IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Rathnayaka Samarakoon
Student Reference Number: 10820263

Module Code: PUSL2019	Module Name: Informational Management and Retrieval
Coursework Title: PUSL2019 Informational Management and Retrieval Coursework	
Deadline Date: 23/01/2023	Member of staff responsible for coursework: 06
Programme: Sql	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Maleesha S Ramasinghe - 10820265
Rankoth Jayasekara - 10820280
Navoda Ravishani - 10820266
Galappaththi Dinujaya - 10820276
Bokaragoda Wanshanatha - 10820934

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group: 

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have not used translation software.

If used, please state name of software……

Overall mark _____% Assessors Initials _____ Date_____

*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

TABLE OF CONTENTS

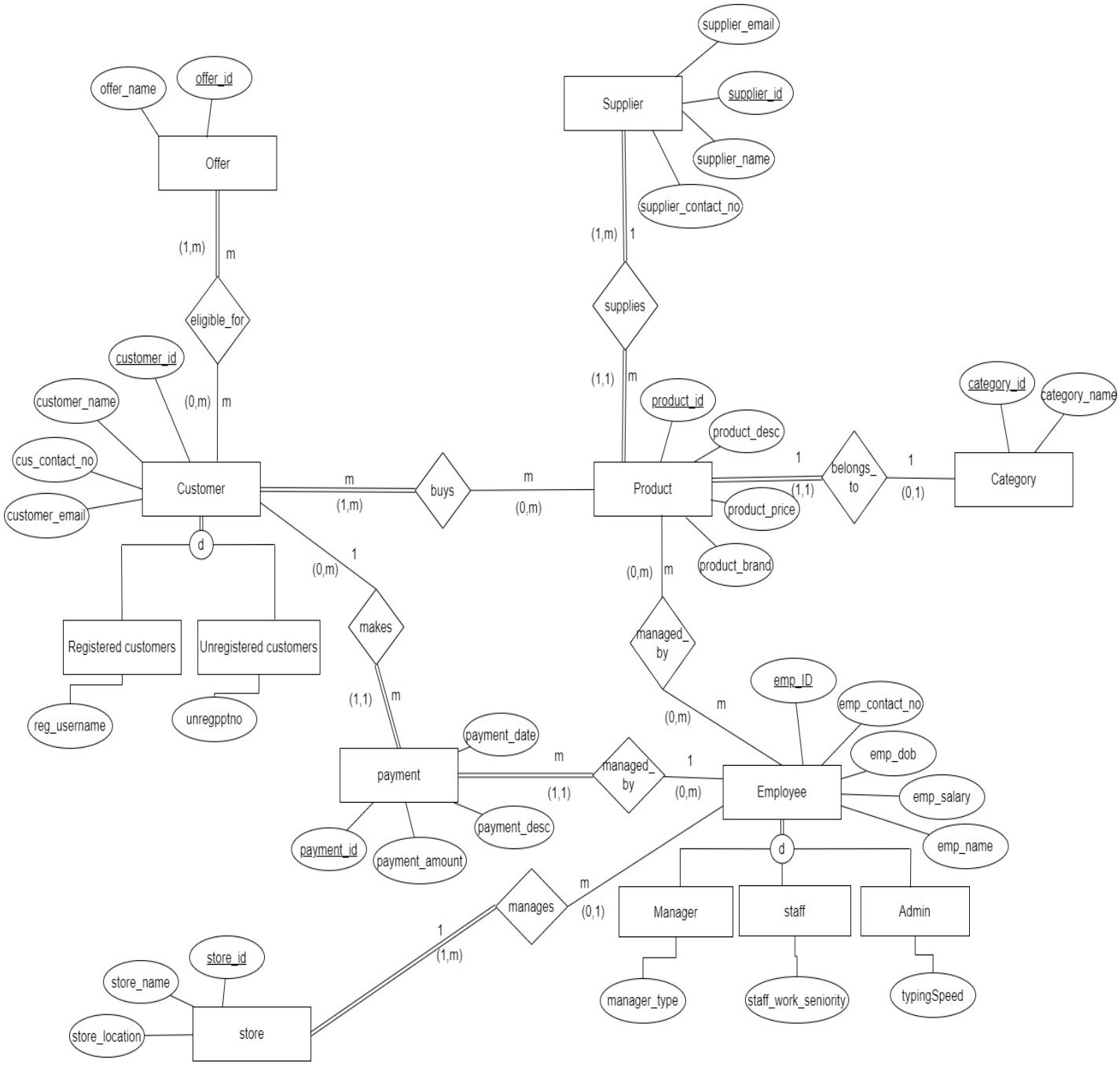
Section 1 - Introduction

This database application is created to a supermarket to manage their total tasks including the stock management operations, billing operations, and finance operations. A supermarket is a large collection of data, thus managing this data requires a complicated and advanced database. This will aid with time management, boost supermarket productivity, and provide a spotless service.

Customers browse the aisles of a supermarket and select the items they want to purchase. A customer takes his selected items to the cashier to pay for them after making the selection. A supermarket needs a database system that can properly determine the cost of a customer's purchases, work to guarantee that the supermarket doesn't run out of things that consumers wish to buy and accept payments from customers.

The database of the system will store all the necessary information such as customer details, employee details, product details, and payment details, category details, store details, offer details, and supplier details. The purpose of the database design for the supermarket management system is to support the management of the process for purchasing goods, billing, and making the purchase transaction quick and accurate while maintaining and protecting the sales records. It makes it easier for both the owner and the employee to control or influence corporate transactions.

EER Diagram


supplier_email
supplier_id
supplier_name
supplier_contact_no
Supplier
offer_name
offer_id
Offer
(1,m)
m
1
supplies
eligible_for
(1,1)
m
customer_id
customer_name
(0,m)
m
product_id
product_desc
category_id
category_name
cus_contact_no
customer_email
Customer
m
(1,m)
buys
m
(0,m)
Product
1
(1,1)
belongs_to
1
(0,1)
Category
product_price
product_brand
d
1
(0,m)
(0,m)
m
makes
managed_by
Registered customers
Unregistered customers
reg_username
unregpptno
(1,1)
m
emp_ID
emp_contact_no
emp_dob
emp_salary
emp_name
payment_date
payment
m
(1,1)
managed_by
1
(0,m)
Employee
m
(0,m)
payment_id
payment_amount
payment_desc
d
manages
m
(0,1)
1
(1,m)
Manager
staff
Admin
store_id
store_name
store_location
store
manager_type
staff_work_seniority
typingSpeed

ER diagram assumptions

Entity related assumptions

- Assuming customer can either be registered or unregistered and cannot belong to both groups at the same time
- Assuming employee can be either a manager or admin or staff and one employee cannot belong to more than one of these groups.
- Assuming every employee should belong to one of the groups either manager or admin or staff
- Assuming every customer should belong to one of the groups either registered or unregistered

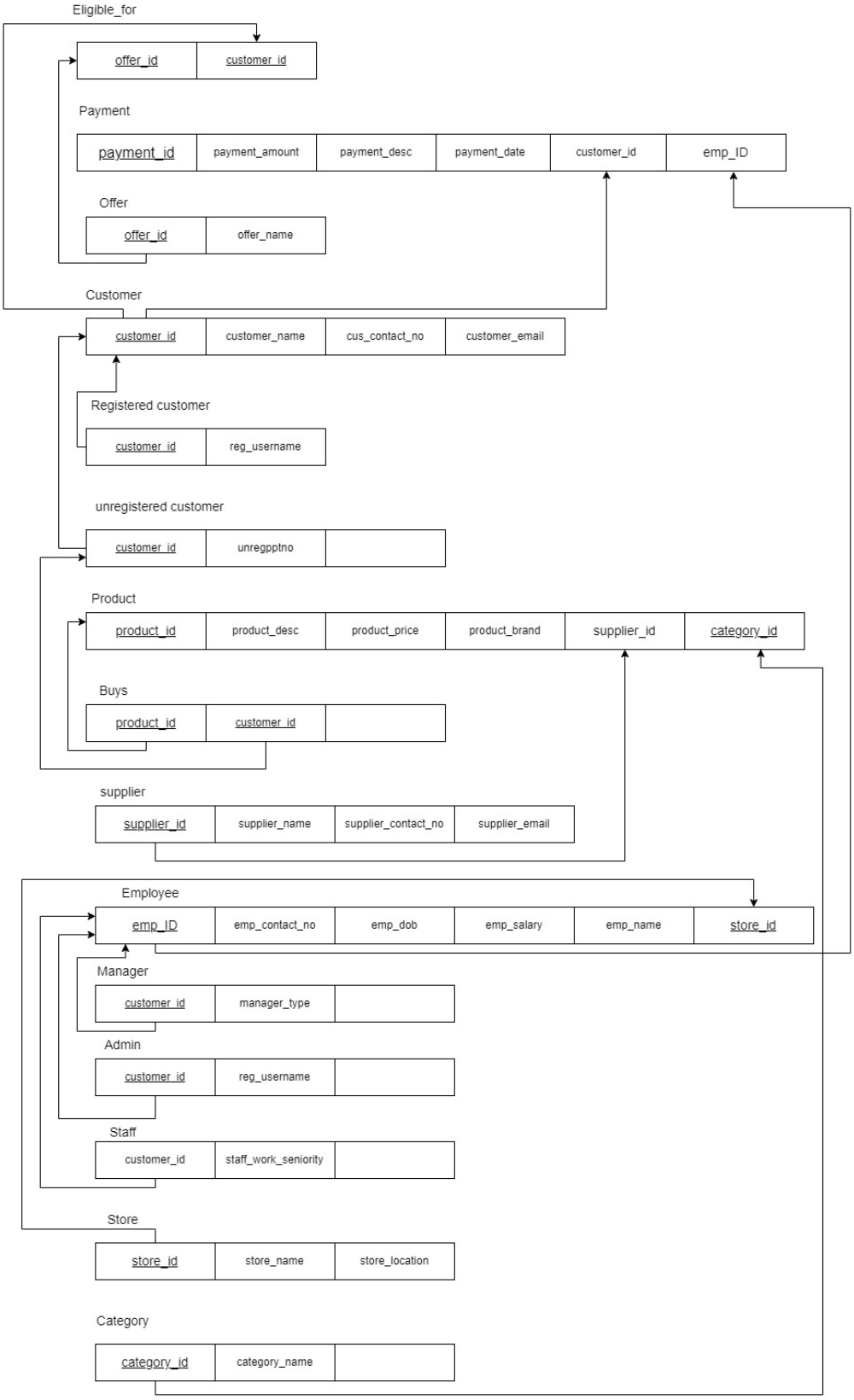
Connectivity related assumptions

- Assuming a customer can be eligible for many offers and one offer may be eligible for many customers
- Assuming a supplier can supply many products a product belongs to only one supplier
- Assuming a customer can buy many products and a product can be purchased by many customers
- Assuming a customer can make many payments but a payment can belong only to one customer
- Assuming a product can be managed by many employees and many employees can manage many products at the same time
- Assuming many employees can manage only one store at a particular time
- Assuming a product should have a category and a category can have a product

Cardinality related assumptions

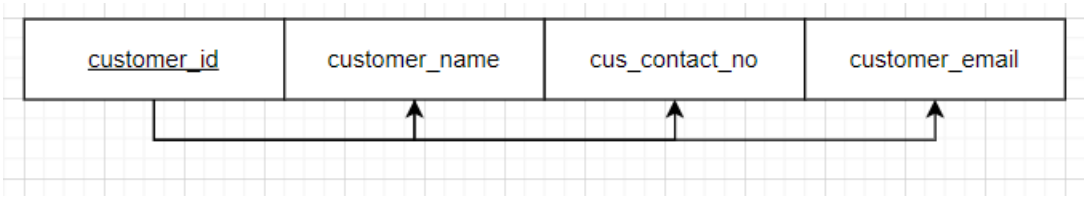
- Assuming every payment has a customer and every customer does not make a payment
- Assuming every payment is handled by an employee but every employee does not handle payments
- Assuming every store is managed by an employee but every employee does not manage stores
- Assuming every product belongs to a category but every category does not have products
- Assuming each and every supplier supplies products and each and every product has a supplier
- Assuming every customer is not eligible for offers but every offer must have customers
- Assuming every customer purchase products but every product is not purchased by a customer

Relational mapping



Data Normalization

Customer



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

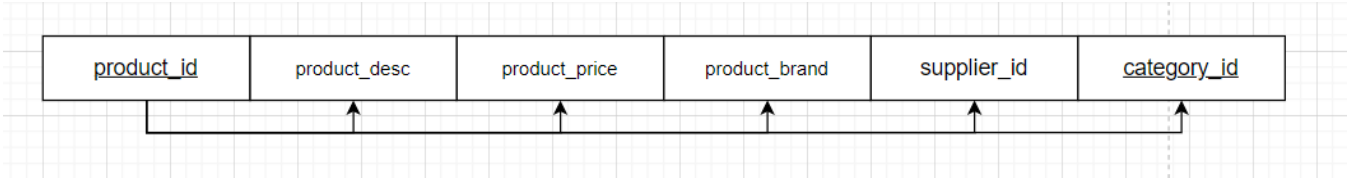
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Product



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

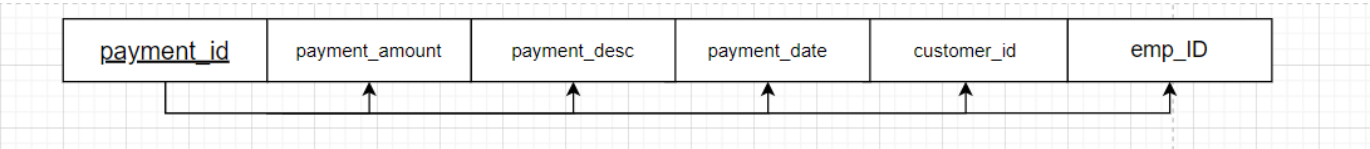
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Payment



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

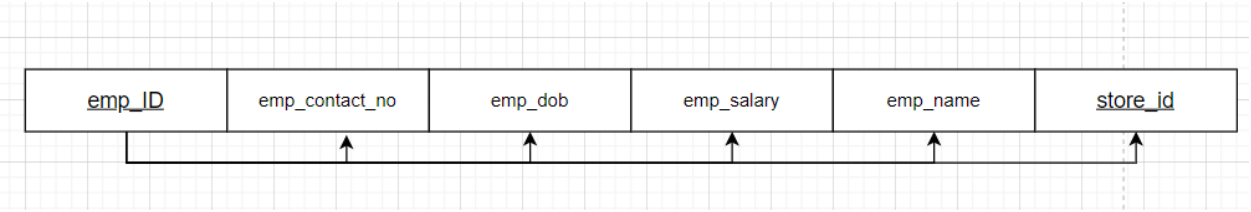
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Employee



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

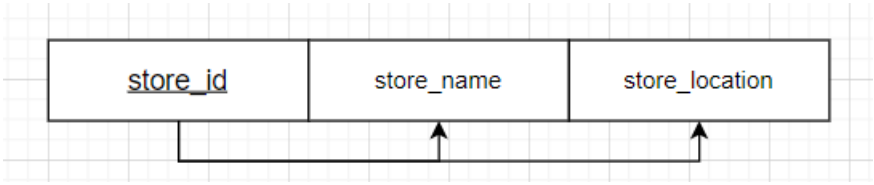
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Store



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

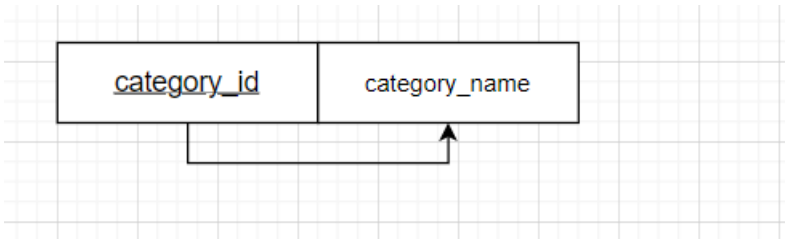
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Category



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

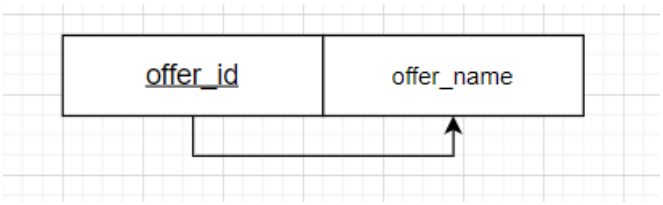
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Offer



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

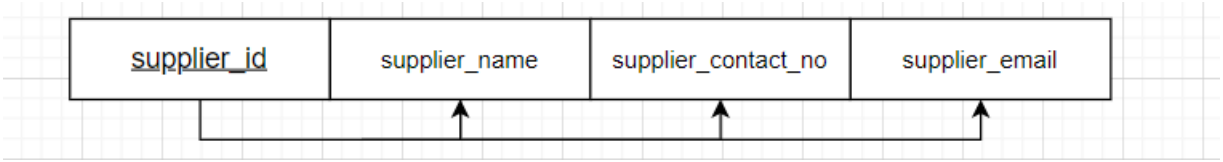
The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Supplier



The above table is 1st normal form because there are,

- No Non atomic values
- No nested relations
- No multi-valued attributes

The above table is in 2nd normal form because there are

- No partial dependencies

The above table is in 3rd normal form because there are

- No transitive dependencies

Data dictionary

customer

Field Name	Data type	Field size	Description	Constraints
Customer_id	int	-	Unique ID for all customers	Primary key
Customer_name	varchar	25	Name of customer	Not null
Cus_contact_no	char	10	Contact no of the customer	Not null
Customer_email	varchar	70	Email address of the customer	Not null

product

Field Name	Data type	Field size	Description	Constraints
Product_id	int	-	Unique ID for all products	Primary key
Product_des	varchar	130	Description of each product	Not null
Product_price	decimal	5	Price of each product	Not null
product_brand	varchar	20	Brand of each product	Not null
Supplier_id	int	-	supplier id for all suppliers	Foreign key Not null
Category_id	int	-	ID for all categories	Foreign key Not null

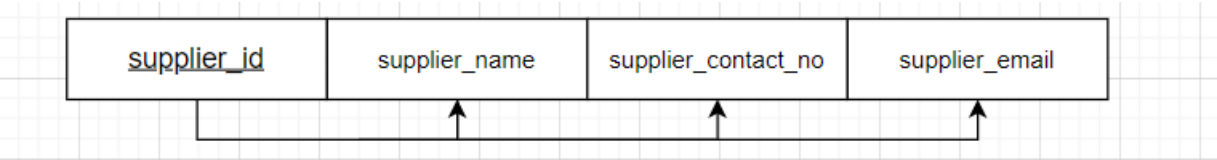
employee

Field Name	Data type	Field size	Description	Constraints
Emp_ID	int	-	Unique ID for all employees	Primary key
Emp_contact_no	char	10	Contact number of each employee	Not null
Emp_dob	date	-	Date of birth of employee	Not null
Emp_salary	decimal	10	Salary of employees	Not null
Emp_name	varchar	30	Name of employees	Not null
Store_id	int	-	ID of the store	Foreign key Not null

payment

Field Name	Data type	Field size	Description	Constraints
Payment_id	int	-	Unique ID for all payments	Primary key
payment_amount	decimal	10	Payment amount for each transaction	Not null
payment_desc	varchar	100	Description of the payment	Not null
payment_date	date	-	Date of the payment	Not null
customer_id	int	-	Id of the customer	Foreign key Not null
emp_ID	int	-	ID of the employee	Foreign key Not null

Supplier



Field Name	Data type	Field size	Description	Constraints
Supplier_id	int	-	Unique ID for all suppliers	Primary key
Supplier_name	varchar	30	Supplier name	Not null
Supplier_contact_no	char	10	Contact number of the supplier	Not null
Supplier_email	varchar	65	Email of the supplier	Not null

store

Field Name	Data type	Field size	Description	Constraints
store_id	int	-	Unique ID for all stores	Primary key
store_name	varchar	25	Store name	Not null
store_location	varchar	20	Location of the store	Not null

category

Field Name	Data type	Field size	Description	Constraints
Category_id	int	-	Unique ID for all categories	Not null Unique
Category_name	varchar	45	Name of the category	Not null

offer

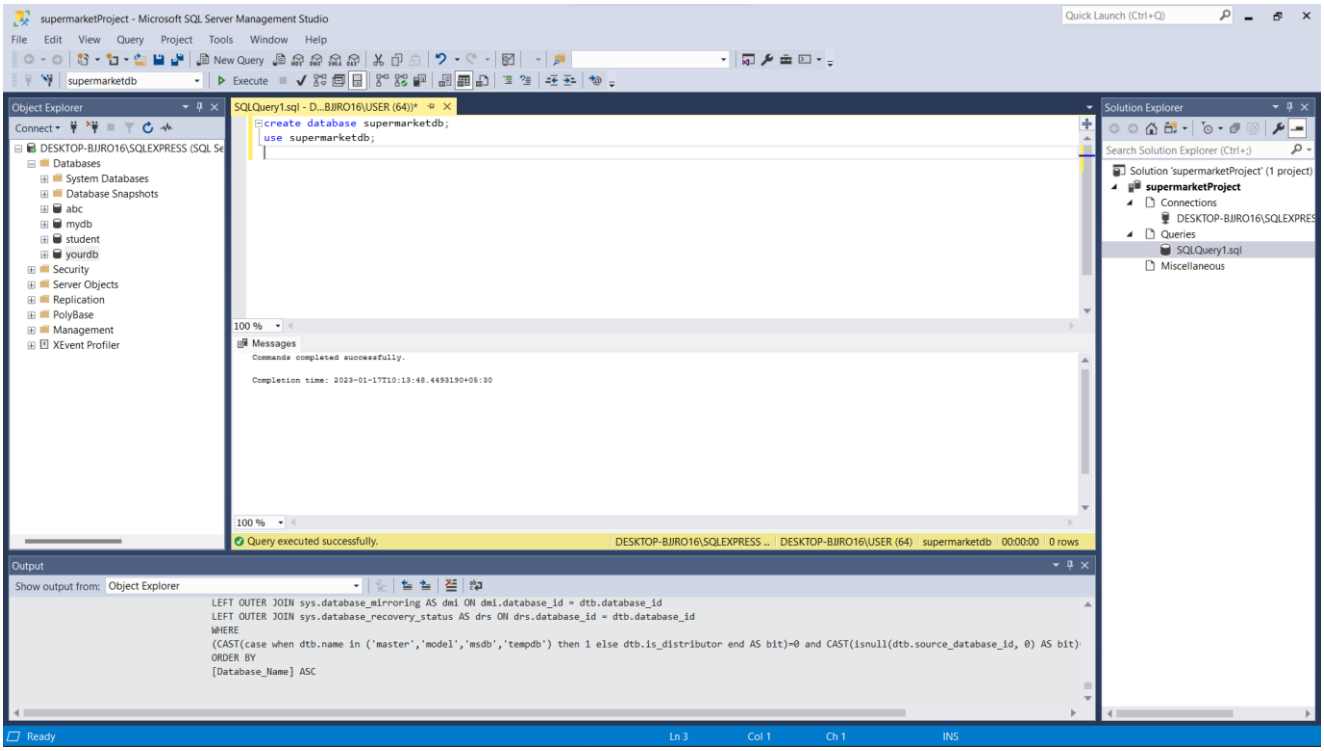
Field Name	Data type	Field size	Description	Constraints
Offer_id	int	-	Unique ID for all offers	Primary key
Offer_name	varchar	45	Name of the offer	Not null

supplier

Field Name	Data type	Field size	Description	Constraints
Supplier_id	int	-	Unique ID for all suppliers	Primary key
Supplier_name	Varchar	25	Name of the supplier	Not null
Supplier_email	varchar	20	Email address of the email	Not null
Supplier_contact_no	char	10	Contact number of the supplier	Not null

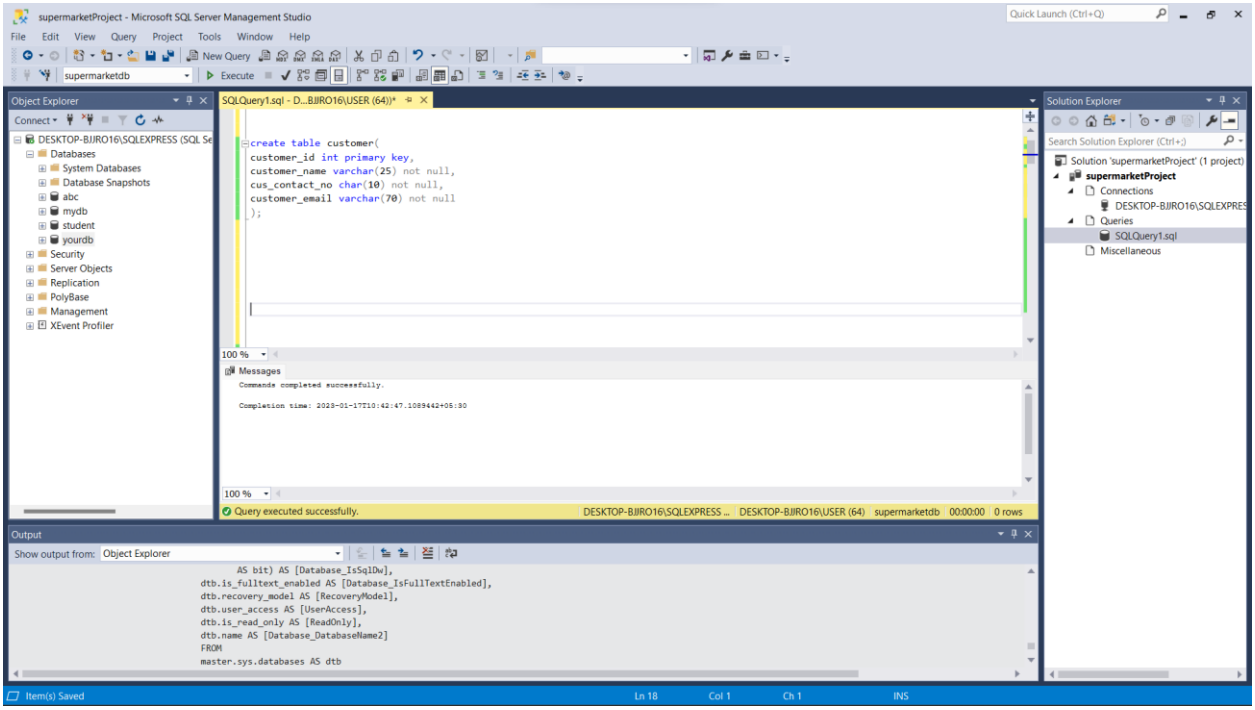
Section 2 - Create table statements

Database setup

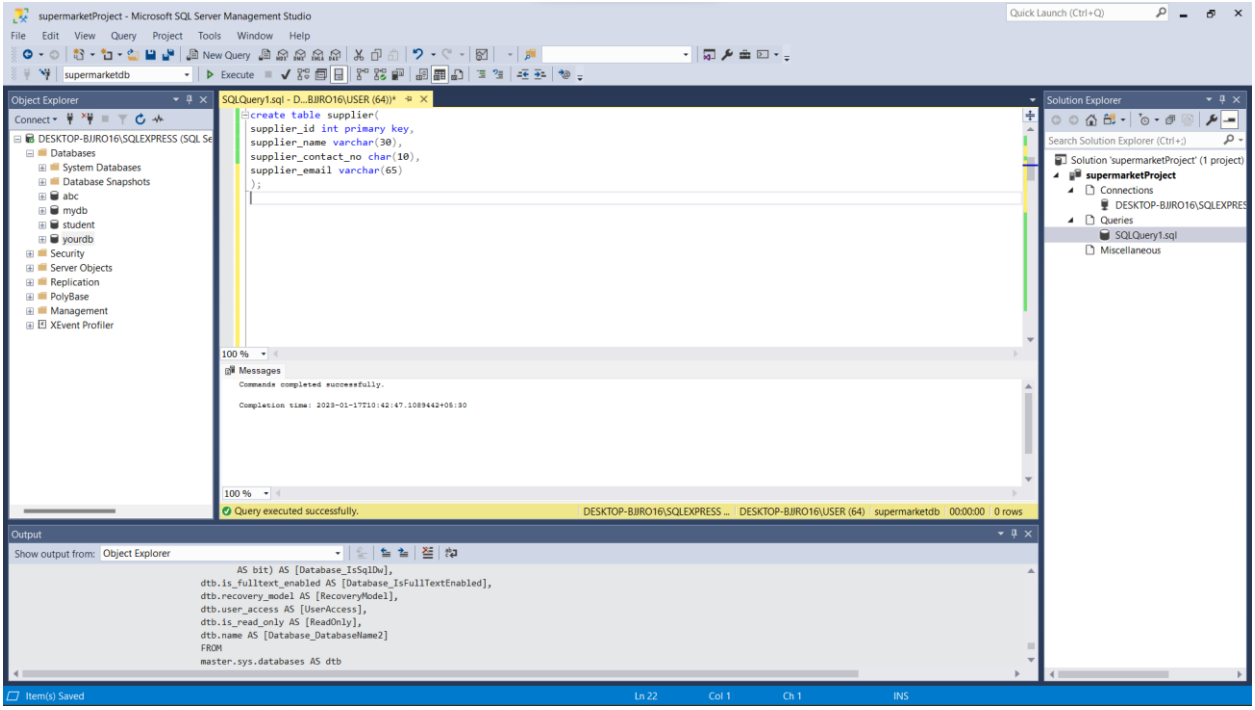


Create table statements

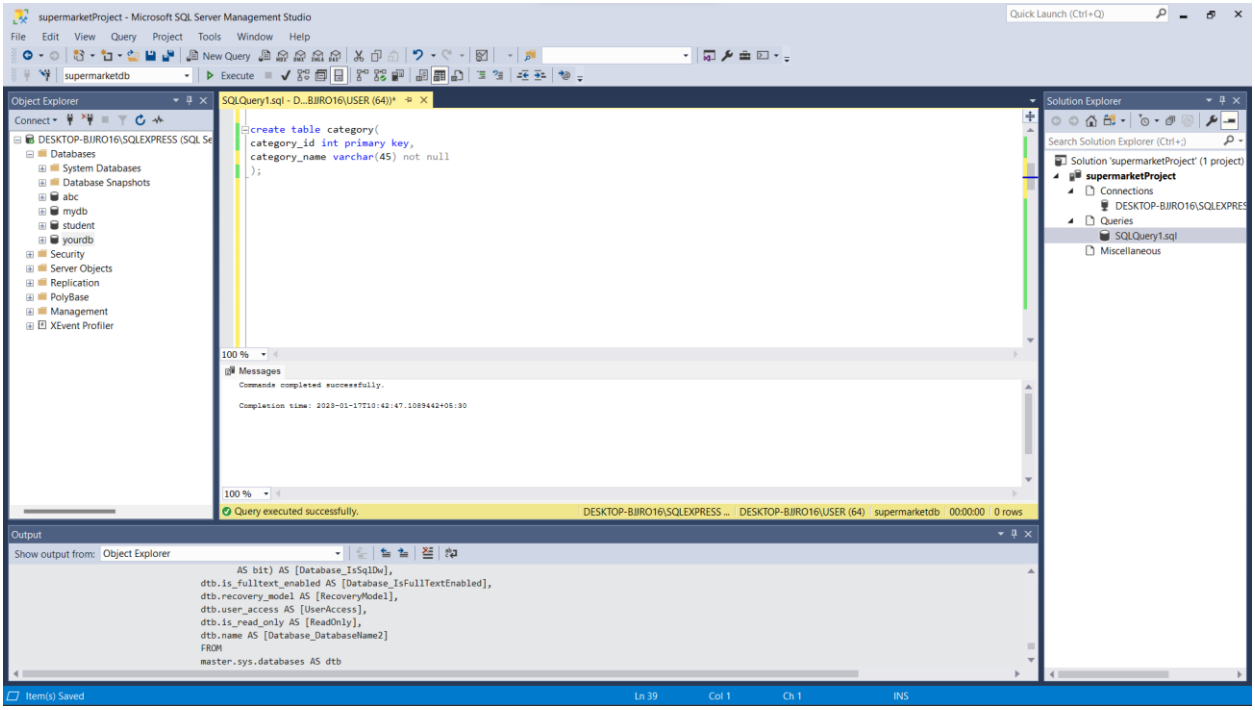
Customer table



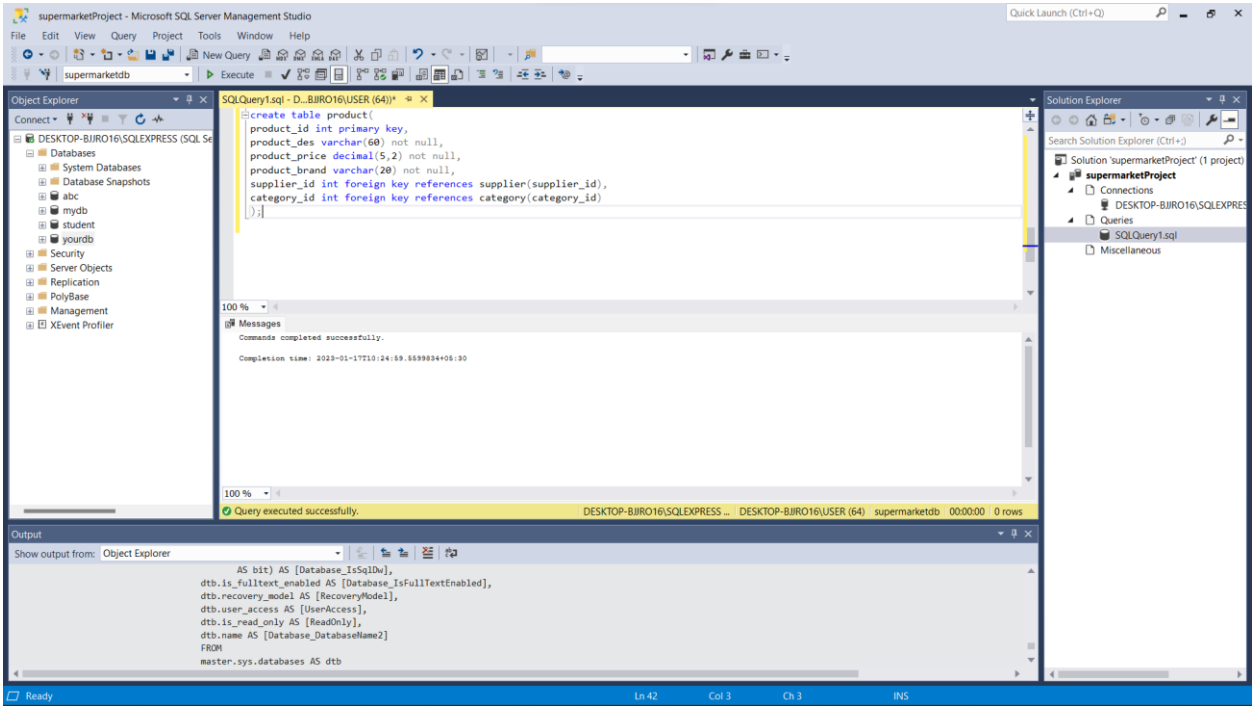
Supplier table



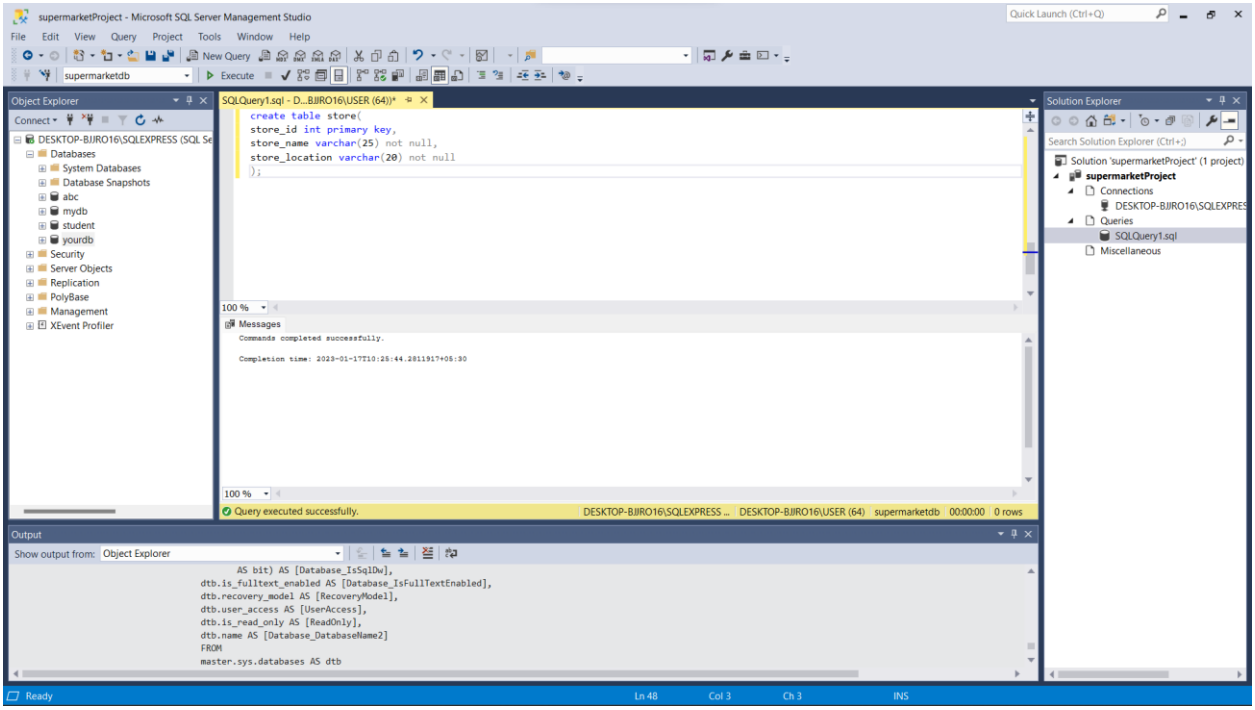
Category table



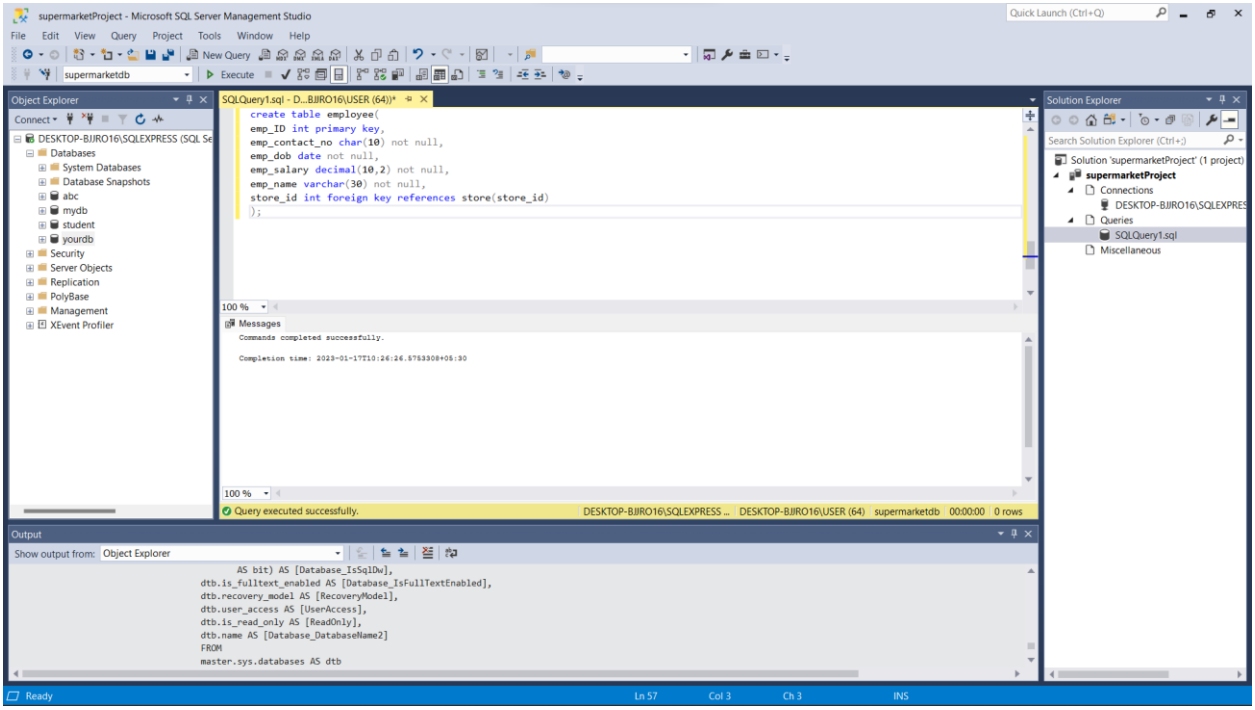
Product



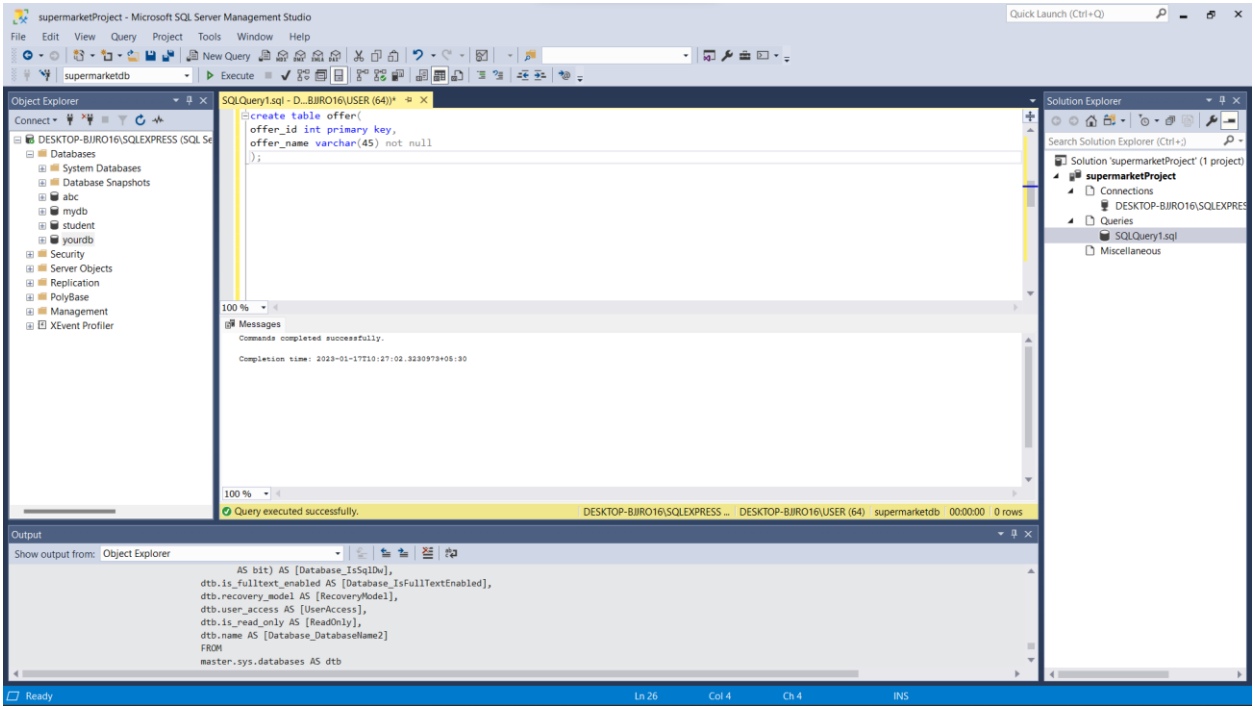
Store table



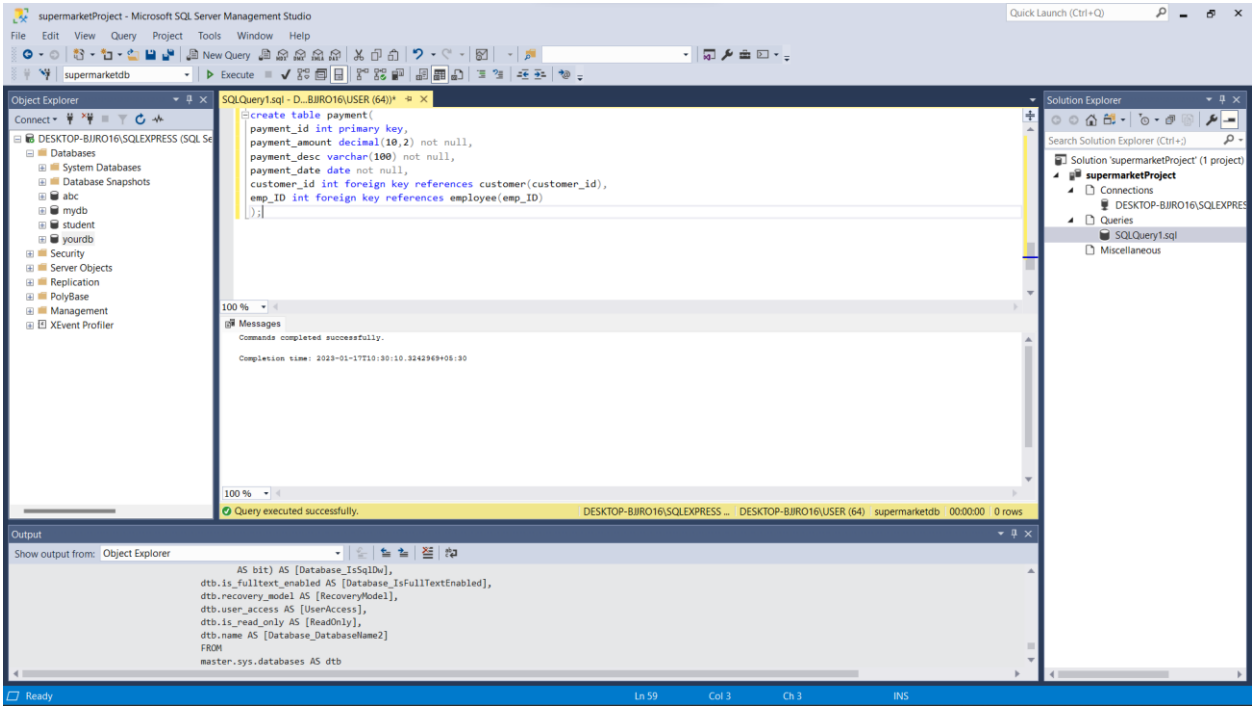
Employee



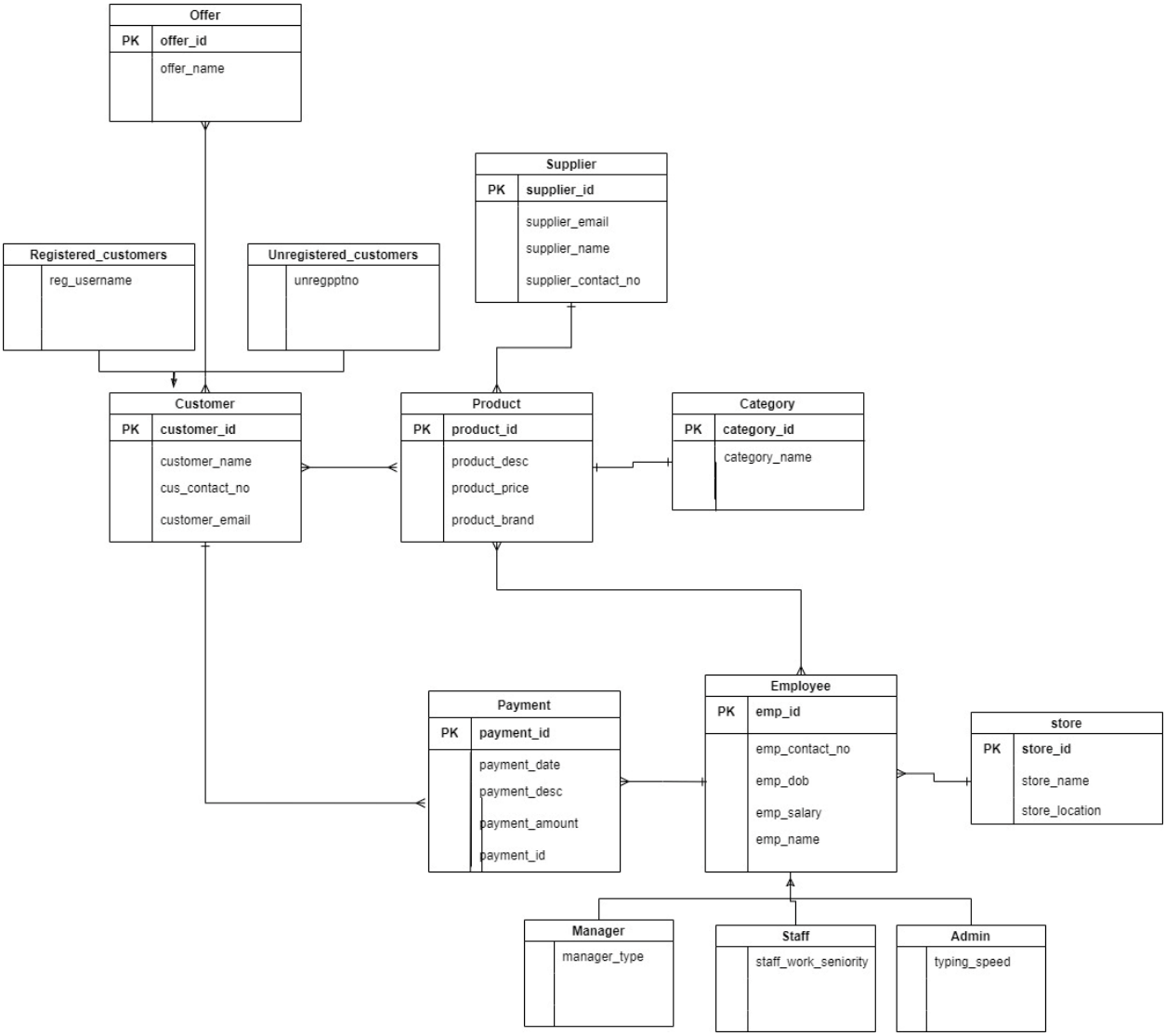
Offer table



Payment table

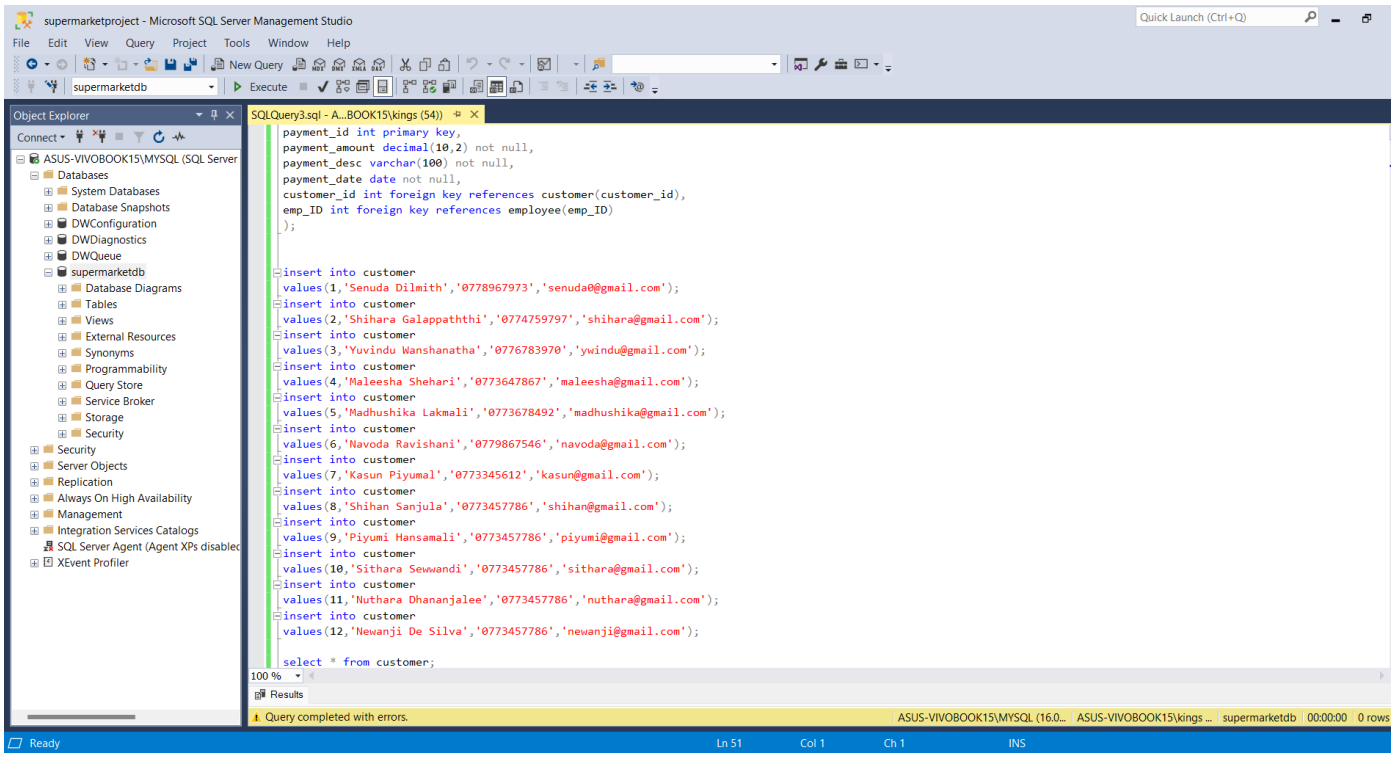


Database Diagram

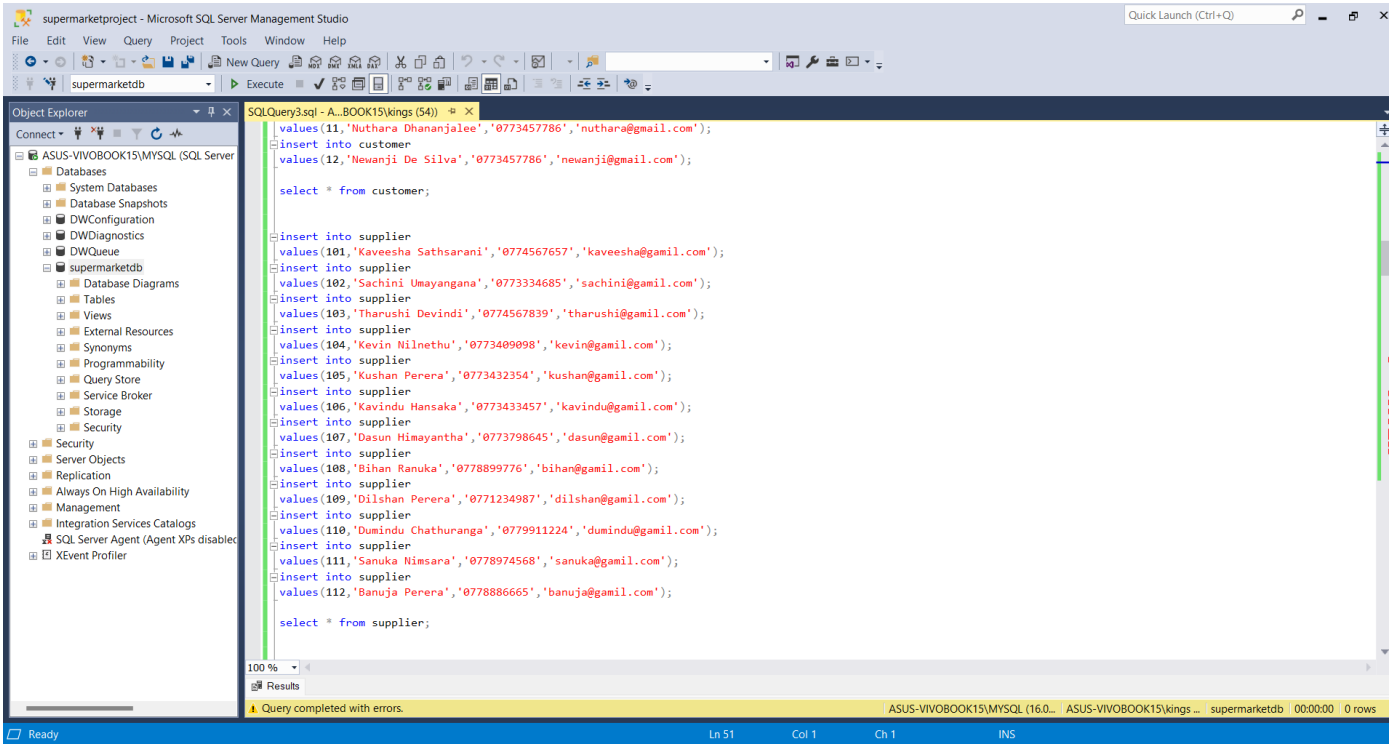


Insert statements

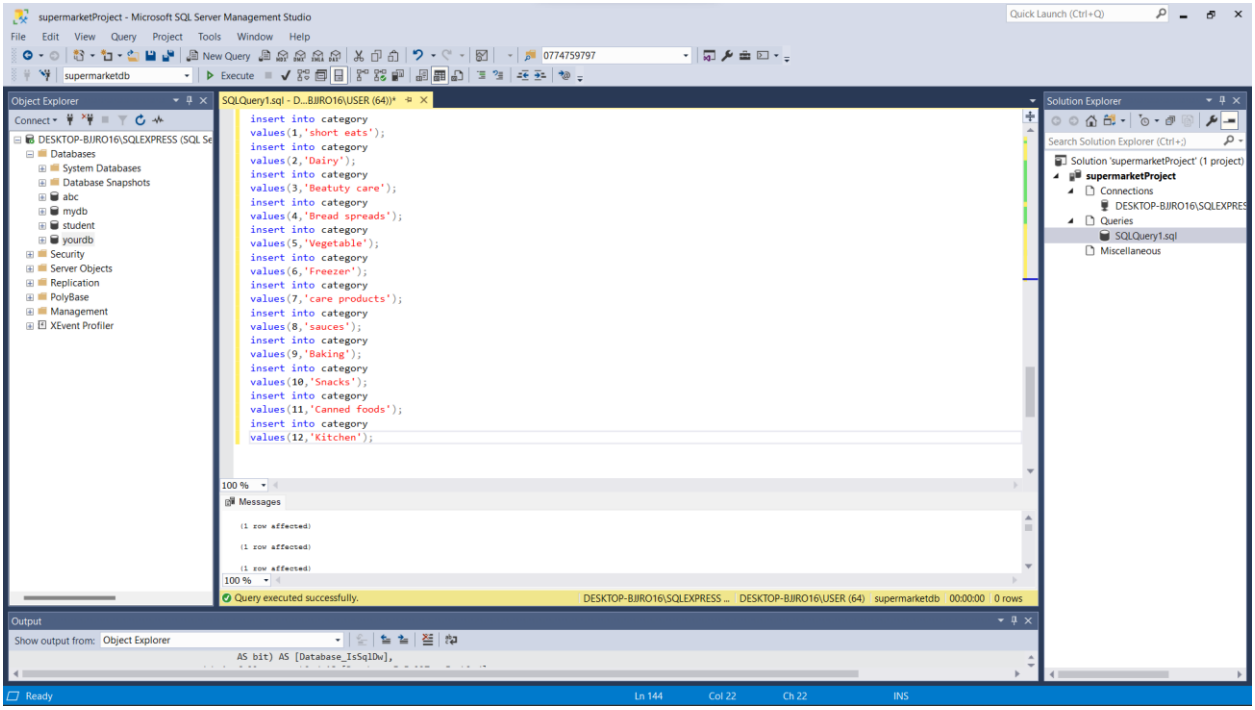
Customer table



Supplier table

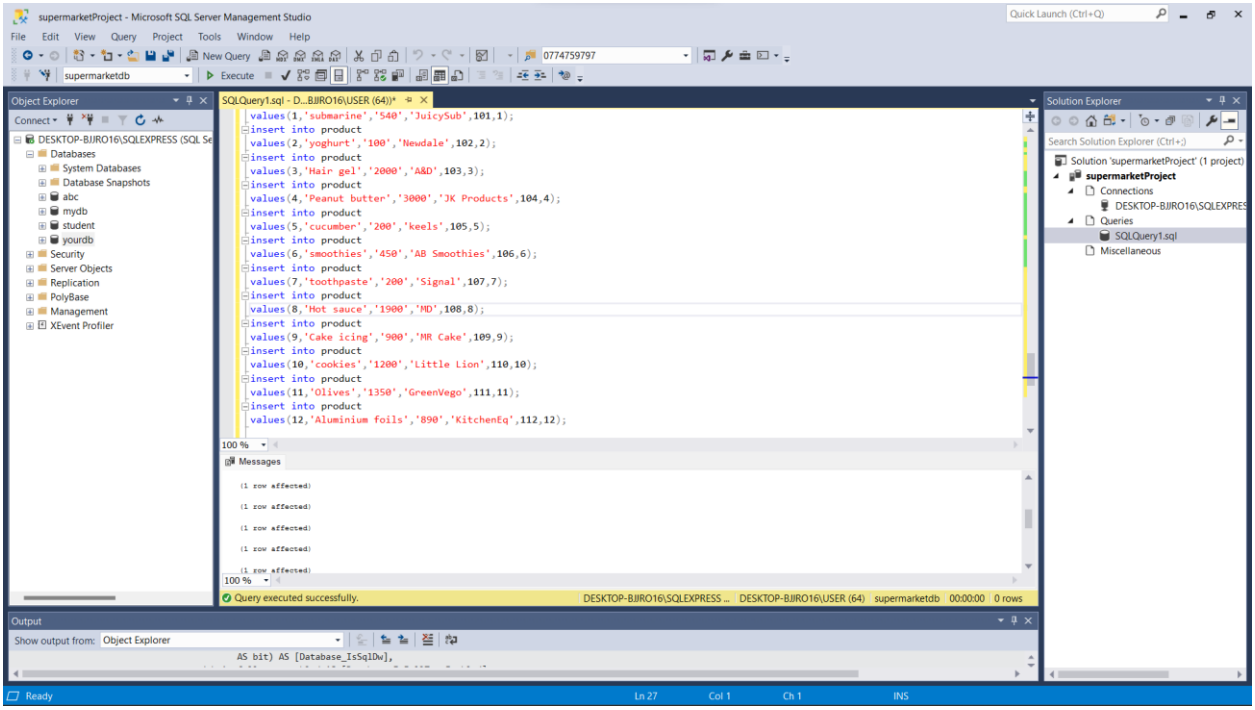


Category table



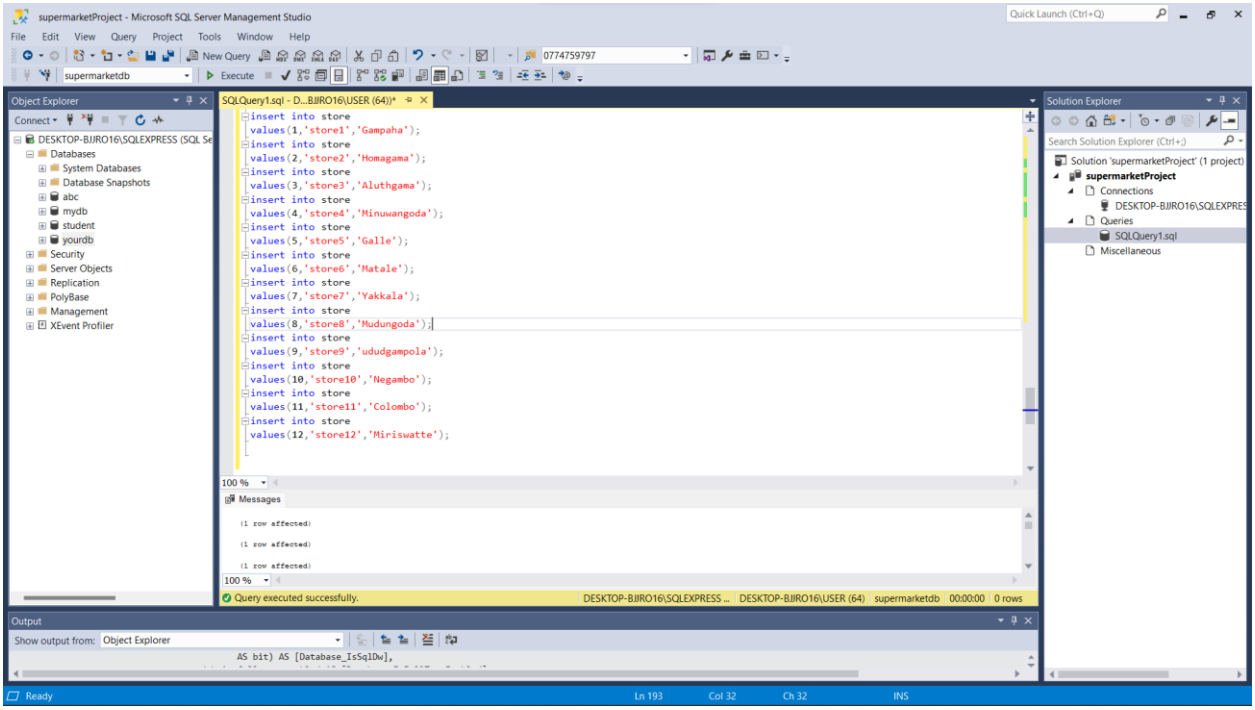
	category_id	category_name
1	1	short eats
2	2	Dairy
3	3	Beatuty care
4	4	Bread spreads
5	5	Vegetable
6	6	Freezer
7	7	care products
8	8	sauces
9	9	Baking
10	10	Snacks
11	11	Canned foods
12	12	Kitchen

Product table



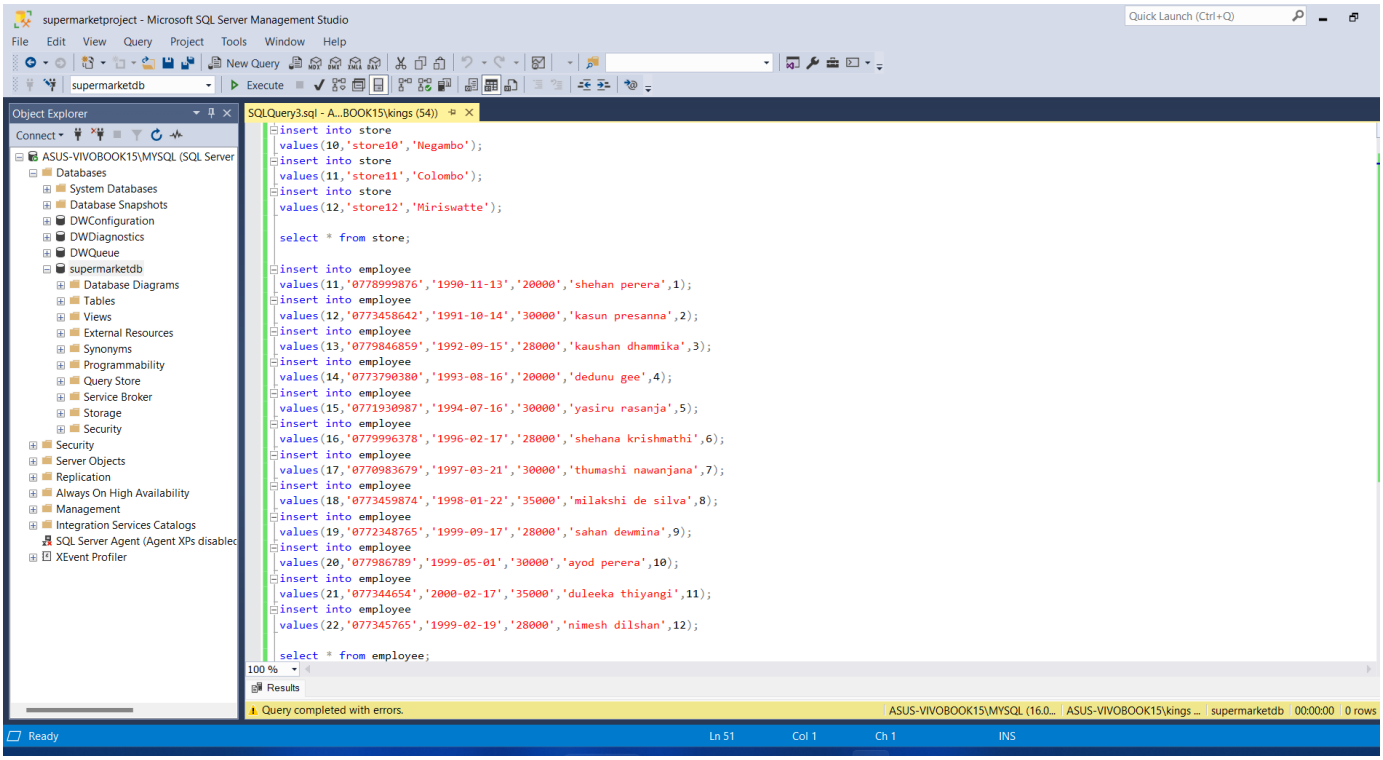
	product_id	product_des	product_price	product_brand	supplier_id	category_id
1	1	submarine	540.00	JuicySub	101	1
2	2	yoghurt	100.00	Newdale	102	2
3	3	Hair gel	2000.00	A&D	103	3
4	4	Peanut butter	3000.00	JK Products	104	4
5	5	cucumber	200.00	keels	105	5
6	6	smoothies	450.00	AB Smoothies	106	6
7	7	toothpaste	200.00	Signal	107	7
8	8	Hot sauce	1900.00	MD	108	8
9	9	Cake icing	900.00	MR Cake	109	9
10	10	cookies	1200.00	Little Lion	110	10
11	11	Olives	1350.00	GreenVego	111	11
12	12	Aluminium f...	890.00	KitchenEq	112	12

Store table

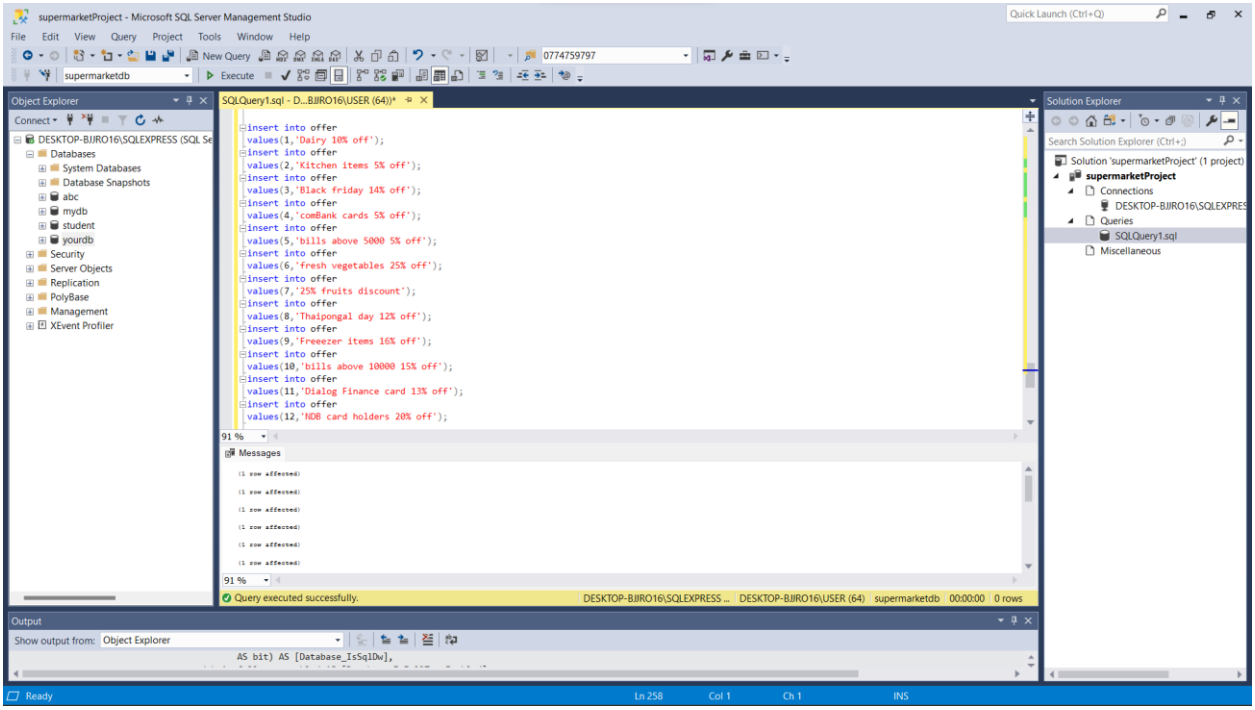


	store_id	store_name	store_location
1	1	store1	Gampaha
2	2	store2	Homagama
3	3	store3	Aluthgama
4	4	store4	Minuwangoda
5	5	store5	Galle
6	6	store6	Matale
7	7	store7	Yakkala
8	8	store8	Mudungoda
9	9	store9	ududgampola
10	10	store10	Negambo
11	11	store11	Colombo
12	12	store12	Miriswatte

Employee table

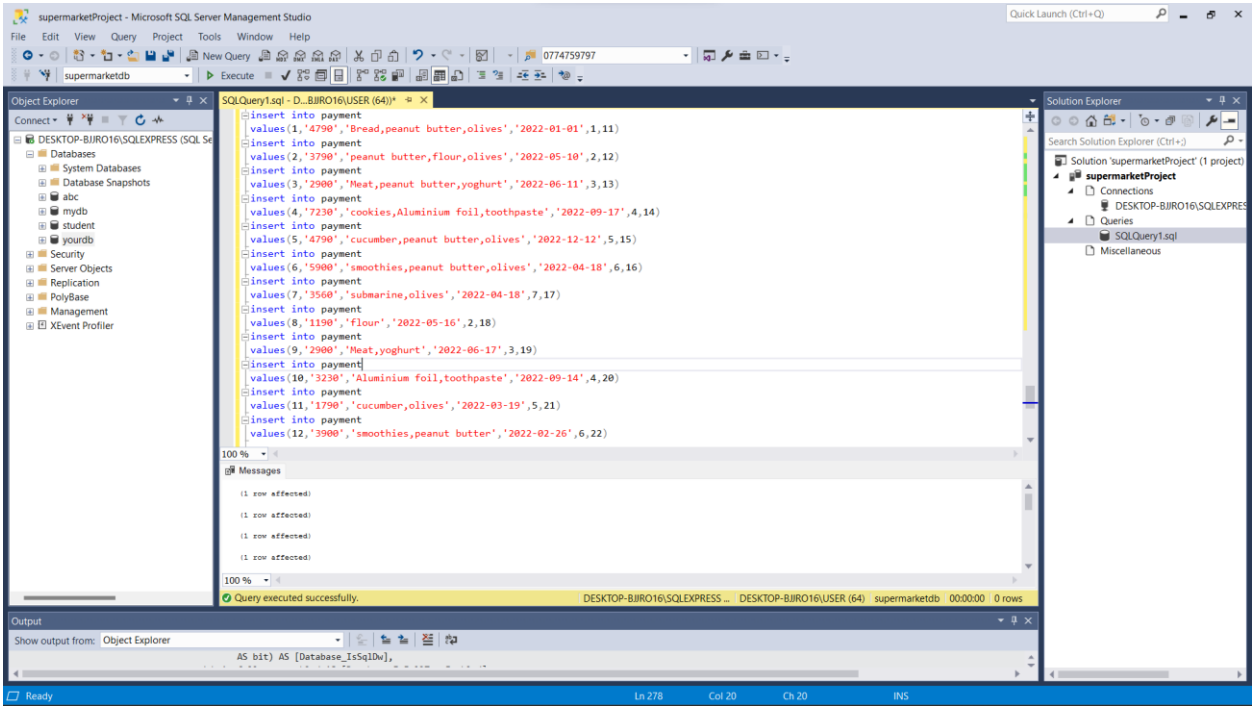


Offer table



	offer_id	offer_name
1	1	Dairy 10% off
2	2	Kitchen items 5% off
3	3	Black friday 14% off
4	4	comBank cards 5% off
5	5	bills above 5000 5% off
6	6	fresh vegetables 25% off
7	7	25% fruits discount
8	8	Thaipongal day 12% off
9	9	Freeezer items 16% off
10	10	bills above 10000 15%...
11	11	Dialog Finance card 1...
12	12	NDB card holders 20%...

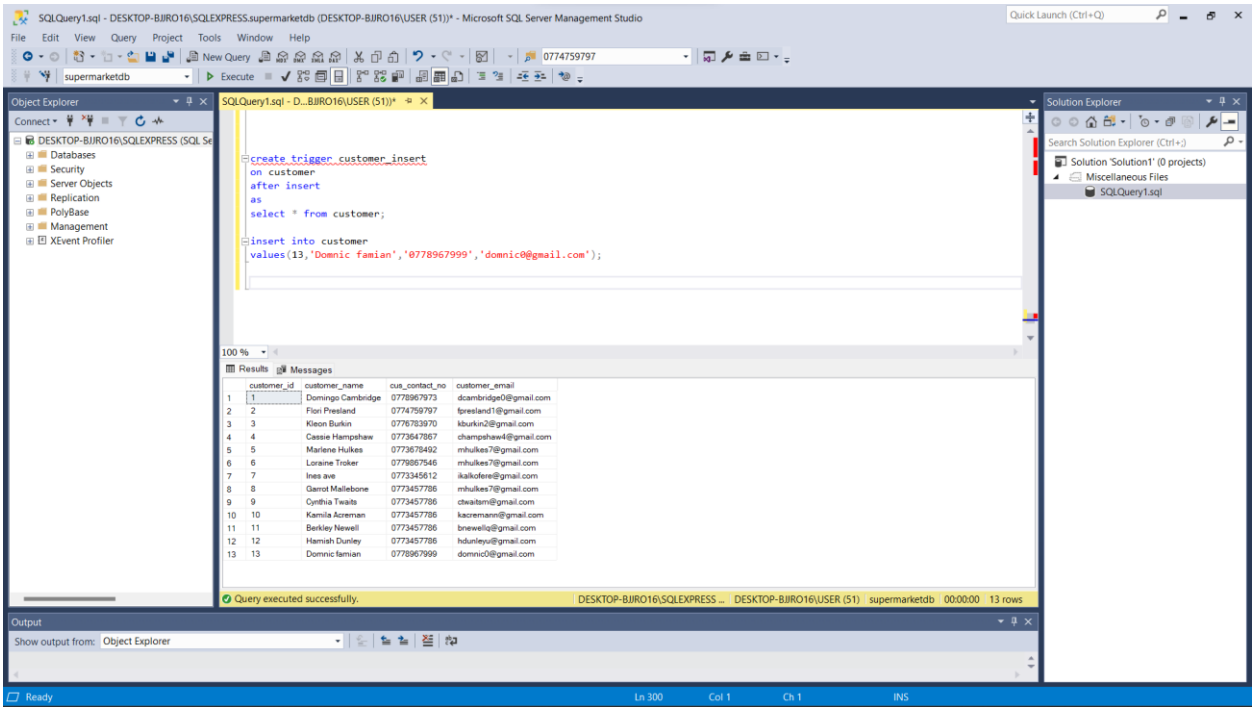
Payment table



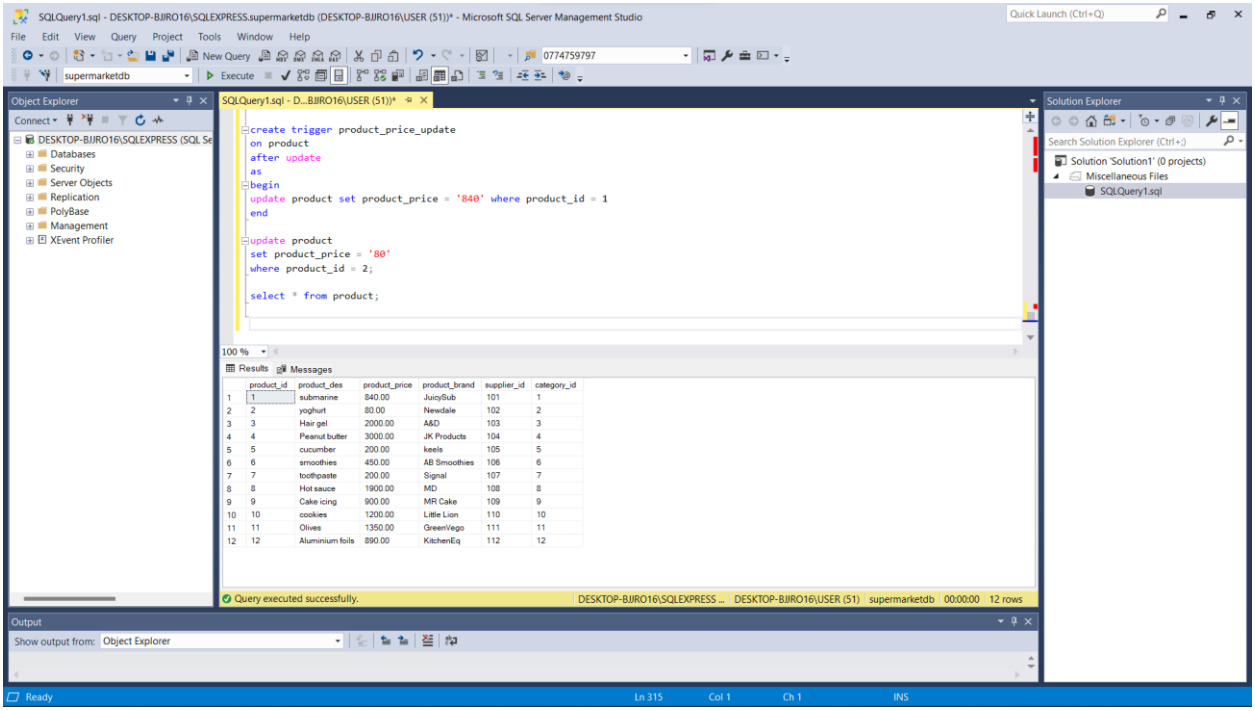
	payment_id	payment_amount	payment_desc	payment_date	customer_id	emp_ID
1	1	4790.00	Bread,peanut butter,olives	2022-01-01	1	11
2	2	3790.00	peanut butter,flour,olives	2022-05-10	2	12
3	3	2900.00	Meat,peanut butter,yoghurt	2022-06-11	3	13
4	4	7230.00	cookies,Aluminium foil,toothpaste	2022-09-17	4	14
5	5	4790.00	cucumber,peanut butter,olives	2022-12-12	5	15
6	6	5900.00	smoothies,peanut butter,olives	2022-04-18	6	16
7	7	3560.00	submarine,olives	2022-04-18	7	17
8	8	1190.00	flour	2022-05-16	2	18
9	9	2900.00	Meat,yoghurt	2022-06-17	3	19
10	10	3230.00	Aluminium foil,toothpaste	2022-09-14	4	20
11	11	1790.00	cucumber,olives	2022-03-19	5	21
12	12	3900.00	smoothies,peanut butter	2022-02-26	6	22

Section 3 - Triggers

Insert trigger – trigger to display all values in customer table when a record is inserted to the customer table

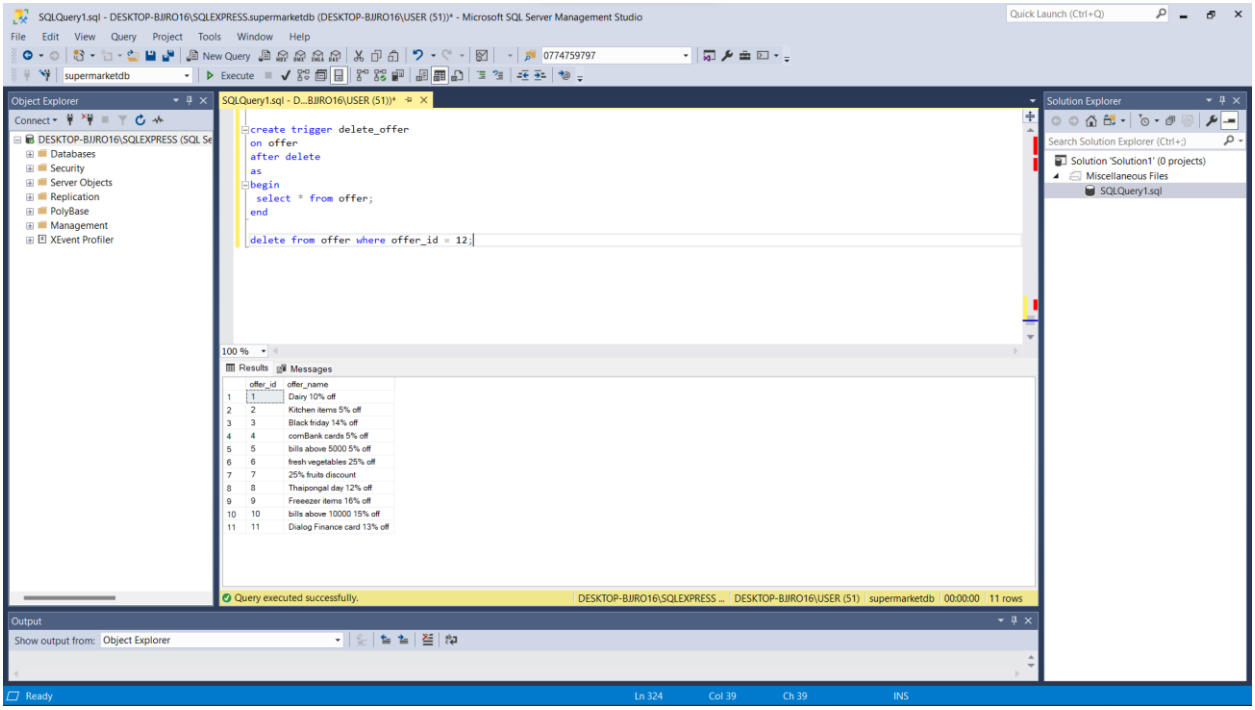


Update trigger – trigger to update values in product table when a record is updated



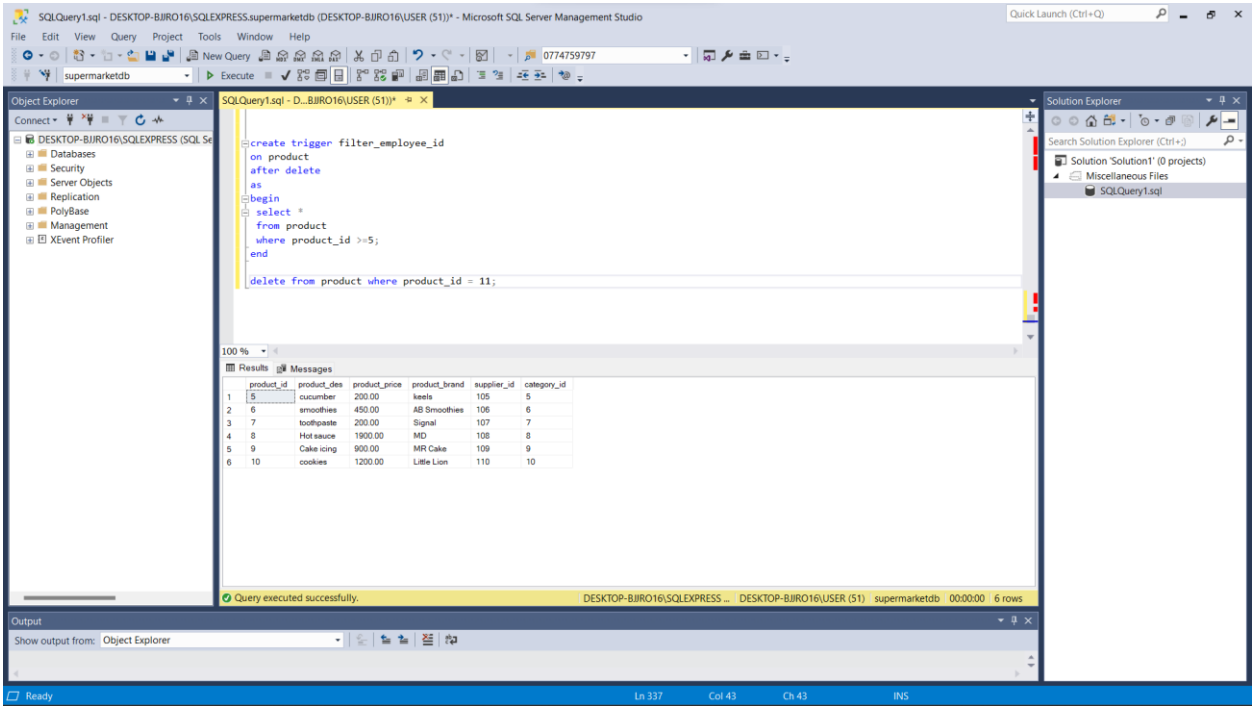
In the above trigger the price of submarine update from Rs.540.00 to Rs.840.00 when the update statement executed

Delete trigger – trigger to show table data once a record gets deleted



Once we delete an offer from offer table the trigger will invoke and display all the details of offer table after deletion.

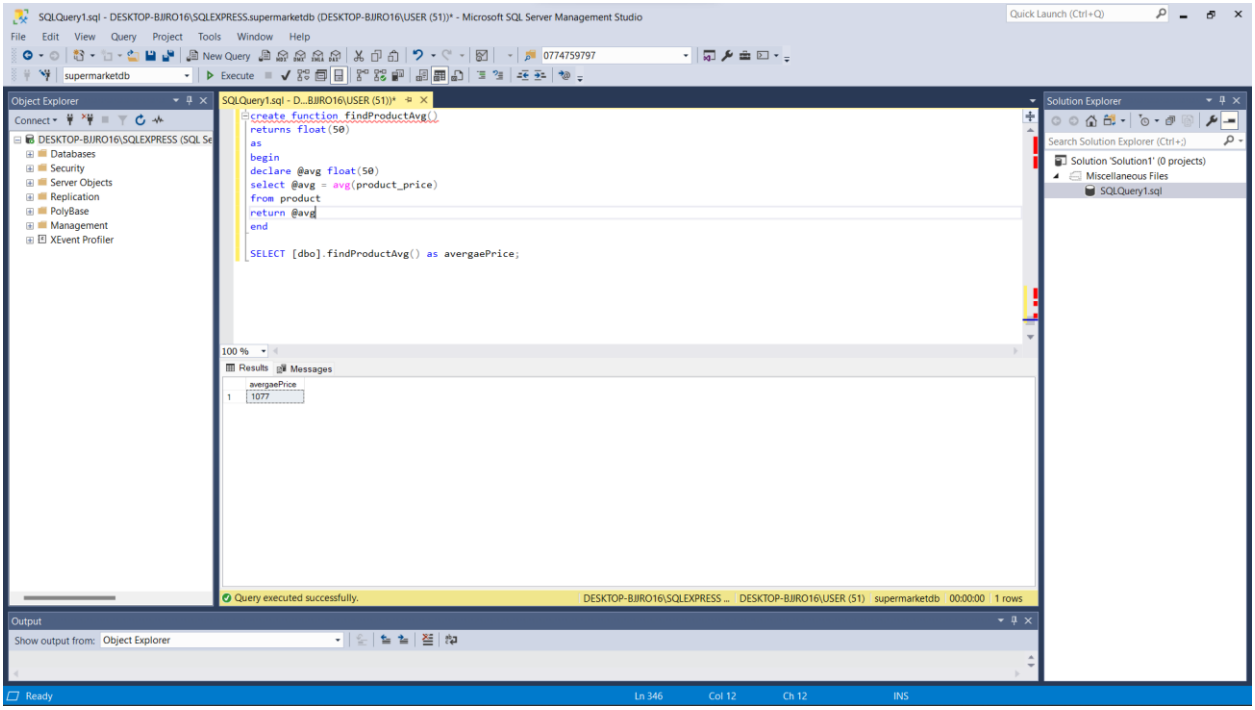
Delete trigger



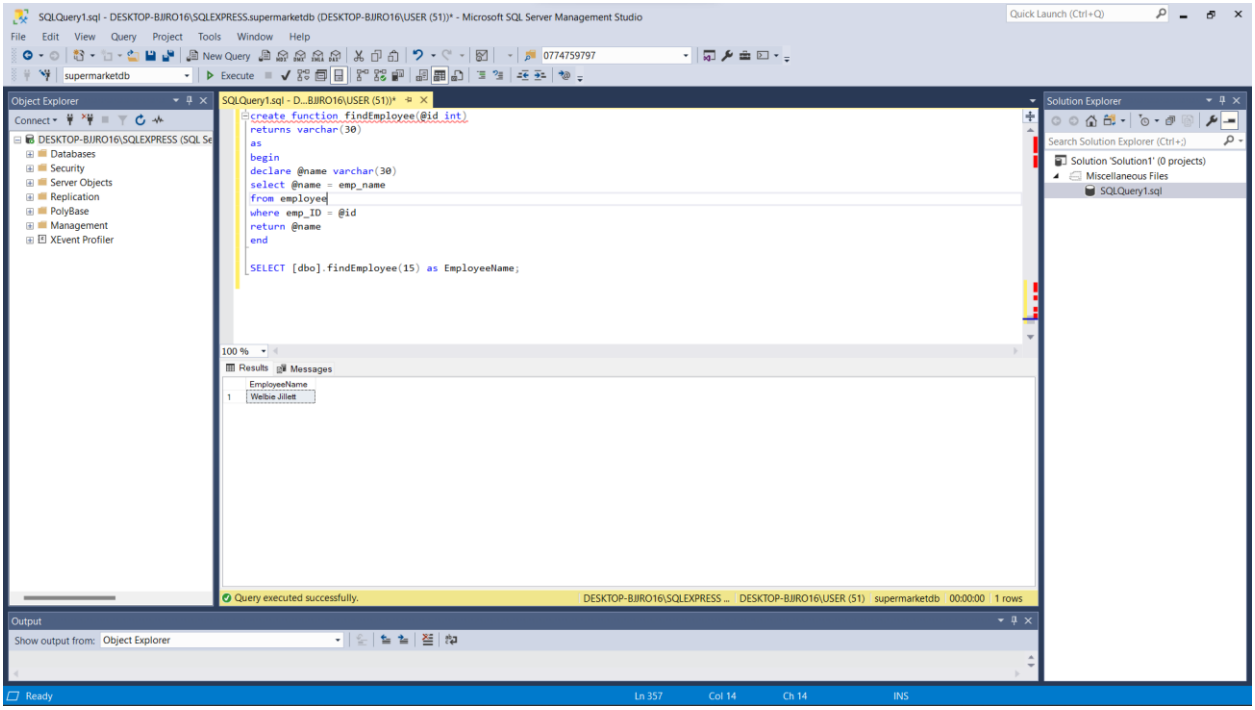
Once your delete a record from product table the trigger will invoke and display all the product records which as an ID greater than or equal to 5.

Functions

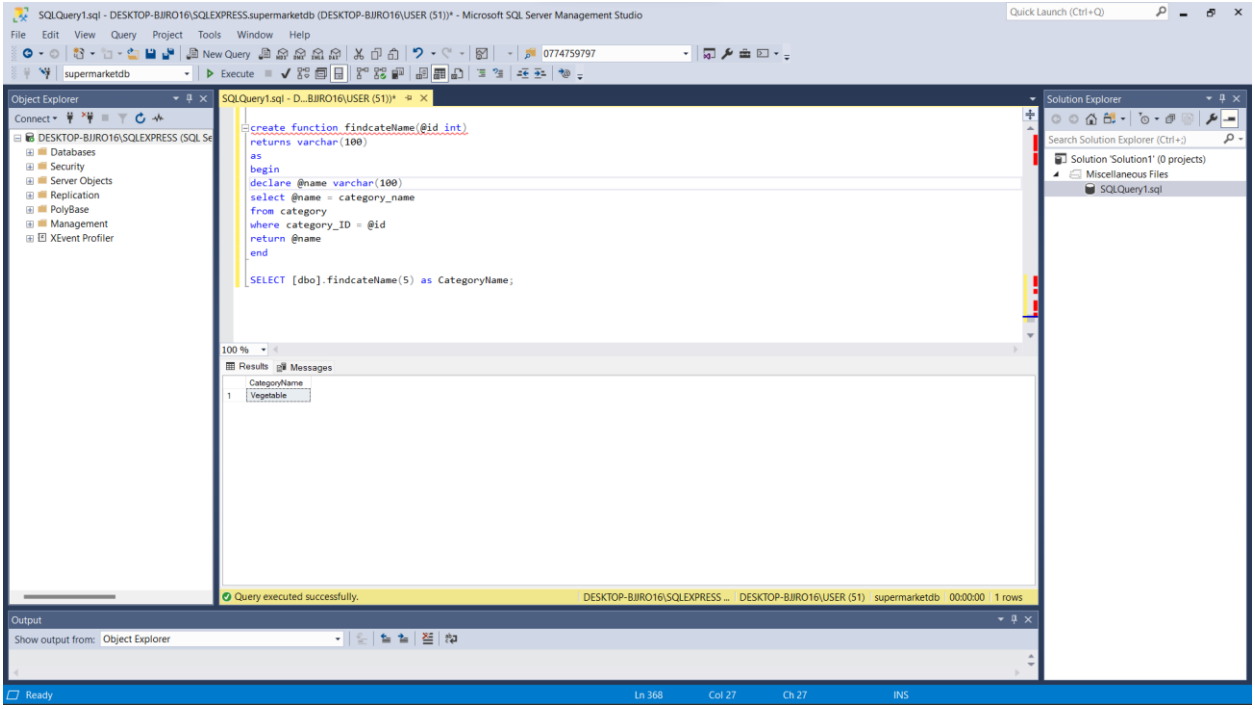
Function to display the average marks of a product (Without parameter type function)



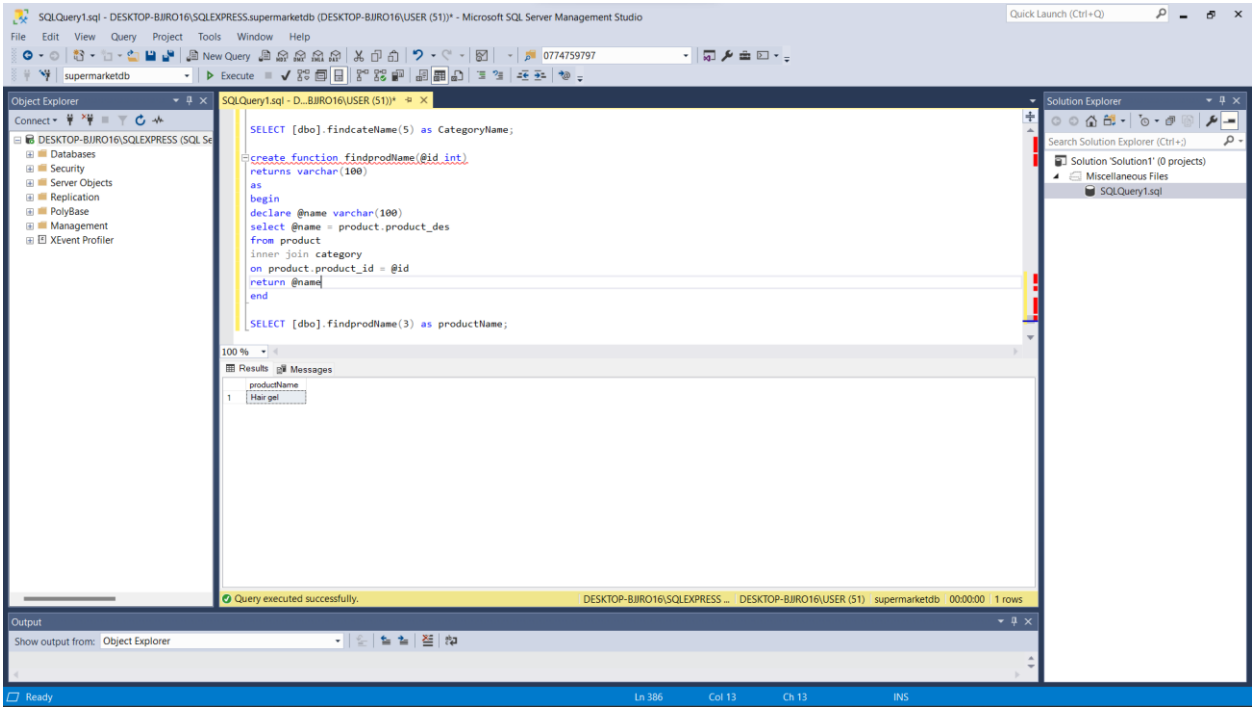
Function to find the employee name when ID is given as input (with return type with parameter)



Function to find category name when category ID is given as input Function to find the employee name when ID is given as input (with return type with parameter)

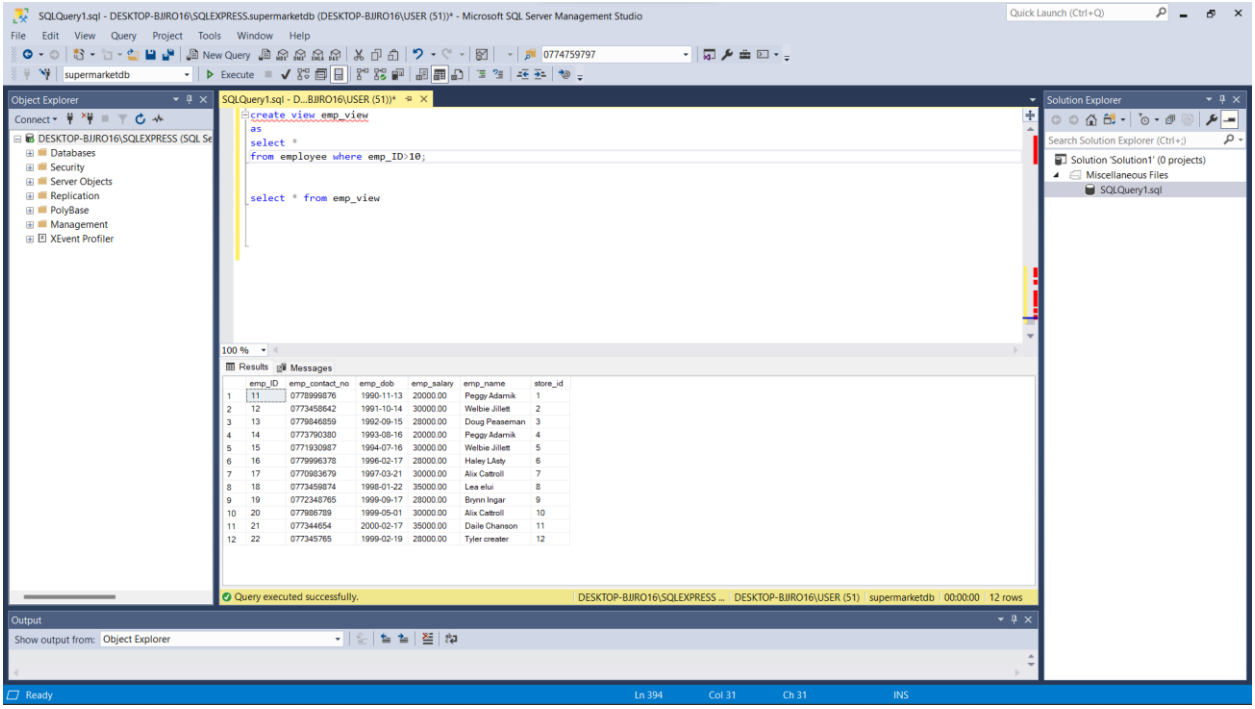


Function that displays the product name if it is available in both product and category tables by considering the user input product ID

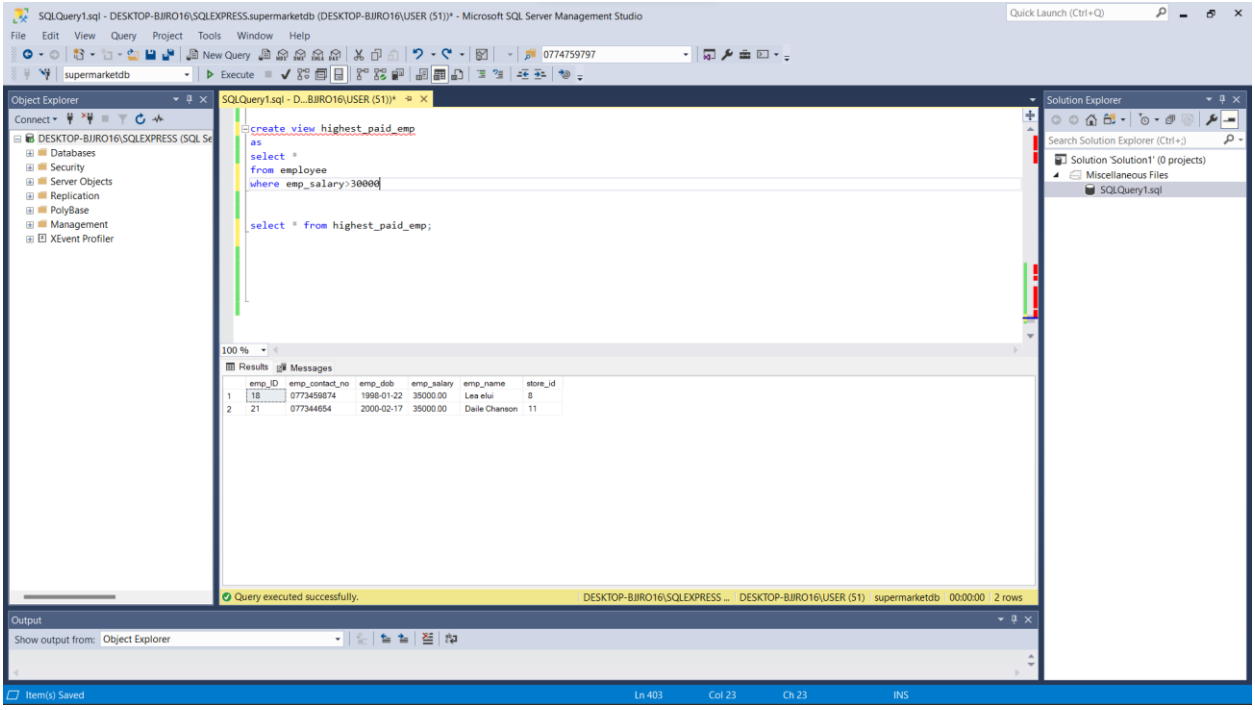


Views

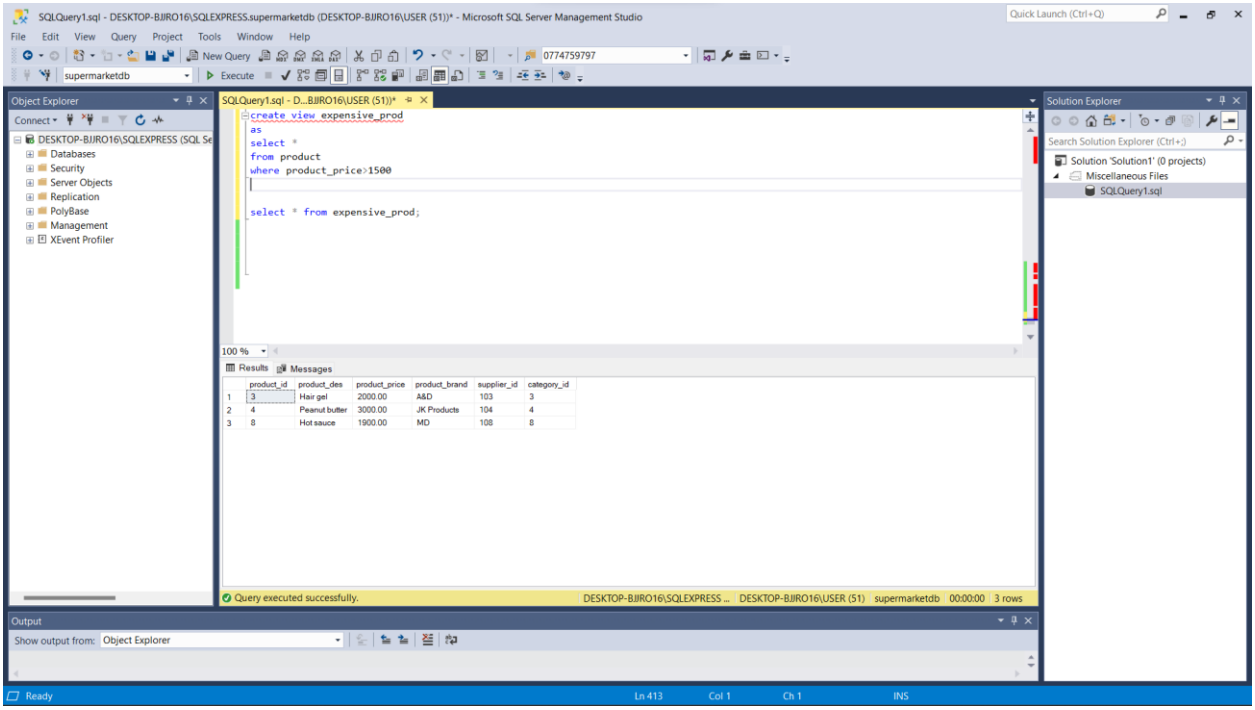
View which displays all employees with an employee ID greater than 10



View which displays employees with a salary greater than Rs.30,000

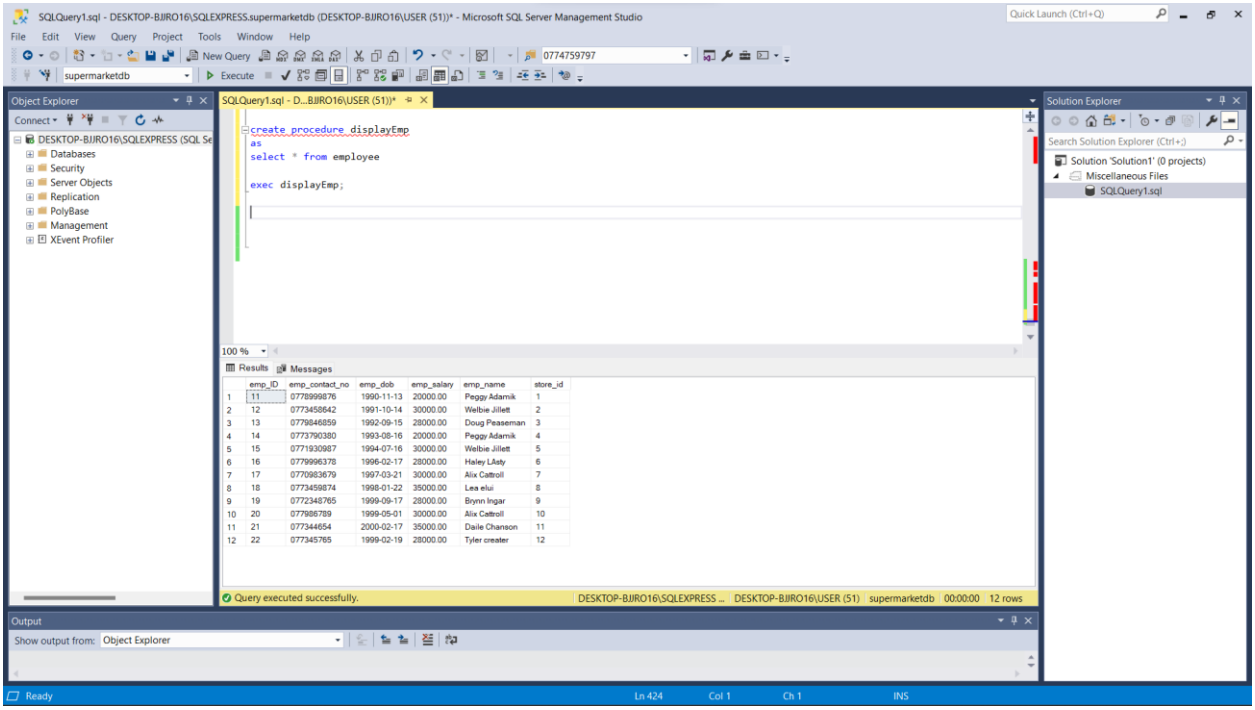


View which displays products that exceed the price Rs.1500

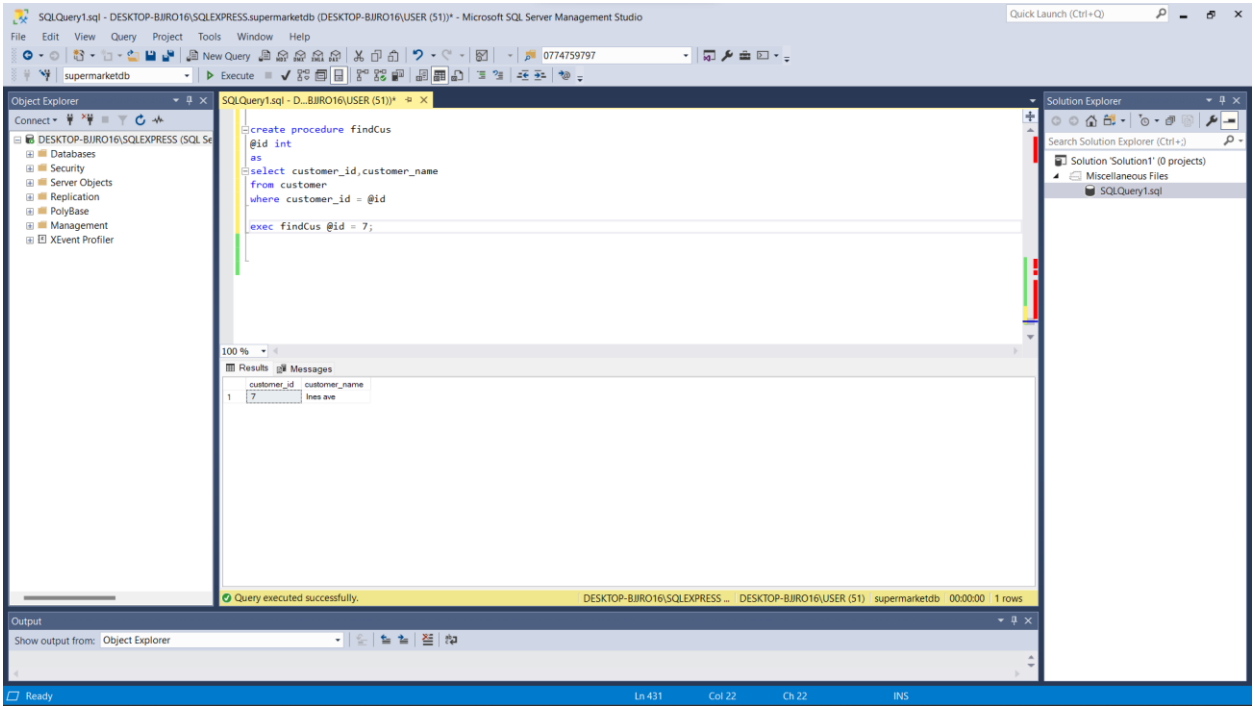


Procedures

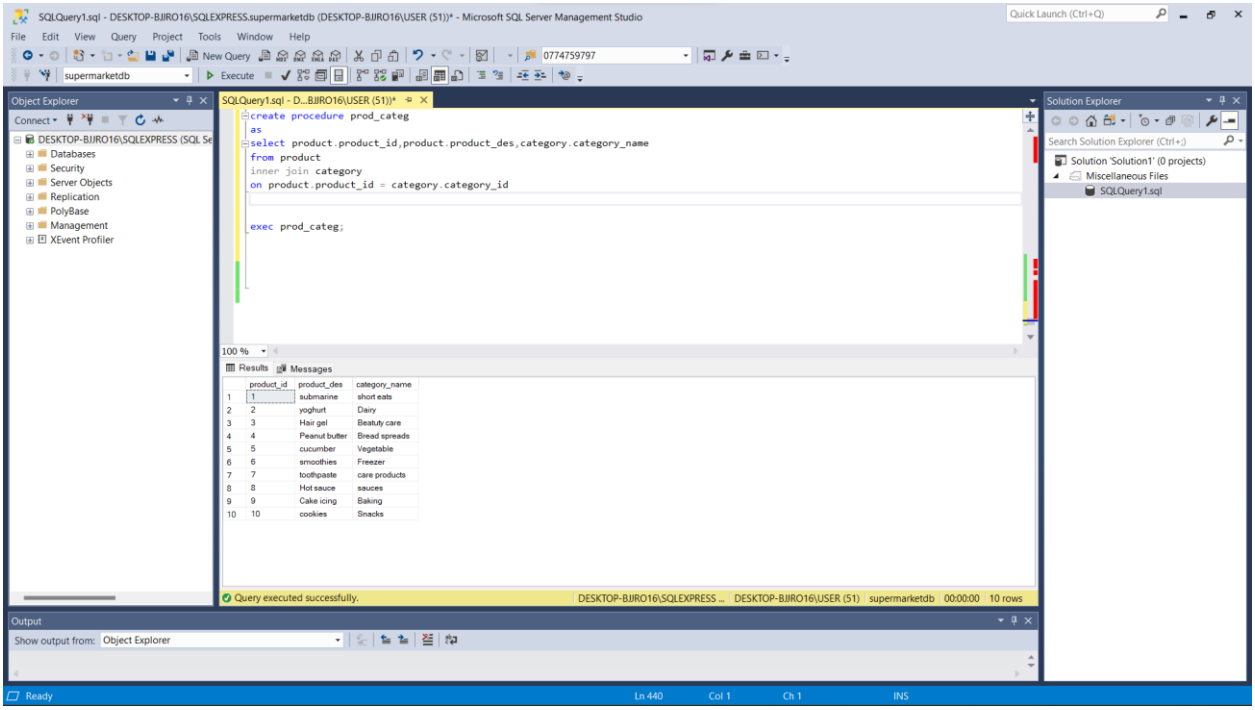
Procedure to display all values in employee table



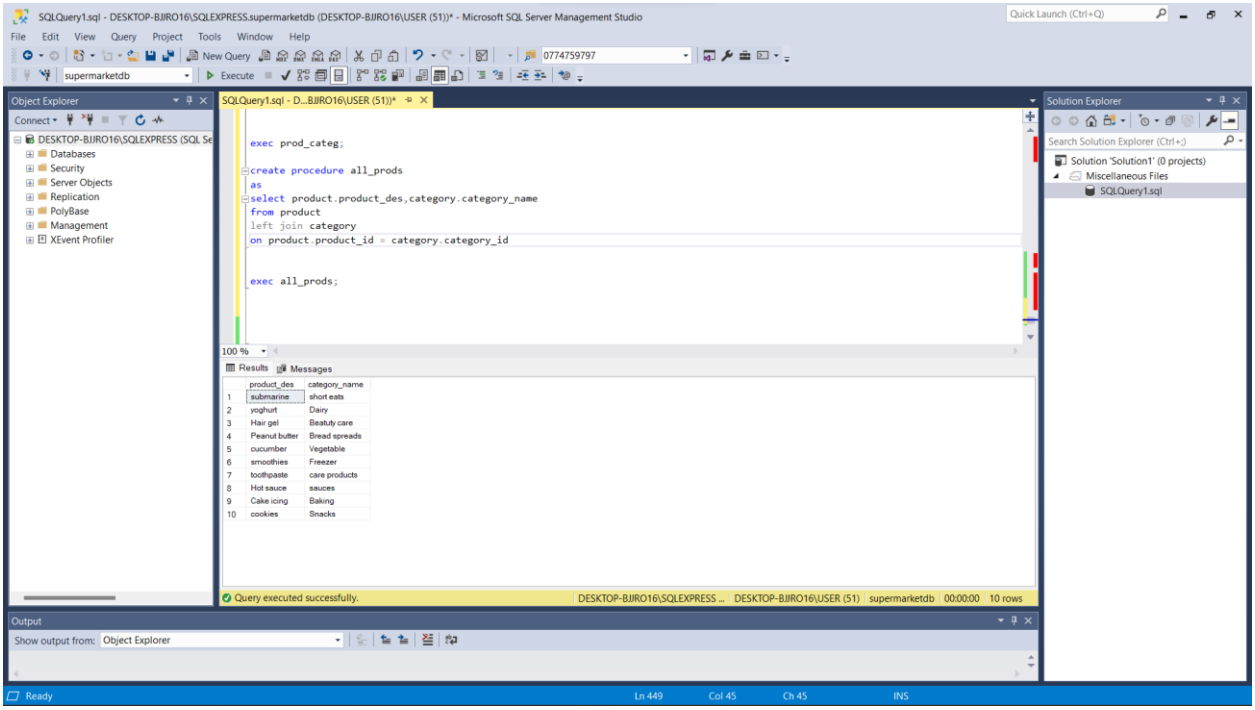
Procedure to display the ID and name of user input customer id



Procedure to display product ID, name and category name of products that are in both product and category table.



Procedure to display all products despite whether it is present in category table or not



Critical Appraisal and Comments on Future Implementation

Critical appraisal

- Everything in this project including the queries is valid and you confirm this by crosschecking with other valid resources and confirm that this project information corroborate with the information of other sources
- Microsoft SQL Server Management Studio does not perform smoothly in low end computers. This can cause various problems such as system lagging and taking more time to fetch the output. The main reason for tis the high RAM usage for SQL server Management Studio

Solutions

- We solved the Microsoft SQL Server Management Studio performance drop by creating the database and implementing the queries in a high-performance pc

Future implementations

The main aim is to develop this system by including more subsystems like cash management systems and to optimize the process in this system to make the tasks more efficient and effective. Once the full optimized project is developed we can implement this system into supermarkets in Sri Lanka. In order to eliminate the need for customers to wait in the checkout line, we can also advance this project by using artificial intelligence to help recognize and process the items that are placed in trolleys.