

A Study on  
Database For “ Mini-mart – Project record keeping System”



PROJECT REPORT



In partial fulfillment of the requirement of the 6th semester, **Bachelor of Business Management (BBM)** of Tribhuwan University.

NATIONAL COLLEGE

Lainchour, Kathmandu

June 2023

**Submitted by:**

Ichhya Singh Khadgi  
(Symbol:4664/19 ,  
reg.:7-2-366-10-2019)

**Submitted to:**

Office of the dean  
Faculty of Management  
Tribhuwan University

## **ACKNOWLEDGEMENT**

I, currently studying in the 6th semester of BBM National College have prepared the report on the topic of database design for “ **Mini-mart – Project record keeping System**” for any mini marts that works on projects as for the partial fulfillment of the course of bachelor in business management (BBM) of Tribhuvan\_University (TU) of Nepal.

Without help and support from our professor Mr. Bikul Koirala, I wouldn't have been able to work on the Mini–mart -Project record system. His constant support and encouragement is the main reason why this report has successfully made and presented.

This project has been the first-ever project I have ever made for the mini mart. This project work has been an encouraging and motivating factor. I would also like to thank my friends and family for their cooperation and support during this project.

## **DECLARATION**

I hereby declare that this project report entitled: "Mini-mart: Project Record Keeping System", submitted to the faculty of management Tribhuvan University comprises my original work under the supervision of Mr. Bikul Koirala, faculty member of National College, Lainchour-26 , Kathmandu. This report is submitted by me in the partial fulfillment of the requirements of project report for the award of BBM degree of Tribhuvan University. This project report has not been submitted to any other university or institution for the award of any degree or diploma.

Signature: .....

Name of the student: Ichhya Singh Khadgi

Exam symbol number: 4664/19

Date: 2080-04-32

## Table of Contents:

### Contents

1.INTRODUCTION:.....	5
2. PROJECT DESCRIPTION:.....	6
2.1 Database:.....	6
2.2 Database Management System:.....	6
2.2.1 Entity-Relational Model:.....	6
2.2.2.Object oriented Model: .....	7
2.2.4. Network Model: .....	8
2.2.5. Object Relational Data Model: .....	9
2.2.6. Relational Model:.....	9
2.3. Schema Diagram: .....	10
2.3.1. MySQL (Structured Query Language):.....	10
2.3.2. Data Definition Language (DDL): .....	11
2.3.3. Data Manipulation Language (DML): .....	11
2.3.4 Transaction Control Language (TCL):.....	12
2.3.5 Data Query Language (DQL): .....	12
3. PROJECT WORK: .....	13
3.1 Method to develop “ Mini-mart – Project record keeping System” .....	13
3.2 Algorithm:.....	13
3.3 Entities and Attributes: .....	14
3.4 Relationships and their description:.....	14
3.5 E-R diagram of "Mini-mart – Project record keeping System” .....	15
3.6 Schema diagram:.....	16
3.7 Structured Query Language: .....	17
3.7.1 DDL statements: .....	17
3.7.2. DML Statements:.....	19
4. SUMMARY .....	25
5. REFERENCES.....	26

# Chapter-1

## 1. INTRODUCTION:

This project is about how a mini mart system can keep a record of its products, customers and their corresponding project. In today's world, keeping everything on paper is time-consuming and costly. So, instead of using the old outdated method of keeping the mini mart system project record, this proposed system would be much more efficient and effective both in terms of cost and time. Using this system, a mart can store various details of its customers and products like name, phone\_number, address, duration, estimated\_cost, brand name etc.

There are so many database systems available in the market, but we have used MySQL because it provides a very easy user interface with fewer chances of making an error while recording data.

The mini mart-Project record keeping system is the place where owner's decision-makers can find customer and product details and the project on which they are working. Just with one click, all the required information can be accessed. Any mart system must have the latest demanding works that must be finished on time. It provides comprehensive details such as customer's names, quantity they purchased, product's name, price, stock etc. So by using these data, a mart owner can assign the particular work to only those who are capable and who have some work experience and using various MySQL Commands keep the track of the project and other details.

# Chapter-2

## 2. PROJECT DESCRIPTION:

### 2.1 Database:

A database is an organized collection of structured information, or data typically stored electronically in a computer system. More specifically, a database is an electronic system that allows data to be easily accessed, manipulated and updated. For example: mini mart system use database to store user, customer and product information, such as names, email, addresses, product name, details, purchase date and so on.

### 2.2 Database Management System:

A database management system (DBMS) is a software tool that enables users to manage a database easily. It enables users to access and interact with the underlying data in the database.

There are different types of database model which are as highlighted below:

1. Entity-Relational Model
2. Object oriented Model
3. Relational Model
4. Hierarchical Model
5. Network Model
6. Object Relational Data Model

#### 2.2.1 Entity-Relational Model:

An entity relational model is a graphical representation that depicts relationship among people, object, places, concepts or events within an information technology (IT) system. It describes interrelated things of interest in a specific domain of knowledge.

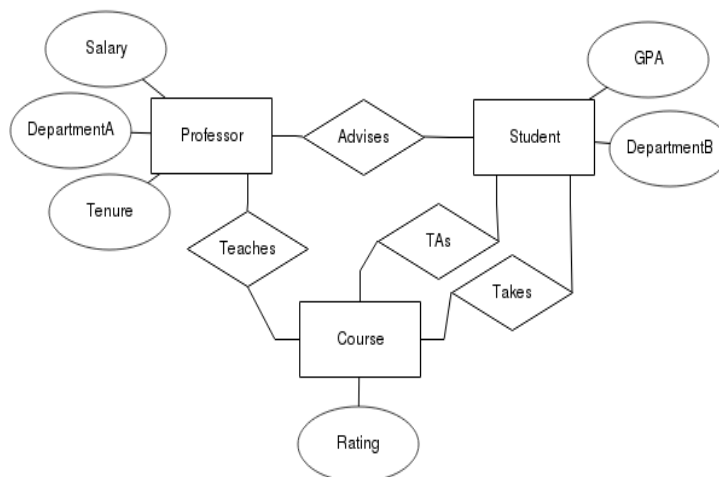


fig:entity relational model

It is used to represent the overall logical structure of the database.

#### i. Entity

ii. Relationship

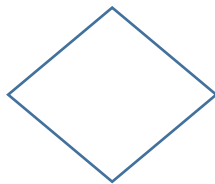
iii. Attributes

iv. Lines

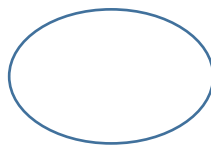
i. **Entity:** An entity is an object or a concept that is identified by the enterprise as having an independent existence. For example, each customer in a mart system is an entity. It is denoted by the Rectangle shape in the ER diagram.



ii. **Relationship:** Association between two or more entities is called a relationship. For example, an employee work on a project is an example of a relationship. It is denoted by a diamond shape.



iii. **Attributes:** Attributes are the information that explains the properties of an entity. For example, a customer entity can have customer id, customer\_name, product\_name, Phone\_number, etc. as attributes. It is denoted by an oval shape.



iv. **Lines:** Lines are used in making the connection between entity – relation, entity – attribute, relation – attribute, and relation – relation.



### 2.2.2. Object oriented Model:

A object oriented model is a database management system in which information is represented in the form of object as used in object oriented programming. It is organized among objects rather than actions, and data rather than logic.

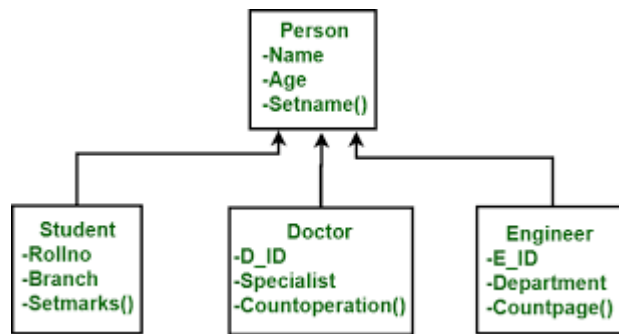


fig:object oriented model

### 2.2.3 Hierarchical Model:

A hierarchical database model is a data model in which the data are organized into a tree like structure. the datas are stored as records which are connected to one another through links.

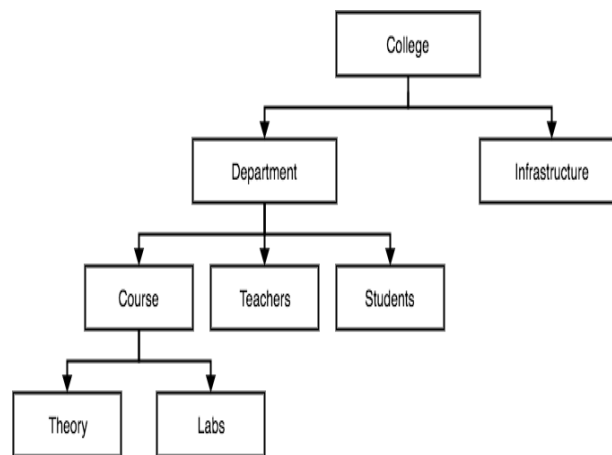


fig:hierarchical model

### 2.2.4. Network Model:

A network database is a type of database model wherein multiple member records or files can be linked to multiple owner files and vice versa.It is conceived as a flexible way of representing objects and their relationships.



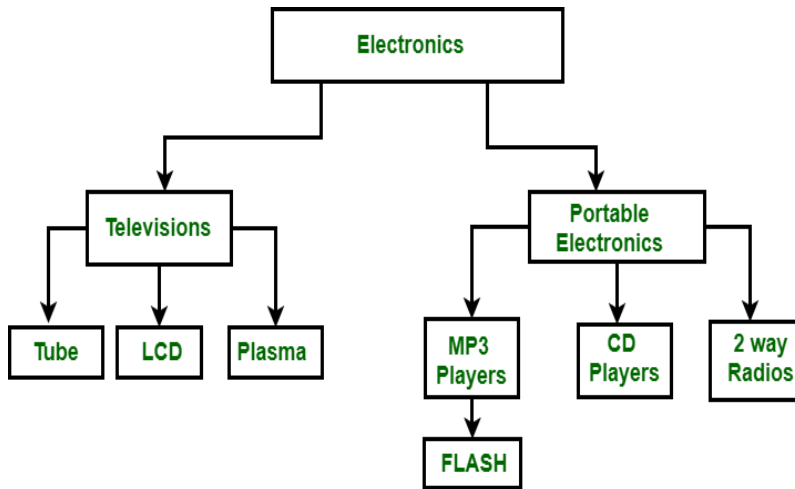


fig:network model

### 2.2.5. Object Relational Data Model:

An object relational model is a combination of a object oriented database model and a relational database model. It supports objects, classes, inheritance etc like object oriented models and has support for data types , tabular structures etc like relational data model.

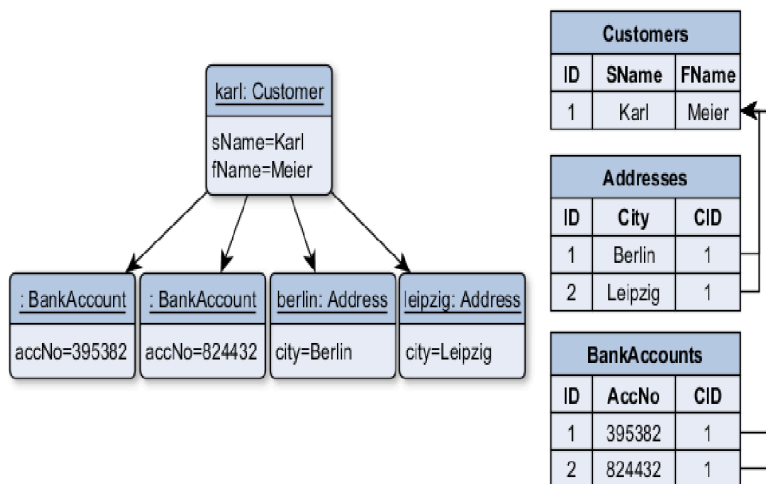


fig: object relational data model

### 2.2.6. Relational Model:

A relational database is a collection of information that organizes data in a predefined relationships where data is stored in one or more tables of columns and rows. It represents the database as a collection of relations.

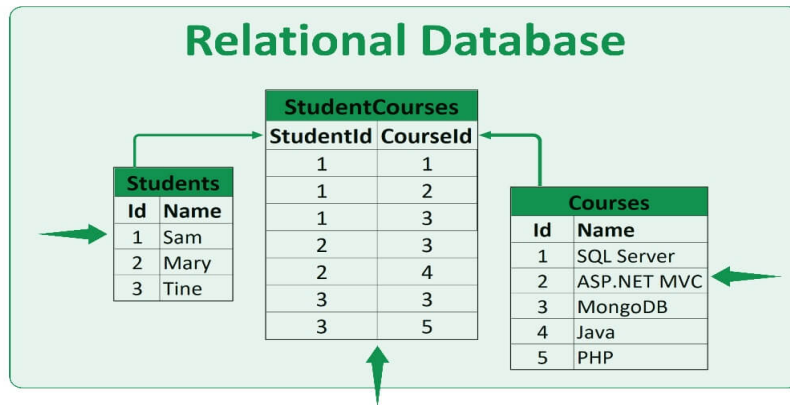


fig:relational database model

There are various languages used in relational database model. some of them are oracle, MYSQL and so on...

### 2.3. Schema Diagram:

There might be a various number of tables and it is important to visualize them. So, schema diagram in DBMS is a visual representation of various tables, their properties, and link among them. It shows how tables are linked to having common attributes or features.

Here we use MySQL among various relational database languages.

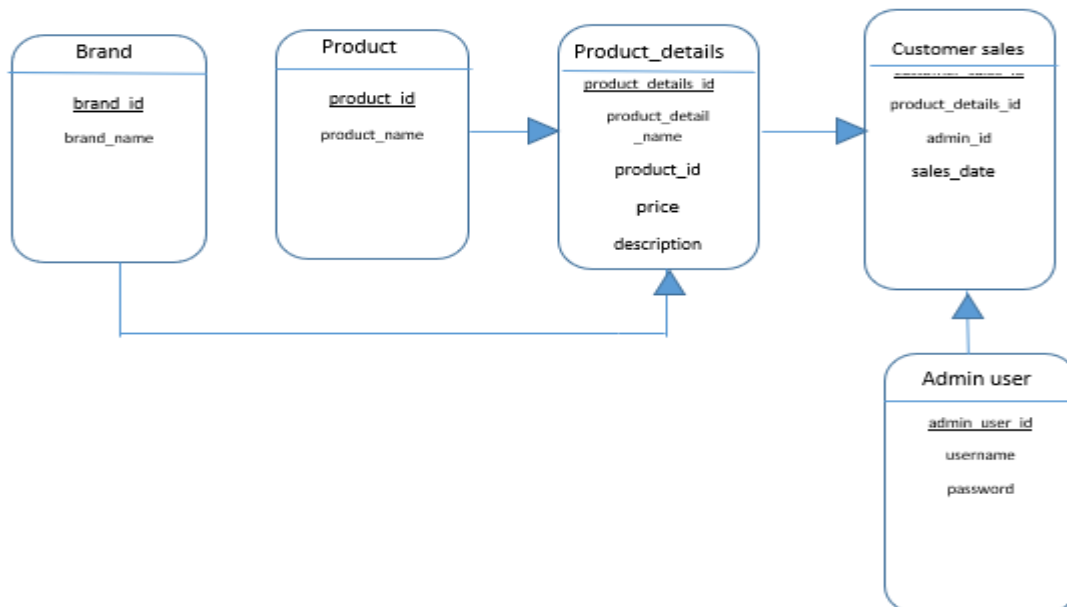


fig: schema diagram

#### 2.3.1. MySQL (Structured Query Language):

- MySQL is one of the most popular Database management systems.
- MySQL stores data in tables. Tables further store data in rows and columns.

- MySQL is a freely available Database system
- MySQL can run on multiple platforms like Linux, Windows, and Unix.
- MySQL is a highly scalable Database system.

### 2.3.2. Data Definition Language (DDL):

Data definition language is a part of SQL commands that helps to create, delete and modify tables. Along with tables, we can also define indexes, specify links between tables and impose some rules(constraints) between database tables. Some of the important DDL statements in SQL are:

- **CREATE statement:**

**Syntax:**

```
CREATE TABLE table name
(
    column1, datatype1, constraint1
    column2, datatype2, constraint2
    column3, datatype3, constraint3
    .....
);
```

- **ALTER statement:**

**Syntax:**

```
ALTER TABLE table name
CHANGE column_name;

ALTER TABLE table name
DROP column_name;

ALTER TABLE table_name
ADD column_name datatype constraint;
```

- **DROP statement:**

**Syntax:**

```
DROP TABLE table name;
```

### 2.3.3. Data Manipulation Language (DML):

Data manipulation language is used to insert, modify or delete data in the tables and columns. Only having a table is not enough, there must be some commands to play with data. Some of the DML statements are:

- **INSERT statement:**

**Syntax:**

```
INSERT into table_name
(col1, col2, col3)
VALUES
('Value1', 'Value2', 'Value3');
```

- **UPDATE statement:**

**Syntax:**

```
UPDATE table_name  
SET col1='value1',  
    col2='value2'  
where ID=(num);
```

- **DELETE statement:**

**Syntax:**

```
DELETE FROM table_name  
where ID=(num);
```

#### 2.3.4 Transaction Control Language (TCL):

Transaction control language or TCL commands deal with the transaction within the database. Some of the TCL commands are as follows,

- **COMMIT:**

**Syntax:**

```
COMMIT;
```

- **ROLLBACK:**

**Syntax:**

```
ROLLBACK;
```

#### 2.3.5 Data Query Language (DQL):

The SQL commands used to retrieve the data from the database are known as data query language. One such SQL command in data query language is the SELECT statement which is used as information retrieval from the database.

## chapter-3

### 3. PROJECT WORK:

#### 3.1 Method to develop “ Mini-mart – Project record keeping System”

In this report, we have used ER database model to design a database system as it is one of the best tools to visualize real-world concepts. By using this tool, we have successfully presented the major component of this “mini-mart – Project record keeping System”.

ER model is mapped into the relational model by representing ER database schema by a collection of relation schema. We have successfully identified all the common attributes most of the customer, products, brand, and admin have and presented them in the ER diagram. As some of the concepts of ER model have already been discussed in chapter 2, here we will show the actual example.

#### 3.2 Algorithm:

Before moving towards the ER diagram and schema diagram , we normalize our data so that it will help us to understand the relationship between various entities.

### NORMALIZATION:

#### 1NF:

product_name	brand_name	price	customer_name	date_of_sale	qty_sold	admin_user
toothpaste	colgate	50	Ram Tamang	2078-05-09	200	Sunil KC
shampoo	sunsilk	10	Sita Gurung	2079-09-18	100	Reema GC
biscuit	oreo	50	Ram Tamang	2079-09-10	20	Reema GC

Though we normalize data for the convenience but in 1NF ,there is high chance of data duplication. So, we move further towards 2NF.

#### 2NF:

##### Table: product

product_id	product_name
1	toothpaste
2	shampoo
3	oreo

##### Table:brand

brand_id	brand_name
1	colgate
2	sunsilk
3	oreo

### Table:customer

customer_id	customer_name
1	Ram Tamang
2	Sita Gurung
3	Raaz Tiwari

### Table:Admin

admin_id	admin_name
1	Reema GC
2	Sunil KC
3	Suraj Karki

After normalizing the data in 2NF, we got primary key in each of the above table so that in mart we don't have to enter the name of customer, product, brand, admin and so on each time they visit . So, after verifying the primary key, it will be easier for the mart to keep the record in an easy and efficient way and also the data consume less storage.

So, to keep the complete record the data is now further normalize into 3NF as in the table below:

### 3NF:

<u>S.N</u>	<u>product_id</u>	<u>brand_id</u>	<u>customer_id</u>	<u>admin_id</u>	<u>price</u>	<u>date_of_sold</u>	<u>qty_sold</u>
1	2	1	3	2	20	2078-09-13	200
2	1	3	2	2	50	2079-08-12	100
3	3	2	1	1	100	2080-01-14	40

Therefore, now 3NF helps the employee to keep the record of customer, brand, product etc in an easy and effective way. It is convenient for both mart and employees.

### 3.3 Entities and Attributes:

- **Brand** entity has attributes: (brand\_name, brand\_id)
- **Admin user** entity has attributes: (admin\_user\_id, username, password)
- **Product** entity has attributes: (product\_id, product\_name)
- **Product details** entity has attributes: (product\_details\_id, product\_details\_name, price, description , product\_id)
- **Customer sales** entity has attributes: (customer\_sales\_id, product\_details\_id, admin\_id, sales\_date)

### 3.4 Relationships and their description:

- Brand has product details(many to many relationship)
- Product has product details(many to many relationship)
- Admin user has customer sales(many to many relationship)
- Product details has customer sales(many to many relationship)

### 3.5 E-R diagram of "Mini-mart – Project record keeping System"

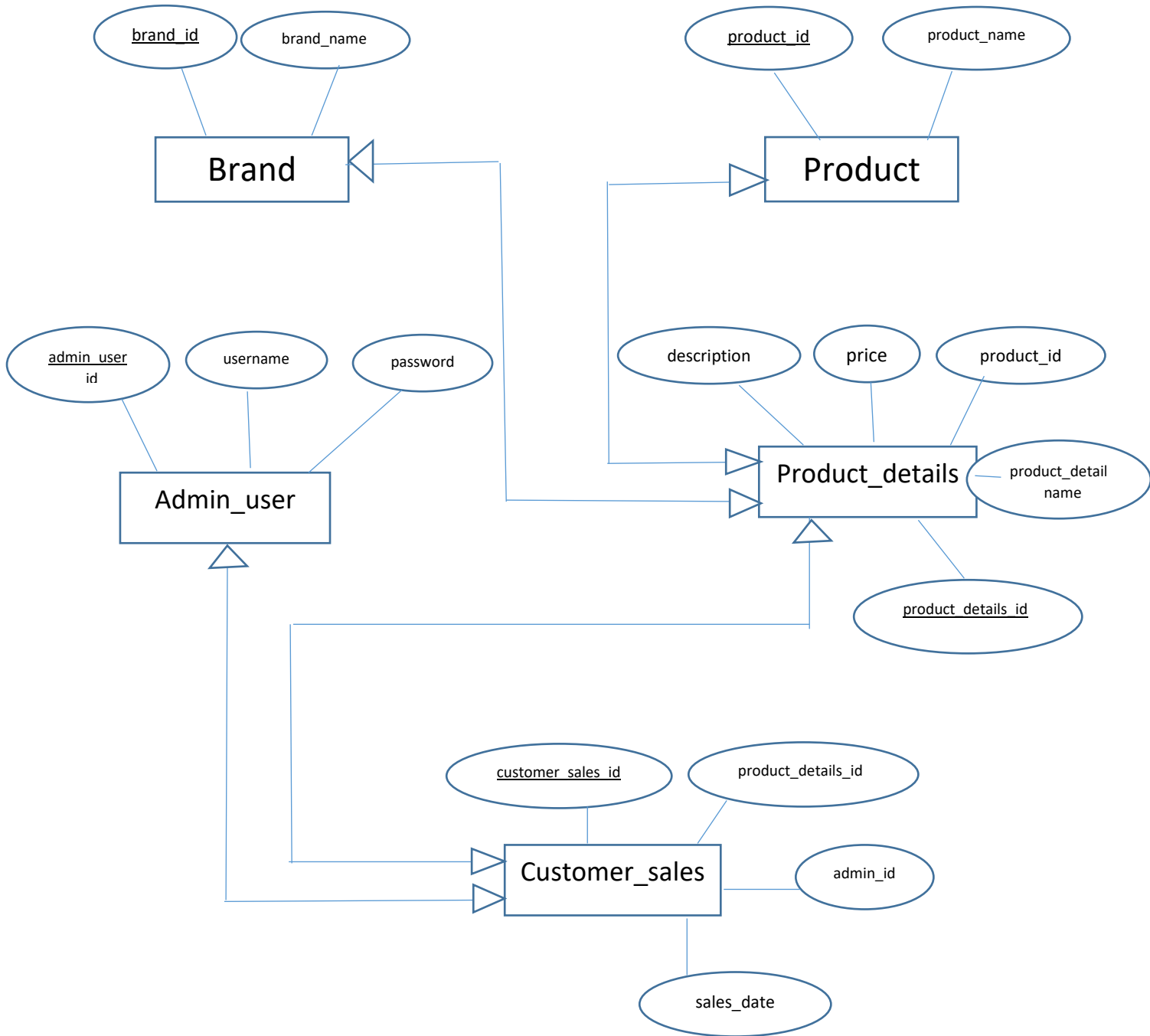
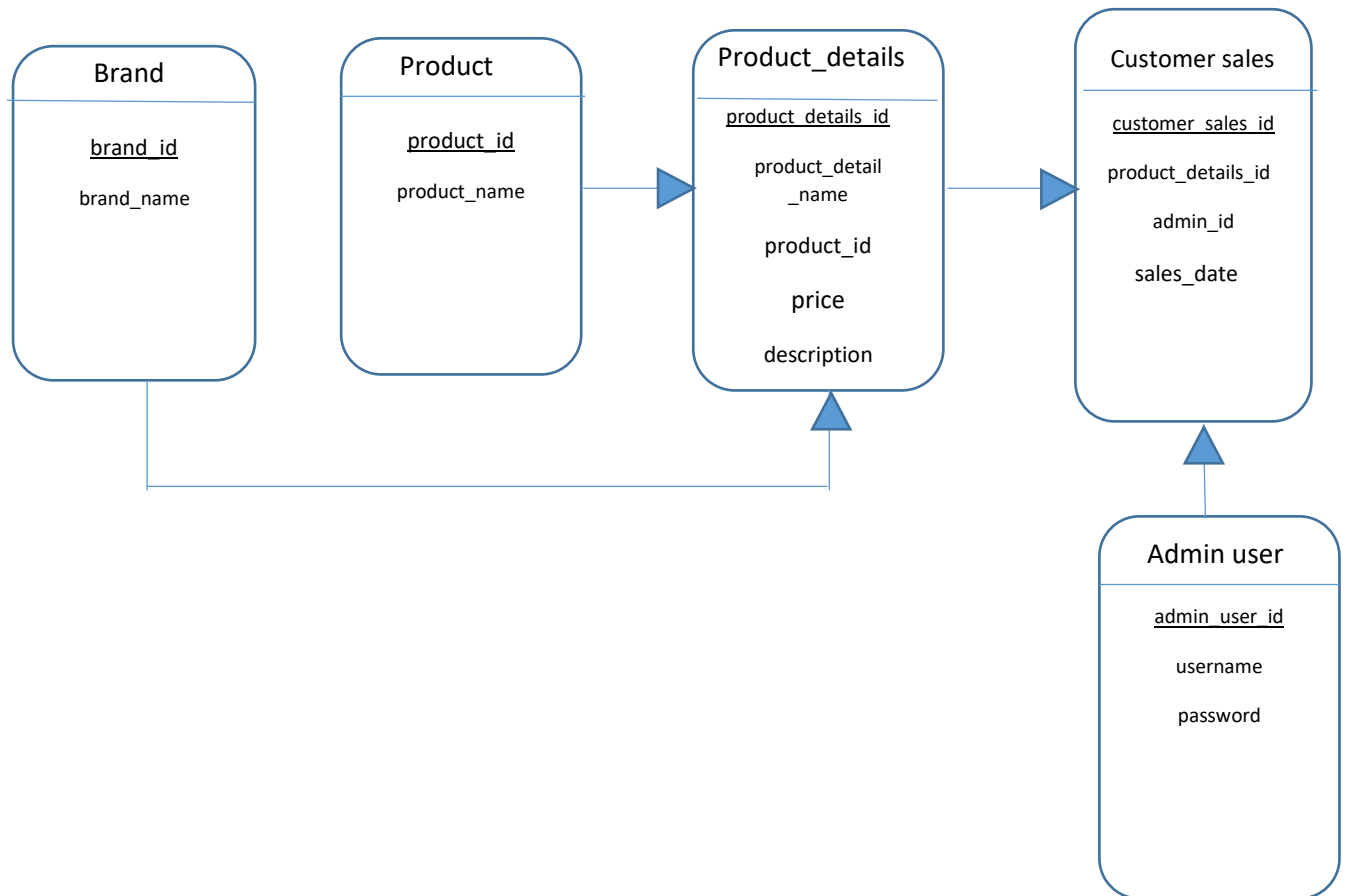


FIG: ER Model

### 3.6 Schema diagram:

The schema diagram of "Mini-mart – Project record keeping System" is presented as below:





### 3.7 Structured Query Language:

Various SQL statements and their corresponding operations are as mentioned below:

#### 3.7.1 DDL statements:

##### 3.7.1.1. CREATE STATEMENTS:

- Brand table creation command:

```
CREATE TABLE Brand
(
    brand_id int primary key AUTO INCREMENT,
    brand_name varchar(100) not null
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 brand_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 brand_name	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More

Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Add to central columns    Remove from central columns

- Product table creation command:

```
CREATE TABLE Product
(
    product_id int primary key AUTO INCREMENT,
    product_name varchar(100) not null
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 product_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 product_name	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More

Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Add to central columns    Remove from central columns

- Product\_details table creation command:

```
CREATE TABLE Product_details
(
    product_details_id int primary key AUTO INCREMENT,
```

```

product_details_name varchar(100) not null,
description varchar(100) not null,
price float(10,2) not null,
FOREIGN KEY ( product_id) REFERENCES Product( product_id)
);

```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 product_details_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 brand_id	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	3 product_id	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	4 description	text	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5 price	float(10,2)			No	None			Change  Drop  More

Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext  
 Add to central columns    Remove from central columns

- Admin\_user table creation command:

```

CREATE TABLE admin_user
(
admin_user_id int primary key AUTO INCREMENT,
username varchar(100) not null,
password varchar(100) not null
);

```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 admin_user_id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 username	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 pass_word	varchar(100)	utf8mb4_general_ci		No	None			Change  Drop  More

Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext  
 Add to central columns    Remove from central columns

- Customer\_sales table creation command:

```

CREATE TABLE customer_sales
(
customer_sales_id int primary key AUTO INCREMENT,
sales_date date not null,
FOREIGN KEY ( product_details_id) REFERENCES Product_details(
product_details_id)
);

```

FOREIGN KEY ( admin\_user\_id) REFERENCES Admin\_user( admin\_user\_id  
);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	customer_sale_id	int(11)		No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2	product_details_id	int(11)		No	None			Change  Drop  More
<input type="checkbox"/>	3	sales_date	timestamp		No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change  Drop  More
<input type="checkbox"/>	4	admin_id	int(11)		No	None			Change  Drop  More

Check all    With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

### 3.7.2. DML Statements:

#### 3.7.2.1. INSERT STATEMENTS:

- Admin user table insert command:

```
INSERT INTO `tbl_admin_user`(`admin_user_id`, `username`, `pass_word`) VALUES ('[value-1]','[value-2]','[value-3]')
```

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

```
SELECT * FROM `tbl_admin_user`
```

Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	admin_user_id	username	pass_word
<input type="checkbox"/>	1	reema	1db0f2eb32f4f462189b6c35a36bab71
<input type="checkbox"/>	2	riya	5ee907831fc55cc73ba2ff954aef357

Check all    With selected: Edit Copy Delete Export

- brand table insert command:

```
INSERT INTO `tbl_brand`(`brand_id`, `brand_name`) VALUES ('[value-1]','[value-2]')
```

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT * FROM `tbl_brand`
```

Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	brand_id	brand_name
<input type="checkbox"/>	1	CG
<input type="checkbox"/>	2	Colgate
<input type="checkbox"/>	3	DDC

Check all    With selected: Edit Copy Delete Export

- product table insert command:

```
INSERT INTO `tbl_product`(`product_id`, `product_name`) VALUES ('[value-1]','[value-2]')
```

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT * FROM `tbl_product`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_id	product_name
<input type="checkbox"/>	1	Biscuit
<input type="checkbox"/>	2	Toothpaste

Check all | With selected: Edit Copy Delete Export

- Product details table insert command:

```
INSERT INTO `tbl_product_details`(`product_details_id`, `brand_id`, `product_id`, `description`, `price`) VALUES ('[value-1]', '[value-2]', '[value-3]', '[value-4]', '[value-5]')
```

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM `tbl_product_details`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_details_id	brand_id	product_id	description	price
<input type="checkbox"/>	1	2	2	give us fresh air	200.00
<input type="checkbox"/>	2	3	1	helpful for infants	20.00

Check all | With selected: Edit Copy Delete Export

- Customer sales table insert command:

```
INSERT INTO `tbl_customer_sales`(`customer_sale_id`, `product_details_id`, `sales_date`, `admin_id`) VALUES ('[value-1]', '[value-2]', '[value-3]', '[value-4]')
```

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM `tbl_customer_sales`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	customer_sale_id	product_details_id	sales_date	admin_id
<input type="checkbox"/>	1	1	2023-06-05 07:30:58	1
<input type="checkbox"/>	2	2	2023-06-05 07:30:58	2

Click the drop-down arrow to toggle column's visibility.

Check all | With selected: Edit Copy Delete Export

### 3.7.2.2. UPDATE STATEMENTS:

- Admin user table update command:**

```
UPDATE `tbl_admin_user` SET `admin_user_id`=[value-1],`username`=[value-2],`pass_word`=[value-3] WHERE 1
```

### before updating:

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

```
SELECT * FROM `tbl_admin_user`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	admin_user_id	username	pass_word
<input type="checkbox"/> Edit Copy Delete	1	reema	1db0f2eb32f4f462189b6c35a36bab71
<input type="checkbox"/> Edit Copy Delete	2	riya	5ee907831fc55cc73ba2ff954aef357

↑  Check all | With selected: Edit Copy Delete Export

### after updating:

Showing rows 0 - 1 (2 total, Query took 0.0003 seconds.)

```
SELECT * FROM `tbl_admin_user`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

```
UPDATE `tbl_admin_user` SET `username` = 'RESMA' WHERE `tbl_admin_user`.`admin_user_id` = 1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	admin_user_id	username	pass_word
<input type="checkbox"/> Edit Copy Delete	1	RESMA	1db0f2eb32f4f462189b6c35a36bab71
<input type="checkbox"/> Edit Copy Delete	2	riya	5ee907831fc55cc73ba2ff954aef357

↑  Check all | With selected: Edit Copy Delete Export

- **Brand table update command:**

```
UPDATE `tbl_brand` SET `brand_id`=[value-1],`brand_name`=[value-2] WHERE 1
```

### before updating:

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT * FROM `tbl_brand`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	brand_id	brand_name
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	CG
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Colgate
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	DDC

Check all | With selected: Edit Copy Delete Export

after updating:

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM `tbl_brand`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

```
UPDATE `tbl_brand` SET `brand_name` = 'LG' WHERE `tbl_brand`.`brand_id` = 1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	brand_id	brand_name
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	LG
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Colgate
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	DDC

Check all | With selected: Edit Copy Delete Export

- **Product table update command:**

```
UPDATE `tbl_product` SET `product_id`='[value-1]',`product_name`='[value-2]' WHERE 1
```

before updating:

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT * FROM `tbl_product`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_id	product_name
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Biscuit
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	Toothpaste

Check all | With selected: Edit Copy Delete Export

after updating:

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM `tbl_product`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

```
UPDATE `tbl_product` SET `product_name` = 'shampoo' WHERE `tbl_product`.`product_id` = 1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_id	product_name
<input type="checkbox"/> Edit Copy Delete	1	shampoo
<input type="checkbox"/> Edit Copy Delete	2	Toothpaste

Check all | With selected: Edit Copy Delete Export

- **Product details table update command:**

```
UPDATE `tbl_product_details` SET `product_details_id`='[value-1]',`brand_id`='[value-2]',`product_id`='[value-3]',`description`='[value-4]',`price`='[value-5]' WHERE 1
```

before updating:

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM `tbl_product_details`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_details_id	brand_id	product_id	description	price
<input type="checkbox"/> Edit Copy Delete	1	2	2	give us fresh air	200.00
<input type="checkbox"/> Edit Copy Delete	2	3	1	helpful for infants	20.00

Check all | With selected: Edit Copy Delete Export

after updating:

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM `tbl_product_details`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

```
UPDATE `tbl_product_details` SET `brand_id` = '1' WHERE `tbl_product_details`.`product_details_id` = 1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	product_details_id	brand_id	product_id	description	price
<input type="checkbox"/> Edit Copy Delete	1	1	2	give us fresh air	200.00
<input type="checkbox"/> Edit Copy Delete	2	3	1	helpful for infants	20.00

Check all | With selected: Edit Copy Delete Export

- **customer sales table update command:**

```
UPDATE `tbl_customer_sales` SET `customer_sale_id`='[value-1]',`product_details_id`='[value-2]',`sales_date`='[value-3]',`admin_id`='[value-4]' WHERE 1
```

**before updating:**

Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

```
SELECT * FROM `tbl_customer_sales`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	customer_sale_id	product_details_id	sales_date	admin_id
<input type="checkbox"/> Delete	1	1	2023-06-05 07:30:58	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	2023-06-05 07:30:58	2

Check all | With selected:  Edit  Copy  Delete  Export

**after updating:**

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM `tbl_customer_sales`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

```
UPDATE `tbl_customer_sales` SET `admin_id` = '2' WHERE `tbl_customer_sales`.`customer_sale_id` = 1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	customer_sale_id	product_details_id	sales_date	admin_id
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	2023-06-05 07:30:58	2
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	2023-06-05 07:30:58	2

Check all | With selected:  Edit  Copy  Delete  Export



## CHAPTER- 4

### 4. SUMMARY

Every marts undertakes various works which are very important for survival in tough market competition. Before undertaking any work, the mart owner must assess its capable employees. Assigning a particular work to a capable person is very important because it can save both time and resources.

Before jumping into the field, employees must be analyzed and assessed based upon various dimensions. If decision-makers have to open the drawer to find the details of each brands , products, customers,etc., it will take more time and it is quite undesirable. So if all the required information is collected in one place, it will make easy access of data whenever required.

Our "**Mini-mart – Project record keeping System**" is the latest database management system that uses MySQL to manage data. Using various SQL commands users can easily retrieve data. This system is equipped with the latest security mechanism that allows to grant and revoke access to the data. Even though this system saves time and energy, but its cost is quite high. Small marts can't afford to use this system.

## CHAPTER- 5

### 5. REFERENCES

- **GOOGLE:**

[https://www.google.com/search?q=database&oq=data&gs\\_lcrp=EgZjaHJvbWUqCggCEAAYsQMYgAQyBggAEEUYOTIGCAEQRRhAMgoIAhAAGLEDGIAEMgoIAxAGLEDGIAEMg0IBBAAGIMBGLEDGIAEMgoIBRAAGLEDGIAEMgclBhAAGIAEMgoIBxAGLEDGIAE0gEINjU2OGowajeoAgCwAgA&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=database&oq=data&gs_lcrp=EgZjaHJvbWUqCggCEAAYsQMYgAQyBggAEEUYOTIGCAEQRRhAMgoIAhAAGLEDGIAEMgoIAxAGLEDGIAEMg0IBBAAGIMBGLEDGIAEMgoIBRAAGLEDGIAEMgclBhAAGIAEMgoIBxAGLEDGIAE0gEINjU2OGowajeoAgCwAgA&sourceid=chrome&ie=UTF-8)

- **Draw.io:**

draw.io is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams.

- **XAMPP:**

XAMPP is a cross-platform web server that is free and open-source. XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl. XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver.