Project Report

Module Name	WSQ Database Design and Implementation (SF)
Course Name	Postgraduate Diploma in Software Engineering
Assignment Title	Plan, Design, and Implement a Database for eCommerce Portal

Learner Name	**Assessor Name**
HOÀNG MINH TRUONG	**DR. NGUYEN TRUNG LAP,** **DR. HO KIM GIAU**

Date Issued	**Completion Date**	**Submitted On**
		20/11/2023

Plagiarism

Plagiarism covers both copying or paraphrasing with the source without proper understanding and citation. The source includes but is not limited to:

- Books, Journal Articles, Online resources
- Others Assignment or project
- Software program
- Images

The learners must submit their work without plagiarism while producing the evidence against the assessment criteria. They must apply proper citation and Harvard referencing. Their work must go through the Turnitin software to check for the similarity index. Please consult the relevant module instructor or mentor if you need any further advice.

Learner declaration
I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. Learner signature: TRUONG Date: 20/11/2023

Table of Contents

LIST OF FIGURE

LIST OF TABLE

1. Project Background

You currently work as a Data Engineer for Brightica design agency, where you design and implement data models for client-centric products. As part of the role, your manager Mr. Andrew assigned the project to develop an optimal database design to deliver Rich Internet Application for Boutiqa. Boutiqa is a marketplace for sellers to promote their products and for consumers to purchase with ease. The company wants to have a consumer-centric application with an enhanced user experience

2. Project Objective

The goal of the project is to build a database system, helping Boutiqua manage information effectively. At Boutiqua admin can view and manage information related to:

- Manage Seller's Data
- Manage Buyer's Data
- Manage Items' information

These entities are related to each other, easy to query according to many different requirements, this access is done based on permissions for different users, supporting the operations department to perform tasks. Helping analysing and evaluating the business situation, managing and maintaining website operations well.

Project are required to demonstrate your capabilities in the following areas:

• Planning of database use group
• Conceptual, Logical, and Physical design of the database
• Writing queries and stored procedures to optimize the system performance and management reports.

The scope of the project in this module is to design and develop and implement the database. The overview of the project is as below:
There are 3 types of users:
1. Sellers
2. Consumers
3. Administrator

3. Functional, Non-functional and Technical Requirements

Sellers should be able to perform following functions in the portal:
1. Register in the portal.
2. Update their Profile after logging in.
3. Maintain the product catalog to promote their products.

Consumers should be able to perform following functions in the portal:
1. Register in the portal.
2. Update their Profile after logging in.
3. Search products.
4. Choose products to view the details.
5. Add, edit, and remove items in the shopping cart.

Administrator should be able to perform following functions in the portal:

1. Administer user data.
2. Send bulk email invite to potential clients to register.

4. Task 1 Define

The goal of the project is to build a database system, helping Boutiqua manage information effectively. At Boutiqua admin can view and manage information related to:

- Manage Seller's Data
- Manage Buyer's Data
- Manage Items' information

These entities are related to each other, easy to query according to many different requirements, this access is done based on permissions for different users, supporting the operations department to perform tasks. Helping analysing and evaluating the business situation, managing and maintaining website operations well.

Base on the requirement we collect the following information

- One Seller can sell many items.
- Items will be identified and categories by system
- One items can be sell by many seller with many difference competitive prices.
- Price for each item will be decided by seller.
- Buyer can buy many items from difference shop.

5. Task 2 Design Database

5.1 Data Normalization

5.1.1 Sample Form

Assuming the data sample contains all the information of a transaction, each line will in turn contain basic information such as transaction time, customer information, item, seller, quantity,amount:

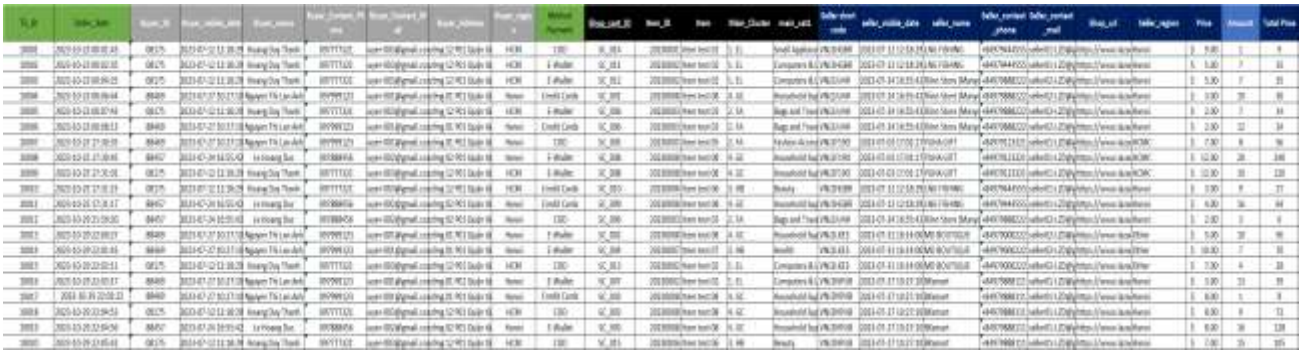


Figure 1 Sample Data Template

Apply the data normalization process to compact the above data sample.

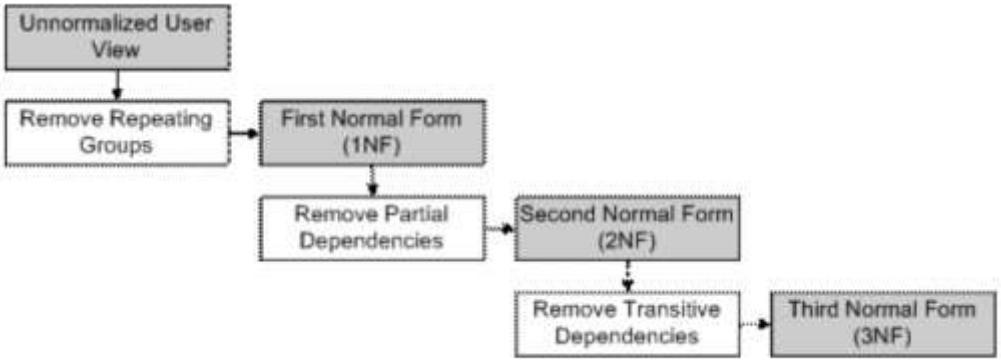


Figure 2 Normalization Process

5.1.2 First Normal Form

Identify repeating attribute.

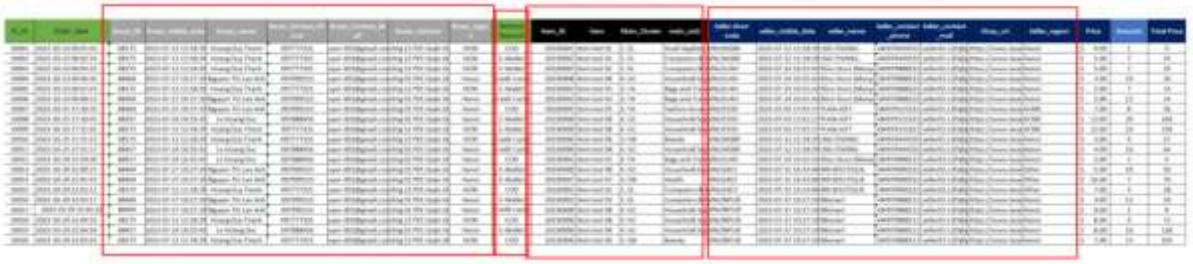


Figure 3 1NF - Identify repeating attribute

Remove these repeating attributes to a new table together with a copy of the key from the UNF table.

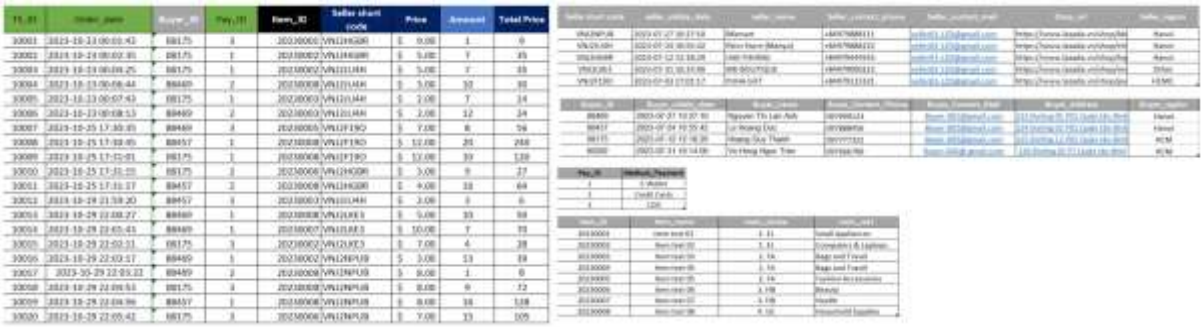


Figure 4 1NF-Final

Assign a key to the new table (and underline it). The key from the original un normalised table always becomes part of the key of the new table.

Dependence Diagram

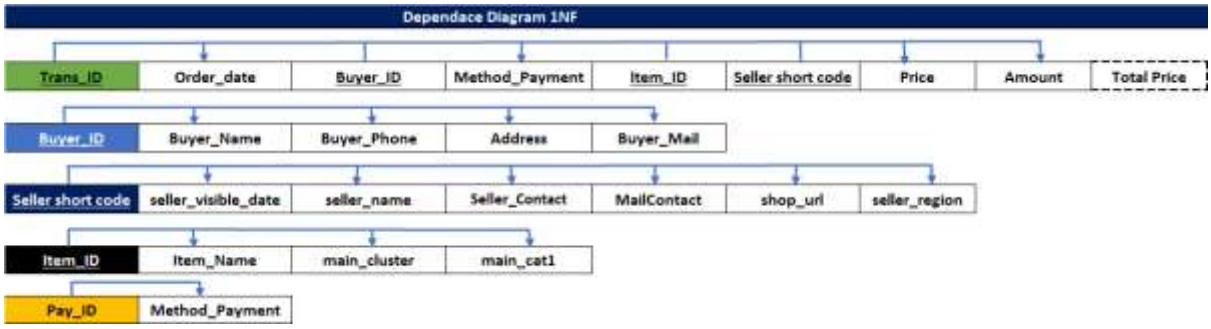


Figure 5 1NF - Dependence Diagram

- Total Price is a derived attribute.
- Buyer_Name, Buyer_Phone, Address, Buyer_Mail depend on Buyer_ID
- Seller_visible_date, Seller_name, Seller_contact, MailContact, shop_url, seller_region depend on Seller_short_code
- Item, main_cluster, main_cat1 depend on Iteam_ID
- Method_payment depend on Pay_ID

5.1.3 Second Normal Form

Figure 6 2NF - Converting to 2NF

Independence Diagram

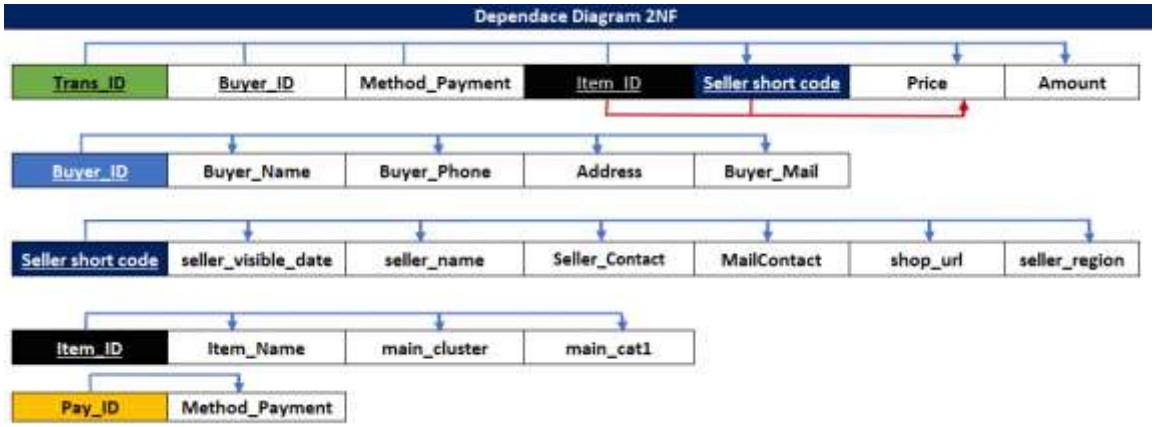


Figure 7 2NF - Dependency Diagram

In this case, there is one transitive dependence, that price will depend on Item_ID and Seller_short_code. That mean, one item will be sold by many seller with many difference price.
So we need to remove transitive dependence in 3NF.

5.1.4 Third Normal Form

Transaction

10001	08175	3	SC_014	1
10002	08175	1	SC_011	7
10003	08175	1	SC_012	7
10004	88469	2	SC_001	10
10005	08175	1	SC_006	7
10006	88469	2	SC_006	12
10007	88469	3	SC_005	8
10008	88457	1	SC_008	20
10009	08175	1	SC_008	10
10010	08175	2	SC_010	9
10011	88457	2	SC_009	16
10012	88457	3	SC_006	3
10013	88469	1	SC_002	10
10014	88469	1	SC_004	7
10015	08175	3	SC_013	4
10016	88469	1	SC_007	13
10017	88469	2	SC_003	1
10018	08175	3	SC_003	9
10019	88457	1	SC_003	16
10020	08175	3	SC_015	15

Method payment

Shop cart

Items

Figure 8 3NF - Breakdown into separate table(1)

Buyer

Seller

Figure 9 3NF - Breakdown into separate table(2)

Dependency Diagram

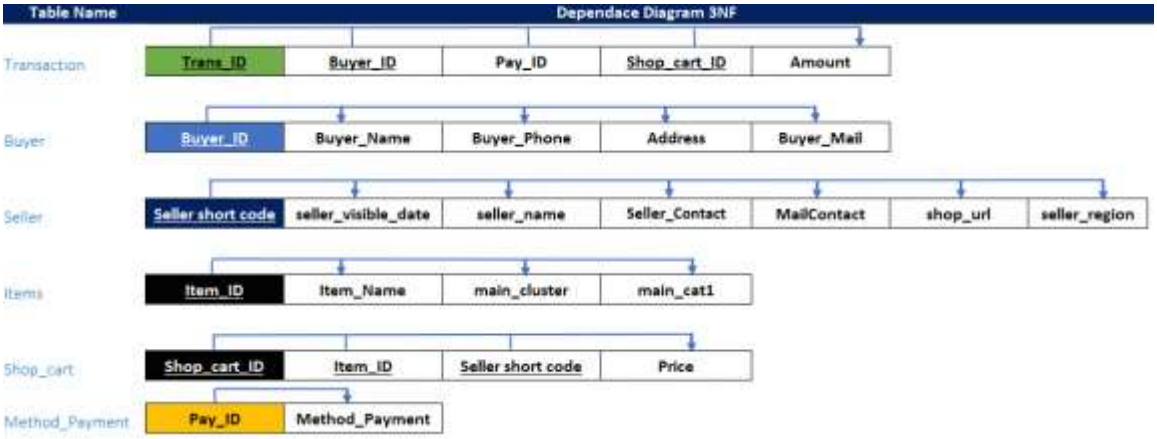


Figure 10 3NF - Dependency Diagram

5.2 Conceptual Design

ER Diagram

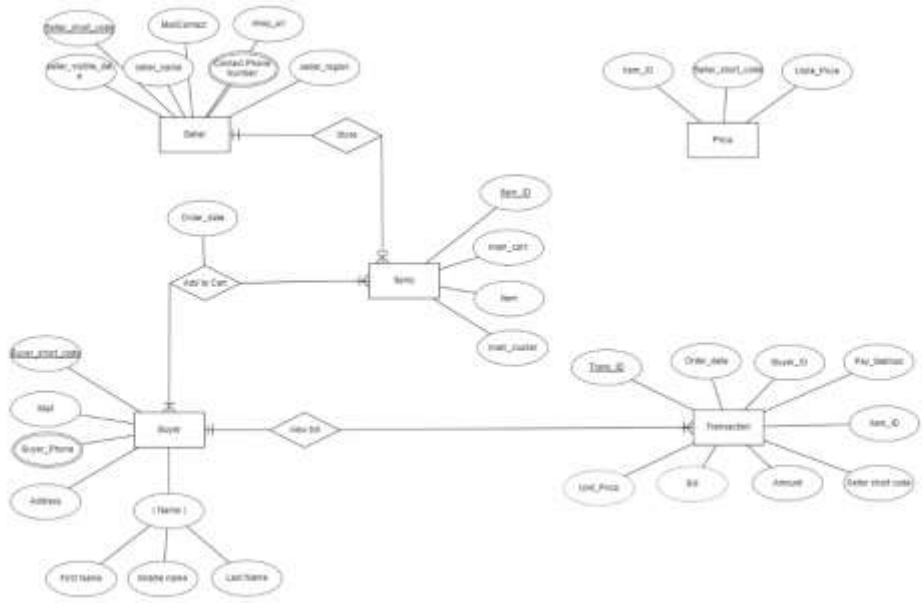


Figure 11 ERD by erdplus.com

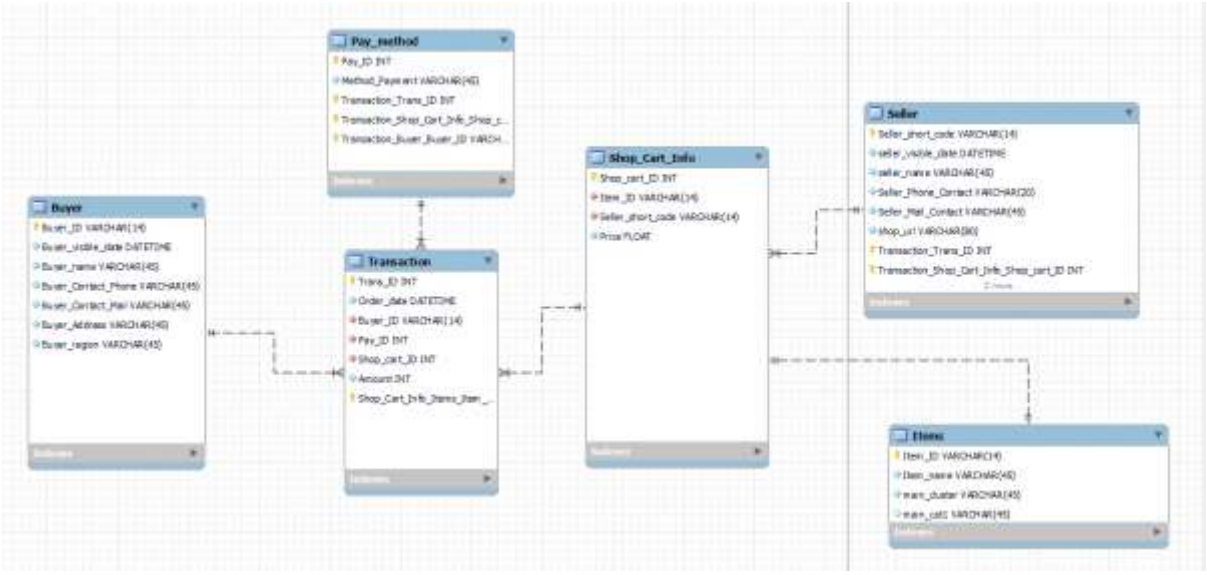


Figure 12 Detail ERD by MySQL WorkBench

5.3 Define Entities

5.3.1 Requirement gathering

Base on the requirement we collect the following information

- One Seller can sell many items.
- Items will be identified and categories by system
- One items can be sell by many seller with many difference competitive prices.
- Price for each item will be decided by seller.

- Buyer can buy many items from difference shop.

5.3.2 Organizing Entities & Attributes

Table 1 Organizing Attributes

Seller	Buyer	Item ID	Shop_cart_info	Transaction
Seller short code	Buyer_short_code	Item_ID	Shop_cart_ID	Trans_ID
seller_visible_date	Name	Item	Item_ID	Order_date
seller_name	Buyer_Phone	main_cluster	Seller_short_code	Buyer_ID
Contact Phone Number	Address	main_cat1	Price	Method Payment
MailContact	Mail			Purchase_ID
shop_url				Amount
seller_region				

6. Task 3 - Database Dictionary

Table 2 Data Dictionary

Data	Attributes	Choosing Appropriate Data Type	Data Types
Seller	Seller_short_code	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	VARCHAR(14)
	seller_visible_date	The first time seller start to appper, need to detail date & time so we can choose DATETIME	DATETIME
	seller_name	This will contain characters of variable length, So we can use VARCHAR(45)	VARCHAR(45)
	Seller_Phone_Contact	Phone will also contain characters of width up to 20 Characters. We can use VARCHAR(20), to Accommodateup to 2 numbers	VARCHAR(20)
	Seller_Mail_Contact	Email is also variable number of characters with size up to 80 Character	VARCHAR(80)
	shop_url	May be a link with long tail so we can choose TEXT	TEXT
	seller_region	This will contain characters of variable length, So we can use	VARCHAR(45)
Buyer	Buyer_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	VARCHAR(14)
	Buyer_visible_date	The first time seller start to appper, need to detail date & time so we can choose DATETIME	DATETIME
	Buyer_name	This will contain characters of variable length, So we can use VARCHAR(45)	VARCHAR(45)
	Buyer_Contact_Phone	Phone will also contain characters of width up to 20 Characters. We can use VARCHAR(20), to Accommodateup to 2 numbers	VARCHAR(20)
	Buyer_region	This will contain characters of variable length, So we can use VARCHAR(45)	VARCHAR(45)
	Buyer_Address	May be a link with long tail so we can choose TEXT	TEXT
	Buyer_Contact_Mail	Email is also variable number of characters with size up to 80 Character	VARCHAR(80)
Items	Item_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	VARCHAR(14)
	Item_name	This will contain characters of variable length, So we can use VARCHAR(45)	VARCHAR(45)

	main_cluster	This will contain characters of variable length, So we can use VARCHAR(45)	VARCHAR(45)
	main_cat1	This will contain characters of variable length, So we can use VARCHAR(45)	VARCHAR(45)
Shop_Cart_Info	Shop_cart_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	INT()
	Item_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	VARCHAR(14)
	Seller_short_code	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	VARCHAR(14)
	Price	Value of unit price, we can choose FLOAT	FLOAT()
Pay_method	Pay_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(2)	VARCHAR(2)
	Method_Payment	Should be one of some method payment, so we can choose VARCHAR(45)	VARCHAR(45)
Transaction	Trans_ID	Unit ID, contain NUMBER so we can choose INT()	INT()
	Order_date	The first time seller start to appper, need to detail date & time so we can choose DATETIME	DATETIME
	Buyer_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	VARCHAR(14)
	Pay_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(2)	VARCHAR(2)
	Shop_cart_ID	Unit ID, contain Text & number with lengh around 8, so we can choose VARCHAR(14)	Int()
	Amount	Amount of each item will be sold, I want to limit in smallin	SMALLINT()

7. Task 4 -MySQL Implement

7.1 Create talbes in MySQL

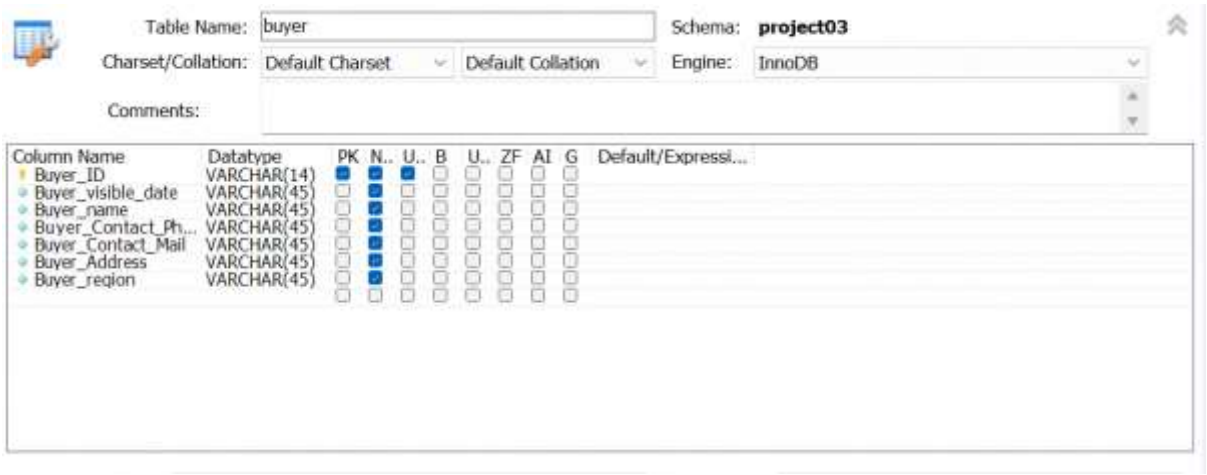


Figure 13 Table Buyer

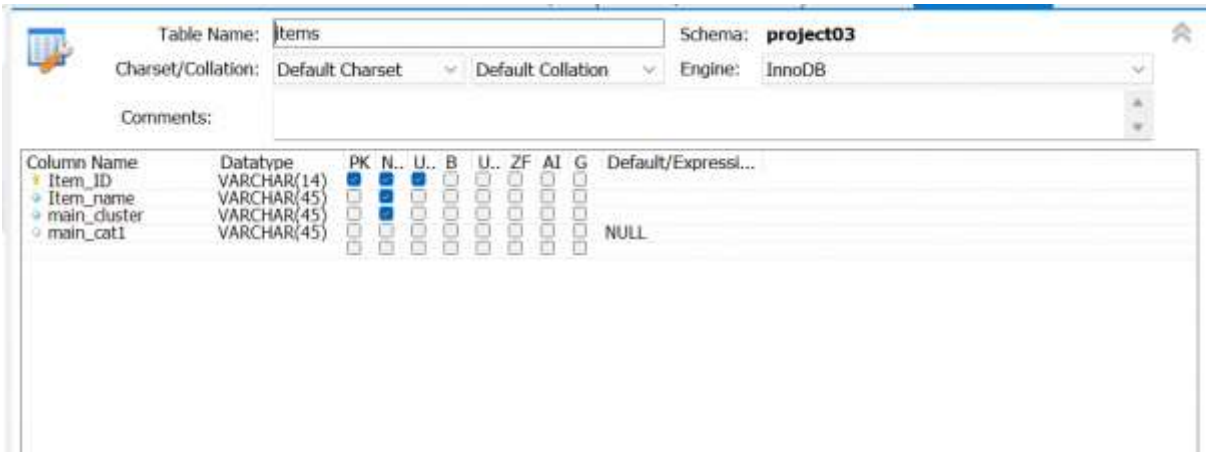


Figure 14 Item Diagram

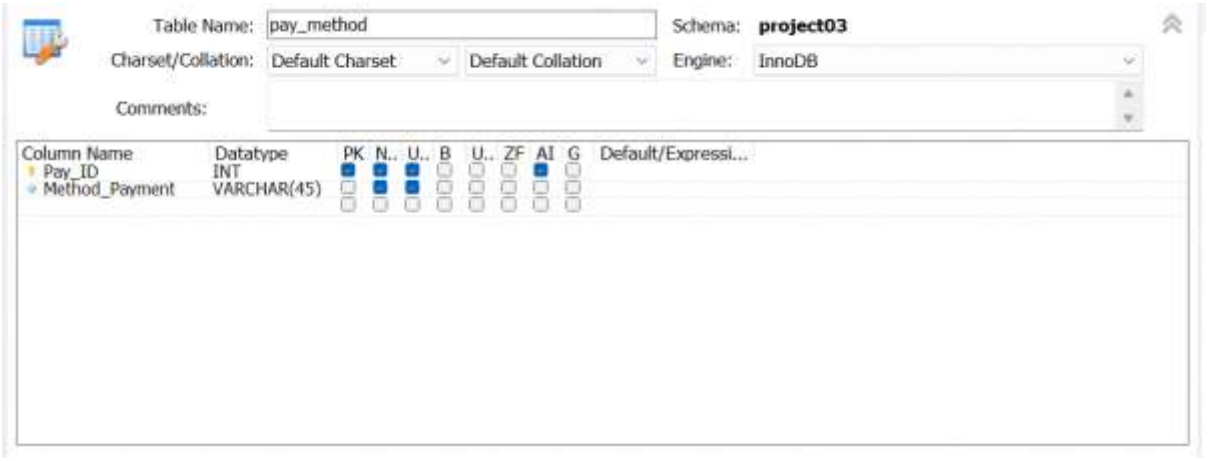


Figure 15 Table Payment Method

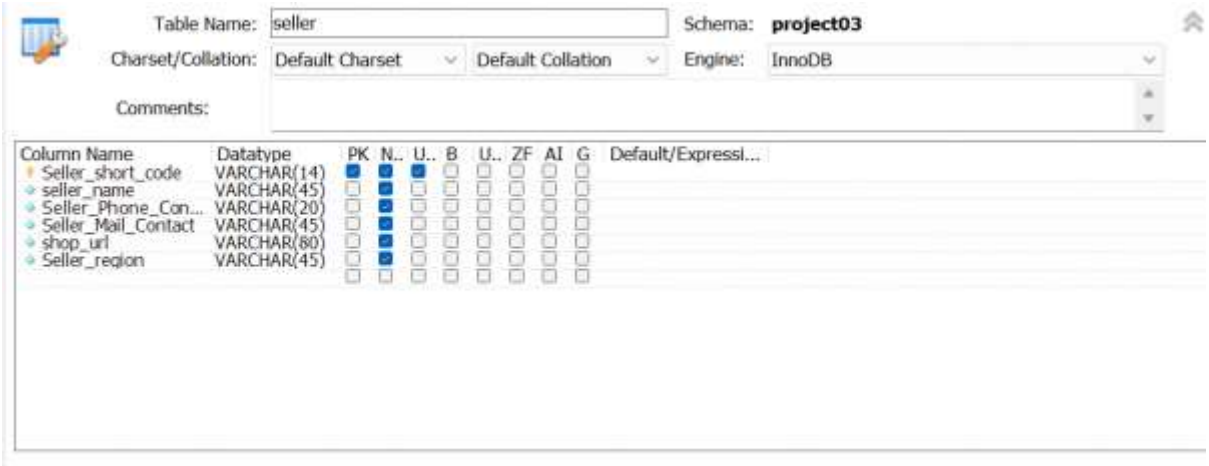


Figure 16 Table Seller

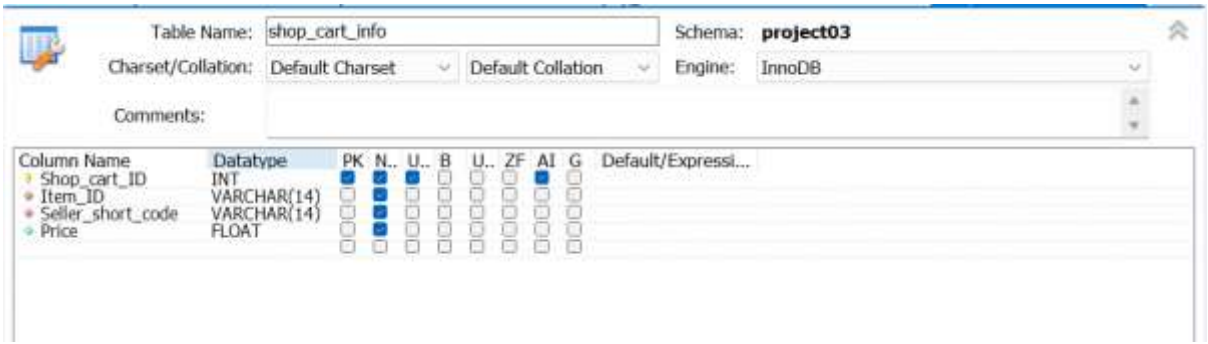


Figure 17 Table Shop_Cart_Info

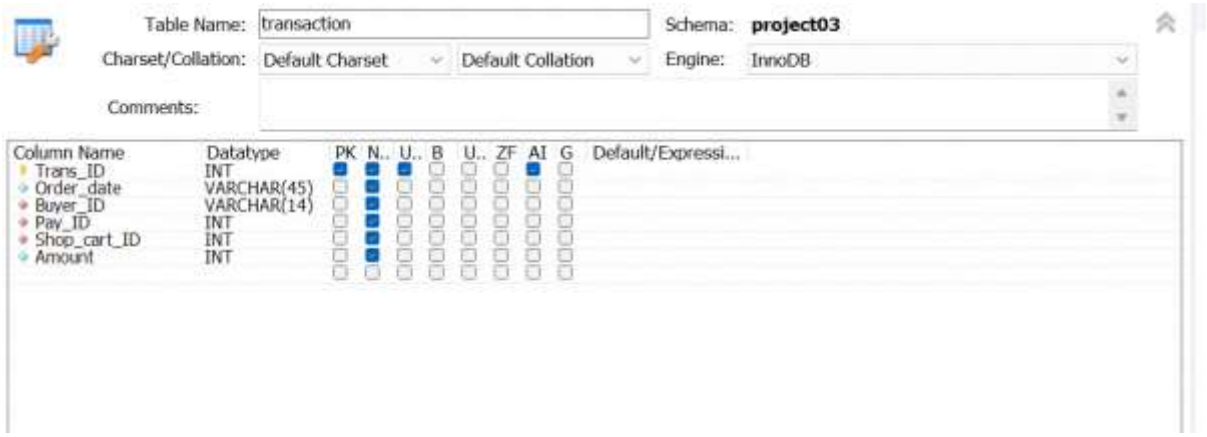


Figure 18 Table Transaction

7.2 List of pages with query and screenshot of result

Table 3 List Page Queries

PAGE	Purpose of Query	Query
Registration	For account creation of new Buyer	INSERT INTO project03.buyer VALUES ('88469','2023-07-27 10:27:10','Nguyen Thi Lan Anh','097999123','Buyer-001@gmail.com','123 Đường 01 P01 Thanh Xuaan','Hanoi');
	For account creation of new Seller	INSERT INTO project03.seller VALUES ('VNJ2NPUB','BKsmart','+84979888111','seller01-LZD@gmail.com','https://www.lazada.vn/shop/bksmart','Hanoi');
Update Profile	From UID Update Profile	UPDATE project03.seller SET seller_name = 'Seller audit 04' WHERE Seller_short_code = 'VNJ2LKE3';
CRUD Items	Maintain Product	INSERT INTO project03.items VALUES ('20230001','Item test 01 ','1. EL','Small Appliances'); UPDATE project03.items SET Item_name = 'Item_test_new'

		WHERE Item_ID = ‘20030001’
Create payment method	Insert one new Pay method	INSERT INTO project03.pay_method VALUES('1','E-Wallet');
Search Product	Search for Product	SELECT * from project03.Items WHERE Item_ID = ‘20030001’;
Create Catalouge	Create 1 catalouge	INSERT INTO project03.shop_cart_info VALUES ('1','20230008','VNJ2JU4H','3');
Create transaction	Create one transaction	INSERT INTO project03.transaction VALUES ('10001','2023-10-23 00:01:43','08175','3','14','1');

7.3 Create User

In this project, we have three main type user: Buyer, Seller, Admin.
Admin will login with ‘root’ account, have the highest privileges.
So we have to create account for users and devide privileges.

```
CREATE USER 'Buyer_Account'@'localhost' IDENTIFIED BY 'minhtruong';
CREATE USER 'Seller_Account'@'localhost' IDENTIFIED BY 'minhtruong';
```

7.2 Grant Privileges

```
-- ALTER ROLE FOR SELLER
GRANT CREATE, DELETE, INSERT, SELECT, UPDATE on project03.seller to
'Seller_Account'@'localhost';
GRANT CREATE, DELETE, INSERT, SELECT, UPDATE on project03.shop_cart_info
to 'Seller_Account'@'localhost';
GRANT SELECT on project03.items to 'Seller_Account'@'localhost';
-- ALTER ROLE FOR BUYER
GRANT CREATE, DELETE, INSERT, SELECT, UPDATE on project03.buyer to
'Buyer_Account'@'localhost';
GRANT SELECT on project03.shop_cart_info to 'Buyer_Account'@'localhost';
GRANT SELECT on project03.seller to 'Buyer_Account'@'localhost';
GRANT SELECT on project03.items to 'Buyer_Account'@'localhost';
GRANT SELECT on project03.pay_method to 'Buyer_Account'@'localhost';
GRANT CREATE, DELETE, INSERT, SELECT, UPDATE on project03.transaction to
'Buyer_Account'@'localhost';
```

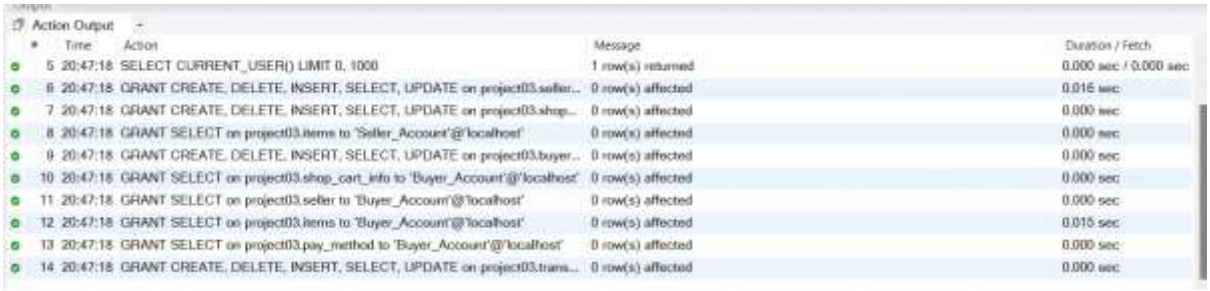


Figure 19 Create & Grant Privileges

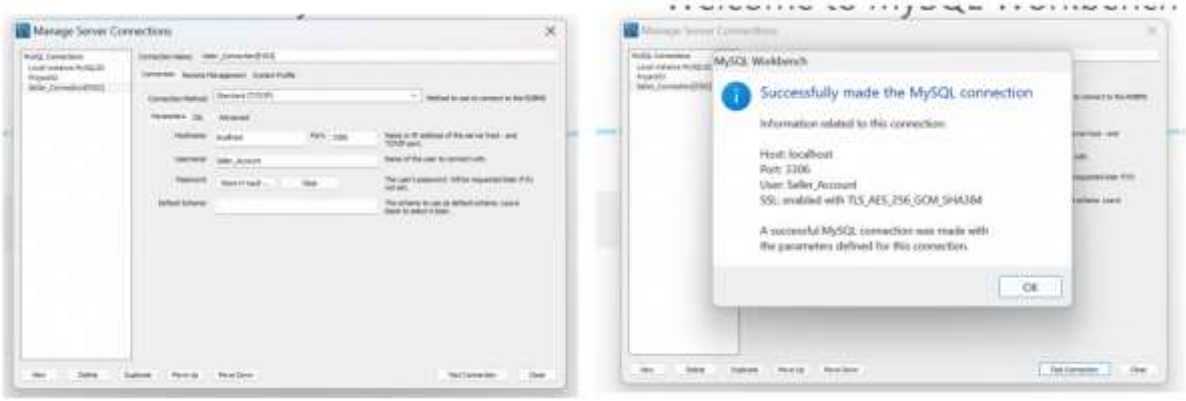


Figure 21Login - Seller Account

Figure 20 Login Buyer Account

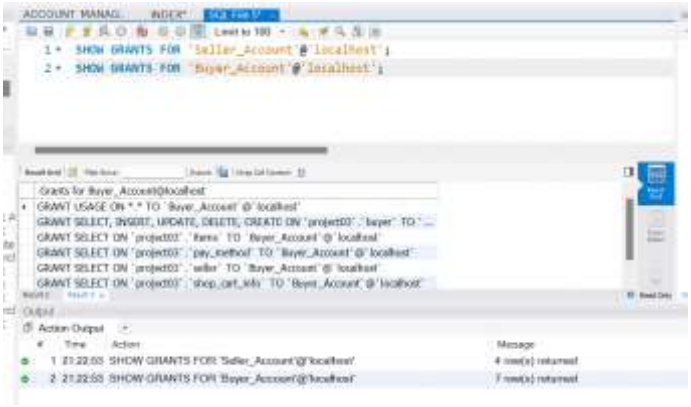
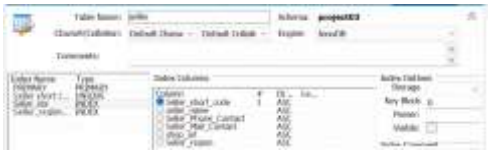


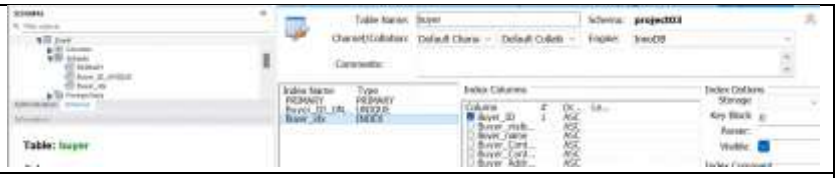
Figure 22 Show Grant for User

8. Task 5 – Index & Backup

8.1. Create Index

Table 4 Table of Index

Table	Index name	Query
project03.seller	Seller_idx Seller_region_idx	CREATE INDEX Seller_idx ON project03.seller(Seller_short_code ASC) VISIBLE; CREATE INDEX Seller_region_idx ON project03.seller(Seller_region ASC) VISIBLE; 
project03.items	Items_idx Items_cluster_idx	CREATE INDEX Items_idx ON project03.items(Item_ID ASC) VISIBLE; CREATE INDEX Items_cluster_idx ON project03.items(main_cluster ASC) VISIBLE; 
project03.pay_method	Pay_idx	CREATE INDEX Pay_idx ON project03.pay_method(Pay_ID ASC) VISIBLE; 
project03.buyer	Buyer_idx	CREATE INDEX Buyer_idx ON project03.buyer(Buyer_ID ASC) VISIBLE;

		
project03.shop_cart_info	Cart_idx	CREATE INDEX Cart_idx ON project03.shop_cart_info(Shop_cart_ID ASC) VISIBLE; 
project03.transaction	Trans_idx	CREATE INDEX Trans_idx ON project03.transaction(Trans_ID ASC) VISIBLE; 

8.2 Backup Database & Schedule Backup

8.2.1 BACKUP

Bakups enables you to recover data after several types ò crashes. Backups mút be scheduled regularly

CREATE USER ONLY FOR BACKUP

```
-- CREATE USER FOR BACKUP
CREATE USER 'User_Backup'@'localhost' IDENTIFIED BY 'minhtruong';
SHOW GRANTS FOR 'User_Backup'@'localhost';
```

Copy the backup command in a text file and save it at cp_backup.bat

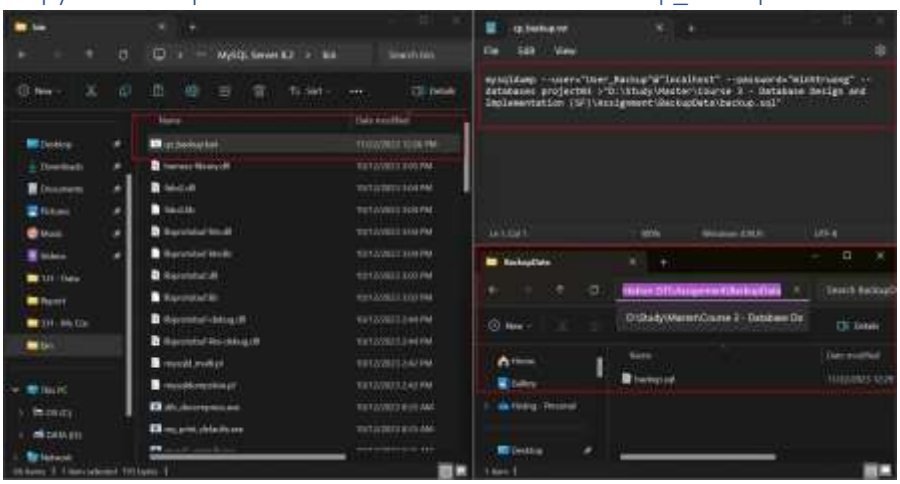


Figure 23 Create Backup DB

Create a bat file, in folder C:\Program Files\MySQL\MySQL Server 8.2\bin

Content:

mysqldump --user="User_Backup"@"localhost" --password="minhtruong" --databases project03 >"D:\Study\Master\Course 3 - Database Design and Implementation (SF)\Assignment\BackupData\backup.sql"

Run As Admin -> This command will auto create a identified file on folder
---- Reference : Doc + https://www.youtube.com/watch?v=lo30JnlRVEU

8.2.2 SCHEDULE BACKUP

Go to Task Schedule

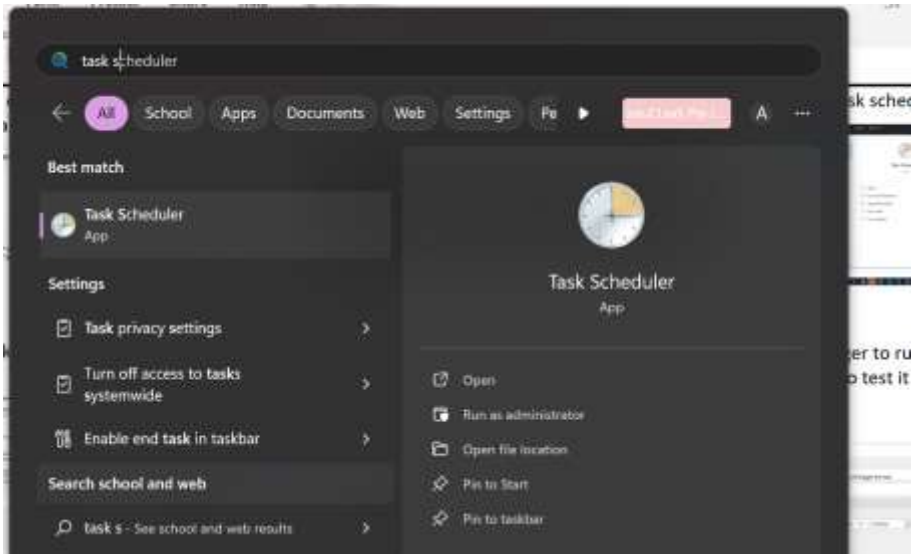


Figure 24 Open Task Schedule

Click Create Task & Enter Task Name

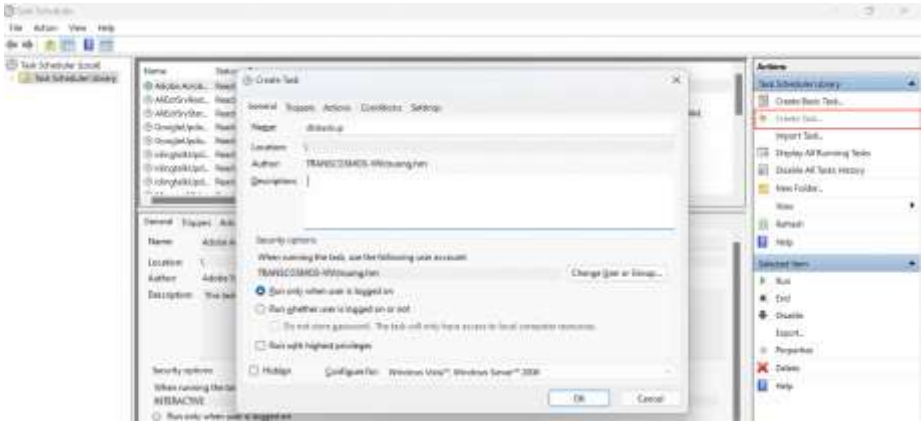


Figure 25 Create Task Name

Create the trigger to run the task at the scheduled time. To test it set as one time and verify it

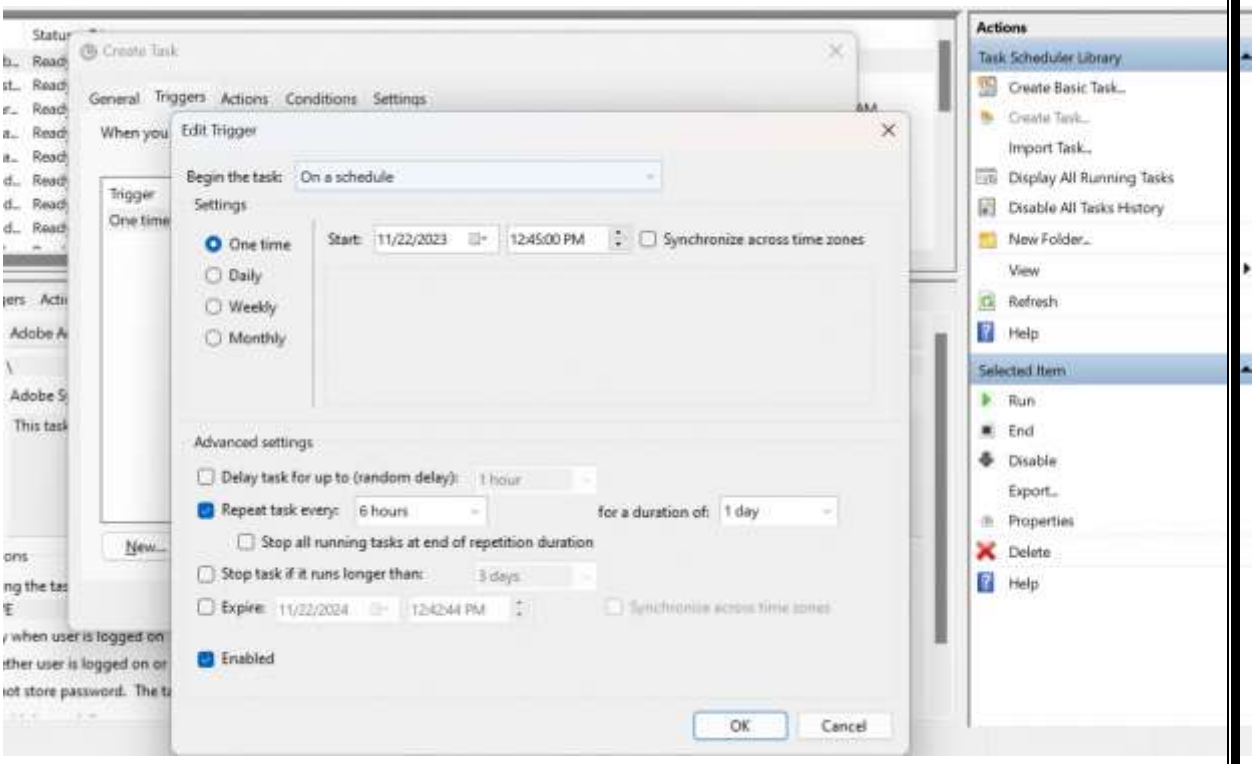


Figure 26 Select Detail for task

Set the action to be triggered. Browse and choose the batch file location

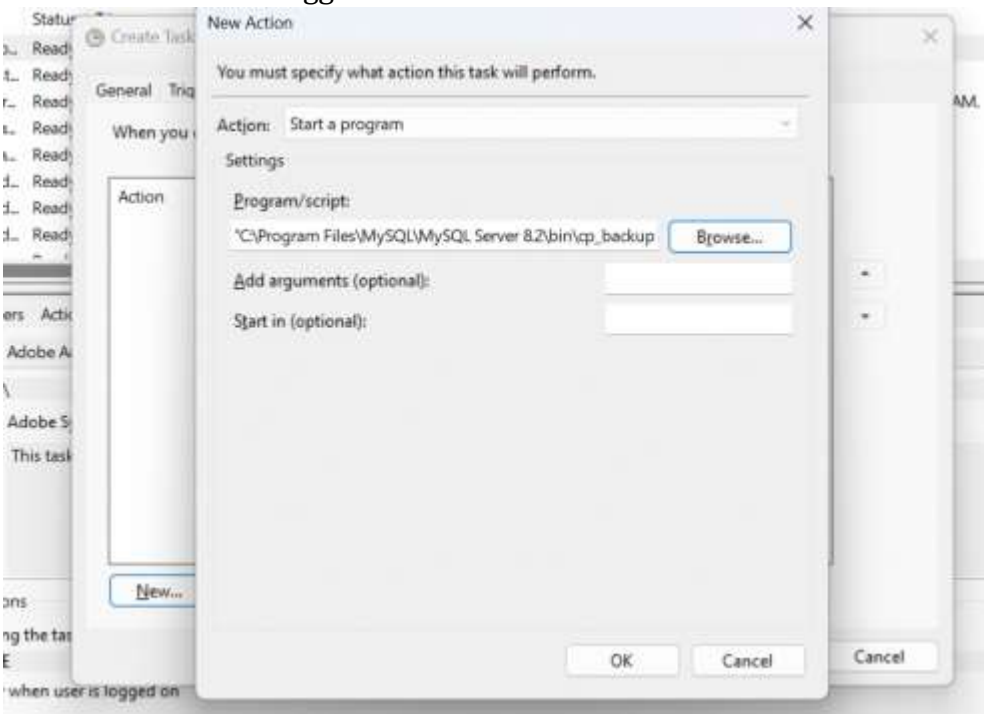


Figure 27 Select Action to be triggered

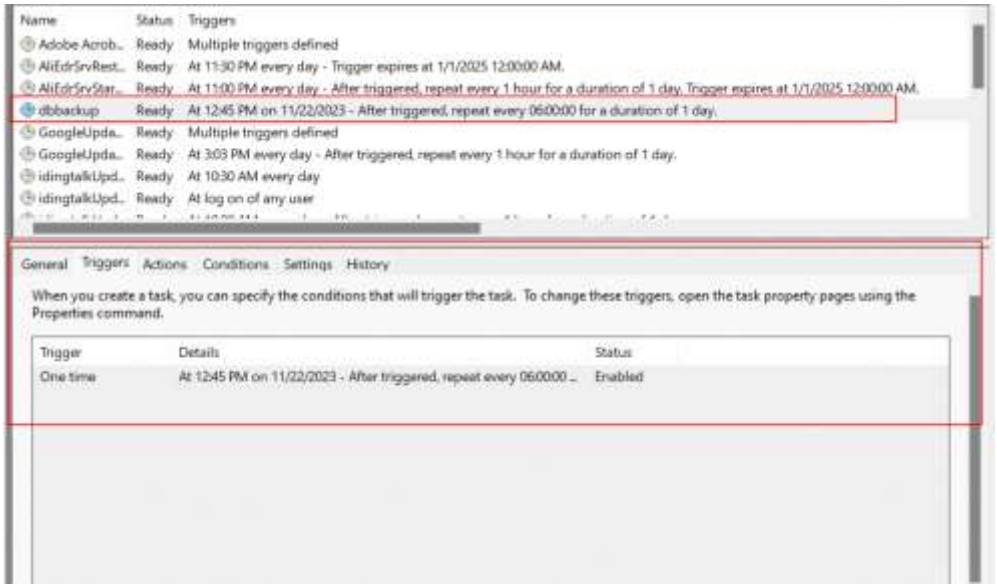


Figure 28 Check after scheduled time

Check the folder after the scheduled time. You will see your backup file

8.2.3 PROVIDE RESTORATION SCRIPT IN CASE OF FAILURE

Create new data base to contain data backup

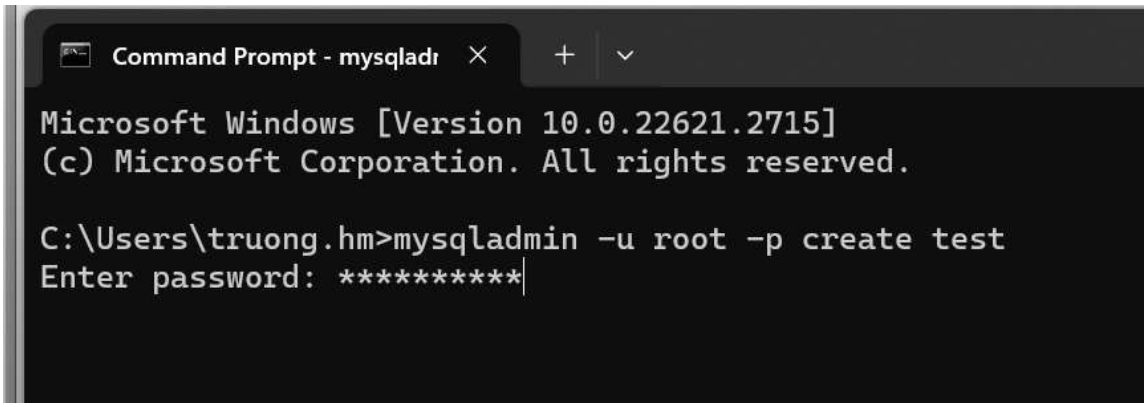


Figure 29 Create new Database

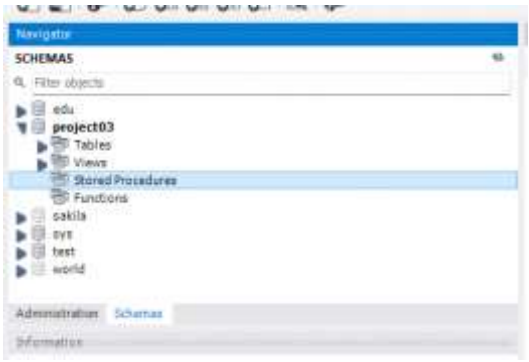


Figure 30 New Database

Use mysql to restore the database.

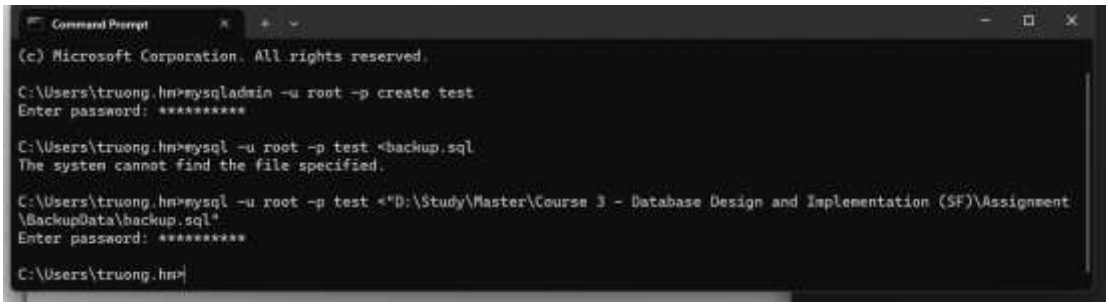


Figure 31 Promt restore the database

9. Task 6 – Apply MySQL

9.1 Create 8 SQL queries

In the report, I need to execute queries to answer the following questions:

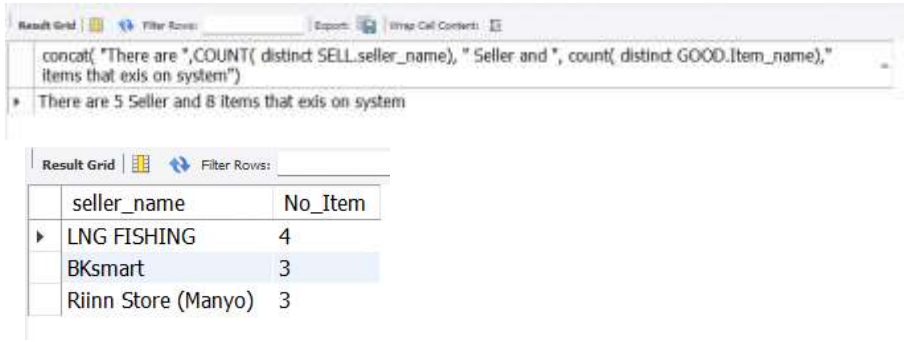
1. Top 3 Main_Cluster that bring the biggest GMV
2. How Many Seller, Items Exist On System
3. Top 3 Seller Have Most Item Buyed.
4. Total No Transaction & Total Gmv?
5. Top 3 Most Sold Products
6. Top 3 Main_Cluster That Bring The Biggest Gmv
7. Top 3 Main_Cluster That Bring The Biggest Gmv
8. How many customers are there in the system?
9. How many customers have purchased?
10. Which areas do most customers come from?
11. Percentage of customers returning to purchase
12. Buyer buys cost most?

I have organized the above questions into the following 3 reports:

9.1.1 RP 01 : OVERALL ABOUT SELLER

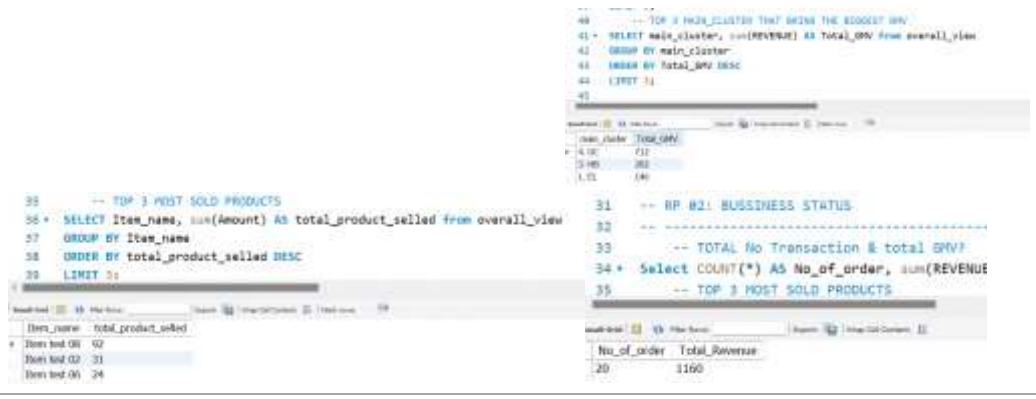
Table 5: Report 1- Overall About Seller

Title of the report	REPORT 01 OVERALL
Description	-- HOW MANY SELLER, ITEMS EXIST ON SYSTEM -- TOP 3 SELLER HAVE MOST ITEM BUYED.
Purpose	Overview about seller & Item have on system
Query	-- HOW MANY SELLER, ITEMS EXIST ON SYSTEM SELECT concat("There are ",COUNT(distinct SELL.seller_name), " Seller and ", count(distinct GOOD.Item_name)," items that exis on system") FROM project03.seller as SELL , project03.items AS GOOD; -- TOP 3 SELLER HAVE MOST ITEM BUYED. SELECT SELL.seller_name, COUNT(DISTINCT CART.Item_ID) AS No_Item FROM shop_cart_info AS CART INNER JOIN seller AS SELL ON CART.Seller_short_code = SELL.Seller_short_code GROUP BY SELL.seller_name ORDER BY No_Item DESC, SELL.seller_name ASC limit 3;

Result screenshot	

9.1.2 RP 02: BUSSINESS STATUS

Table 6 Report 2 - Bussiness Status

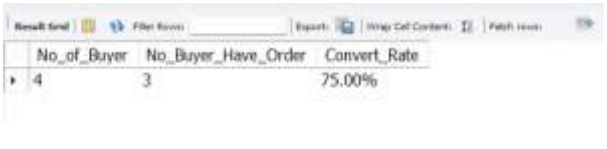
Title of the report	BUSSINESS STATUS
Description	-- TOTAL No Transaction & total GMV? -- TOP 3 MOST SOLD PRODUCTS -- TOP 3 MAIN_CLUSTER THAT BRING THE BIGGEST GMV
Purpose	To Deepdive more about business at this time
Query	-- TOTAL No Transaction & total GMV? Select COUNT(*) AS No_of_order, sum(REVENUE) AS Total_Revenue from overall_view; -- TOP 3 MOST SOLD PRODUCTS SELECT Item_name, sum(Amount) AS total_product_selled from overall_view GROUP BY Item_name ORDER BY total_product_selled DESC LIMIT 3; -- TOP 3 MAIN_CLUSTER THAT BRING THE BIGGEST GMV SELECT main_cluster, sum(REVENUE) AS Total_GMV from overall_view GROUP BY main_cluster ORDER BY Total_GMV DESC LIMIT 3;
Result screenshot	

9.1.2 RP 03: BUYER BEHAVIOR

Table 7 Report 03 - Buyer Behavior

Title of the report	03: BUYER BEHAVIOR
Description	-- How many customers are there in the system? -- How many customers have purchased? -- Which areas do most customers come from? -- Percentage of customers returning to purchase -- Buyer buys cost most?
Purpose	To understand more about customer's behavior
Query	-- How many customers are there in the system? -- How many customers have purchased? select (Select count(*) from buyer) as No_of_Buyer, (Select count(Distinct Buyer_ID) from transaction) as No_Buyer_Have_Order, Concat(ROUND(((Select count(Distinct Buyer_ID) from transaction)/(Select count(*) from buyer)),2)*100,"%") as Convert_Rate from buyer, transaction limit 1; -- Which areas do most customers come from? Select BUY.Buyer_region, count(*) as No_of_order from transaction AS TRANS inner join buyer AS BUY on TRANS.Buyer_ID = BUY.Buyer_ID GROUP BY BUY.Buyer_region ORDER BY No_of_order DESC LIMIT 1; -- Percentage of customers returning to purchase CREATE VIEW TEMP1 AS Select transaction.Buyer_ID, count(*) AS No_of_Order from transaction GROUP BY transaction.Buyer_ID -- ------- SELECT (SELECT COUNT(*) FROM TEMP1) AS No_Buyer, (SELECT COUNT(*) FROM TEMP1 WHERE No_of_Order > 1) AS No_Buyer_Re_Buy, CONCAT(ROUND(((SELECT COUNT(*) FROM TEMP1 WHERE No_of_Order > 1)/(SELECT COUNT(*) FROM TEMP1))*100,0),"%") AS RETURN_RATE FROM TEMP1 LIMIT 1; -- Buyer buys cost most? SELECT Buyer_name, sum(REVENUE) AS Total_Revenue FROM overall_view GROUP BY Buyer_name ORDER BY Total_Revenue LIMIT 1;

Result screenshot



No_of_Buyer	No_Buyer_Have_Order	Convert_Rate
4	3	75.00%

Buyer region	No. of order
Hanoi	11

9.2 Import Data through.csv file

Import data through .csv in My SQL Workbench

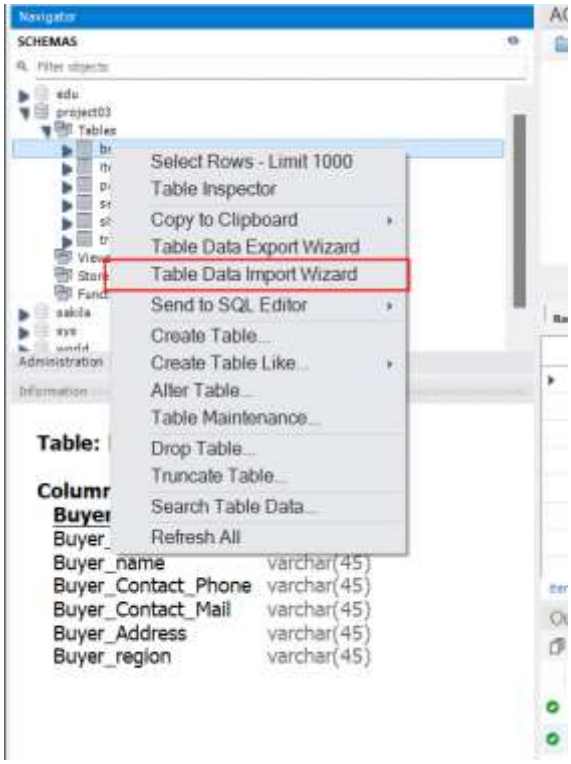


Figure 32 Step1-Import .csv file

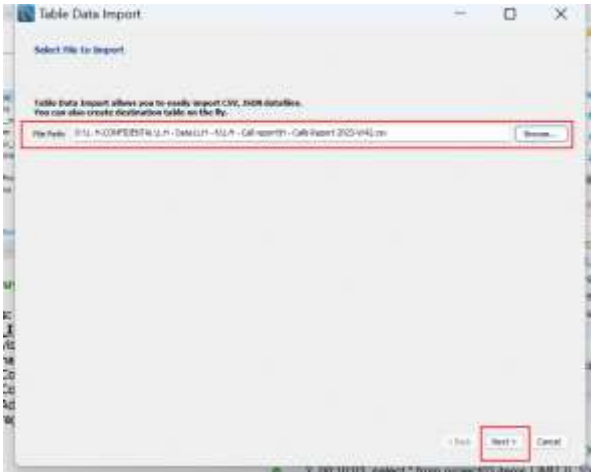


Figure 33 Step2 - Import .csv file

Import file into a new table

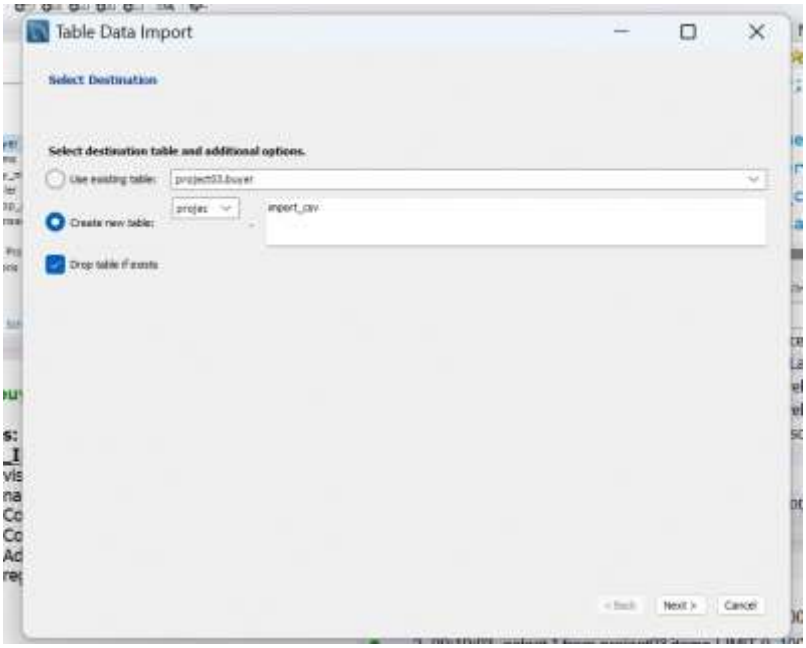


Figure 34 Step3 - insert csv into a new table

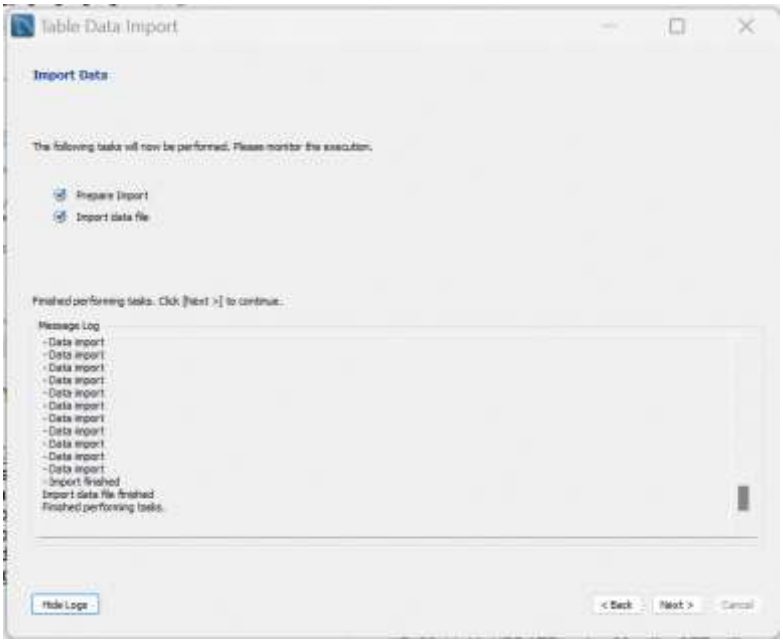


Figure 35 Import .csv processing

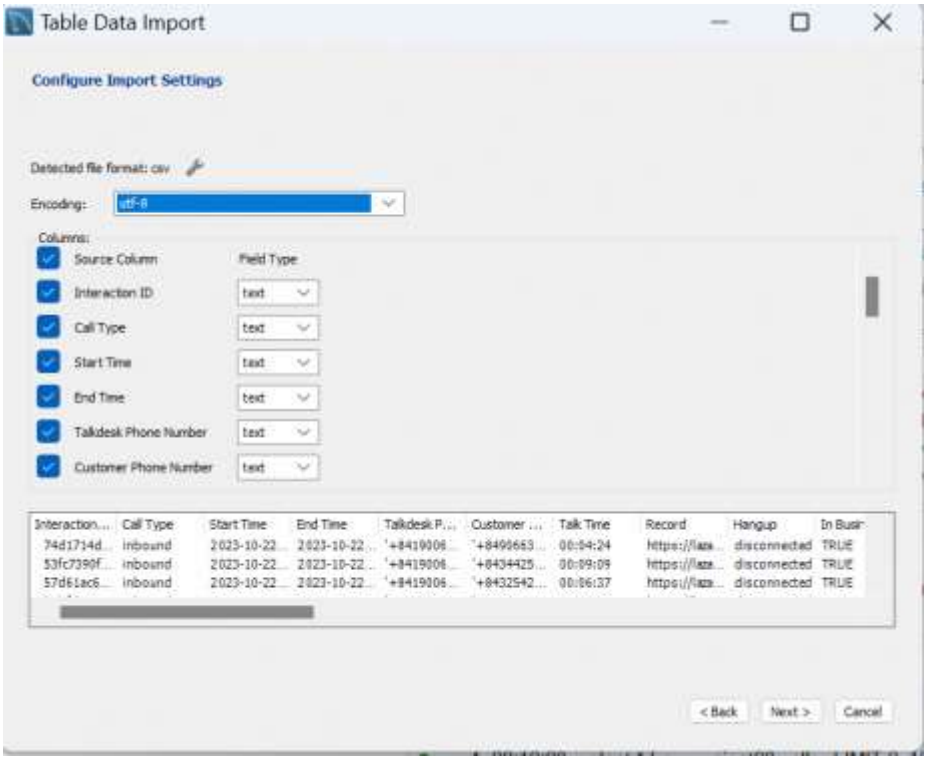


Figure 36 Declare the data format loaded from the .csv file

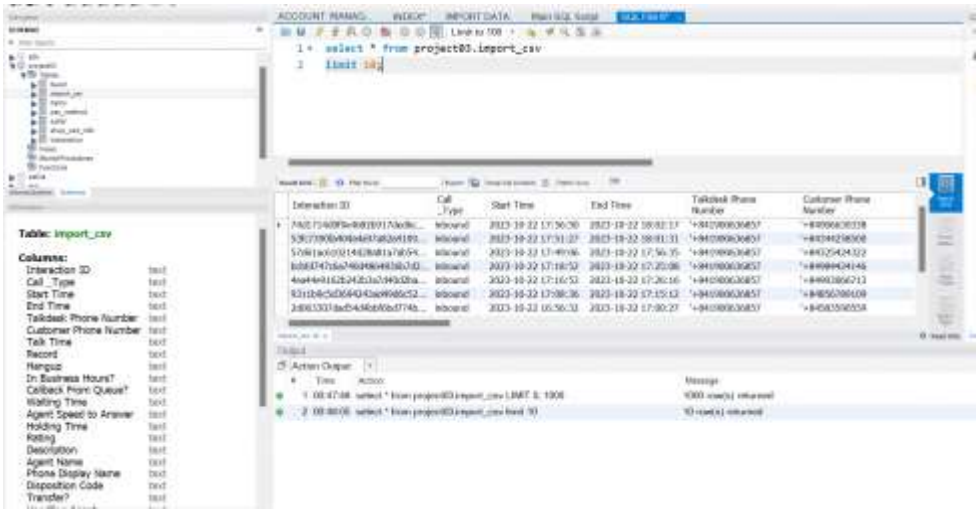


Figure 37 Complete the file loading process