INSTABOOK

ONLINE BOOK STORE DATABASE

FNU ANIRUDH, MADHURI UPADRASTA

TABLE OF CONTENTS

1. **PROJECT DESCRIPTION**---------- **2**

2. **PROJECT MEMBERS**---------- **3**

3. **DATABASE ENVIRNOMENT**---------- **4**

A) Client Profile---------- 4

B) Data Flow Diagram---------- 5

C) User Profile---------- 7

4. **GOALS AND OBJECTIVES**---------- 8

A) Client/User Goals---------- **8**

B) Client/User Requirements---------- 8

C) Database Goals---------- 8

5. DATABASE PROFILE---------- **9**

A) Business Rules---------- 9

B) Entity Relationship Diagram---------- 9

C) Entities and Relationships---------- 10

6. DATABASE ASSESSMENT---------- **12**

A) Validation of Database System against Objectives---------- 12

B) Weakness & Limitations of Current Implementation---------- 13

C) Future Changes and Upgrades---------- 13

7. QUERIES---------- **14**

8. ADDITONAL:HOW TO VISUALIZE DATA IN R USING SQL QUERIES----------
---------- **23**

Project Team – FNU Anirudh, Madhuri Upadrasta

Project Description

InstaBook Inc., simply referred as **InstaBook**, is an American online bookstore startup with its location at Sacramento, California. It is a one-stop shopping for true bibliophiles, with carefully curated collections from independent booksellers. If you don't have a local bookstore or if you need a specific title (like a textbook), the web is your best bet and even though it's online, it's still got the feel of what may be the world's coolest neighborhood bookstore. The customers can opt for the delivery with free shipping. The site is easy to search, but clicking around gives you the feel of a virtual independent bookstore.

This startup needs a database management system to track of their orders, reports and queries in order to make their process simple and data easily accessible. A database will help customers track their order and allow employee to access specific information at any time. This solution will enhance their work efficiency. The goal of this project is to create a database that can add and remove data easily, in addition, provide more efficient storage, avoid data duplication, avoid inconsistent record, easy to change information and maintain security.

Our database will provide up-to-date information of the following : enter and edit product history, update customers and contacts; reports - with lists of daily orders, lists of orders by Zip Code for Tax Info, lists of recurring customer orders and queries to retrieve specific data from the tables.

Customers can track their order and edit their information, whereas Employees can access and track specific orders and analyze the reading patterns of users. InstaBook will use this system to track the activities for analyzing user behavior through history of records maintained for the future business development.

In conclusion, we believe that implementing such a database system will automate a lot of processes and will save effort, time and money.

FNU Anirudh (aanirudh@iu.edu), Madhuri Upadrasta (mupadras@umail.iu.edu)

Project Members

Instabook Database Management System provided a learning opportunity for us. We were able to apply database concepts we learned in class.

Project Contributions

1. **FNU ANIRUDH**

a) Documented User requirements, Flow diagram and business rule.
b) Created Database, Created all the tables and defined Primary key, foreign keys.
c) Inserted Data into all the tables manually.
d) Implemented Trigger which checks quantity ordered against total quantity.
e) Came up with 10 queries for information retrieval which improves efficiency of the organization.
f) Showed how SQL queries can be used in R to visualize data.

Challenges Faced:-

i) Constraint fails to display error message in MYSQL though implemented correctly, MYSQL accepts data even though constraint fails.
ii) Moved entire database by re-writing queries in Postgresql and re-applying constraint.
iii) Realized that Postgresql doesn't support subqueries in constraints.
iv) Implemented Trigger in MYSQL again to solve the problem.

2. **MADHURI UPADRASTA**

i) Helped documenting Project Description and Initial draft.
ii) Provided data for Customer table.
iii) Helped in Preparing and presenting PPT.

DATABASE ENVIORNMENT

A. CLIENT PROFILE

InstaBook Inc., simply referred as InstaBook, is an American online bookstore startup with its location at Sacramento, California. It is a one-stop shopping for true bibliophiles, with carefully curated collections from independent booksellers. If you don't have a local bookstore or if you need a specific title (like a textbook), the web is your best bet and even though it's online, it's still got the feel of what may be the world's coolest neighborhood bookstore. The customers can opt for the delivery with free shipping. The site is easy to search, but clicking around gives you the feel of a virtual independent bookstore.

InstaBook would benefit from database in several ways:

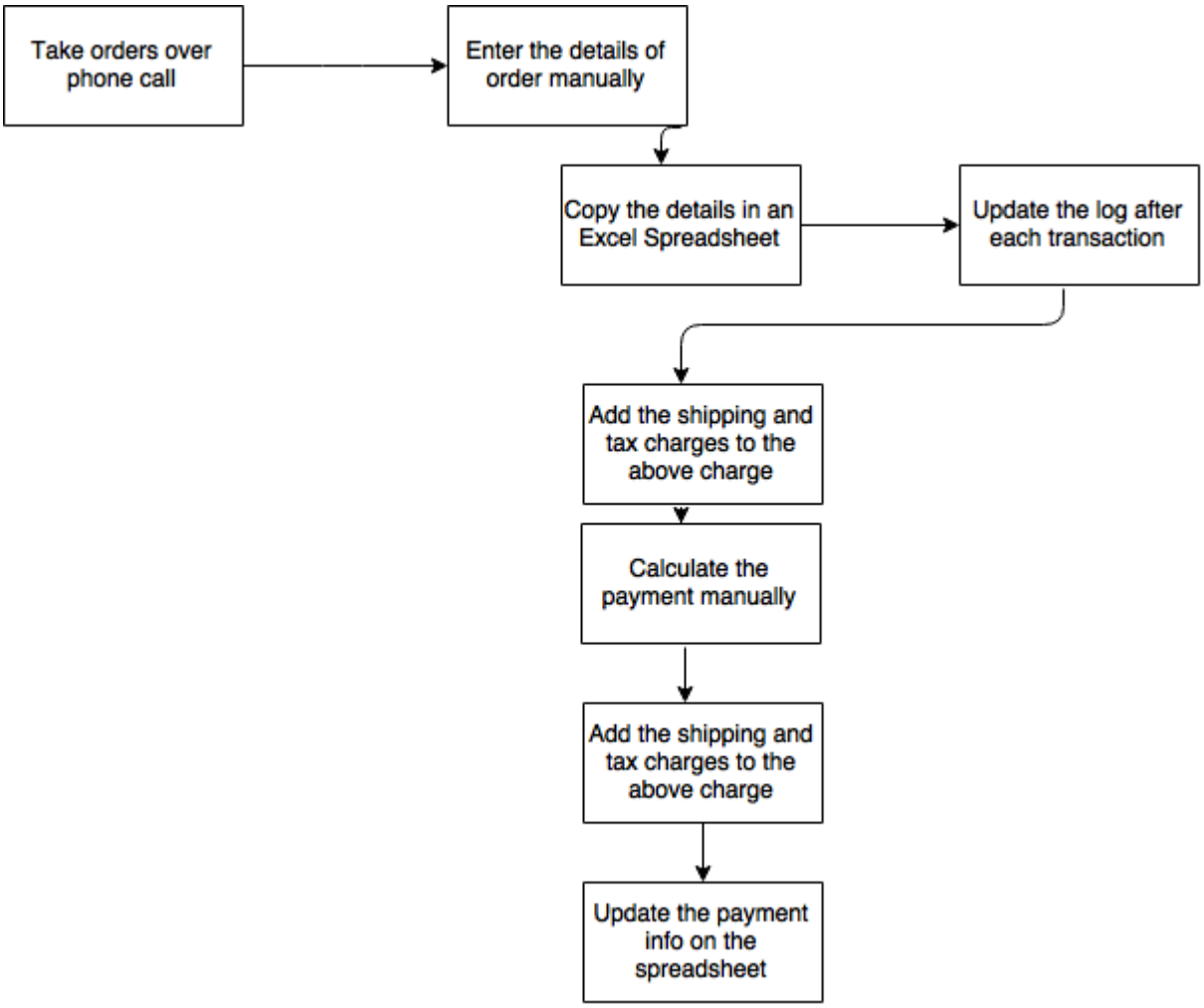
- To track orders.
- Allow customers to view their orders.
- To track their reports.
- Allow employee to access specific information at any time.
- It will reduce the risk of errors they were susceptible to while using a paper-based system.
- Improve efficiency by reducing time taken for processing information.

The database will store large amounts of data

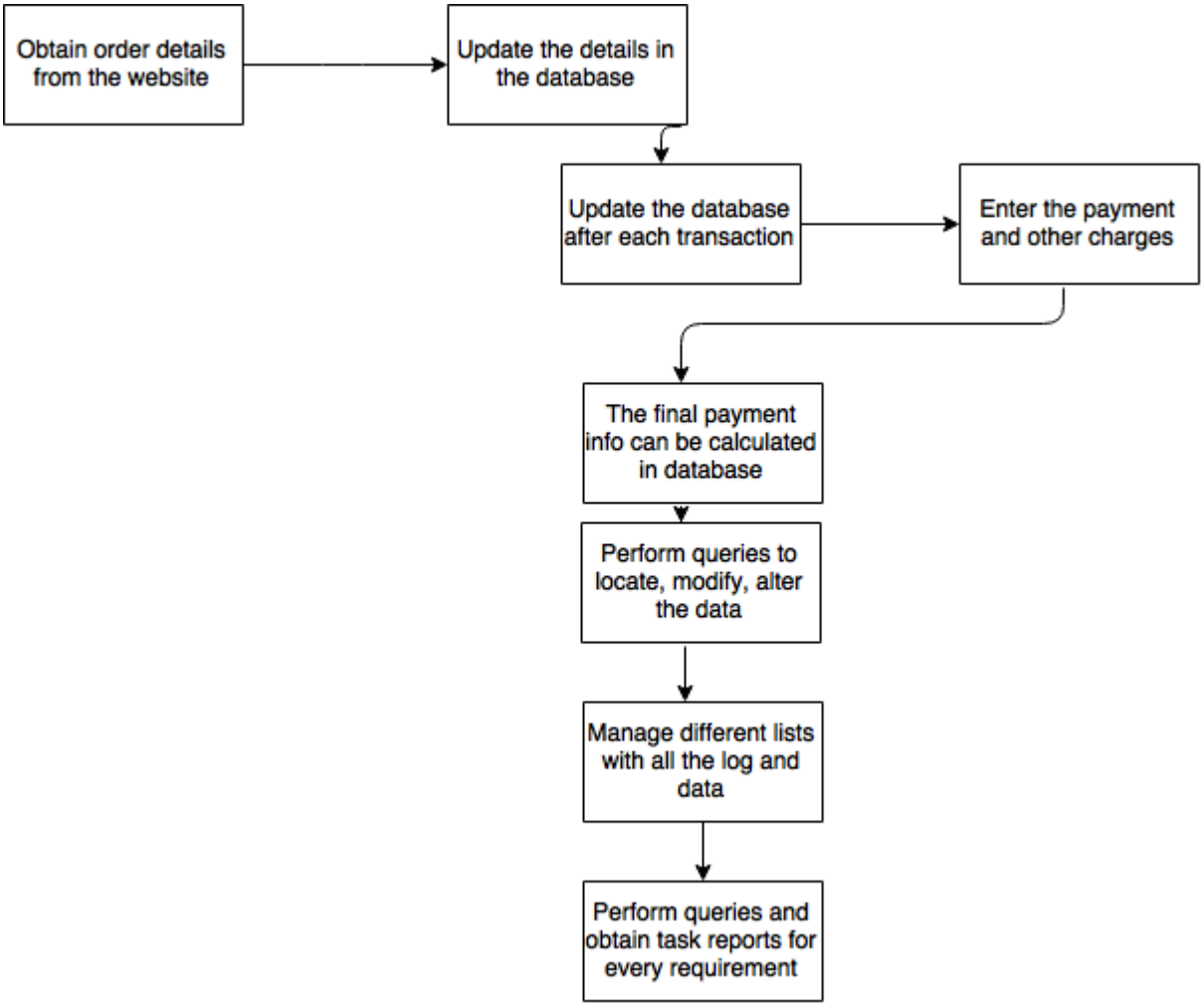
- Customer Information: - Information related to customer (including items such as name, address and phone number) will be stored in the database.
- Book Information: - Information related to book in store (including Author name, ISBN, Price, Category and Publisher) will be stored in the database.
- Order Information: - Information related to order placed (including Order_id, Order_date, Shipping address, ordered quantity) will be stored in the database.

B. DATA FLOW DIAGRAM

BEFORE THE UTILIZATION OF A DATABASE MANAGEMENT SYSTEM



AFTER THE UTILIZATION OF A DATABASE MANAGEMENT SYSTEM



C. User Profile

Customers: - The largest and most varied group will be the customers. Customers can be from any state or city and will vary widely in age, gender and location. Customers will access database to view their orders, edit and read information.

Employees: - Employees are staff members who will use the database for multiple reasons. Employees will have ability to both read, write and modify information. Employees can track specific orders and help improve business by understanding user's reading pattern and by estimating demand and supply.

Goals & Objectives

The Database will be used by the employees of the company to carry out their day-to-day activities and the customers to check the view their orders. The database is all about order management and information management. The database is created as a powerful interface that is easy to access and manage also being effective and powerful in implementation. The goal of this project is to create a database that can add and remove data easily, in addition, provide more efficient storage, avoid data duplication, avoid inconsistent record, easy to change information and maintain security.

Client/User Goals

The database provides up-to-date information of the following:

- enter and edit product history, update customers and contacts;
- reports - with lists of daily orders,
- Lists of orders by Zip Code for Tax Info,
- List of recurring customer orders and queries to retrieve specific data from the tables.
- Customers can view their order and edit personal information.
- Employees can access and track specific orders and analyze following:
    1. Most ordered book
    2. Reading Pattern of user
    3. Frequent user
    4. State or City with most number of orders.

Client/User Requirements

- A field to indicate order date i.e. data on which order is placed.
- A field indicating address and zip code of customer which can be modified.
- A field indicating name of the book.
- A field indicating quantity.
- A field indicating Order ID or Transaction ID.

Database Goals

- Create a database that is easy/intuitive for users with various levels of computers skills since data will be entered by various members including customers.
- Allow the employees or staff members to quickly and easily generate the report.
- Eliminate paper work completely to keep track of orders.
- Allow Client to rank customers based on number of orders or amount spent.

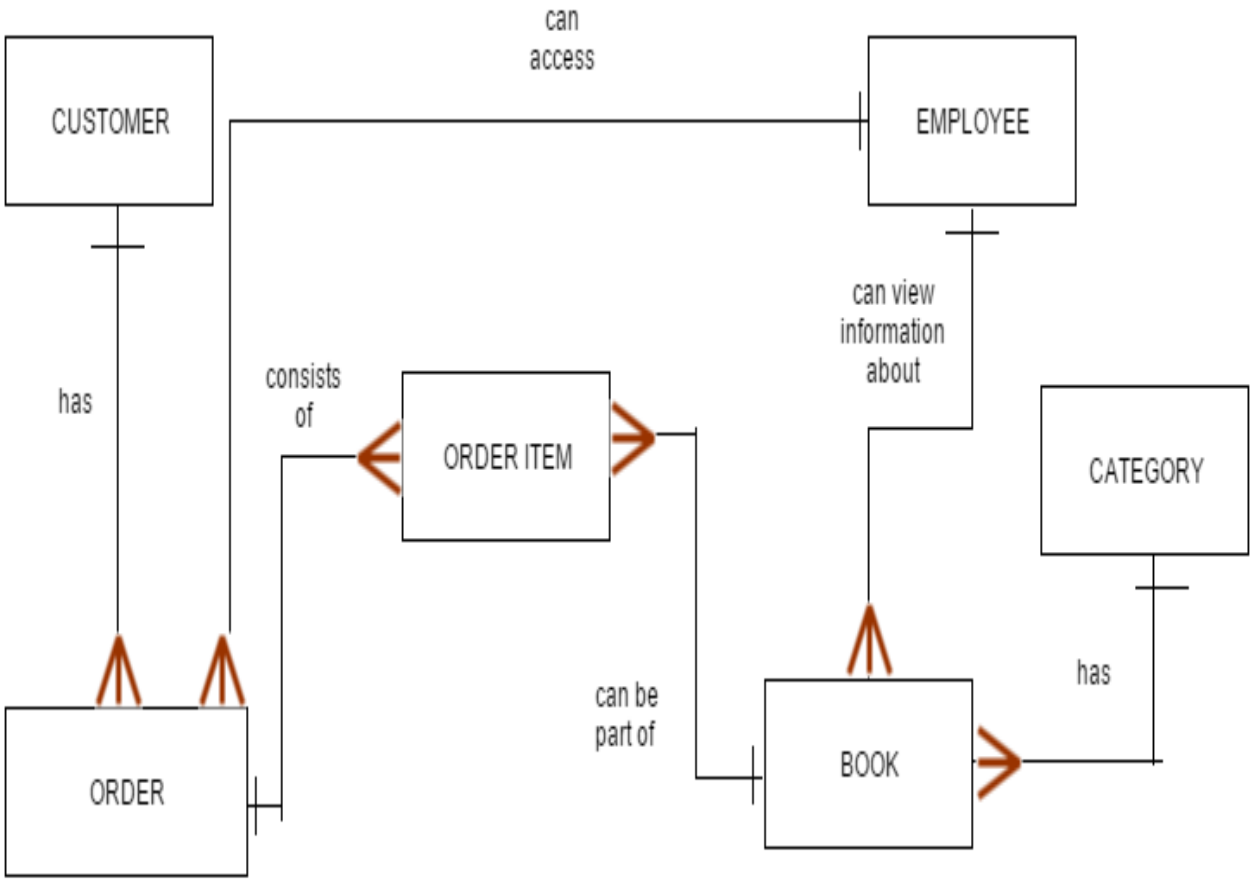
- Create more uniform efficient and uniform method of storing and accessing information about orders and customers.
- Allow customers to track and know the status of their order.

DATABASE GOALS

BUSINESS RULE

- One customer can place many orders, one order can be placed by one customer only
- One category can have many books, one book can have only one category
- One book can have many orders, one order can have many books.

ENTITY RELATIONSHIP DIAGRAM



Note: - This E-R Diagram is different from one we implemented. We implemented many to one relationship between Book and Order but keeping this diagram since many to one relationship is generally avoided.

ENTITIES AND RELATIONSHIP

Customer Table

Customer table stores information related to customer. It stores individual information about customer i.e. their First Name, Last Name, Address and E-mail (PRIMARY KEY). The table stores list of customers and their information.

PRIMARY KEY: - Email

FOREIGN KEY: - None

Order Table

Order table stores information related to order. It stores individual order of each customer. I.e. Order Id, Order date, User information, Book information, quantity ordered and order_date. The table stores all the orders placed by customers.

PRIMARY KEY: - None

FOREIGN KEY: - Email, ISBN

Book Table

Book Table stores information related to books. It stores information related to each book i.e. Book name, ISBN, Author name, category_id, Publisher, price, quantity and edition.

PRIMARY KEY: - ISBN

FOREIGN KEY: - category_id

Category Table

Category Table stores information related to category of each book. It stores and classifies book based on category or genre i.e. Category_id and Category Name.

PRIMARY KEY: - Category_id

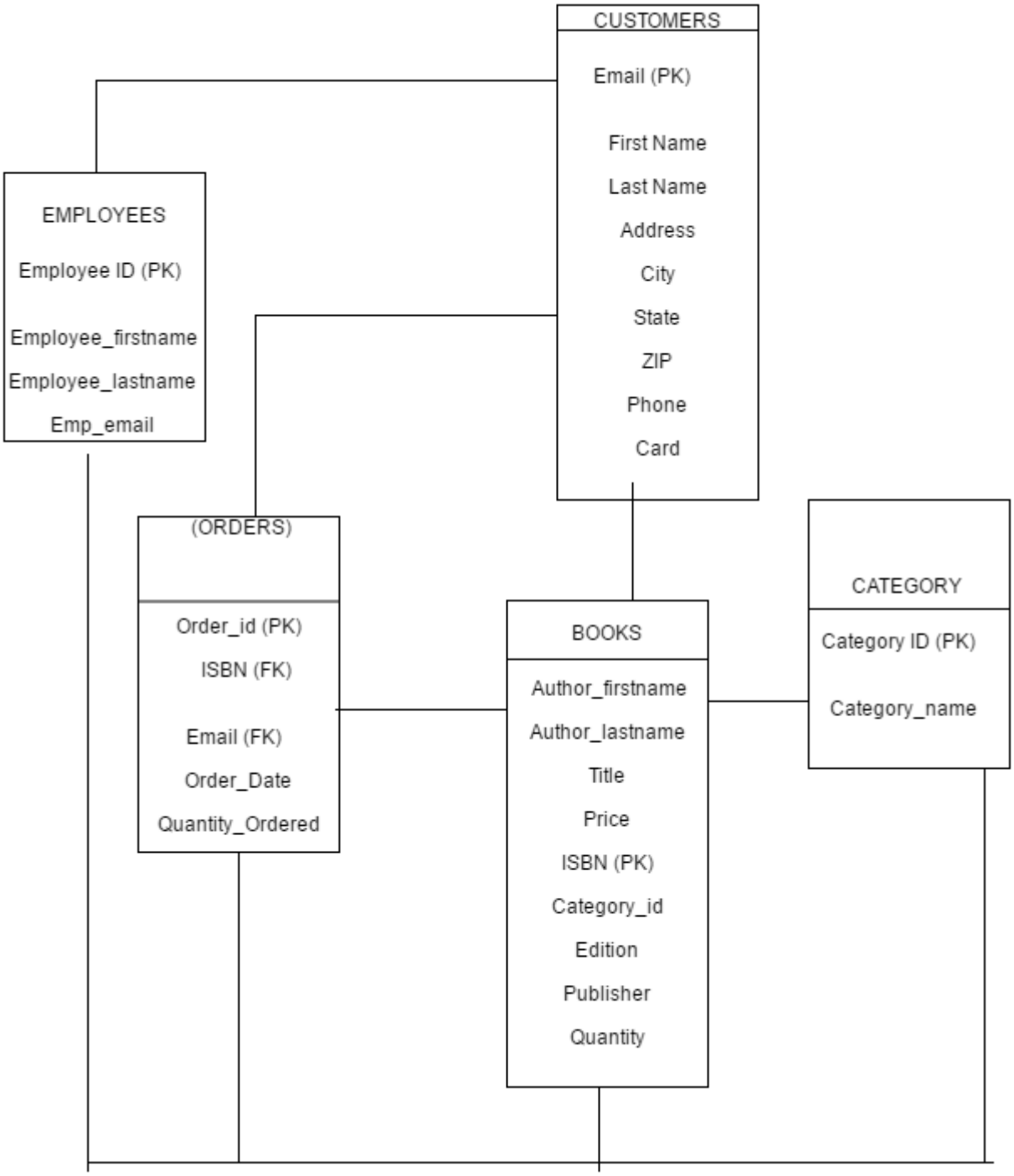
FOREIGN KEY: - None

Employee Table

Employee Table stores information about employee who can view or edit orders. It stores list of employees in company i.e. Employee id, First Name, Last Name, Employee email.

PRIMARY KEY: - Employee_id

FOREIGN KEY: - None



Database Assessment

A. Validation of Database System against Objectives

- Database has ability to store and access information about each customer and orders.

- System is capable of generating reports that provide an overview of all of the applications.
- Met client/user requirements as described in above, with the exception of the Creation of fields as asked.
- System has eliminated the need for a separate tracking spreadsheet while collocating
- application information in one place
- Database offers efficient and uniform method of storing and accessing applicant information
- The structure of this database directly improved the efficiency of system and achievement of results. Also on other hand, this database structure improved the data integrity, storage and consistency.

B. Weaknesses & Limitations of Current Implementation

- The customers cannot access their log information about their previous transactions.
- Customers can just only check the status of the transaction
- Customers do not have the access to track their packages when out for delivery
- The employees can use the dbms only as a source to save data of the customers but do not have any personalized information of their timesheets and any other log information
- It is not possible to toggle between multiple transactions made by the same customer in a single query.

C. Future Changes and Upgrades

- Can update the entire database to create employee login interface and a customer login interface
- We can perform a query to determine the future demand of the books using an exponential moving average where forecasted demand can be given by

$$\textit{forecasted demand} = \sum_{i=0}^{t=1} \propto\ (1-\propto)^{-1} D_{t-i}$$

- We can have more attributes and attribute values under each category
- We can track the customer's' data and create a log for recurring customers
- We can reduce many to one relationship and implement better process to guarantee better efficiency.

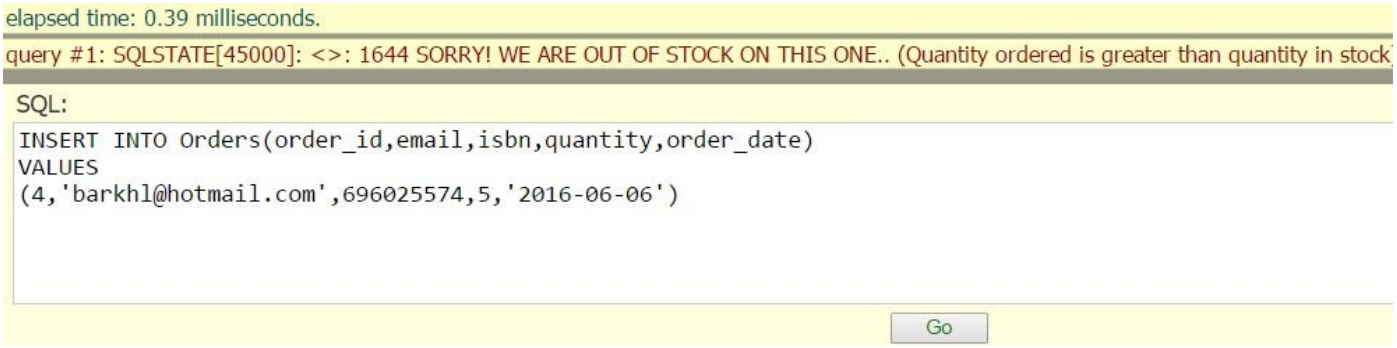
QUERIES

1. HOW TRIGGER CAN HELP DATABASE WHEN ORDERED QUANTITY IS GREATER THAN IN STOCK QUANTITY.

TRIGGER IMPLEMENTATION

```
DELIMITER //

CREATE or REPLACE TRIGGER check_order

BEFORE INSERT ON orders

FOR EACH ROW

BEGIN

  IF (NEW.Quantity > 0) AND (SELECT quantity from book where
isbn=NEW.isbn)>NEW.quantity THEN

    SET NEW.Quantity = NEW.Quantity;

  ELSE

    signal sqlstate '45000' set message_text = "SORRY! WE ARE OUT OF STOCK ON THIS
ONE.. (Quantity ordered is greater than quantity in stock)";

  END IF;

END

//
```

Description: - We are trying to order 5 quantities of particular book but there are only 3 books which are available in stock.



Result: - Database throws error message when user tries to order more quantity than in stock.

2. CALCULATING TOTAL AMOUNT TO BE PAID BY CUSTOMER WHO HAS ORDERED MORE 1 OR MORE BOOK OR ORDERED IN LARGE QUANTITY

Description: - This query calculates the sum or total amount to be paid by customer when he/she orders more than 1 book or orders books in large quantity.

SQL:

```
SELECT SUM(o.quantity*b.price)AS Total_Amount  FROM Orders o
INNER JOIN Book b
USING(isbn)
WHERE order_id=1;
SELECT *FROM Orders
WHERE order_id=1;
```

Go

Query 1:

Total_Amount
38

Query 2:

order_id	Email	ISBN	Quantity	order_date
1	jholloway@hotmail.com	812821823	2	2015-12-10
1	jholloway@hotmail.com	716313197	2	2015-12-10

Result: - Josh ordered 4 books where 2 books are priced at $ 5 each and other two at $ 14 each. The total sum is $ 38.

3. LISTING ALL BOOKS ACCORDING TO CATEGORY NAME IN ASCENDING ORDER OF CATEGORY

Description: - This query lists all the books in stock according to category name which is connected to books table via category id and lists them according to category name.

```
SELECT author_fname,author_lname,title,c.category_name,price,edition,publisher,quantity FROM Book
INNER JOIN Category c
USING (category_id)
ORDER BY c.category_name;
```

Go

author_fname	author_lname	title	category_name	price	edition	publisher	quantity
Frank	Rosenow	Manual Art	Arts	50	Hardcover	Norton	25
Stephen	Bugg	Understanding Art	Arts	10	Paperback	Scholastic	5
Daniel	Vettori	Modern Art (Essential Art)	Arts	50	Hardcover	Parragon Publishing	12
RS	Bharadwaj	Business Statistics	Business	30	Paperback	Schand	15
John D	Radebaugh	International Business	Business	20	Paperback	McGraw	7
Kenneth	Brookbank	Educational administration	Education	8	Paperback	Longman	14
Clive	Beck	Moral education:interdisciplinary approaches	Education	10	Paperback	Newman P	1
Jean	Ferris	Across the grain	Fiction	10	Paperback	Farrar Straus Giroux	4
Kristin	Hannah	Angel Falls	Fiction	30	Hardcover	Crown Publishers	16
Siv Cedering	Fox	Joys of fantasy:the book for loving couples	Fiction	5	Paperback	Stein and Day	7
Carolyn	Gage	The second coming of Joan of Arc and other plays	Fiction	12	Paperback	Herbooks	9
John D	Roscecrance	Gambling without guilt:the legitimation of an American pastime	Gambling	60	Paperback	Brooks/Cole Pub. Co	50
Bruce Edward	Moon	Existential art therapy: the canvas mirror	Health	20	Hardcover	C.C. Thomas	10
Garfield	Sobers	Health Today	Health	6	Paperback	Angus & Robertson	2
Stewart	Lynette	Palms for the home and garden	Home & Garden	15	Hardcover	Angus & Robertson	8
David	Bowie	Better Homes and Gardens New Garden Book	Home & Garden	5	Paperback	Meredith	4

Result: - Query fetches list of all the books arranged category wise in alphabetical order of category name. It helps employees and customer to go through books available in each category.

4. COUNTING NUMBER OF CUSTOMER IN EACH STATE

Description: - This query counts number of customers in each state from whom we have received orders.

```
SELECT state, COUNT(state)AS Number_of_Customers FROM Customer
GROUP BY state;
```

Go

state	Number_of_Customers
AL	1
AR	1
AZ	3
CA	4
IL	1
LA	1
MA	1
MD	1
MN	2
NV	2
NY	1
OK	1
OR	3
UT	2
WA	1

Result: - This query helps organization in finding states with most number of customers, results can be used to target future potential customers in other states or more customers in particular state.

5. LIST ALL BOOK TITLES BETWEEN ‘A’ AND ‘M’ WHOSE PRICE IS MORE THAN ‘AVERAGE PRICE’.

Description: - This query helps find book names between ‘A’ and ‘M’ whose price is greater than average price.

SQL: Database: -- Select Database

```
select author_fname,author_lname,title,edition,publisher,price from book  WHERE title BETWEEN 'A' AND 'M'AND price> (select avg(price
from book)
ORDER BY title;
SELECT AVG(price) FROM Book;
```

Go

Query 1:

author_fname	author_lname	title	edition	publisher	price
Kristin	Hannah	Angel Falls	Hardcover	Crown Publishers	30
RS	Bharadwaj	Business Statistics	Paperback	Schand	30
John D	Roscecrance	Gambling without guilt:the legitimation of an American pastime	Paperback	Brooks/Cole Pub. Co	60
Riccardo	Poli	Genetic Programming	Hardcover	Springer	23
Joyce	Farrell	Java Programming	Paperback	ButterWorths	25

Query 2:

AVG(price)
21.4000

Result: - Queries displays all the books whose price is greater than average price and where title name is between ‘A’ and ‘M’.

6. VIEWING DISTINCT CUSTOMER INFORMATION ALONG WITH DATE THEY PLACED THEIR ORDER AND ORDER ID.

Description: - This query displays customer information with their order date.

4 rows returned; elapsed time: 0.82 milliseconds.

SQL: Database:

```
SELECT DISTINCT o.order_id,cus_fname,cus_lname,email,address,city,state,zip,phone,card,o.order_date FROM Customer
INNER JOIN Orders o
USING (email)
ORDER BY o.order_id;
```

Go

order_id	cus_fname	cus_lname	email	address	city	state	zip	phone	card	order_date
1	Jescie	Holloway	jholloway@hotmail.com	Ap #757-1842 Odio Street	Chicago	IL	60290	2147483647	Credit	2015-12-10
2	Anika	Hopper	ahopper@gmail.com	754-742 Non Av	Fayetteville	AR	72701	2147483647	Credit	2014-06-10
3	Isadora	Armstrong	isarmstrong@gmail.com	5396 Auctor Av	Phoenix	AZ	85005	2147483647	Debit	2012-08-09
4	Barclay	Hale	barkhl@hotmail.com	5889 Vel, Road	Saint Paul	MN	55104	2147483647	Debit	2016-06-06

Result: - This query helps employees view distinct customer information along with order details.

7. LISTING ALL CUSTOMERS WHO ARE FROM CALIFORNIA AND WHO USE CREDIT CARD FOR PAYMENT

Description: - This query helps employees to find out list of customer who are from state of California and use credit card for payment.

SQL:

```
SELECT *FROM Customer
WHERE state='CA' and card='Credit'
ORDER BY cus_fname;
```

Go

cus_fname	cus_lname	email	address	city	state	zip	phone	card
Briar	Hutchinson	sagittis.lobortis@tortor.ca	P.O. Box 694, 284 Odio Rd	San Francisco	CA	94249	2147483647	Credit
Rana	Livingston	ranliving@yahoo.com	3261 Feugiat Rd.	San Francisco	CA	94130	2147483647	Credit
Wendy	Ryan	wenry@usc.com	765-8607 Nulla. Street	Los Angles	CA	90007	2147483647	Credit

Result: - This query fetches list of all the customers from state of California who use credit card for payment. Instabook can use this information to reward cash points to these customers on purchase and keep them loyal to the brand.

8. FINDING LIST OF ORDERS BETWEEN BEGINNING OF MARCH IN 2011 TILL END OF DECEMBER 2015

Description: - This query finds list of orders that were placed between March 2011 to December 2015.

SQL:

```
SELECT order_id,email,b.title,isbn,b.price,b.edition,b.publisher,order_date FROM Orders
INNER JOIN Book b
USING(isbn)
WHERE order_date>='2011-03-01'
AND order_date<='2015-12-31' ;
```

Go

order_id	email	title	isbn	price	edition	publisher	order_date
1	jholloway@hotmail.com	Joys of fantasy:the book for loving couples	812821823	5	Paperback	Stein and Day	2015-12-10
1	jholloway@hotmail.com	Building Societies	716313197	14	Hardcover	Fabian Society	2015-12-10
2	ahopper@gmail.com	Angel Falls	609605925	30	Hardcover	Crown Publishers	2014-06-10
3	isarmstrong@gmail.com	Genetic Programming	2147483647	23	Hardcover	Springer	2012-08-09

Result: - This query finds all the orders that were placed in between March 2011 till December 2015. This query helps management see revenue made during last few years and also gives information about books that were widely sold and frequent customers.

9. DISPLAY ALL THE BOOKS ALONG WITH CATEGORY WHOSE MAXIMUM PRICE IS GREATER THAN $ 25

Description: - This query displays all the books along with their title, author,price and category name which are greater than $ 25.

SQL:

```
SELECT title,author_fname,author_lname,c.category_name, max(price) FROM Book
INNER JOIN Category c
USING(category_id) GROUP BY title HAVING max(price)>25;
```

Go

title	author_fname	author_lname	category_name	max(price)
Angel Falls	Kristin	Hannah	Fiction	30
Business Statistics	RS	Bharadwaj	Business	30
Gambling without guilt:the legitimation of an American pastime	John D	Roscecrance	Gambling	60
Manual Art	Frank	Rosenow	Arts	50
Modern Art (Essential Art)	Daniel	Vettori	Arts	50

Result: - This query displays all the book titles, author_name, category along with price for all the books which are marked more than $25. It helps employees and management know which books can actually bring more profit if sold with offer or with discount.

10. QUERY THAT JOINS ALL THE 4 TABLES AND DISPLAYS CUSTOMER DETAILS ALONG WITH BOOK DETAILS, CATEGORY NAME AND ORDER DETAILS

Part I

Description: - This query displays all the details about customer, his book order and category from where he ordered and order details.

Query:-

```
SELECT o.Order_id,c.cus_fname AS Firstname,c.cus_lname AS Lastname,c.email AS
Email,b.title AS Bookname,a.category_name AS Category,o.quantity AS Quantityordered,o.isbn
AS ISBN,c.address AS Address,c.city AS City,c.state AS State,c.zip AS ZIP,c.phone AS
Phone,c.card AS Payment FROM Customer c

INNER JOIN Orders o
```

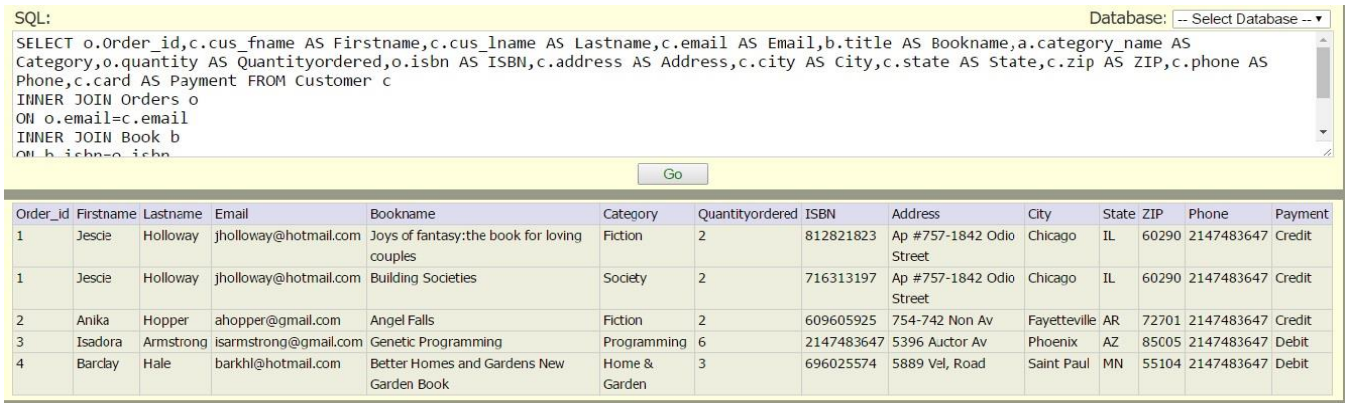
ON o.email=c.email

INNER JOIN Book b

ON b.isbn=o.isbn

INNER JOIN Category a

ON a.category_id=b.category_id;



SQL: Database: -- Select Database --

```
SELECT o.Order_id,c.cus_fname AS Firstname,c.cus_lname AS Lastname,c.email AS Email,b.title AS Bookname,a.category_name AS
Category,o.quantity AS Quantityordered,o.isbn AS ISBN,c.address AS Address,c.city AS City,c.state AS State,c.zip AS ZIP,c.phone AS
Phone,c.card AS Payment FROM Customer c
INNER JOIN Orders o
ON o.email=c.email
INNER JOIN Book b
ON b.isbn=o.isbn
```

Go

Order_id	Firstname	Lastname	Email	Bookname	Category	Quantityordered	ISBN	Address	City	State	ZIP	Phone	Payment
1	Jescie	Holloway	jholloway@hotmail.com	Joys of fantasy:the book for loving couples	Fiction	2	812821823	Ap #757-1842 Odio Street	Chicago	IL	60290	2147483647	Credit
1	Jescie	Holloway	jholloway@hotmail.com	Building Societies	Society	2	716313197	Ap #757-1842 Odio Street	Chicago	IL	60290	2147483647	Credit
2	Anika	Hopper	ahopper@gmail.com	Angel Falls	Fiction	2	609605925	754-742 Non Av	Fayetteville	AR	72701	2147483647	Credit
3	Isadora	Armstrong	isarmstrong@gmail.com	Genetic Programming	Programming	6	2147483647	5396 Auctor Av	Phoenix	AZ	85005	2147483647	Debit
4	Barclay	Hale	barkhl@hotmail.com	Better Homes and Gardens New Garden Book	Home & Garden	3	696025574	5889 Vel, Road	Saint Paul	MN	55104	2147483647	Debit

Result:- This query can be used by employee to see all the information regarding a customer like his name, email, address, phone, payment method, book ordered, category to which book belongs and order_id. This query joins all the 4 tables to fetch result.

Part II

Description: - This query is similar to previous query but is specific to customer where customer can specify his email to see all the orders made by him.

Query:-

SELECT o.Order_id,c.cus_fname AS Firstname,c.cus_lname AS Lastname,c.email AS Email,b.title AS Bookname,a.category_name AS Category,o.quantity AS Quantityordered,o.isbn AS ISBN,c.address AS Address,c.city AS City,c.state AS State,c.zip AS ZIP,c.phone AS Phone,c.card AS Payment FROM Customer c

INNER JOIN Orders o

ON o.email=c.email

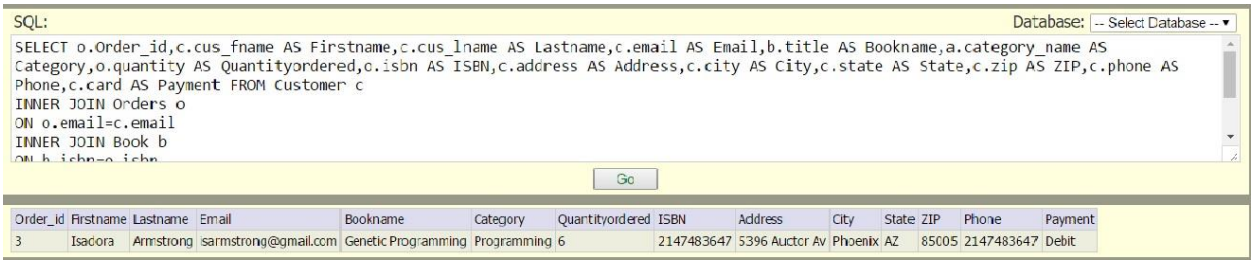
INNER JOIN Book b

ON b.isbn=o.isbn

INNER JOIN Category a

ON a.category_id=b.category_id

WHERE c.email='isarmstrong@gmail.com';



Result: - This query allows customer to view their orders and keep track their activities. This query joins all the 4 tables to display result and customer uses his email id to view his/her order.

ADDITIONAL: - HOW SQL QUERIES CAN BE USED TO VISUALIZE DATABASE IN R

MYSQL Database can be connected to RStudio using dbConnect library and then specifying user name and password to make connection. Once connection is established one can run SQL Queries in R and perform data visualization.

Example 1:- In this example I am taking all the data from Customer table and then just selecting city column from table and use geccode to get latitude and longitude of all the cities. Later, I plot all the cities on Map.

```
library(dbConnect)
library(ggmap)
library(ggplot2)
library(maps)
mydb = dbConnect(MySQL(), user='sid', password='saibaba',
                 dbname='Instabook', host='localhost')
rs = dbSendQuery(mydb, "select *from customer")
data = fetch(rs, n=-1)
cities=data[5] // Take all the cities as input
a=geocode(cities$city)
latitude<-a$lat
longitude<-a$lon
map("world","usa",fill = TRUE,col="yellow2",bg="lightblue", ylim=c(25, 49), m
ar=c(0,5,5,3))
points(longitude,latitude, col="red", pch=16)
```

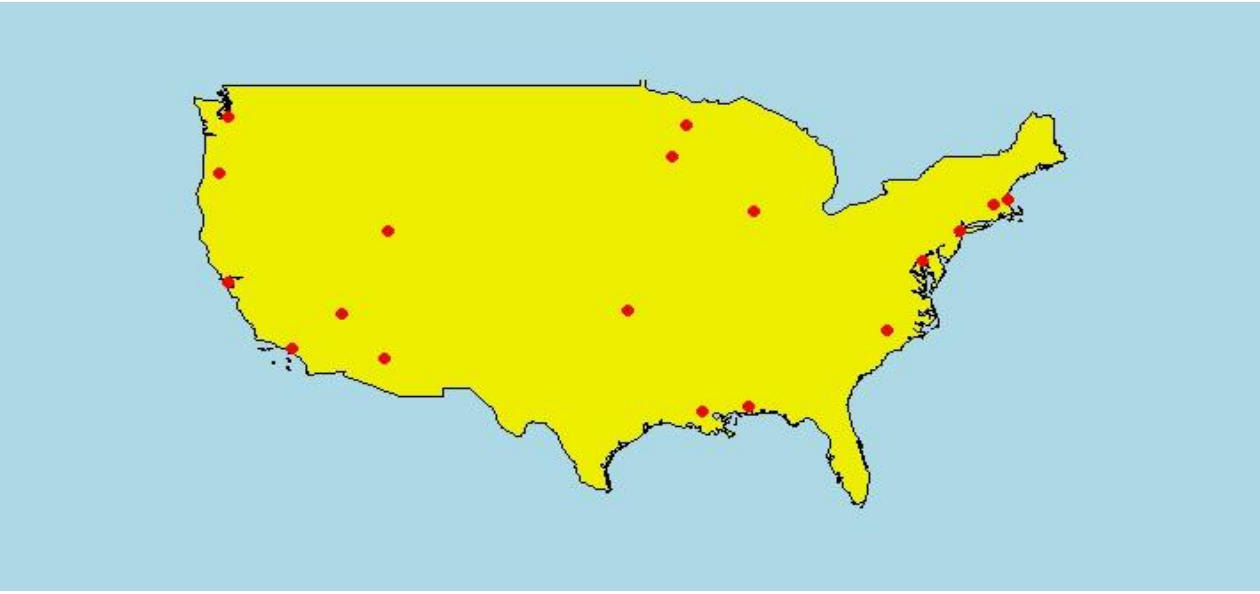


Figure: - List of all the cities from where InstaBook has customers.

Example 2:- In this example, I am using SQL Query to count number of paperback and hardcover books and then plotting them in graph instead of printing it.

```
ps = dbSendQuery(mydb, "SELECT edition,count(edition) AS count from book
GROUP BY edition")
data2=fetch(ps,n=-1)
ggplot(data = data2, aes(edition, count,fill = edition)) + geom_bar(stat="ide
ntity")
```

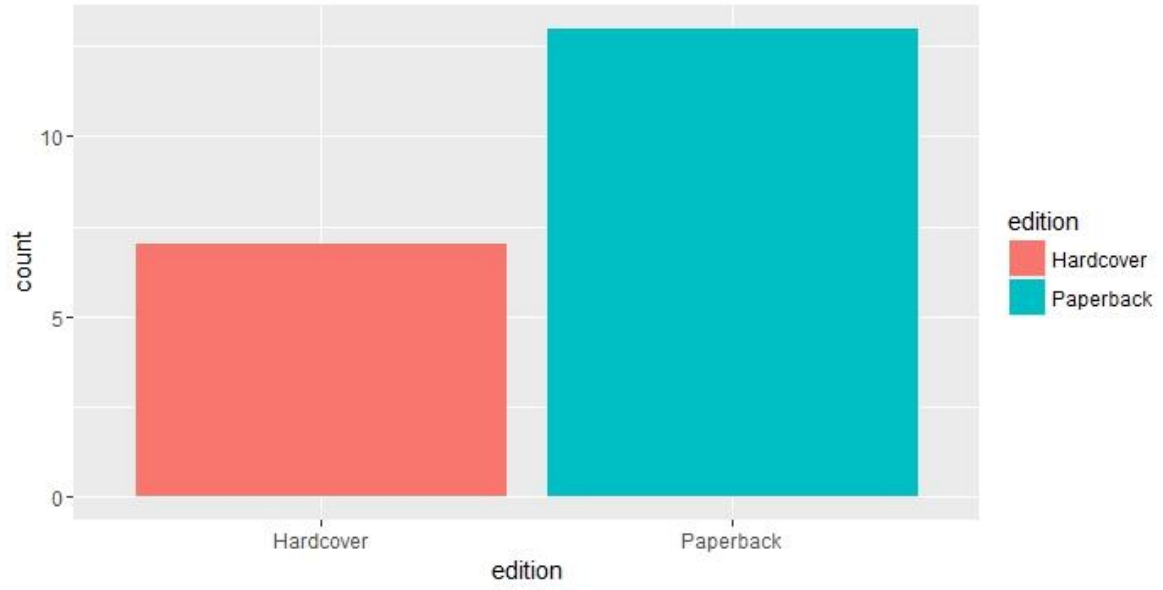


Figure: - Paperback vs Hardcover (Edition Preference)

Example 3:- In this example I am using SQL query to count customers with debit and credit card from customer table and then using the result to visualize by plotting graph.

```
ms=dbSendQuery(mydb, "SELECT card,count(card) AS count from Customer
GROUP BY card")
data3=fetch(ms,n=-1)
ggplot(data = data3, aes(card, count,fill = card)) + geom_bar(stat="identity"
)
```

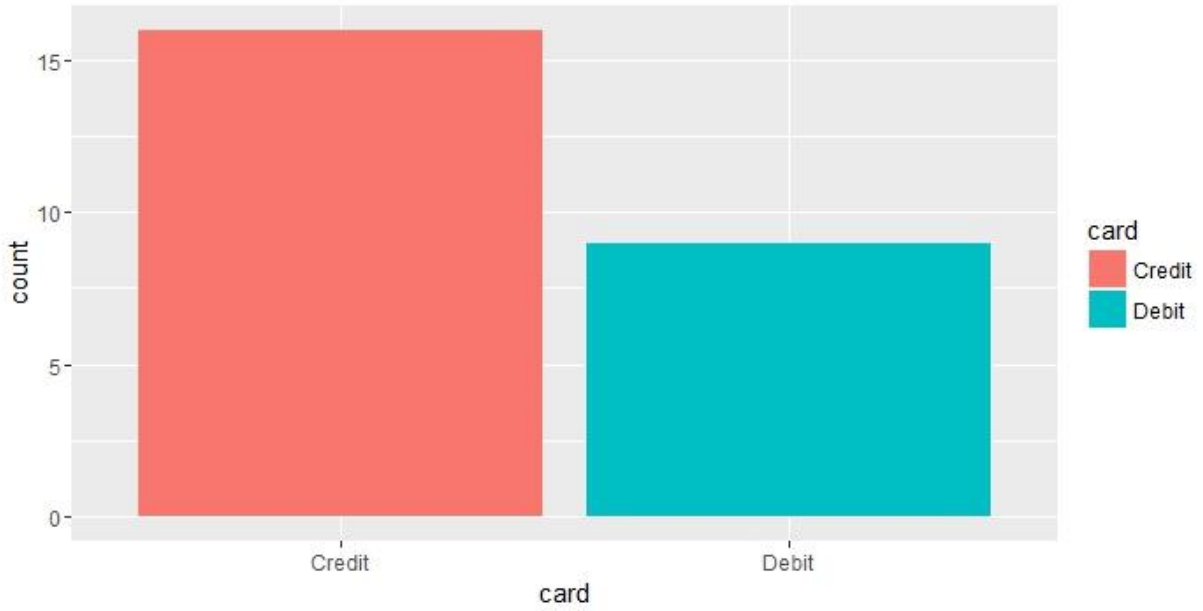


Figure: - Credit VS Debit (Payment Preference)

Example 4:- In this example, I am using SQL Query to count number of customer in each state from customer table and then visualize by plotting graph.

```
ts=dbSendQuery(mydb, "SELECT state, COUNT(state) AS count FROM Customer GROUP
 BY state;")
data4=fetch(ts,n=-1)
ggplot(data = data4, aes(state, count,fill = state)) + geom_bar(stat="identit
y")
```

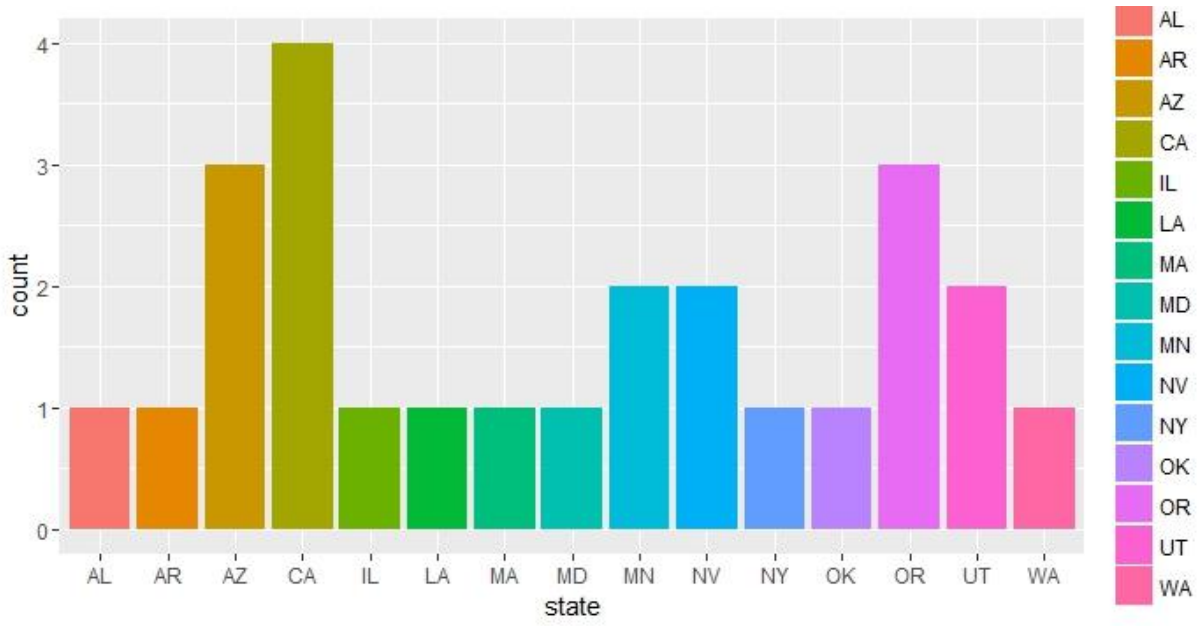


Figure:- Number of InstaBook Customers in each state