Symbiosis Institute of Technology

A DBMS Project Report on

Blue Dart Express

(Parcel Delivery Database Management)

Submitted by

Hridyesh Singh Bisht

(18070124030)

Kavya Suthar

(18070124037)

Sejal Shrestha

(18070124064)

Under the Guidance of

Prof. Shruti Patil

Department of Computer Science

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Index

1.Introduction:

We decided to develop a courier delivery management system, very similar to Bluedart. As due to COVID pandemic, people have relied heavily on courier services from sending an important parcel from one customer to another, to buying basic amenities from e-commerce sites.

Our solution has a few new and better functionalities, such as tracking your parcel, getting information about the employee from the branch responsible for picking or delivering your parcel to you.

2.Problem Statement:

Developing a database management system for a courier delivery management system with the functionality of entering data into a form, adding or removing tables, finding and replacing data, and running queries,

From picking up the parcel from the customer to delivering the parcel to the branch, transferring parcel from one branch to another branch, and then finally delivering the parcel to the receiving customer.

Modules:

A.Company Module:

1. Company Login Module: This enables the company administrator to create, read, update, delete or make any adjustments to the Database of the system.
2. Add Branch from Company Module: This enables the company administrator to add a new branch in the database.
3. Add Employee from Company Module: The add employee module enables the company administrator to add employee information in a branch if not present.

B.Customer Module:

1. Customer Parcel Pickup Module: The Customer delivery module helps the customer to request the branch for a parcel transfer.
2. Add Payment from Customer Module: The add payment module enables the customer to add payment information about the parcel if not present.
3. Customer Parcel delivery Module: The tracking module enables the Customer to receive the parcel.
4. Add Parcel from Customer Module: To add information about the parcel, such as sending customer address, receiving customer address, weight.

C.Employee Module:

1. Add Parcel from Employee Module: The add parcel from module enables the branch administrator to add parcel information that is not present in the database.
2. Can Track the parcel from Employee Module: branch administrator can check the current whereabouts of a particular parcel.
3. Update Transfer Module: The update transfer module, let's the branch administrator update the information of the employee picking up and delivering the parcels.
4. Responsible for Transferring the parcel from customer to branch and vice versa.

Functional Requirements:

Functional Requirement for the Solution describes different activities involved in the program modules,

1.Customer:

1. Fill information about the sending customer, receiving customer and parcel.
2. Make payment to the transfer for the parcel delivery.
3. Request a parcel pickup from the transfer.
4. Parcel gets delivered by the transfer.

2.Transfer:

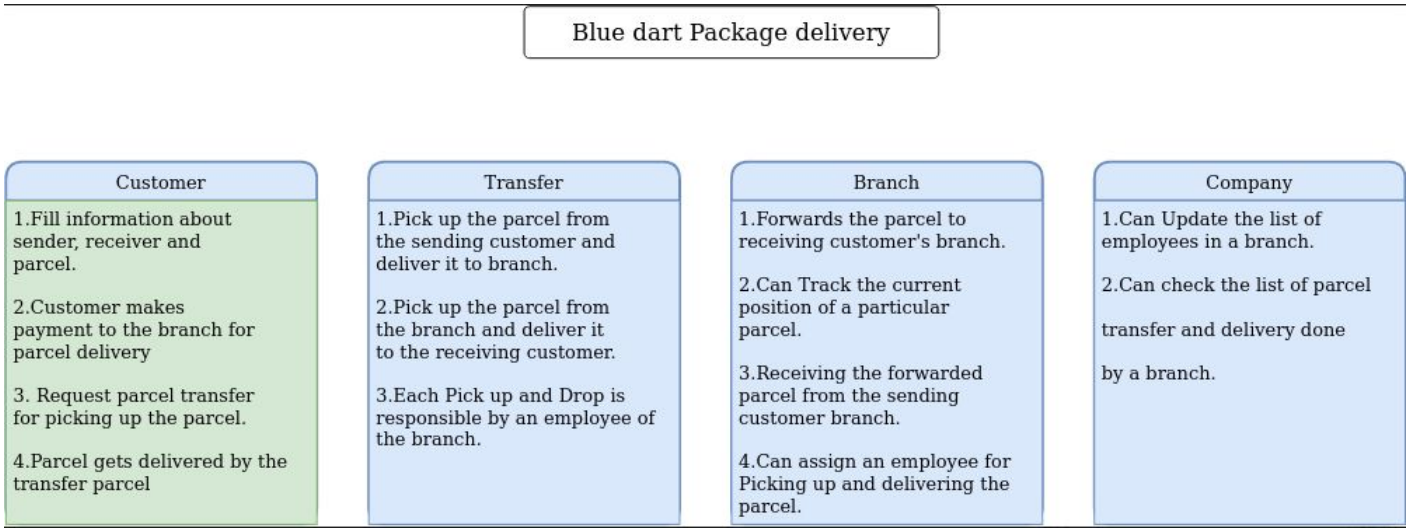
1. Pick up the parcel from a customer and deliver it to the branch.
2. Pick up the parcel from the branch and deliver it to a customer.
3. Each pickup/delivery is responsible by an employee from the branch.
4. Accepting payment information.

3.Branch:

1. Forward the parcel to the receiving customer's branch.
2. Can track the current location of the parcel.
3. Receive the parcel from the sending customer's branch.
4. Can assign an employee a designation.

4.Company:

1. Can update the list of employees in a branch.
2. Can check the list of transfer of parcels done by a branch.



3. Find out entities, attributes and their relationships.

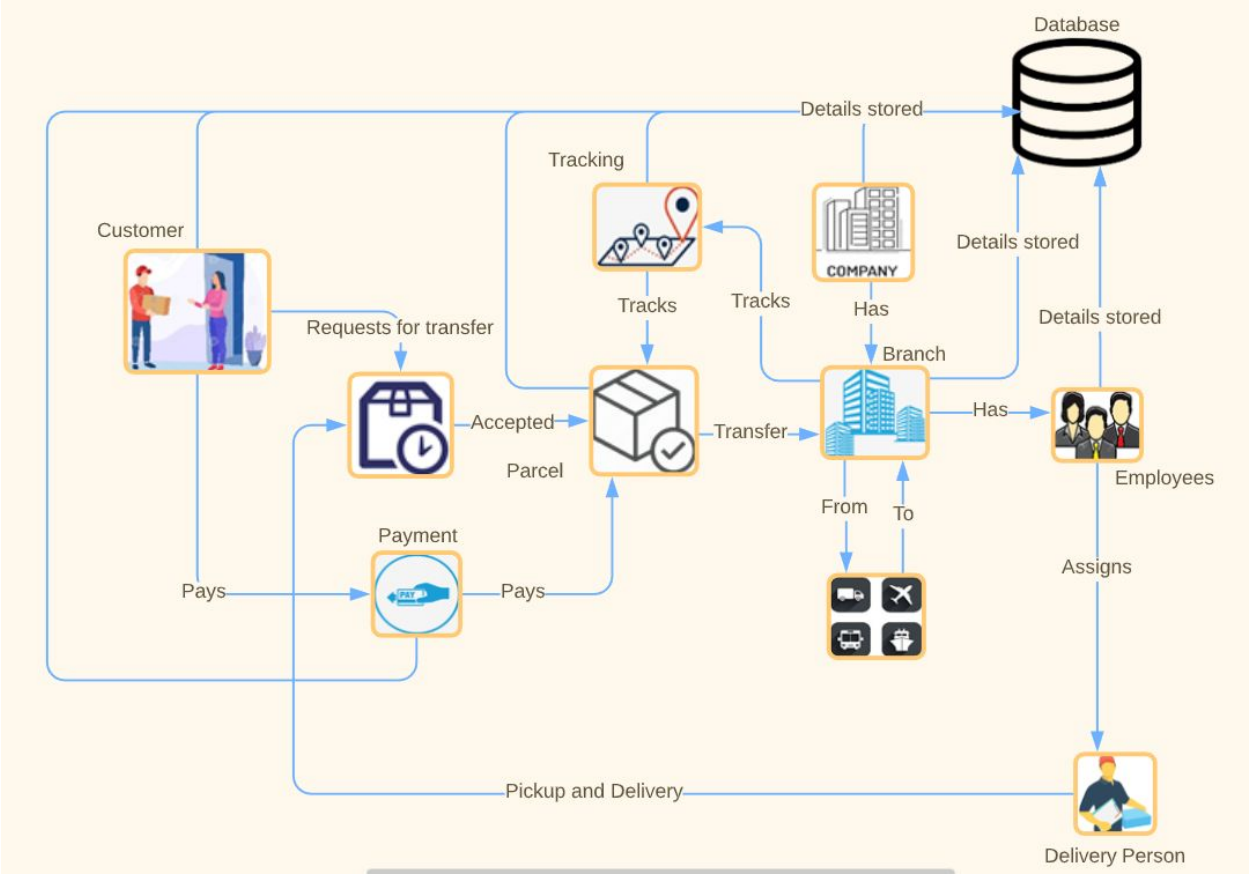
Entities and Attributes

1. Customer: CustomerID, Name, Address, Email, Phone Number, Zip Code.
2. Zipcode: ZipcodeID, City, State.
3. Payment: PaymentID, Payment verification, Mode of paymentID, Mode of deliveryID, Discount.
4. Mode of Payment: ModeofPaymentID, Mode of payment type.
5. Mode of Delivery: ModeofDeliveryID, Mode of delivery type.
6. Parcel: ParcelID, Weight, From address, To address.
7. Transfer: TransferID, Date, Signature, Pickup or Drop, Vehicle Number.
8. Branch: BranchID, Name, Email, Address, Phone number, Zip Code, Vehicles.
9. Employee: EmployeeID, Email, Name, Address, Phone Number, DesignationID.
10. Designation: DesignationID, Designation type.
11. Company: CompanyID, Name, Address, Email, Phone Number.
12. Tracking: TrackingID, Date of sending, Date of Receiving, Current Transfer, Sending branchid, receiving branchid.

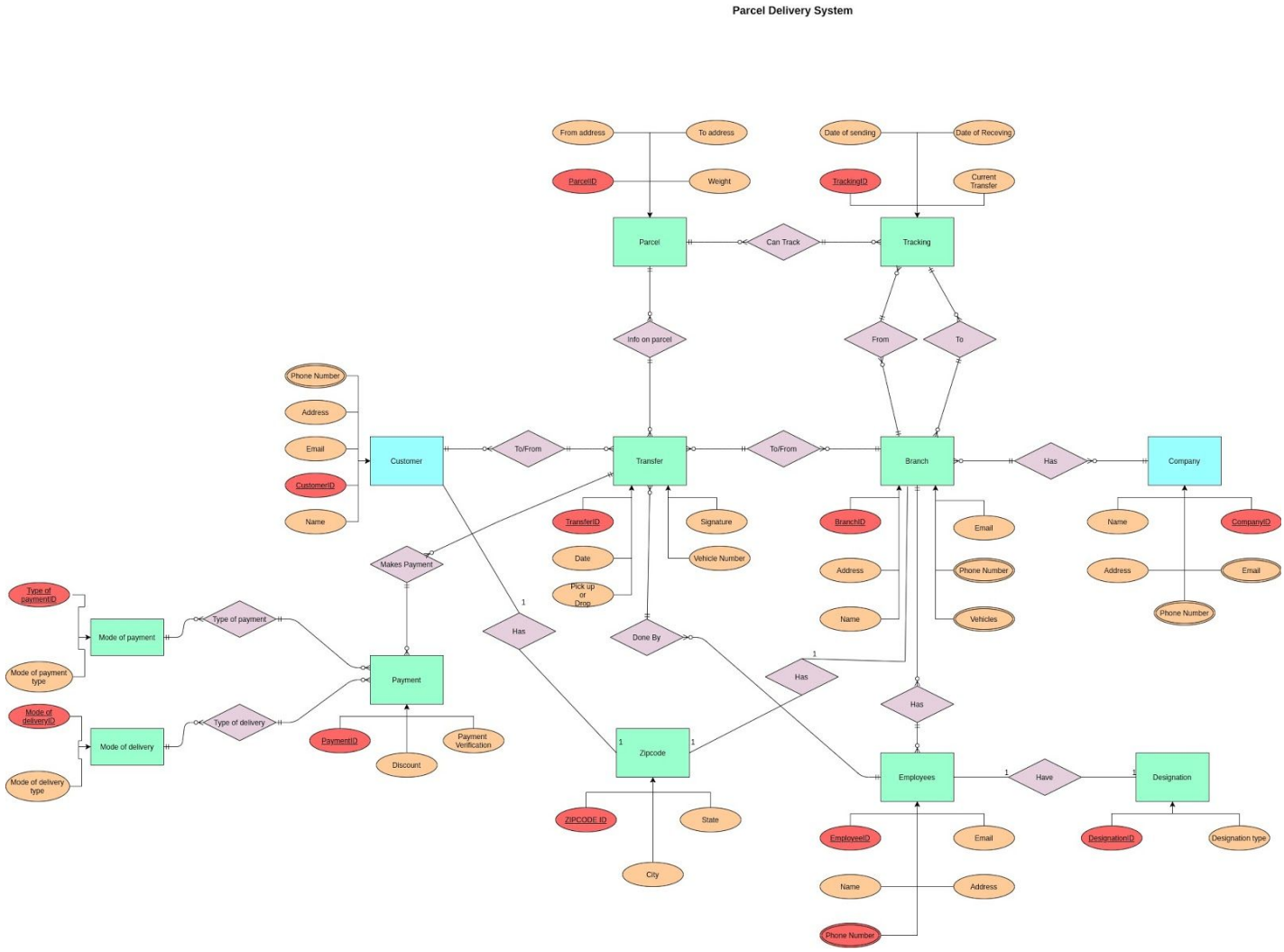
Relationships

Entity	Relationship	Entity
Customer	Has zipcode	Zipcode
Customer	Gives parcel to	Transfer
Customer	Can make payment to	Transfer
Transfer	Has information about	Parcel
Transfer	Records payment	Payment
Payment	Type of payment	Mode of payment
Payment	Type of delivery	Mode of delivery
Transfer	Parcel is taken care by	Employees
Transfer	Gives parcel to	Branch
Branch	Has	Employees
Employees	Have	Designations
Brach	Has	Zipcode
Branch	Can track the transfer parcel from one branch to another	Tracking
Tracking	Can track	Parcel
Branch	Sends parcel to another branch	Branch
Branch	Gives parcel to Transfer	Transfer
Transfer	Parcel is taken care by	Employees
Transfer	Gives parcel to	Customer
Company	Has	Branches

4. System Architecture:



5. Prepare an E-R (or EER if required) diagram.



6. Convert E-R diagram into a relational schema.

1. **Customer:**

CustomerID	Name	Address	E-mail	ZipcodeID

2. Customer_phonenumber:

CustomerID	Phone Number

3. Parcel:

ParcelID	Weight	From Address	To Address

4. Payment:

PaymentID	TransferID	Discount	Payment Verification	TypeOfPaymentID	TypeOfDeliveryID

5. Mode of Payment:

TypeOfPaymentID	Mode of payment type

6. Mode of Delivery:

TypeOfDeliveryID	Mode of delivery type

7. Company:

CompanyID	Name	Address name

8. Company_phonenumber:

CompanyID	Phone Number

9. Company_email:

CompanyID	Email

10. Zipcode:

ZipcodeID	State	City

11. Branch:

BranchID	Name	Address	Email	CompanyID	ZipcodeID

12. Branch_phonenumber:

BranchID	Phone Number

13. Branch_vehicles:

BranchID	Vehicles

14.Employees

EmployeeID	Name	Address	Email	DesignationID	BranchID

15. Employees_phonenumber

EmployeeID	Phone Number

16. Designation:

DesignationID	Designation Type

17. Transfer:

TransferID	Signature	Date	VehicleNumber	Pickup or Drop	CustomerID	ParcelID	EmployeeID

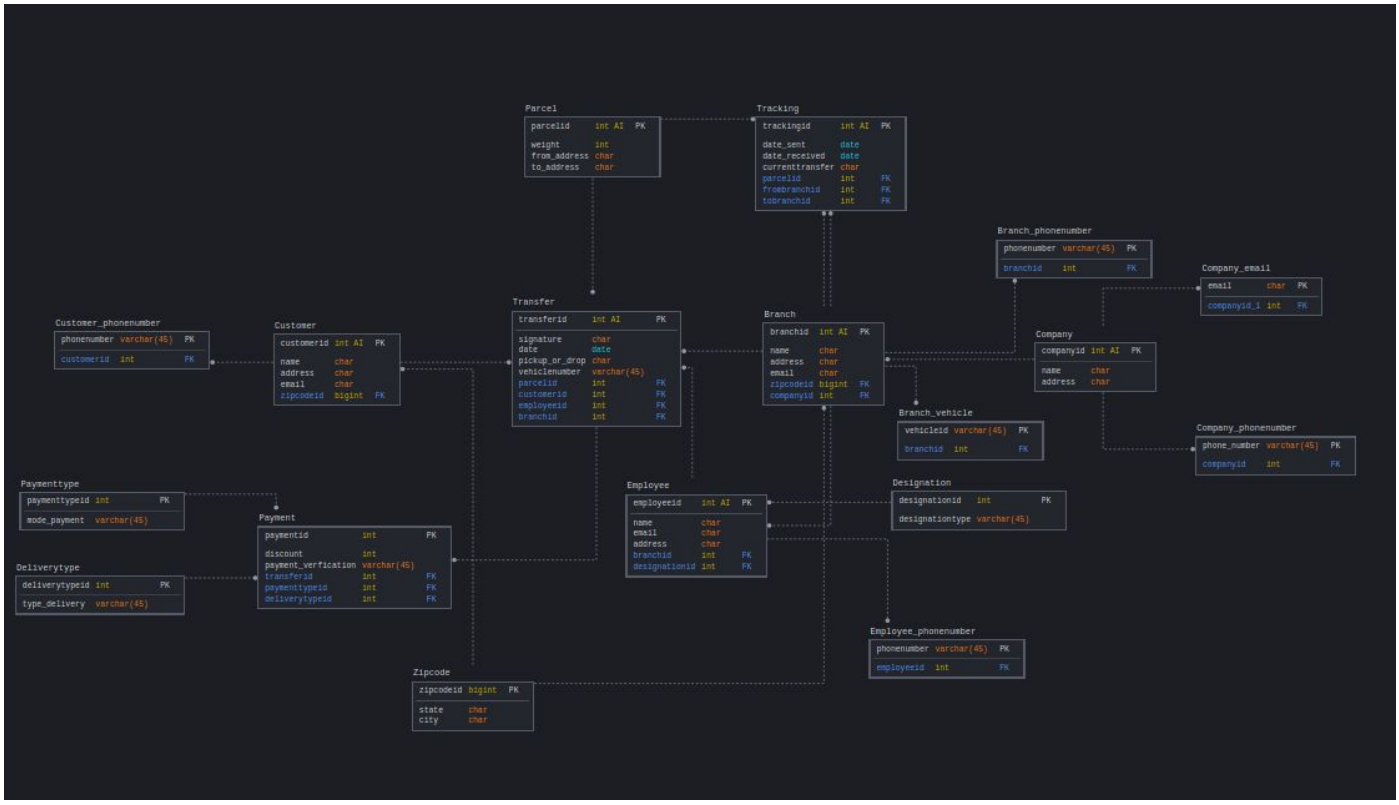
18. Tracking :

TrackingID	BranchID	Date of Sending	Current Transfer	Date of Receiving	ParcelID	Tobranchid	Frombranchid

7. Mention all the keys (primary, foreign key, candidate keys, alternate key) of each relation.

Table	Primary key	Foreign keys	Candidate keys	Alternate keys
Customer	CustomerID	1.ZipcodeID	1.CustomerID 2.Email	1.Email
Zipcode	ZipcodeID	Nil	1.ZipcodeID 2.State	1.State
Transfer	TransferID	1.CustomerID 2.ParcelID 3.BranchID 4.EmployeeID	1.TransferID 2.Signature 3.Date	1.Signature 2.Date
Payment	PaymentID	1.TransferID 2.PaymenttypeID 3.DeliverytypeID	1.PaymentID 2.PaymentVerification	1.PaymentVerification

Mode of payment	TypeofpaymentID	Nil	1.TypeofpaymentID	Nil
Mode of delivery	TypeofdeliveryID	Nil	1.TypeofdeliveryID	Nil
Parcel	ParcelID	Nil	1.ParcelD 2.From address 3.To address	1.From address 2.To address
Branch	BranchID	1.CompanyID 2.ZipcodeID	1.BranchID 2.Name 3.Email 4.Phone Number 5.Address	1.Name 2.Email 3.Address 4.Phone Number
Employee	EmployeeID	1.DesignationID 2.BranchID	1.EmployeeID 2.Name 3.Email 4.Phone Number 5.Address	1.Name 2.Email 3.Address 4.Phone Number
Designation	DesignationID	Nil	1.DesignationID	Nil
Company	CompanyID	Nil	1.CompanyID 2.Name 3.Email 4.Phone Number 5.Address	1.Name 2.Email 3.Phone Number 4.Address
Tracking	TrackingID	1.FrombranchID 2.TobranchID 3.ParcelID	1.TrackingID 2.Date of sending 3.Date of Receiving 4.Current transfer	1.Current transfer 2.Date of sending 3.Date of Receiving



8. Explain how the Codd's rules are applied for the project.

1.THE INFORMATION RULE:

Everything in a database must be stored in a table format, this rule indicates that every piece of data that we store in a database should be stored in a table. As we are using MySQL Server which is a relational database management system, all information is stored in a table.

2.THE GUARANTEED ACCESS RULE:

All data must be accessible, this rule states that all the data should be accessible and also states the importance of primary keys for locating data in a database. Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value).

In MySQL, we can search for the primary key value, and once we have the row, the data is accessed via the column name. We can also access data by any of the columns in the table, the values returned can be singular or multiple.

3.SYSTEMATIC TREATMENT OF NULL VALUES:

The DBMS allows each field to remain null (or empty). Specifically, it must support a representation of "missing information and inapplicable information", distinct from all regular values, and independent of the data type. This is a very important rule because a NULL can be interpreted as one the following – data is missing, data is not known, or data is not applicable.

This rule requires that the RDBMS supports a distinct NULL placeholder, regardless of the data type. NULLs must propagate through mathematical operations as well as string operations.

MySQL provides several useful functions that handle NULL effectively such as IFNULL, COALESCE, and NULLIF.

5.THE COMPREHENSIVE DATA SUBLANGUAGE RULE:

The comprehensive data sublanguage rule: The system must support at least one relational language that,

1. *Has a linear syntax*
2. *Can be used both interactively and within application programs,*
3. *Supports data definition operations (including view definitions), data manipulation operations (update as well as retrieval), security and integrity constraints, and transaction management operations (begin, commit, and rollback).*

This rule mandates the existence of a relational database language, such as MySQL, to manipulate data. The language must be able to support all the central functions of a DBMS: creating a database, retrieving and entering data, implementing database security, and so on.

6.THE VIEW UPDATING RULE:

All views that are theoretically updatable must be updatable by the system, this rule deals with views, which are virtual tables used to give various users of a database different views of its structure.

In MySQL, views are not only query-able but also updatable. It means that you can use the INSERT or UPDATE statement to insert or update rows of the base table through the updatable view. Also, we can use a DELETE statement to remove rows of the underlying table through the view.

7.HIGH-LEVEL INSERT, UPDATE AND DELETE:

The system must support set-at-a-time insert, update, and delete operators. This means that data can be retrieved from a relational database in sets constructed of data from multiple rows and/or multiple tables.

MySQL supports high-level insertion, updates, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

8.PHYSICAL DATA INDEPENDENCE:

Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure.

MySQL implies that the way the data is stored physically must be independent of the logical manner in which it's accessed. This is saying that users shouldn't be concerned about how the data is stored or how it's accessed.

9.LOGICAL DATA INDEPENDENCE:

Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure. Logical data independence is more difficult to achieve than physical data independence.

Rule 8 and 9 specify that specific access or storage techniques used by the RDBMS—and even changes to the structure of the tables in the database—shouldn't affect the user's ability to work with the data.

This can be implemented in MySQL using views

10.INTEGRITY INDEPENDENCE:

Integrity constraints must be specified separately from application programs and stored in the catalogue. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications. This allows,

- The integrity rules filters to allow correct data, stored in a data dictionary.
- Key and check constraints, triggers should be stored in a data dictionary.
- Makes RDBMS independent of front-end

MySQL Server does a great job of providing the tools to make this rule a reality. We can protect our data from invalid values for most any possible case using constraints and triggers.

9. State which kind of anomalies could occur in your relational schema

There could be three kind of anomalies in our database if there is any redundancy in the relations in our database,

1.**Insertion anomaly:** The insertion anomaly occurs when a new record is inserted in a relation. In this anomaly a user cannot insert a fact about an entity until he has an additional fact about another entity.

There are insertion anomalies present in,

1. In the Customer table, we have to first create the Zip Code table.
2. In the Transfer table, we have to first create the Customer, Parcel, Branch and Employee table.
3. In the Payment, we have to first create the Transfer, Paymenttype and Deliverytype table.
4. In the Branch, we have to first create the Company and Zipcode table.
5. In the Employee, we have to create the Designation and Branch table.
6. In the Tracking, we have to create the Branch and Parcel table.

2.**Deletion anomaly**: The deletion anomaly occurs when a record is deleted from the relation. In this anomaly, the deletion of facts about an entity automatically deletes the fact of another entity.

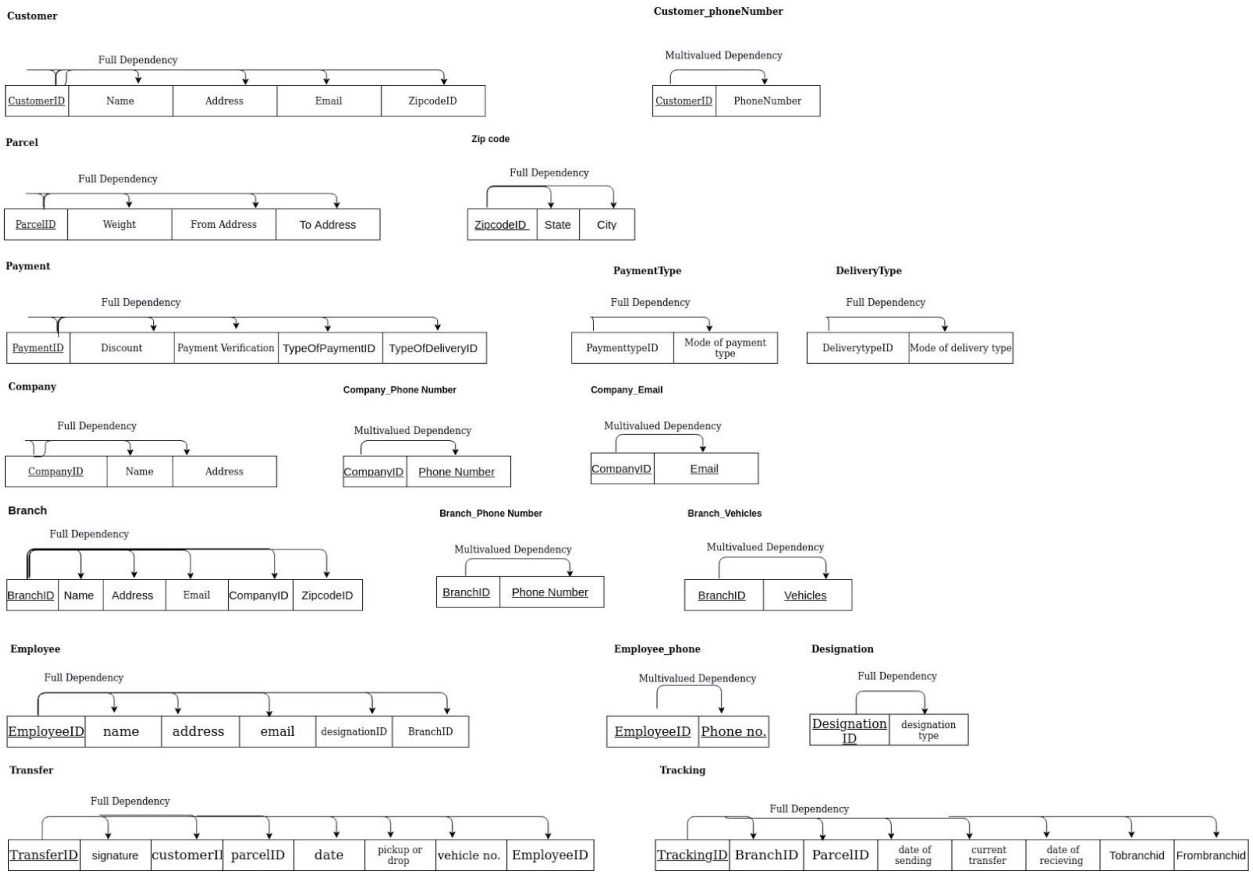
There are no deletion anomalies in our relational schema.

3.**Modification Anomaly**: The modification anomaly occurs when the record is updated in the relation. In this anomaly, the modification in the value of a specific attribute requires modification in all records in which that value occurs.

There are no modification anomalies in our relational schema.

10. Identify and mention the functional dependencies of the schema

Functionality Decomposition



11. Normalize the tables if anomalies are present.

We could check this using 3 NF form,

1.First normal form (1NF):

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Since an attribute of a table holds atomic values. Hence, it is in 1 NF form.

2.Second normal form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Since there are no partial Dependencies only full and multiple dependencies. Hence, it is in 2 NF form.

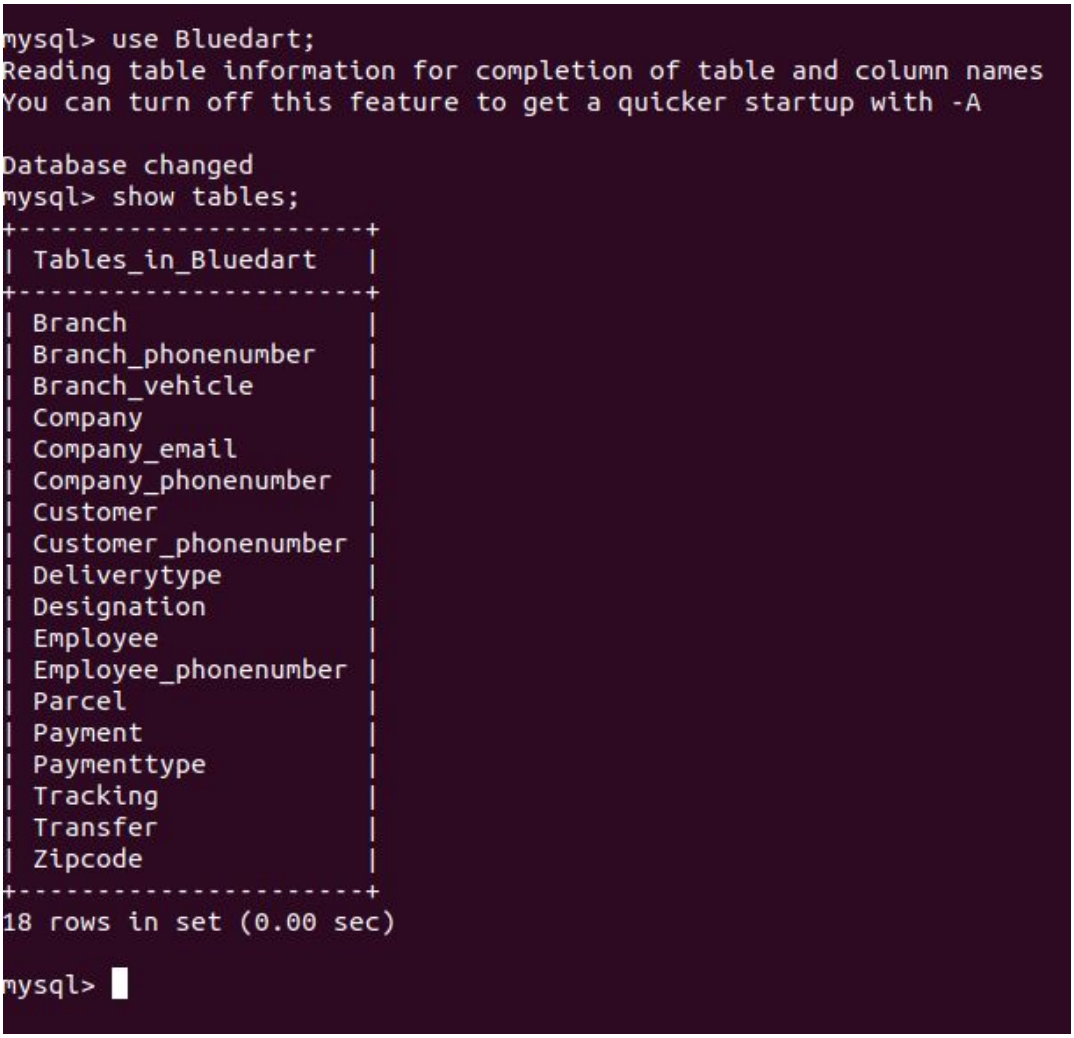
3.Third Normal form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

Since there are no transitive Dependencies. Hence, it is in 3 NF form.

12. Implement the tables in mysql/oracle.



1.Table Description:

1. Company, Company email and Company phone number:

```
mysql> desc Company;
+-----------+----------+------+-----+---------+----------------+
| Field     | Type     | Null | Key | Default | Extra          |
+-----------+----------+------+-----+---------+----------------+
| companyid | int(11)  | NO   | PRI | NULL    | auto_increment |
| name      | char(20) | YES  |     | NULL    |                |
| address   | char(20) | YES  |     | NULL    |                |
+-----------+----------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> mysql> desc Company_email;
+-----------+----------+------+-----+---------+-------+
| Field     | Type     | Null | Key | Default | Extra |
+-----------+----------+------+-----+---------+-------+
| email     | char(20) | NO   | PRI | NULL    |       |
| companyid | int(11)  | NO   | MUL | NULL    |       |
+-----------+----------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc Company_phonenumber;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| companyid    | int(11)     | NO   | MUL | NULL    |       |
| phone_number | varchar(45) | NO   | PRI | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

2.Zipcode:

```
mysql> desc Zipcode;
+-----------+------------+------+-----+---------+-------+
| Field     | Type       | Null | Key | Default | Extra |
+-----------+------------+------+-----+---------+-------+
| zipcodeid | bigint(20) | NO   | PRI | NULL    |       |
| state     | char(20)   | YES  |     | NULL    |       |
| city      | char(20)   | YES  |     | NULL    |       |
+-----------+------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

3.Customer and Customer phone number:

```
mysql> desc Customer;
+------------+------------+------+-----+---------+----------------+
| Field      | Type       | Null | Key | Default | Extra          |
+------------+------------+------+-----+---------+----------------+
| customerid | int(11)    | NO   | PRI | NULL    | auto_increment |
| name       | char(20)   | YES  |     | NULL    |                |
| address    | char(20)   | YES  |     | NULL    |                |
| email      | char(20)   | YES  |     | NULL    |                |
| zipcodeid  | bigint(20) | NO   | MUL | NULL    |                |
+------------+------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc Customer_phonenumber;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| phonenumber | varchar(45) | NO   | PRI | NULL    |       |
| customerid  | int(11)     | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

4.Parcel:

```
mysql> desc Parcel;
+--------------+----------+------+-----+---------+----------------+
| Field        | Type     | Null | Key | Default | Extra          |
+--------------+----------+------+-----+---------+----------------+
| parcelid     | int(11)  | NO   | PRI | NULL    | auto_increment |
| weight       | int(11)  | NO   |     | NULL    |                |
| from_address | char(20) | YES  |     | NULL    |                |
| to_address   | char(20) | YES  |     | NULL    |                |
+--------------+----------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

5.Branch, Branch phone number and Branch vehicles

```
mysql> desc Branch;
+-----------+------------+------+-----+---------+----------------+
| Field     | Type       | Null | Key | Default | Extra          |
+-----------+------------+------+-----+---------+----------------+
| branchid  | int(11)    | NO   | PRI | NULL    | auto_increment |
| name      | char(30)   | YES  |     | NULL    |                |
| address   | char(30)   | YES  |     | NULL    |                |
| email     | char(30)   | YES  |     | NULL    |                |
| zipcodeid | bigint(20) | NO   | MUL | NULL    |                |
| companyid | int(11)    | NO   | MUL | NULL    |                |
+-----------+------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> desc Branch_phonenumber;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| phonenumber | varchar(45) | NO   | PRI | NULL    |       |
| branchid    | int(11)     | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc Branch_vehicle;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| vehicleid | varchar(45) | NO   | PRI | NULL    |       |
| branchid  | int(11)     | NO   | MUL | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

6.Designation:

```
mysql> desc Designation;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| designationid   | int(11)     | NO   | PRI | NULL    |       |
| designationtype | varchar(45) | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

7.Employee and Employee phone number:

```
mysql> desc Employee;
+---------------+----------+------+-----+---------+----------------+
| Field         | Type     | Null | Key | Default | Extra          |
+---------------+----------+------+-----+---------+----------------+
| employeeid    | int(11)  | NO   | PRI | NULL    | auto_increment |
| name          | char(20) | YES  |     | NULL    |                |
| email         | char(20) | YES  |     | NULL    |                |
| address       | char(20) | YES  |     | NULL    |                |
| branchid      | int(11)  | NO   | MUL | NULL    |                |
| designationid | int(11)  | NO   | MUL | NULL    |                |
+---------------+----------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> desc Employee_phonenumber;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| phonenumber | varchar(45) | NO   | PRI | NULL    |       |
| employeeid  | int(11)     | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

8.Transfer:

```
mysql> desc Transfer;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| transferid    | int(11)     | NO   | PRI | NULL    | auto_increment |
| signature     | char(20)    | YES  |     | NULL    |                |
| date          | date        | NO   |     | NULL    |                |
| pickup_or_Drop | char(20)   | YES  |     | NULL    |                |
| vehiclenumber | varchar(45) | NO   |     | NULL    |                |
| parcelid      | int(11)     | NO   | MUL | NULL    |                |
| customerid    | int(11)     | NO   | MUL | NULL    |                |
| employeeid    | int(11)     | NO   | MUL | NULL    |                |
| branchid      | int(11)     | NO   | MUL | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)
```

9.Payment, type of payment and type of delivery:

```
mysql> desc Payment;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| paymentid          | int(11)     | NO   | PRI | NULL    |       |
| discount           | int(11)     | NO   |     | NULL    |       |
| payment_verfication | varchar(45) | NO   |     | NULL    |       |
| transferid         | int(11)     | NO   | MUL | NULL    |       |
| paymenttypeid      | int(11)     | NO   | MUL | NULL    |       |
| deliverytypeid     | int(11)     | NO   | MUL | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> desc Paymenttype;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| paymenttypeid | int(11)     | NO   | PRI | NULL    |       |
| mode_payment  | varchar(45) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> desc Deliverytype;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| deliverytypeid | int(11)     | NO   | PRI | NULL    |       |
| type_delivery  | varchar(45) | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

10.Tracking:

```
mysql> desc Tracking;
+-----------------+----------+------+-----+---------+----------------+
| Field           | Type     | Null | Key | Default | Extra          |
+-----------------+----------+------+-----+---------+----------------+
| trackingid      | int(11)  | NO   | PRI | NULL    | auto_increment |
| date_sent       | date     | NO   |     | NULL    |                |
| date_received   | date     | NO   |     | NULL    |                |
| currenttransfer | char(20) | YES  |     | NULL    |                |
| parcelid        | int(11)  | NO   | MUL | NULL    |                |
| frombranchid    | int(11)  | NO   | MUL | NULL    |                |
| tobranchid      | int(11)  | NO   | MUL | NULL    |                |
+-----------------+----------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

2.Table Data:

1. Company, Company email and Company phone number:

```
mysql> select * from Company;
+-----------+----------+-------------+
| companyid | name     | address     |
+-----------+----------+-------------+
|         1 | Bluedart | Maharashtra |
+-----------+----------+-------------+
1 row in set (0.00 sec)

mysql> select * from Company_phonenumber;
+-----------+--------------+
| companyid | phone_number |
+-----------+--------------+
|         1 | 11           |
|         1 | 111          |
|         1 | 2323453232   |
|         1 | 233215534    |
|         1 | 27349209     |
|         1 | 913478211    |
+-----------+--------------+
6 rows in set (0.00 sec)

mysql> select * from Company_email;
+----------------------+-----------+
| email                | companyid |
+----------------------+-----------+
| adasd@gmail.com      |         1 |
| admin@bluedart.com   |         1 |
| ceo@bluedart.com     |         1 |
| cto@bluedart.com     |         1 |
| gfssde@gmail.com     |         1 |
| manager@bluedart.com |         1 |
+----------------------+-----------+
6 rows in set (0.00 sec)
```

2.Zipcode:

```
mysql> select * from Zipcode;
+-----------+----------------+-----------+
| zipcodeid | state          | city      |
+-----------+----------------+-----------+
|         1 | Uttrakhand     | Doon      |
|         2 | karnataka      | Bangalore |
|         3 | Rajasthan      | Udaipur   |
|         4 | Madhya Pradesh | Indore    |
|         5 | Gujrat         | Ahmedabad |
|         6 | Maharashtra    | Nagpur    |
+-----------+----------------+-----------+
6 rows in set (0.00 sec)
```

3.Customer and Customer phone number:

```
mysql> select * from Customer;
+------------+----------+--------------+--------------------+-----------+
| customerid | name     | address      | email              | zipcodeid |
+------------+----------+--------------+--------------------+-----------+
|          1 | KB       |  Vasant vihar | KB@gmail.com       |         1 |
|          2 | HSB      | HAL road     | HSB@gmail.com      |         2 |
|          3 | harry    | sector 14    | harryop@gmail.com  |         1 |
|          4 | hermioeni | patodi villa | hermioni@gmail.com |         2 |
|          5 | Tina     |  Itwari      | tina@gmail.com     |         5 |
|          6 | Aisha    |  Chaoni      | aisha@gmail.com    |         5 |
+------------+----------+--------------+--------------------+-----------+
6 rows in set (0.00 sec)

mysql> select * from Customer_phonenumber;
+-------------+------------+
| phonenumber | customerid |
+-------------+------------+
| 201         |          1 |
| 202         |          1 |
| 203         |          2 |
| 33445       |          3 |
| 775645      |          3 |
| 7766        |          4 |
| 4534653432  |          5 |
| 54232564522 |          5 |
| 54334367676 |          6 |
+-------------+------------+
9 rows in set (0.00 sec)
```

4.Parcel:

```
mysql> select * from Parcel;
+----------+--------+--------------+------------+
| parcelid | weight | from_address | to_address |
+----------+--------+--------------+------------+
|        1 |     20 | Doon         |  Bangalore |
|        2 |     30 |  Bangalore   |  Doon      |
|        3 |     24 | UDR          | hyderabad  |
|        4 |     31 |  Bangalore   | Indore     |
|        5 |     45 | Bandra       |  Andheri   |
|        6 |     34 |  Chaoni      |  Juhu      |
+----------+--------+--------------+------------+
6 rows in set (0.00 sec)
```

5.Branch, Branch phone number and Branch vehicles

```
mysql> select * from Branch;
+----------+------------------+-------------+-----------------------+-----------+-----------+
| branchid | name             | address     | email                 | zipcodeid | companyid |
+----------+------------------+-------------+-----------------------+-----------+-----------+
|        1 | BluedartDoon     | EC road     | Doon@bluedart.com     |         1 |         1 |
|        2 | BluedartBangalore | Hal road    | Bangalore@bluedart.com |         2 |         1 |
|        3 | Bluedartudaipur  | DN road     | UDR@bluedart.com      |         1 |         1 |
|        4 | BluedartHyderabad | Queen circle | hyderabad@bluedart.com |         2 |         1 |
|        5 | BluedartAhmedabad | Dholka      | Ahmedabad@bluedart.com |         5 |         1 |
|        6 | BluedartNagpur   | Mahal       | Nagpur@bluedart.com   |         6 |         1 |
+----------+------------------+-------------+-----------------------+-----------+-----------+
6 rows in set (0.00 sec)

mysql> select * from Branch_phonenumber;
+-------------+----------+
| phonenumber | branchid |
+-------------+----------+
| 101         |        1 |
| 102         |        1 |
| 123445      |        1 |
| 345678      |        1 |
| 103         |        2 |
| 6522345     |        2 |
| 33525133434 |        5 |
| 98233987229 |        5 |
| 6535345335  |        6 |
+-------------+----------+
9 rows in set (0.00 sec)

mysql> select * from Branch_vehicle;
+--------------+----------+
| vehicleid    | branchid |
+--------------+----------+
| UK D 7879    |        1 |
| KN B 8798    |        2 |
| RJ 14 CR 7879 |        3 |
| AP GT 8798   |        4 |
| BA S 6675    |        5 |
| MH A 7689    |        5 |
+--------------+----------+
6 rows in set (0.00 sec)
```

6.Designation:

```
mysql> select * from Designation;
+----------------+-----------------+
| designationid | designationtype |
+----------------+-----------------+
|              1 | Manager         |
|              2 | Accountant      |
|              3 | Packer          |
|              4 | Receiptionist   |
|              5 | Driver          |
|              6 | Delivery Boy    |
+----------------+-----------------+
6 rows in set (0.00 sec)
```

7.Employee and Employee phone number:

```
mysql> select * from Employee;
+------------+--------+------------------+-----------+----------+---------------+
| employeeid | name   | email            | address   | branchid | designationid |
+------------+--------+------------------+-----------+----------+---------------+
|          1 | Raju   | raju@gmail.com   | Doon      |        1 |             2 |
|          2 | Kaju   | kaju@gmail.com   | Bangalore |        2 |             1 |
|          3 | guddu  | guddu@gmail.com  | UDR       |        1 |             2 |
|          4 | Bablu  | Bablu@gmail.com  | Hyderabad |        2 |             1 |
|          5 | Ramesh | ramesh@gmail.com | Haryana   |        5 |             6 |
|          6 | Shyam  | shyam@gmail.com  | Pune      |        6 |             5 |
+------------+--------+------------------+-----------+----------+---------------+
6 rows in set (0.00 sec)

mysql> select * from Employee_phonenumber;
+-------------+------------+
| phonenumber | employeeid |
+-------------+------------+
| 300         |          1 |
| 301         |          1 |
| 302         |          2 |
| 34543       |          3 |
| 88776       |          3 |
| 33445       |          4 |
| 4542456767  |          5 |
| 545453464   |          5 |
| 4545654544  |          6 |
+-------------+------------+
9 rows in set (0.00 sec)
```

8.Transfer:

```
mysql> select * from Transfer;
+------------+-----------+------------+---------------+--------------+----------+------------+------------+----------+
| transferid | signature | date       | pickup_or_Drop | vehiclenumber | parcelid | customerid | employeeid | branchid |
+------------+-----------+------------+---------------+--------------+----------+------------+------------+----------+
|          1 | a         | 2020-10-06 | Pickup         | UK D 7879    |        1 |          1 |          2 |        1 |
|          2 | b         | 2020-12-04 | Drop           | KN B 8798    |        2 |          2 |          1 |        1 |
|          3 | a         | 2020-10-15 | Pickup         | RJ 14 D 7879 |        3 |          4 |          4 |        3 |
|          4 | b         | 2020-12-04 | Drop           | AP SE 8798   |        4 |          3 |          3 |        4 |
|          5 | a         | 2020-08-06 | Pickup         | MH A 7689    |        5 |          5 |          6 |        5 |
|          6 | b         | 2020-02-04 | Drop           | BA S 6675    |        6 |          6 |          5 |        5 |
+------------+-----------+------------+---------------+--------------+----------+------------+------------+----------+
6 rows in set (0.00 sec)
```

9.Payment, type of payment and type of delivery:

```
mysql> select * from Payment;
+-----------+----------+--------------------+------------+--------------+---------------+
| paymentid | discount | payment_verfication | transferid | paymenttypeid | deliverytypeid |
+-----------+----------+--------------------+------------+--------------+---------------+
|         1 |        0 | Yes                |          1 |            2 |             2 |
|         2 |       10 | Yes                |          2 |            1 |             1 |
|         3 |        4 | Yes                |          4 |            4 |             4 |
|         4 |        3 | Yes                |          4 |            3 |             3 |
|         5 |       10 | Yes                |          5 |            6 |             6 |
|         6 |        5 | Yes                |          5 |            5 |             5 |
+-----------+----------+--------------------+------------+--------------+---------------+
6 rows in set (0.00 sec)

mysql> select * from Paymenttype;
+--------------+---------------+
| paymenttypeid | mode_payment   |
+--------------+---------------+
|            1 | Cash          |
|            2 | Card          |
|            3 | Paytm         |
|            4 | Google Pay    |
|            5 | UPI           |
|            6 | BlueDart Coins |
+--------------+---------------+
6 rows in set (0.01 sec)

mysql> select * from Deliverytype;
+---------------+--------------+
| deliverytypeid | type_delivery |
+---------------+--------------+
|             1 | Base         |
|             2 | Prime        |
|             3 | Drone        |
|             4 | Underground  |
|             5 | Speed Post   |
|             6 | SIT          |
+---------------+--------------+
6 rows in set (0.00 sec)
```

10.Tracking:

```
mysql> select * from Tracking;
+------------+------------+--------------+-----------------+----------+-------------+------------+
| trackingid | date_sent  | date_received | currenttransfer | parcelid | frombranchid | tobranchid |
+------------+------------+--------------+-----------------+----------+-------------+------------+
|          1 | 2020-10-11 | 2020-08-20    | Flight          |        1 |            2 |          3 |
|          2 | 2020-08-11 | 2020-06-15    | Train           |        2 |            1 |          4 |
|          3 | 2020-10-12 | 2020-08-25    | Travel bus      |        1 |            1 |          5 |
|          4 | 2020-06-16 | 2020-07-18    | Train           |        2 |            2 |          6 |
+------------+------------+--------------+-----------------+----------+-------------+------------+
4 rows in set (0.01 sec)
```

13. Execute the 25 queries, 5 functions, 5 procedures, and 5 triggers.

1. Queries:

1. SELECT * FROM Customer WHERE email="HSB@gmail.com";
2. SELECT * FROM Customer WHERE zipcodeid=2;
3. SELECT * FROM Customer_phonenumber WHERE customerid=3

```
mysql> SELECT * FROM Customer WHERE email="HSB@gmail.com";
+------------+------+----------+--------------+-----------+
| customerid | name | address  | email        | zipcodeid |
+------------+------+----------+--------------+-----------+
|          2 | HSB  | HAL road | HSB@gmail.com |         2 |
+------------+------+----------+--------------+-----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM Customer WHERE zipcodeid=2;
+------------+-----------+-------------+-------------------+-----------+
| customerid | name      | address     | email             | zipcodeid |
+------------+-----------+-------------+-------------------+-----------+
|          2 | HSB       | HAL road    | HSB@gmail.com     |         2 |
|          4 | hermioeni | patodi villa | hermioni@gmail.com |         2 |
+------------+-----------+-------------+-------------------+-----------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM Customer_phonenumber WHERE customerid=3;
+-------------+------------+
| phonenumber | customerid |
+-------------+------------+
| 33445       |          3 |
| 775645      |          3 |
+-------------+------------+
2 rows in set (0.00 sec)
```

4. SELECT COUNT(*) FROM Parcel;
5. SELECT * FROM Parcel WHERE weight >=40;

```
mysql> SELECT COUNT(*) FROM Parcel;
+----------+
| COUNT(*) |
+----------+
|        6 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM Parcel WHERE weight >=40;
+----------+--------+--------------+------------+
| parcelid | weight | from_address | to_address |
+----------+--------+--------------+------------+
|        5 |     45 | Bandra       | Andheri    |
+----------+--------+--------------+------------+
1 row in set (0.00 sec)
```

6. SELECT * FROM Payment WHERE paymenttypeid=1 and deliverytypeid=2;
7. SELECT * FROM Payment WHERE paymenttypeid=2 and deliverytypeid=2;

```
mysql> SELECT * FROM Payment WHERE paymenttypeid=1 and deliverytypeid=2;
Empty set (0.01 sec)

mysql> SELECT * FROM Payment WHERE paymenttypeid=2 and deliverytypeid=2;
+-----------+----------+--------------------+------------+--------------+----------------+
| paymentid | discount | payment_verfication | transferid | paymenttypeid | deliverytypeid |
+-----------+----------+--------------------+------------+--------------+----------------+
|         1 |        0 | Yes                |          1 |            2 |              2 |
+-----------+----------+--------------------+------------+--------------+----------------+
1 row in set (0.00 sec)
```

8. SELECT * FROM Company ORDER BY name;
9. SELECT * FROM Company_email WHERE companyid=1;
10. SELECT * FROM Company_phonenumber WHERE companyid=2;

```
mysql> SELECT * FROM Company ORDER BY name;
+-----------+----------+-------------+
| companyid | name     | address     |
+-----------+----------+-------------+
|         1 | Bluedart | Maharashtra |
+-----------+----------+-------------+
1 row in set (0.00 sec)

mysql> SELECT * FROM Company_email WHERE companyid=1;
+---------------------+-----------+
| email               | companyid |
+---------------------+-----------+
| adasd@gmail.com     |         1 |
| admin@bluedart.com  |         1 |
| ceo@bluedart.com    |         1 |
| cto@bluedart.com    |         1 |
| gfssde@gmail.com    |         1 |
| manager@bluedart.com |         1 |
+---------------------+-----------+
6 rows in set (0.00 sec)

mysql> SELECT * FROM Company_phonenumber WHERE companyid=2;
Empty set (0.00 sec)
```

11. SELECT * from Employee where name= “Shyam”;
12. SELECT * from employee WHERE branchid = (SELECT branchid FROM Branch WHERE address="Dholka");
13. SELECT * FROM Employee WHERE Address='Mahal' OR Address='Haryana';

```
mysql> SELECT * from Employee where name= "Shyam";
+------------+-------+-----------------+---------+----------+---------------+
| employeeid | name  | email           | address | branchid | designationid |
+------------+-------+-----------------+---------+----------+---------------+
|          6 | Shyam | shyam@gmail.com | Pune    |        6 |             5 |
+------------+-------+-----------------+---------+----------+---------------+
1 row in set (0.27 sec)

mysql> SELECT * from employee WHERE branchid = (SELECT branchid  FROM Branch WHERE address="Dholka");
+------------+--------+-----------------+---------+----------+---------------+
| employeeid | name   | email           | address | branchid | designationid |
+------------+--------+-----------------+---------+----------+---------------+
|          5 | Ramesh | ramesh@gmail.com | Haryana |        5 |             6 |
+------------+--------+-----------------+---------+----------+---------------+
1 row in set (0.31 sec)

mysql> SELECT * FROM Employee WHERE Address='Mahal' OR Address='Haryana';
+------------+--------+-----------------+---------+----------+---------------+
| employeeid | name   | email           | address | branchid | designationid |
+------------+--------+-----------------+---------+----------+---------------+
|          5 | Ramesh | ramesh@gmail.com | Haryana |        5 |             6 |
+------------+--------+-----------------+---------+----------+---------------+
1 row in set (0.00 sec)
```

14. SELECT * FROM Transfer WHERE parcelid = (SELECT parcelid FROM Parcel WHERE weight=45);
15. SELECT * FROM Transfer WHERE employeeid = (SELECT employeeid FROM employee WHERE address='Haryana');

```
mysql> SELECT * FROM Transfer WHERE parcelid = (SELECT parcelid  FROM Parcel WHERE weight=45);
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
| transferid | signature | date       | pickup_or_Drop | vehiclenumber | parcelid | customerid | employeeid | branchid |
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
|          5 | a         | 2020-08-06 | Pickup         | MH A 7689     |        5 |          5 |          6 |        5 |
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
1 row in set (0.34 sec)

mysql> SELECT * FROM Transfer WHERE employeeid = (SELECT employeeid  FROM employee WHERE address='Haryana');
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
| transferid | signature | date       | pickup_or_Drop | vehiclenumber | parcelid | customerid | employeeid | branchid |
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
|          6 | b         | 2020-02-04 | Drop           | BA S 6675     |        6 |          6 |          5 |        5 |
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
1 row in set (0.04 sec)
```

16. SELECT * FROM Transfer WHERE Date='2020-08-06';
17. SELECT * FROM Tracking ORDER BY CurrentTransfer;

```
mysql> SELECT * FROM Transfer WHERE Date='2020-08-06';
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
| transferid | signature | date       | pickup_or_Drop | vehiclenumber | parcelid | customerid | employeeid | branchid |
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
|          5 | a         | 2020-08-06 | Pickup         | MH A 7689     |        5 |          5 |          6 |        5 |
+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
1 row in set (0.07 sec)

mysql> SELECT * FROM Tracking ORDER BY CurrentTransfer;
+------------+------------+---------------+------------------+----------+--------------+------------+
| trackingid | date_sent  | date_received | currenttransfer  | parcelid | frombranchid | tobranchid |
+------------+------------+---------------+------------------+----------+--------------+------------+
|          1 | 2020-10-11 | 2020-08-20    | Flight           |        1 |            2 |          3 |
|          5 | 2020-08-11 | 2020-08-20    | Road             |        3 |            4 |          2 |
|          6 | 2020-06-08 | 2020-06-15    | Road             |        6 |            5 |          2 |
|          2 | 2020-08-11 | 2020-06-15    | Train            |        2 |            1 |          4 |
|          4 | 2020-06-16 | 2020-07-18    | Train            |        2 |            2 |          6 |
|          3 | 2020-10-12 | 2020-08-25    | Travel bus       |        1 |            1 |          5 |
+------------+------------+---------------+------------------+----------+--------------+------------+
6 rows in set (0.11 sec)
```

18. select * from employee e inner join transfer t on e.employeeid=t.employeeid where t.parcelid=4;
19. SELECT * from employee where name like 'S%' and exists (select * from Designation where DesignationType ='Driver');
20. select * from designation d inner join employee e on d.designationid=e.designationid where e.branchid=2;

```
mysql> select * from employee e inner join transfer t on e.employeeid=t.employeeid where t.parcelid=4;
+------------+------+-----------------+---------+----------+---------------+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
| employeeid | name | email           | address | branchid | designationid | transferid | signature | date       | pickup_or_Drop | vehiclenumber | parcelid | customerid | employeeid | branchid |
+------------+------+-----------------+---------+----------+---------------+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
|          3 | guddu | guddu@gmail.com | UDR     |        1 |             2 |          4 | b         | 2020-12-04 | Drop           | AP SE 8798    |        4 |          3 |          3 |        4 |
+------------+------+-----------------+---------+----------+---------------+------------+-----------+------------+----------------+---------------+----------+------------+------------+----------+
1 row in set (0.02 sec)

mysql> SELECT * from employee where name like 'S%' and exists (select * from Designation where DesignationType ='Driver');
+------------+------+----------------+---------+----------+---------------+
| employeeid | name | email          | address | branchid | designationid |
+------------+------+----------------+---------+----------+---------------+
|          6 | Shyam | shyam@gmail.com | Pune    |        6 |             5 |
+------------+------+----------------+---------+----------+---------------+
1 row in set (0.15 sec)

mysql>  select * from designation d inner join employee e on d.designationid=e.designationid where e.branchid=2;
+---------------+-----------------+------------+-------+---------------+----------+----------+---------------+
| designationid | designationtype | employeeid | name  | email         | address  | branchid | designationid |
+---------------+-----------------+------------+-------+---------------+----------+----------+---------------+
|             1 | Manager         |          2 | Kaju  | kaju@gmail.com | Bangalore |        2 |             1 |
|             1 | Manager         |          4 | Bablu | Bablu@gmail.com | Hyderabad |        2 |             1 |
+---------------+-----------------+------------+-------+---------------+----------+----------+---------------+
2 rows in set (0.00 sec)
```

21. SELECT * FROM Branch WHERE name="Bluedartudaipur"
22. SELECT * FROM Branch WHERE branchid="1"
23. SELECT branchID FROM Branch WHERE name = "BluedartDoon"

```
mysql> SELECT * FROM Branch WHERE name="Bluedartudaipur";
+----------+-----------------+---------+------------------+-----------+-----------+
| branchid | name            | address | email            | zipcodeid | companyid |
+----------+-----------------+---------+------------------+-----------+-----------+
|        3 | Bluedartudaipur | DN road | UDR@bluedart.com |         1 |         1 |
+----------+-----------------+---------+------------------+-----------+-----------+
1 row in set (0.00 sec)

mysql> SELECT * FROM Branch WHERE BranchID="1";
+----------+--------------+---------+------------------+-----------+-----------+
| branchid | name         | address | email            | zipcodeid | companyid |
+----------+--------------+---------+------------------+-----------+-----------+
|        1 | BluedartDoon | EC road | Doon@bluedart.com |         1 |         1 |
+----------+--------------+---------+------------------+-----------+-----------+
1 row in set (0.00 sec)

mysql> SELECT branchID FROM Branch WHERE name = "BluedartDoon";
+----------+
| branchID |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)
```

24. SELECT branchID FROM Branch_vehicle WHERE vehicleid="UK D 7879"
25. SELECT phonenumber FROM Branch_phonenumber WHERE branchid = "2"
26. SELECT name FROM Branch WHERE email = "UDR@bluedart.com"

```
mysql> SELECT branchid FROM Branch_vehicle WHERE vehicleid="UK D 7879";
+----------+
| branchid |
+----------+
|        1 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT phonenumber FROM Branch_phonenumber WHERE branchid = 2;
+-------------+
| phonenumber |
+-------------+
| 103         |
| 6522345     |
+-------------+
2 rows in set (0.00 sec)

mysql> SELECT name FROM Branch WHERE email = "UDR@bluedart.com";
+-----------------+
| name            |
+-----------------+
| Bluedartudaipur |
+-----------------+
1 row in set (0.00 sec)
```

27. SELECT zipcodeID FROM Branch WHERE name="Bluedartudaipur"

```
mysql> SELECT * FROM Branch WHERE name="Bluedartudaipur";
+----------+-----------------+---------+-----------------+-----------+-----------+
| branchid | name            | address | email           | zipcodeid | companyid |
+----------+-----------------+---------+-----------------+-----------+-----------+
|        3 | Bluedartudaipur | DN road | UDR@bluedart.com |         1 |         1 |
+----------+-----------------+---------+-----------------+-----------+-----------+
1 row in set (0.00 sec)

mysql>
```

2. Function:

Total Number Of Employees

1. delimiter $
2. create function totalemployees()
    a. returns int
    b. begin
    c. declare employees int unsigned;
    d. select count(*) into employees from employee;
    e. return employees;
    f. end$
3. delimiter ;
4. select totalemployees() as "Total Number of Employees";

```
mysql> delimiter $
mysql>  create function totalemployees()
    -> returns int
    -> begin
    -> declare employees int unsigned;
    -> select count(*) into employees from employee;
    -> return employees;
    -> end$
Query OK, 0 rows affected (1.25 sec)

mysql> delimiter ;
mysql> select totalemployees() as "Total Number of Employees";
+---------------------------+
| Total Number of Employees |
+---------------------------+
|                         6 |
+---------------------------+
1 row in set (0.44 sec)
```

2. Total Number Of Parcel

1. delimiter $
2. create function totalparcel()
    a. returns int
    b. begin
    c. declare parcels int unsigned;
    d. select count(*) into parcels from parcel;
    e. return parcels;
    f. return parcels;
    g. end$
3. delimiter ;
4. select totalparcel() as "Total Number of Parcel";

```
mysql> delimiter $
mysql> create function totalparcel()
    -> returns int
    -> begin
    -> declare parcels int unsigned;
    -> select count(*) into parcels from parcel;
    -> return parcels;
    -> return parcels;
    -> end$
Query OK, 0 rows affected (0.18 sec)

mysql>   delimiter ;
mysql> select totalparcel() as "Total Number of Parcel";
+------------------------+
| Total Number of Parcel |
+------------------------+
|                      6 |
+------------------------+
1 row in set (0.08 sec)
```

3. function to find the name of customer

1. delimiter $
2. create function customers()
    a. returns text
    b. begin
    c. declare customername text;
    d. select name into customername
    e. from customer
    f. where name="Aisha";
    g. return customername;
    h. end$
3. delimiter ;
4. select customers() as " Name of Customer";

```
mysql> delimiter $
mysql> create function customers()
    -> returns text
    -> begin
    -> declare customername text;
    -> select name into customername
    -> from customer
    -> where name="Aisha";
    -> return customername;
    -> end$
Query OK, 0 rows affected (0.14 sec)

mysql> delimiter ;
mysql> select customers() as " Name of Customer";
+------------------+
| Name of Customer |
+------------------+
| Aisha            |
+------------------+
1 row in set, 1 warning (0.25 sec)
```

4. **function to retrieve if pickup or drop was done on given date**

1. delimiter $
2. create function pickdrop()
    a. returns text
    b. begin
    c. declare pd text;
3. select pickup_or_Drop into pd
    a. from transfer
    b. where date='2020-10-15';
    c. return pd;
    d. end$
4. delimiter ;
5. select pickdrop() as " PICKUP OR DROP";

```
mysql> delimiter $
mysql> create function pickdrop()
    -> returns text
    -> begin
    -> declare pd text;
    ->  select pickup_or_Drop into pd
    -> from transfer
    -> where date='2020-10-15';
    -> return pd;
    -> end$
mysql> delimiter ;
mysql> select pickdrop() as " PICKUP OR DROP";
+----------------+
| PICKUP OR DROP |
+----------------+
| NULL           |
+----------------+
1 row in set, 1 warning (0.15 sec)
```

5. function to retrieve the weight for given address

1. delimiter $
    a. create function weight()
2. returns int
3. begin
4. declare wt int;
5. select weight into wt
6. from parcel p where p.from_address="Doon";
    a. return wt;
    b. end$$
    c. delimiter ;
    d. select weight() as "Weight";

```
mysql> delimiter $
mysql> create function weight()
    ->  returns int
    ->      begin
    ->     declare wt int;
    ->      select weight into wt
    ->     from parcel p where p.from_address="Doon";
    -> return wt;
    -> end$$

mysql> delimiter ;
mysql> select weight() as "Weight";
+--------+
| Weight |
+--------+
|     20 |
+--------+
1 row in set (0.00 sec)
```

3.Procedure:

1. Display_customers():

1. DELIMITER $$
2. USE `Bluedart`$$
3. CREATE PROCEDURE `display_customers` ()
4. BEGIN
5. SELECT * FROM Customer ;
6. END$$
7. DELIMITER ;

```
mysql> CALL `Bluedart`.`display_customers`();
+------------+-----------+--------------+--------------------+-----------+
| customerid | name      | address      | email              | zipcodeid |
+------------+-----------+--------------+--------------------+-----------+
|          1 | KB        |  Vasant vihar | KB@gmail.com       |         1 |
|          2 | HSB       | HAL road     | HSB@gmail.com      |         2 |
|          3 | harry     | sector 14    | harryop@gmail.com  |         1 |
|          4 | hermioeni | patodi villa | hermioni@gmail.com |         2 |
|          5 | Tina      |  Itwari      | tina@gmail.com     |         5 |
|          6 | Aisha     |  Chaoni      | aisha@gmail.com    |         5 |
+------------+-----------+--------------+--------------------+-----------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

2.Update_adress (customerid, new address):

1. DELIMITER $$
2. USE `Bluedart`$$
3. CREATE PROCEDURE update_adress (IN temp_customerid int, IN new_address char(20))
4. BEGIN
5. UPDATE Customer SET address = new_address WHERE customerid=temp_customerid;
6. END$$
7. DELIMITER ;

```
mysql> CALL `Bluedart`.`display_customers`();
+------------+-----------+--------------+--------------------+-----------+
| customerid | name      | address      | email              | zipcodeid |
+------------+-----------+--------------+--------------------+-----------+
|          1 | KB        |  Vasant vihar | KB@gmail.com       |         1 |
|          2 | HSB       | HAL road     | HSB@gmail.com      |         2 |
|          3 | harry     | sector 14    | harryop@gmail.com  |         1 |
|          4 | hermioeni | patodi villa | hermioni@gmail.com |         2 |
|          5 | Tina      |  Itwari      | tina@gmail.com     |         5 |
|          6 | Aisha     |  Chaoni      | aisha@gmail.com    |         5 |
+------------+-----------+--------------+--------------------+-----------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> CALL `Bluedart`.`update_adress`(1, "Rajpur road");
Query OK, 1 row affected (0.10 sec)

mysql> CALL `Bluedart`.`display_customers`();
+------------+-----------+--------------+--------------------+-----------+
| customerid | name      | address      | email              | zipcodeid |
+------------+-----------+--------------+--------------------+-----------+
|          1 | KB        | Rajpur road  | KB@gmail.com       |         1 |
|          2 | HSB       | HAL road     | HSB@gmail.com      |         2 |
|          3 | harry     | sector 14    | harryop@gmail.com  |         1 |
|          4 | hermioeni | patodi villa | hermioni@gmail.com |         2 |
|          5 | Tina      |  Itwari      | tina@gmail.com     |         5 |
|          6 | Aisha     |  Chaoni      | aisha@gmail.com    |         5 |
+------------+-----------+--------------+--------------------+-----------+
6 rows in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

3.weight_max(Output Highest weight present):

1. DELIMITER $$
2. USE `Bluedart`$$
3. CREATE PROCEDURE weight_max(OUT highestweight int)
4. BEGIN
5. select max(weight) into highestweight from Parcel;
6. END$$
7. DELIMITER ;

```
mysql> CALL `Bluedart`.`weight_max`(@M);
Query OK, 1 row affected (0.00 sec)

mysql> select @M;
+------+
| @M   |
+------+
|   45 |
+------+
1 row in set (0.00 sec)
```

4.disp_no_of_transfers(Output No of transfer, Input Parcel ID):

1. DELIMITER $$
2. USE `Bluedart`$$
3. CREATE PROCEDURE disp_no_of_transfers(INOUT no_of_transfer integer, IN temp_parcelid int)
4. BEGIN
5. select COUNT(temp_parcelid)
6. INTO no_of_transfer FROM Transfer where parcelid = temp_parcelid;
7. END$$
8. DELIMITER ;

```
mysql> CALL `Bluedart`.`disp_no_of_transfers`(@M, 1);
Query OK, 1 row affected (0.00 sec)

mysql> select @M;
+------+
| @M   |
+------+
|    1 |
+------+
1 row in set (0.00 sec)

mysql>
```

5.Payment_in_cash():

1. DELIMITER $$
2. USE `Bluedart`$$
3. CREATE PROCEDURE payment_in_cash ()
4. BEGIN
5. SELECT * FROM Payment where paymenttypeid=1 ;
6. END$$

7. DELIMITER ;

```
mysql> CALL `Bluedart`.`payment_in_cash`();
+-----------+----------+--------------------+------------+--------------+----------------+
| paymentid | discount | payment_verfication | transferid | paymenttypeid | deliverytypeid |
+-----------+----------+--------------------+------------+--------------+----------------+
|         2 |       10 | Yes                |          2 |            1 |              1 |
+-----------+----------+--------------------+------------+--------------+----------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

4.Triggers :

1. For email verification:

1. DELIMITER //
2. CREATE TRIGGER EMAIL_VERIFICATION
3. BEFORE INSERT
4. ON Customer
5. FOR EACH ROW
6. BEGIN
7. IF NEW.email NOT LIKE '%@%.__%' THEN
8. SIGNAL SQLSTATE VALUE '45000'
9. SET MESSAGE_TEXT = '[table:Customer] - email column is not valid';
10. END IF;
11. END;//

```
mysql> insert into Customer values(7,"Demo","SIT","Demogmail",1);
ERROR 1644 (45000): [table:Customer] - email column is not valid
mysql>
```

2.For valid phone number:

1. DELIMITER //
2. CREATE TRIGGER VALID_PH
3. BEFORE INSERT
4. ON Customer_phonenumber
5. FOR EACH ROW
6. BEGIN
7. IF LENGTH(NEW.phonenumber)<10 THEN
8. SIGNAL SQLSTATE VALUE '45000'

```
9.          SET MESSAGE_TEXT = '[table:Customer_phonenumber] - PHONE NO. column is
    not valid';
10.         END IF;
11. END;//
```

```
mysql> insert into Customer_phonenumber values ("555",3);
ERROR 1644 (45000): [table:Customer_phonenumber] - PHONE NO. column is not valid
mysql>
```

3.For valid name:

```
1. DELIMITER //
2. CREATE TRIGGER VALID_NAME
3. BEFORE INSERT
4. ON Customer
5. FOR EACH ROW
6. BEGIN
7.          IF  ISNULL(NEW.name) THEN
8.          SIGNAL SQLSTATE VALUE '45000'
9.          SET MESSAGE_TEXT = '[table:Customer] - NAME column is not valid';
10.         END IF;
11. END;//
```

```
mysql> insert into Customer values(8,NULL,"Pune","demo1@gmail.com",5);
ERROR 1644 (45000): [table:Customer] - NAME column is not valid
mysql>
```

4.For checking if the State is valid:

```
1. DELIMITER //
2. CREATE TRIGGER VALID_Zipcode
3. BEFORE INSERT
4. ON Zipcode
5. FOR EACH ROW
6. BEGIN
7.          IF NOT LENGTH(NEW.zipcodeid)=10 THEN
8.          SIGNAL SQLSTATE VALUE '45000'
```

```
9.          SET MESSAGE_TEXT = '[table:Zipcode] - zipcode is not valid';
10.         END IF;
11. END;//
```

```
mysql> INSERT INTO Zipcode (zipcodeid,state,city) values(7,"maharashtra","Pune");
ERROR 1644 (45000): [table:Zipcode] - zipcode is not valid
mysql>
```

5.To check for a valid date:

```
1. DELIMITER //
2. CREATE TRIGGER VALID_DATE
3.          BEFORE INSERT
4.          ON Transfer
5.          FOR EACH ROW
6. BEGIN
7.          IF NOT ISDATE(NEW.date) THEN
8.          SIGNAL SQLSTATE VALUE '45000'
9.          SET MESSAGE_TEXT = '[table:Tranfer] - Date is not valid';
10.         END IF;
11. END;//
```

```
mysql> insert into Transfer values (6,"a",'404',"Pickup","RJ 7879",3,4,4,3);
ERROR 1292 (22007): Incorrect date value: '404' for column 'date' at row 1
mysql>
```

GitHub Repository containing our schema, data population and queries.