define a view over the pj database consisting of all customer (cusId and name only) who bought bookId ‘S001’.

VAR Project VIEW

(((Buy WHERE bookId=bookId(‘S001’)){cusId})JOIN Customer)

{cusId,name};

**26**

**2.4.3 User Account Creation**

mysql> create user 'James'@'localhost' identified by 'James';

Query OK, 0 rows affected (0.03 sec)

mysql> create user 'HninCho'@'localhost' identified by 'HninCho';

Query OK, 0 rows affected (0.00 sec)

1)User 'James'(Admin)full privileges(RETRIVE,INSERT,UPDATE,DELETE) over all relvar.

mysql> grant all on pjdatabase.\* to 'James'@'localhost';

Query OK, 0 rows affected (0.00 sec)

User 'HninCho'(Customer) privileges(RETRIVE) over all relvar.

mysql> grant select on pjdatabase.\* to 'HninCho'@'localhost';

Query OK, 0 rows affected (0.00 sec)

**27**

2)User 'James'(Admin)update for BookStore Price to '9000' for BookStore 'S006'.

mysql> update BookStore set Price=9000 where Bookid='S006';

Query OK, 1 row affected (0.03 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from BookStore;

**+--------+--------------------------------------+------------------------+------**

-------------+-----------------+------------+----------------+-------+------+

| bookId | bookTitle | author | isbn

| edition | pubdate | category | price | qty |

+--------+--------------------------------------+------------------------+------

**28**

-------------+-----------------+------------+----------------+-------+------+

| S001 | ahs | Ellen Monk Bret Wagner | 978-1

-111-82039-8 | fourth edition | 2010-12-5 | IT Fiction | 13000 | 170 |

| S002 | Traction Startup Guide | Gabriel Weinberg | 978-0

-976-33969-1 | seventh edition | 2013-01-20 | IT Fiction | 20000 | 100 |

| S003 | A Man like Him | Journal Kyaw Ma Ma Lay | 978-1

-501-71935-6 | third edition | 2017-04-20 | Novel | 5500 | 150 |

| S004 | Mg KoKo and Mya Nandar | Kyi Aye | 978-0

-523-43589-8 | second edition | 2015-05-7 | Novel | 300 | 100 |

| S005 | A Kiss Before Dying | Stephen King | 978-0

**29**

-671-79-1 | first edition | 1953-06-18 | Horror Fiction | 3500 | 100 |

| S006 | Carrie | Stephen King | 978-0

-385-08695-0 | nineth edition | 2014-11-14 | Horror Fiction | 9000 | 200 |

| S007 | Mulan | Xu Ming Lee | 978-0

-414-07-2 | first edition | 1992-7-16 | Comic Fiction | 7000 | 300 |

| S008 | Mickey Mouse:The Greatest Adventures | Walt Disney | 978-1

-68396-122-2 | lastest edition | 2018-11-21 | Comic Fiction | 12000 | 200 |

+--------+--------------------------------------+------------------------+------

-------------+-----------------+------------+----------------+-------+------+

**30**

3)User 'James'(Admin)insert for BookStore .

mysql> insert into BookStore values('S009','A Yate','Ma Sandar','978-0-424-08-3'

,'first edition','2002-12-3','Novel','6500','200');

Query OK, 1 row affected (0.03 sec)

mysql> select \* from BookStore;

+--------+--------------------------------------+------------------------+------

-------------+-----------------+------------+----------------+-------+------+

| bookId | bookTitle | author | isbn

| edition | pubdate | category | price | qty |

+--------+--------------------------------------+------------------------+------

-------------+-----------------+------------+----------------+-------+------+

**31**

| S001 | ahs | Ellen Monk Bret Wagner | 978-1

-111-82039-8 | fourth editin | 2010-12-5 | IT Fiction | 13000 | 170 |

| S002 | Traction Startup Guide | Gabriel Weinberg | 978-0

-976-33969-1 | seventh edition | 2013-01-20 | IT Fiction | 20000 | 100 |

| S003 | A Man like Him | Journal Kyaw Ma Ma Lay | 978-1

-501-71935-6 | third edition | 2017-04-20 | Novel | 5500 | 150 |

| S004 | Mg KoKo and Mya Nandar | Kyi Aye | 978-0

-523-43589-8 | second edition | 2015-05-7 | Novel | 300 | 100 |

| S005 | A Kiss Before Dying | Stephen King | 978-0

-671-79-1 | first edition | 1953-06-18 | Horror Fiction | 3500 | 100 |

**32**

| S006 | Carrie | Stephen King | 978-0

-385-08695-0 | nineth edition | 2014-11-14 | Horror Fiction | 9000 | 200 || S007 | Mulan | Xu Ming Lee | 978-0

-414-07-2 | first edition | 1992-7-16 | Comic Fiction | 7000 | 300 |

| S008 | Mickey Mouse:The Greatest Adventures | Walt Disney | 978-1

-68396-122-2 | lastest edition | 2018-11-21 | Comic Fiction | 12000 | 200 |

| S009 | A Yate | Ma Sandar | 978-0

-424-08-3 | first edition | 2002-12-3 | Novel | 6500 | 200 |

+--------+--------------------------------------+------------------------+------

-------------+-----------------+------------+----------------+-------+------+

**33**

4)User 'HninCho'(Customer)update for BookStore Price to '300' for BookStore 'S005'.

mysql> update BookStore set Price=300 where Bookid='S005';

ERROR 1142 (42000): UPDATE command denied to user 'HninCho'@'localhost' for table '

bookstore'

5)User 'HninCho'(Customer)insert for BookStore.

mysql> insert into Bookstore values('S018','MMMM','jdshg','192-12-12-12','first'

,'2000','200');

ERROR 1142 (42000): INSERT command denied to user 'HninCho'@'localhost' for table '

bookstore'

### **34**

**2.5.1 Integrity Constraint**

**Type Constraint**

**Type adminId POSSREP (CHAR)**

**CONSTRAINT THE\_adminId(adminId)<>Null;**

**Type custId POSSREP (CHAR)**

**CONSTRAINT THE\_custId(custId)<>Null;**

**Type bookId POSSREP (CHAR)**

**CONSTRAINT THE\_bookId(bookId)<>Null;**

**Type priceId POSSREP (CHAR)**

**CONSTRAINT THE\_priceId(priceId)<>Null;**

**Type pId POSSREP (CHAR)**

**CONSTRAINT THE\_pId(pId)<>Null;**

**35**

**Type deliId POSSREP (CHAR)**

**CONSTRAINT THE\_deliId(deliId)<>Null;**

**Type payId POSSREP (CHAR)**

**CONSTRAINT THE\_payId(payId)<>Null;**

**Attribute Constraint**

**VAR Admin BASE RELATION**

**{ adminId VARCHAR ,**

**Name VARCHAR,**

**Rank VARCHAR,**

**adminEmail VARCHAR,**

**adminPassword VARCHAR}**

**PRIMARY KEY {adminId};**

**36**

**VAR Customer BASE RELATION**

**{ custId VARCHAR ,**

**Name VARCHAR,**

**Address VARCHAR,**

**Phone VARCHAR,**

**Email VARCHAR**

**custPassword VARCHAR}**

**PRIMARY KEY {custId};**

**VAR Bookstore BASE RELATION**

**{ bookId VARCHAR ,**

**bookTitle VARCHAR,**

**Author VARCHAR,**

**Isbn VARCHAR,**

**pubDate VARCHAR**

**Category VARCHAR,**

**Price INT,**

**Qty INT}**

**PRIMARY KEY {bookId};**

**37**

**VAR Purchase BASE RELATION**

**{ pId VARCHAR ,**

**custId VARCHAR,**

**bookId VARCHAR,**

**Qty VARCHAR,**

**PRIMARY KEY {pId}**

**FOREIGN KEY (custId) REFRENCES Customer**

**FOREIGN KEY (bookId) REFRENCES Bookstore;**

**VAR Delivery BASE RELATION**

**{ deliId VARCHAR ,**

**NameVARCHAR,**

**Email VARCHAR,**

**PRIMARY KEY {deliId};**

**38**

**VAR Payment BASE RELATION**

**{ payId VARCHAR ,**

**custId VARCHAR,**

**deliId VARCHAR,**

**payAmt INT,**

**payDate VARCHAR,**

**PRIMARY KEY {payId}**

**FOREIGN KEY (custId) REFRENCES Customer**

**FOREIGN KEY (deliId) REFRENCES Delivery;**

**Relvar Constraint**

**CONSTRAINT Bookstore**

**COUNT(Bookstore)=COUNT(Bookstore{bookId});**

**CONSTRAINT Admin**

**COUNT(Admin where Rank="Moderator");**

**39**

**CONSTRAINT Deli**

**COUNT(Delivery)=COUNT(Delivery(deliId));**

**Database Constraint**

**CONSTRAINT SuccessOrder**

**Delivery(deliId)=Payment(deliId);**

**2.5.2 Security Constraint**

**AUTHORITY Admin**

**GRANT ALL**

**ON Admin,Customer,Bookstore,Purchase,Delivery,Payment**

**TO James;**

**AUTHORITY Mod**

**GRANT RETRIEVE**

**ON Admin,Customer,Bookstore,Purchase,Delivery,Payment**

**TO Ella;**

**40**

**Chapter(3)**

**Conclusion**

    This system is efficient in maintaining customer’s details and can easily perform operations on customer’s records. This system also reduces the work load of the shop keeper to know the quantity of books available and also keep the records of how many books are purchased and sold. The system perfectly met all of our objectives. We also analyzed online bookstores. We searched books’ titles, authors, etc. from the browser.

**Future Amendments**

    If we have to do this project again in the future, we want to create more effective and useful system. We want to add more functions. For example, our bookstore will give customers discount for many books purchased. Also, we want to develop a website for this system by using a programming language.