Create a Database Schema and Table Relationships for a Logistic Company's Data

Table of Contents

ABSTRACT

Logistics is the support function of an organization and it means having the right object, at the right place, in the right time. Logistics deals with various kinds of methods to control the flow of resources from one place to another. One of the major and the mostimportant factors that is costing is being dealt with utmost attention. The project is being designed keeping in attention the details of the various requirements of logistics such as keeping records of the goods i.e. their details and the kind of content that is stored in the shipment which is to be delivered.

A Relational Database Management System (RDBMS) is similar to DBMS. The difference is that in RDBMS, the entities and values in tables are related to one another. Also the tables are related to each other. Thus, it is called “Relational”.

PROBLEM DESCRIPTION

The logistics company provides services in both the international and domestic sectors. The logistics management takes into consideration every facility that has an impact on cost. It plays an important role in making the product conform to customer requirements. Also it involves efficient integration of suppliers, manufacturers, Import & export and other activities at many levels, from the strategic level through the tacticalto the operational level.

Customers can send different types of shipping contents. Payment is to be paid at the same time the product is delivered to the client. Delivery boy and centre head can update the status of the shipment. Create a database schema and table relationships that can be used in any technology.

SCOPE

It is of critical importance to the organization how it delivers products & services to the customer, whether the product is tangible or intangible. Effective and efficient physical movement of the tangible product will speak of intangible services associated with theproduct and the organization which is delivering it.

In case of intangible products, the delivery of tangibles at the right place & right time will speak about its quality. On the macro level infrastructure such as various modes of transport, transportation equipment, storage facilities, connectivity and information processing are contributing to a large extent in the physical movement of goods produced in manufacturing, mining and agriculture Sectors.

This speed and reliability in distribution of products and services contribute to a great extent in the growth of a country's domestic and international trade.

TABLE DEFINITIONS

1) Employee_Details Table:
   This table contains the information of the employees.

Column Name	Data Type	Description
Emp_ID	INT (5)	Employee ID (Primary Key)
Emp_NAME	VARCHAR (30)	Name of the employee
Emp-BRANCH	VARCHAR (15)	Branch name
Emp_DESIGNATION	VARCHAR (40)	Designation of the employee
Emp_ADDR	VARCHAR (100)	Address of the employee
Emp_CONT_NO	VARCHAR (10)	Contact Number of the employee

2) Membership Table:
   This table contains the membership details of the customer or client.

Column Name	Data Type	Description
M_ID	INT	Membership ID associated with the client (Primary Key)
START_DATE	TEXT	Start date of the membership
END_DATE	TEXT)	End date of the membership

3) Customer Table:
   This table contains the information of the customers or clients.

Column Name	Data Type	Description
Cust_ID	INT (4)	Client ID (Primary Key)
Cust-NAME	VARCHAR (30)	Name of the client
Cust-EMAIL_ID	VARCHAR (50)	Email of the client
Cust_CONT_NO	VARCHAR (10)	Contact Number of the client
Cust_ADDR	VARCHAR (100)	Address of the client
Cust_TYPE	VARCHAR (30)	Type of client (Wholesale, Retail, Internal Goods)
Membership_M_ID	INT	Membership ID (Foreign Key)

4) Payment_Details Table:
   This table contains the payment details.

Column Name	Data Type	Description
PAYMENT_ID	VARCHAR (40)	Payment Unique ID (Primary Key)
AMOUNT	INT	Price to be paid by the client
PAYMENT_STATUS	VARCHAR (10)	Payment status (Paid / Not Paid)
PAYMENT_DATE	TEXT	Date when payment is made by the client
PAYMENT_MODE	VARCHAR (25)	Mode of payment (COD / Card Payment)
Shipment_SH_ID	VARCHAR (6)	Shipment ID (Foreign Key)
Shipment_Client_C_ID	INT (4)	Client ID (Foreign Key)

5) Shipment_Details Table:
   This table contains the shipment details.

Column Name	Data Type	Description
SD_ID	VARCHAR (6)	Shipment ID (Primary Key)
SD_CONTENT	VARCHAR (40)	Type of shipping content
SD_DOMAIN	VARCHAR (15)	Shipment Domain (International / Domestic)
SD_TYPE	VARCHAR (15)	Service Type (Express / Regular)
SD_WEIGHT	VARCHAR (10)	Shipment Weight
SD-CHARGES	INT (10)	Shipment Charges
SD-ADDR	VARCHAR (100)	Source Address
DS_ADDR	VARCHAR (100)	Destination Address
Customer_Cust_ID	INT (4)	Client ID (Foreign Key)

6) Status table:

This table contains the details about the delivery status.

Column Name	Data Type	Description
CURRENT_ST	VARCHAR (15)	Current status of the shipment
SENT_DATE	TEXT	Date when shipment was sent
DELIVERY_DATE	TEXT	Date when the product was/will be delivered
SH_ID	VARCHAR (6)	Shipment ID (Primary Key)

7) Employee Manages Shipment Table:

This is a relationship table between the employee and the shipment table.

Column Name	Data Type	Description

Employee_E_ID	INT (5)	Employee ID (Foreign Key)
Shipment_SH_ID	VARCHAR (6)	Shipment ID (Foreign Key)
Status_SH_ID	VARCHAR (6)	Shipment_ID from status table (Foreign Key)

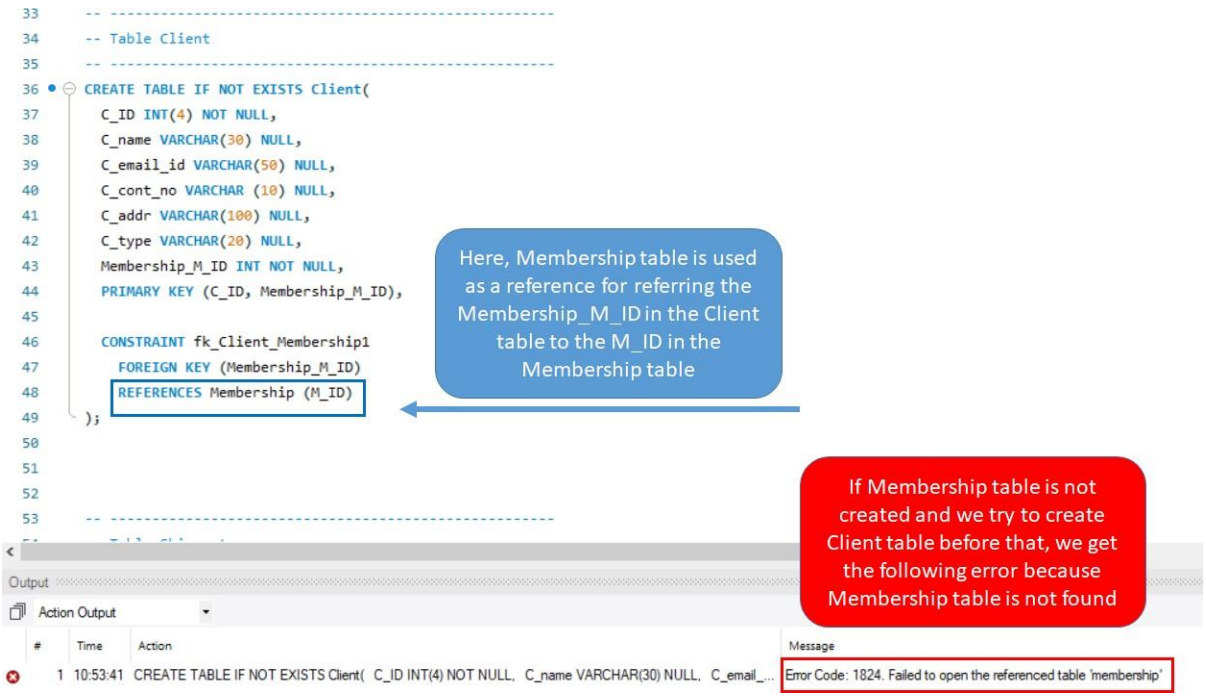
CLASS DIAGRAM

CREATING TABLES AND RELATIONSHIPS

We need to create the tables in the order:

1. Employee
2. Membership
3. Client
4. Shipment
5. Payment
6. Status
7. Employee_Manages_Shipment

Otherwise, simply create the referred tables first and then create other tables. If the referred table is not created first, then the workbench will throw an error. For example, the 'Membership' table is referred in the 'Client' table. Now, if we try to create the Client table before creating the Membership table, we will get the following error:



This also applies to the insertion of the data. When inserting the data in the tables follow the same.

```
--- ----------------------------------------------------------------------------
-- Schema LOGISTICS
```

```
--- -----------------------------------------------------------------------
CREATE SCHEMA IF NOT EXISTS GLLOGISTICS;
USE GLLOGISTICS;

--- -----------------------------------------------------------------------
-- Table Employee_Details
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Employee_Details(
       Emp_ID INT(5) NOT NULL,
       Emp_name VARCHAR(30) NULL,
       Emp_branch VARCHAR(15) NULL,
       Emp_designation VARCHAR(40) NULL,
       Emp_addr VARCHAR(100) NULL,
       Emp_Cont_no VARCHAR(10) NULL,
       PRIMARY KEY (Emp_ID)
);


--- -----------------------------------------------------------------------
-- Table Membership
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Membership(
       M_ID INT NOT NULL,
       Start_Date TEXT NULL,
       End_Date TEXT NULL,
       PRIMARY KEY (M_ID)
);
```

```
--- -----------------------------------------------------------------------
-- Table Customer
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Client(
       Cust_ID INT(4) NOT NULL,
       Cust_name VARCHAR(30) NULL,
       Cust_email_id VARCHAR(50) NULL,
       Cust_cont_no VARCHAR (10) NULL,
       Cust_addr VARCHAR(100) NULL,
       Cust_type VARCHAR(20) NULL,
       Membership_M_ID INT NOT NULL,
       PRIMARY KEY (Cust_ID, Membership_M_ID),

       CONSTRAINT fk_Customer_Membership1
            FOREIGN KEY (Membership_M_ID)
            REFERENCES Membership (M_ID)
);


--- -----------------------------------------------------------------------
-- Table Shipment_Details
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Shipment(
       Sd_id VARCHAR(6) NOT NULL,
       Sd_content VARCHAR(40) NULL,
       Sd_domain VARCHAR(15) NULL,
       Sd_type VARCHAR(15) NULL,
       Sd_weight VARCHAR(10) NULL,
       Sd_charges INT(10) NULL,
       Sd_addr VARCHAR(100) NULL,
       Ds_Addr VARCHAR(100) NULL,
       Customer_Cust_ID INT(4) NOT NULL,
       PRIMARY KEY (Sd_id, Customer_C_ID),

       CONSTRAINT fk_Shipment_Client1
             FOREIGN KEY (Customer_Cust_ID)
             REFERENCES Customer(Cust_ID)
);
```

```
--- -----------------------------------------------------------------------
-- Table Payment_Details
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Payment(
       PAYMENT_ID VARCHAR(40) NOT NULL,
       Amount INT NULL,
       Payment_Status VARCHAR(10) NULL,
       Payment_Date TEXT NULL,
       Payment_Mode VARCHAR(25) NULL,
       Shipment_Details_Sd_id VARCHAR(6) NOT NULL,
       Shipment_Details_Customer_Cust_ID INT(4) NOT NULL,
       PRIMARY KEY (PAYMENT_ID, Shipment_Sd_id,
     Shipment_Details_Customer_C_ID),

       CONSTRAINT fk_Payment_Shipment1
              FOREIGN KEY (Shipment_Sh_id ,
          Shipment_Customer_C_ID)
              REFERENCES Shipment (Sh_id , Customer_C_ID)
);



--- -----------------------------------------------------------------------
-- Table Status
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Status(
       Current_ST VARCHAR(15) NOT NULL,
       Sent_date TEXT NULL,
       Delivery_date TEXT NULL,
       Sh_id VARCHAR(6) NOT NULL,
       PRIMARY KEY (Sh_id)
);
```

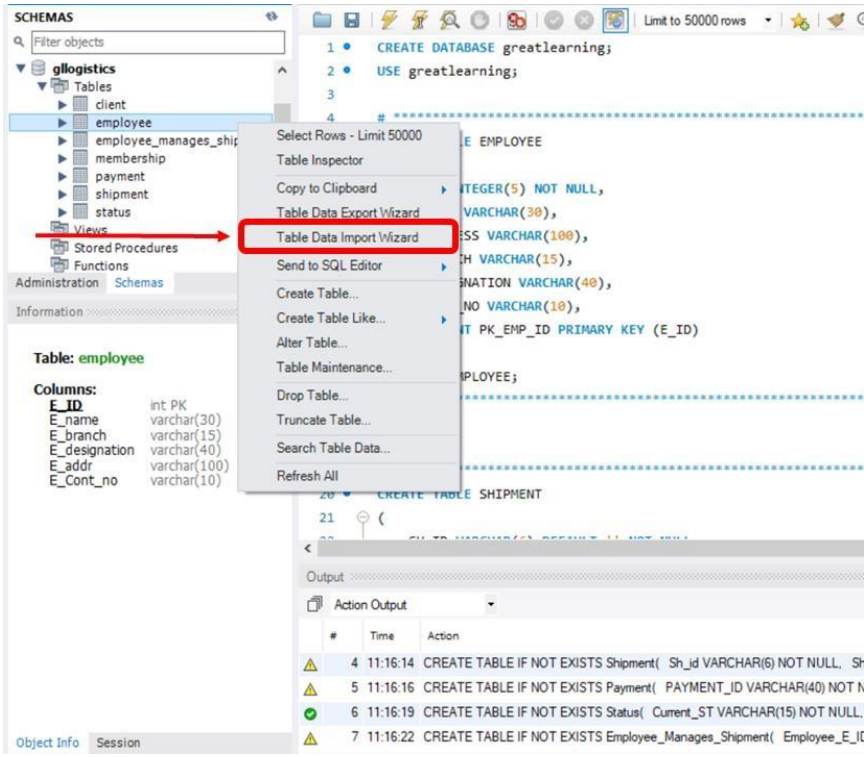
```
--- -----------------------------------------------------------------------
-- Table Employee_Manages_Shipment
--- -----------------------------------------------------------------------
CREATE TABLE IF NOT EXISTS Employee_Manages_Shipment(
  Employee_E_ID INT(5) NOT NULL,
  Shipment_Sh_id VARCHAR(6) NOT NULL,
  Status_Sh_id VARCHAR(6) NOT NULL,


  PRIMARY KEY (Employee_Details_Emp_ID,
Shipment_Details_Sd_id,  Status_Sh_id),
Key constraint on Employee_Details_Emp_ID,
Shipment_Deatils_Sh_id and Status_Sh_id

  CONSTRAINT fk_Employee_Manages_Shipment_Employee
          Employee_Details_Emp_ID
    FOREIGN KEY (Employee_Details_Emp_ID)
    REFERENCES Employee_Details (Emp_ID),
  CONSTRAINT fk_Employee_Manages_Shipment_Shipment1

    FOREIGN KEY (Shipment_Details_Sd_id)
    REFERENCES Shipment_Details (Sd_id),
  CONSTRAINT fk_Employee_Manages_Shipment_Status1

    FOREIGN KEY (Status_Sh_id)
    REFERENCES Status (Sh_id)
);


--- -----------------------------------------------------------------------
-- Describe Tables
--- -----------------------------------------------------------------------
DESCRIBE Customer;
DESCRIBE Employee_Details;
DESCRIBE Shipment_Details;
DESCRIBE Payment_Details;
DESCRIBE Membership;
DESCRIBE STATUS;
DESCRIBE employee_manages_shipment;
--- -----------------------------------------------------------------------
```

```
--- ----------------------------------------------------------------------
-- Selecting the contents from the tables
--- ----------------------------------------------------------------------
SELECT * FROM Employee_Details;
SELECT * FROM Membership;
SELECT * FROM Customer;
SELECT * FROM Payment_Details;
SELECT * FROM Shipment_Details;
SELECT * FROM STATUS;
SELECT * FROM employee_manages_shipment;
--- ----------------------------------------------------------------------
```

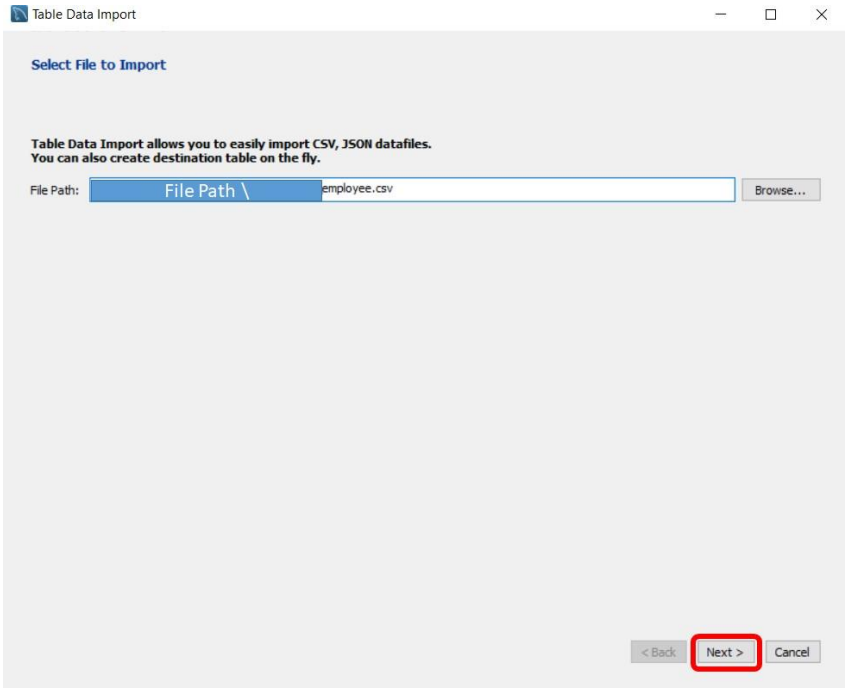
LOADING DATA

MySQL Workbench provides the facility to upload the csv file into the tables usingthe graphical user interface (GUI). Below are the steps for the same:

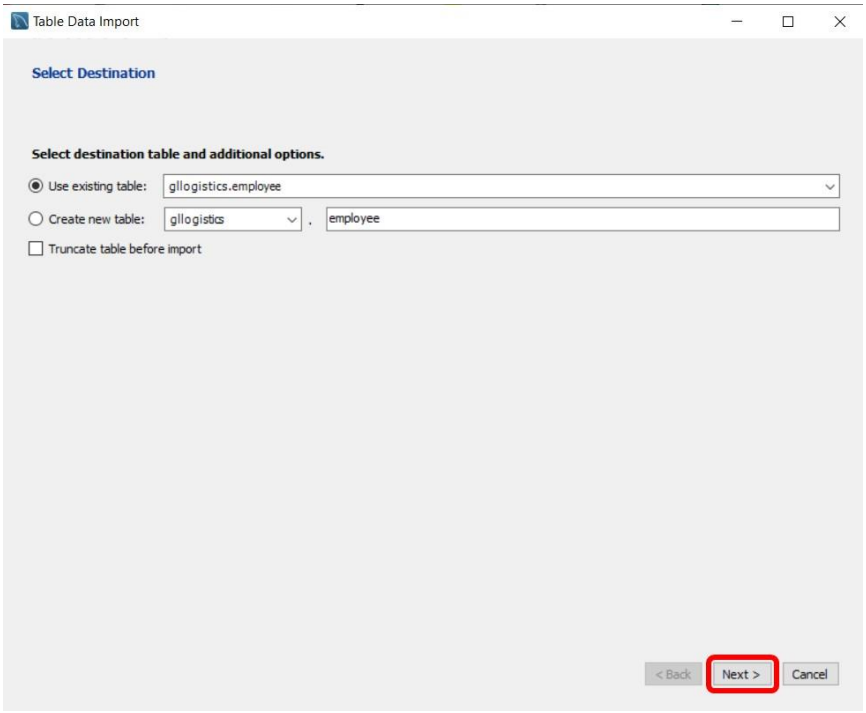
Step 1: Right-click on the table and click on “Table Data Import Wizard”.



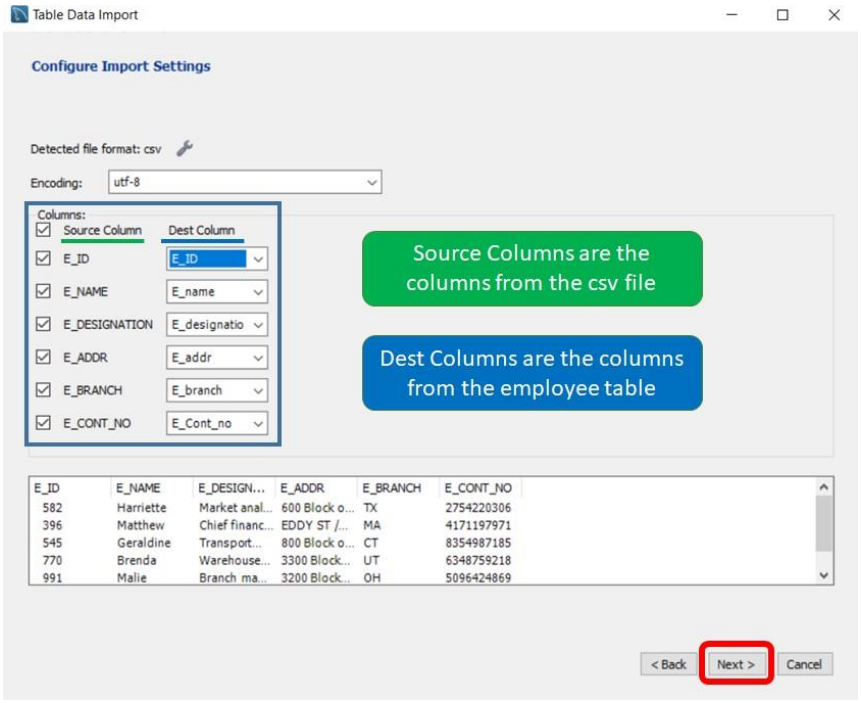
Step 2: Browse and select the csv file and click “Next” button.



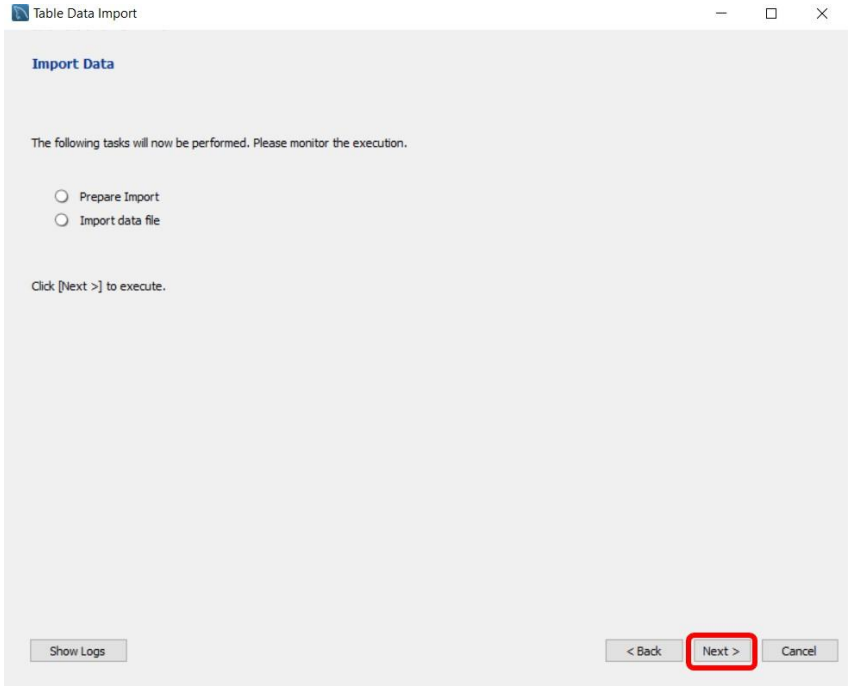
Step 3: Click “Next”



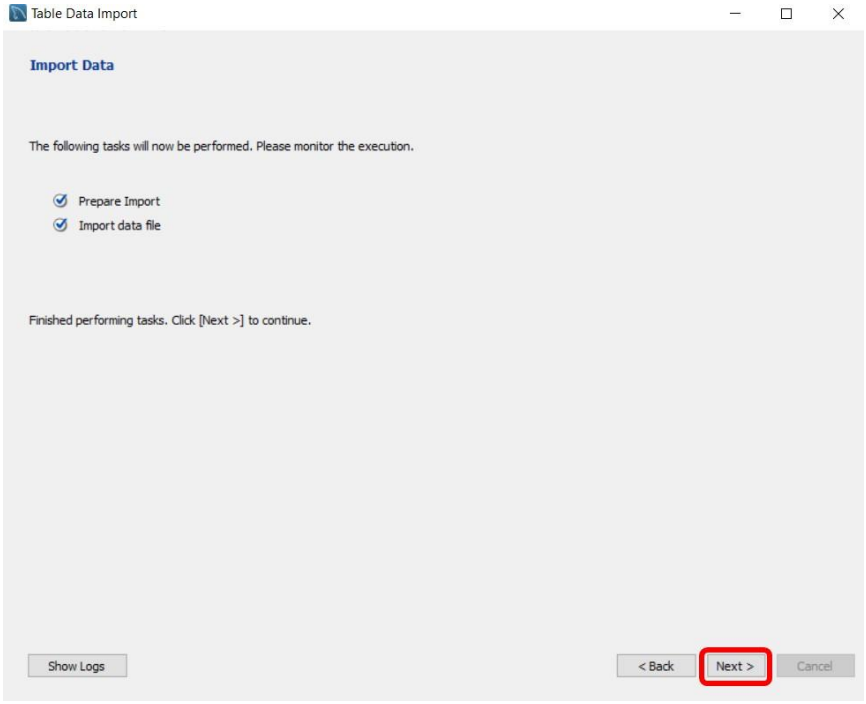
Step 4: Map the source and destination columns correctly and click “Next”.



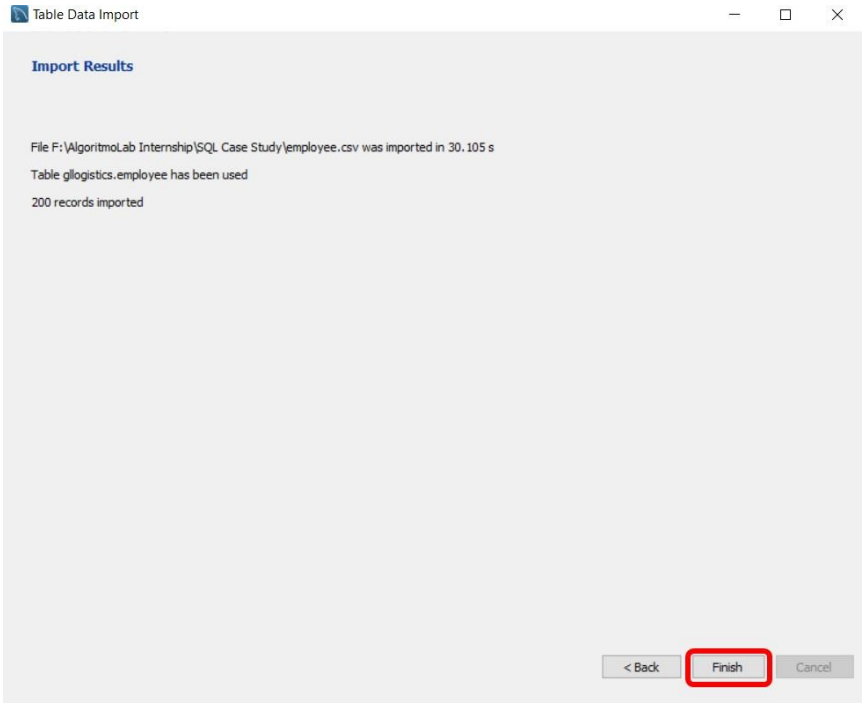
Step 5: Click “Next”



Step 6: Click “Next”



Step 7: Data imported successfully. Click “Finish”



NOTE: Monitor every step so that you don’t run into any error.

A glimpse of Employee_Details table:

E_ID	E_name	E_branch	E_designation	E_addr	E_Cont_no
2	Zoya	TN	Transport manager	400 Block of MASON ST	9250747856
7	Guy	TN	Transport manager	NATOMA ST / 6TH ST	9744154055
11	Rita	WI	Manager	300 Block of GOLDEN GATE AV	7945949825
18	Jennie	WA	Project director	2400 Block of SAN BRUNO AV	8311555346
23	Yasmeen	TX	In House logistics executive	1200 Block of HOWARD ST	2261795235
26	Dawn	PA	Non-executive director	200 Block of DIVISION ST	5544881818
31	Elliana	LA	Office manager	COLUMBUS AV / LOMBARD ST	9741124956
33	Jennifer	MA	Engineering department manager	FELL ST / STEINER ST	6522949251
47	Kaitlyn	MS	Chief finance officer	800 Block of MARKET ST	6223189308
48	Lawrence	CO	Inventory manager	100 Block of APTOS AV	3072705157
49	Maeve	MA	Chief finance officer	GENEVA AV / MISSION ST	5956508519
54	Johnnie	RI	In House logistics executive	100 Block of BRITTON ST	8682770474

DATA PREPROCESSING (Working with dates)

It is not necessary that the dates obtained from csv files will always be in the same format. They can be separated by '/' (slash) or by '-'. Also dates can be in any formatlike 'dd-mm-yy', 'dd-mm-yyyy', 'yyyy-mm-dd', etc and many more.

In tables PAYMENT and MEMBERSHIP, the date is in the format "%Y-%m-%d". Inthe STATUS table the date is of the format "%m/%d/%Y".

There can be some dates that are entered erroneously like 02/31/1999. There areonly 28 or 29 days in the month of February. But the date '31' can be wrongly entered.

Steps to perform while dealing with the dates after importing the data from csv files:

1. Look for erroneous dates

   There can be dates where the month is greater than 12.
   For example: Find the erroneous date from the column 'DELIVERY_DATE' inthe 'STATUS' table where the month is greater than 12.

```
SELECT DELIVERY_DATE FROM STATUS
     WHERE CAST(substring_index(DELIVERY_DATE, '/', 1) AS
     UNSIGNED) > 12;
```

```
--- ------------------------------------------------------------------------
```

   Search for the records where the month is February but the date is erroneously entered as 30 and 31.

```
SELECT * FROM STATUS
     WHERE CAST(substring_index(DELIVERY_DATE, '/', 1) AS
     UNSIGNED) = 2
     AND
     CAST(substring_index(substring_index(DELIVERY_DATE,
     '/', 2), '/', -1) AS UNSIGNED)  > 29;
```

```
SELECT * FROM STATUS
     WHERE CAST(substring_index(SENT_DATE, '/', 1) AS
     UNSIGNED) = 2
```

```
    AND
    CAST(substring_index(substring_index(SENT_DATE, '/',
    2), '/', -1) AS UNSIGNED)  > 29;

SELECT * FROM PAYMENT_DETAILS
    WHERE
    CAST(substring_index(substring_index(PAYMENT_DATE,
    '-', 2), '-', -1) AS UNSIGNED) = 2
    AND
    CAST(substring_index(PAYMENT_DATE, '-', -1) AS
    UNSIGNED) > 29;
```

2. Convert the string in the date format

```
UPDATE PAYMENT_DETAILS
    SET Payment_Date = STR_TO_DATE(Payment_Date,'%Y-%m-
    %d');
UPDATE STATUS
    SET Delivery_Date =
    STR_TO_DATE(Delivery_Date,'%m/%d/%Y'),
    Sent_Date = STR_TO_DATE(Sent_Date,'%m/%d/%Y');
UPDATE MEMBERSHIP
    SET Start_Date = STR_TO_DATE(Start_Date,'%Y-%m-%d'),
    End_Date = STR_TO_DATE(End_Date,'%Y-%m-%d');
```

3. Change the data type of the column to DATE

```
ALTER TABLE PAYMENT_DETAILS
    MODIFY COLUMN Payment_Date Date;
ALTER TABLE STATUS
    MODIFY COLUMN Delivery_Date Date,
    MODIFY COLUMN Sent_Date Date ;
ALTER TABLE MEMBERSHIP
    MODIFY COLUMN Start_Date Date,
    MODIFY COLUMN End_Date Date ;
```

CREATING A SINGLE SOURCE OF TRUTH (SSOT)

SSOT means creating a new table by joining all the available tables. Here we willcreate a new table ‘LOGISTICS’ by joining the tables EMPLOYEE, CLIENT,

MEMBERSHIP, SHIPMENT, PAYMENT, Employee_Manages_Shipment and STATUS.

```
CREATE TABLE logistics_Emp AS
SELECT
     emp.Emp_ID, ship.SD_ID, Cust.Cust_ID, pmt.PAYMENT_ID,
memb.M_ID,
    emp.Emp_NAME, emp.Emp_ADDR, emp.Emp_BRANCH,
emp.Emp_DESIGNATION, emp.Emp_CONT_NO,
    ship.SD_DOMAIN, ship.SD_CONTENT, ship.SD_ADDR,
ship.SD_ADDR, ship.SD_WEIGHT, ship.SD_TYPE, ship.SD_CHARGES,
    cust.Cust_NAME, cust.Cust_TYPE, cust.Cust_ADDR,
cust.Cust_CONT_NO, cust.Cust_EMAIL_ID,
    stat.SENT_DATE, stat.DELIVERY_DATE, stat.CURRENT_ST,
    pmt.AMOUNT, pmt.PAYMENT_STATUS, pmt.PAYMENT_DATE,
pmt.PAYMENT_MODE,
    memb.Start_Date, memb.End_Date

FROM
    EMPLOYEE_Details AS emp
         INNER JOIN
     employee_manages_shipment AS ems ON emp.Emp_ID =
ems.Employee_deatils_Emp_ID
         INNER JOIN
    SHIPMENT_Details AS ship ON ship.SH_ID =
ems.Shipment_SH_ID
           INNER JOIN
     CLIENT AS cust ON Cust.C_ID = ship.Customer_Cust_ID
           INNER JOIN
     STATUS AS stat ON ship.SH_ID = stat.SH_ID
           INNER JOIN
     PAYMENT AS pmt ON ship.SH_ID = pmt.Shipment_SH_ID
           INNER JOIN
     MEMBERSHIP AS memb ON memb.M_ID = cust.Membership_M_ID
;
select * from logistics_Emp;
```

EXPLORATORY DATA ANALYSIS (EDA)

1) Extract all the employees whose name starts with A and ends with A.

```
SELECT
    Emp_name
FROM
    Employee_Deatils
```

```
WHERE
    Emp_name LIKE 'A%A';
```

2) Find all the common names from Employee_Details names and Customernames.

```
SELECT DISTINCT(Emp_name) FROM Employee_Details WHERE
Emp_name IN (SELECT Cust_name FROM Customer AS cus);
```

3) Create a view 'PaymentNotDone' of those customers who have not paidthe amount.

```
CREATE VIEW PaymentNotDone AS
SELECT * FROM logistics_Emp
WHERE PAYMENT_STATUS = 'NOT PAID';
```

-- Selecting all the observations of the newly created view 'PaymentNotDone'SELECT * FROM PaymentNotDone;

4) Find the frequency (in percentage) of each of the class of the paymentmode

```
SET @total_count = 0;
SELECT COUNT(*) INTO @total_count FROM Pyament_Deatils;
SELECT
    PAYMENT_MODE,
    ROUND((COUNT(PAYMENT_MODE) / @total_count) * 100,2)
          AS Percentage_Contribution
FROM
    Payment_Details
GROUP BY PAYMENT_MODE;
```

5) Create a new column 'Total_Payable_Charges' using shipping cost andprice of the product.

```
ALTER TABLE logistics_Emp
     ADD COLUMN TOTAL_PAYABLE_CHARGES FLOAT AFTER AMOUNT;

UPDATE logistics_Emp
     SET TOTAL_PAYABLE_CHARGES = SH_CHARGES + AMOUNT;
SELECT TOTAL_PAYABLE_CHARGES FROM logistics_Emp;
```

6) What is the highest total payable amount ?

```
SELECT MAX(TOTAL_PAYABLE_CHARGES) FROM logistics_Emp;
```

7) Extract the customer id and the customer name of the customers whowere or will be the member of the branch for more than 10 years

```
SELECT Cust_ID, Cust_NAME, START_DATE, END_DATE,
ROUND(DATEDIFF(END_DATE, START_DATE)/365,0)
     AS Membership_Years FROM logistics_Emp
HAVING Membership_Years > 10;
```

8) Who got the product delivered on the next day the product was sent?

```
SELECT * FROM logistics_Emp
     HAVING DELIVERY_DATE-SENT_DATE = 1;
SELECT * FROM logistics_Emp
     HAVING DATEDIFF(DELIVERY_DATE, SENT_DATE)=1;
```

9) Which shipping content had the highest total amount (Top 5).

```
SELECT
    SH_CONTENT, SUM(AMOUNT) AS Content_Wise_Amount
FROM
    logistics_Emp
GROUP BY (SH_CONTENT)
ORDER BY Content_Wise_Amount DESC
LIMIT 5;
```

10) Which product categories from shipment content are transferred more?

```
SELECT SH_CONTENT, COUNT(SH_CONTENT)
     AS Content_Wise_Count
FROM logistics_Emp
GROUP BY(SH_CONTENT)
ORDER BY Content_Wise_Count DESC
LIMIT 5;
```

11) Create a new view 'TXLogistics' where employee branch is Texas.

```
CREATE VIEW TXLogistics AS
     SELECT * FROM logistics_Emp
          WHERE E_BRANCH = 'TX';

SELECT * FROM TXLogistics;
```

12) Texas(TX) branch is giving 5% discount on total payable amount. Createa new column 'New_Price' for payable price after applying discount.

```
ALTER VIEW TXLogistics
   AS SELECT *, AMOUNT - ((AMOUNT * 5)/100) AS New_Price
   FROM logistics_Emp
```

```
   WHERE E_BRANCH = 'TX';
SELECT * FROM TXLogistics;
```

13) Drop the view TXLogistics

```
DROP VIEW TXLogistics;
```

14) The employee branch in New York (NY) is shutdown temporarily. Thus,the the branch needs to be replaced to New Jersy (NJ).

```
SELECT * FROM logistics_Emp WHERE E_BRANCH = 'NY';

UPDATE  logistics_Emp
     SET E_BRANCH = 'NJ'
WHERE E_BRANCH = 'NY';

SELECT * FROM logistics_Emp;
```

15) Finding the unique designations of the employees.

```
SELECT DISTINCT(Emp_DESIGNATION) FROM Employee_Details;
```

16) We will see the frequency of each customer type (in percentage).

```
SET @total_count = 0;
SELECT COUNT(*) INTO @total_count FROM logistics_Emp;
SELECT Cust_TYPE, (COUNT(Cust_TYPE)/@total_count)*100
     AS Contribution FROM logistics_Emp
GROUP BY Cust_TYPE;
```

18) Rename the column SER_TYPE to SERVICE_TYPE.

```
ALTER TABLE logistics_Emp
CHANGE SER_TYPE SERVICE_TYPE VARCHAR (15);
```

19) Which service type is preferred more?

```
SELECT SERVICE_TYPE, COUNT(SERVICE_TYPE)
     AS Frequency
FROM logistics_Emp
GROUP BY SERVICE_TYPE
ORDER BY Frequency DESC;
```

20) Find the shipment id and shipment content where the weight is greaterthan the average weight.

```
SELECT SH_ID, SH_CONTENT, SH_WEIGHT FROM Shipment_Details
WHERE SH_WEIGHT > (SELECT AVG(SH_WEIGHT) FROM
Shipment_Details);
```

CONCLUSION

The rise in the demand of transportation of shipment from one place to another and due to development of better transportation facilities all around the globe, logistics hastaken a vital position in business processes all around the world. The increase in the amount of transfer of contents has also contributed to the development of logistics thisrapidly.

Due to all these factors it became necessary to keep track of all whereabouts of the shipments. Logistics system not only helps us to keep track of them but also providesus with better solutions and helps us to get maximum utilization of the available resources. Keeping track of any shipment and knowing its current status becomes easy.