VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF ECONOMICS AND LAW

MIDTERM PROJECT

# DESIGN MANAGEMENT DATABASE FOR THAO CAM VIEN

COURSE: DATABASE

COURSE ID: 232IS930

LECTURER: LAM HONG THANH

GROUP 3

# GROUP 3: MEMBERS

| Full Name | Student ID |
|---|---|
| Nguyễn Trần Thanh Huyền | K224111450 |
| Phạm Tuyết Nhung | K224111460 |
| Vũ Quỳnh Như | K224111461 |
| Lê Nguyễn Minh Thảo | K224111462 |
| Nguyễn Ngọc Diểm Thúy | K224111466 |

HCM CITY, MAY 2024

# ACKNOWLEDGEMENT

First of all, group 3 would like to thank Ms. Lam Hong Thanh - lecturer in charge of guiding the Database course (232IS930) for teaching enthusiastically support group 3 in the whole course as well as completing the essay. After learning about Database and conducting a research paper on the topic "DESIGN MANAGEMENT DATABASE FOR THAO CAM VIEN", our team has accumulated a lot of knowledge and experience to continue developing future orientation.

However, with the limited amount of knowledge and limited time to carry out the project, the results of the research paper are inevitably flawed. Our group is looking forward to receiving suggestions Ms.Thanh in order to improve our work and possibly apply it in the future.

Once again, we sincerely thank you for your feedback and help.

**GROUP 3**

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAP 1. OVERVIEW

## 1.1 Reason

Currently, the Thao Cam Vien is not only a tourist attraction but also an important environment for animal and plant management. Building a database system for the Thao Cam Vien will play a crucial role in optimizing the management of information related to animals and plants. The system not only records specific information about them but also monitors their health status and diet. In addition, the system also has the ability to support the management of entertainment activities and games at the Thao Cam Vien, as well as manage the staff working there. Through this, the Thao Cam Vien's database system will not only bring benefits to current management but also optimize management in an effective and productive way. For these reasons, the group chose the topic "Design of the Thao Cam Vien Management Database."

## 1.2 Objective

Using data from the system, Thao Cam Vien can evaluate operational performance, reduce errors, and identify problems and opportunities for improvement in the management of plant and animal species here. They can identify inefficient or weak processes and take the necessary measures to optimize them. This can include optimizing operational schedules, improving customer experience, making changes to plant and animal care and optimizing resource use.

The system will record financial transactions, including revenue from ticket sales, collaborations with third parties, and other services such as recreational activities. Simultaneously, it will also log operational expenses such as various purchasing invoices, employee salaries, etc. This data will be used for analysis and decision-making regarding financial strategy, aiding in improving financial performance and enhancing revenue. Through financial monitoring and revenue tracking, Thao Cam Vien can enhance services and amenities while ensuring reasonable pricing, including improving infrastructure or expanding relaxation and entertainment areas.

The human resources information management and organization system will help Thao Cam Vien have an effective team and bring satisfaction to employees to increase employee engagement with the business. Besides, the system contains information about employees, including personal information, skills, departments, etc. This makes it easy for human resource management to monitor and evaluate performance, and make plans. Plan training and capacity development and manage employee work schedules effectively.

In addition, this system can also help optimize the sightseeing experience for customers through online ticket booking and providing information about flora and fauna. Customers can book tickets for themselves or for relatives. From here, many different customer files can easily access the business

### 1.3 Result

Built management processes and database design for the management and business system of Thao Cam Vien Sai Gon.

# CHAP 2. BUSINESS MODEL INTRODUCTION

## 2.1 Business overview:

The Thao Cam Vien Limited Liability Company is the owner of Thao Cam Vien, a place dedicated to nurturing, conserving, researching, educating, and exhibiting various species of wild fauna and flora. It organizes recreational activities, eco-tourism, and provides entertainment services. It also undertakes planting, caring for, protecting, and developing greenery, as well as providing tree planting and care services. The business model encompasses both B2C and B2B operations.

Business activities of Thao Cam Vien include:

- Selling tickets for visiting Thao Cam Vien.
- Thao Cam Vien offers rental space for food and beverage stalls, souvenir shops, or for organizing events and festivals.
- Thao Cam Vien has a children's play area with games such as swings, slides, and inflatable houses.

### 2.1.1 Overview of the Thao Cam Vien's business process

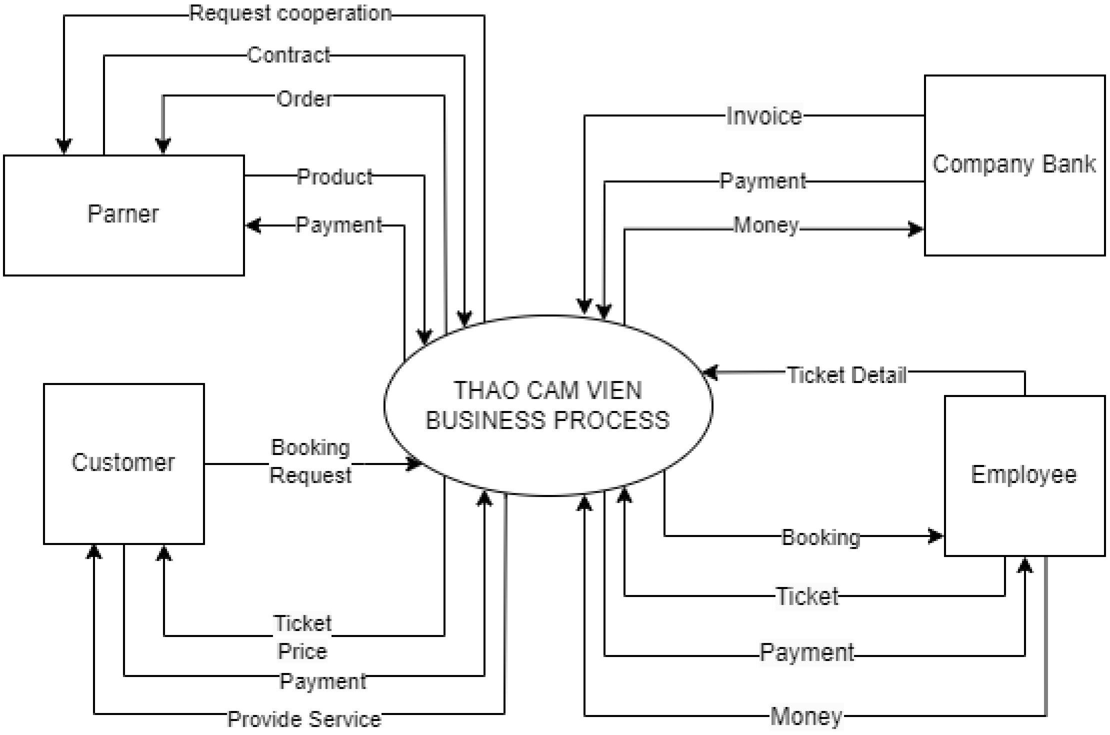


*Figure 2.1 Overview of the Thao Cam Vien's business process*

***Includes entities:***

- *Entity "Customer": (for online customers):*

- Manage customer information when booking online tickets: Store customer personal information, including name, address, contact information, and ticket purchase history.
- Customer relationship management: Track ticket purchase history, customer interactions and feedback to improve service and create promotions or offers.
- Transaction management: Track ticket purchases and related payment information.

- *Entity "Employee":*

- Employee information management: Store employee personal information, including name, position, contact information, and work schedule.
- Access rights management: Determine access rights and delegate permissions to employees to ensure information security and compliance with internal regulations.
- Work schedule management: Monitor employee work schedules and manage work shifts.

- *Entity "Partners":*

- Partner information management: Store information about the zoo's partners, including company name, address, contact information and cooperation agreements.
- Contract and commitment management: Track contracts and commitments between zoos and partners, including terms for services, products or technical support.
- Partner relationship management: Track interactions and activities involving partners to ensure smooth and effective collaboration.

- *Entity "Company bank":*

- Account management: Track company bank accounts, including account numbers, balances, and transaction history.
- Manage financial transactions: Track company financial transactions such as fund transfers, bill payments, and income and expenditure.
- Financial project management: Track the company's projects or financial strategies, including budgets, investment plans, and other financial strategies.
- Management of information related to tax and accounting: Includes storing tax information, financial reports, and other accounting-related information of the company to ensure compliance with legal regulations and internal regulations.

### 2.1.2 Organizational structure

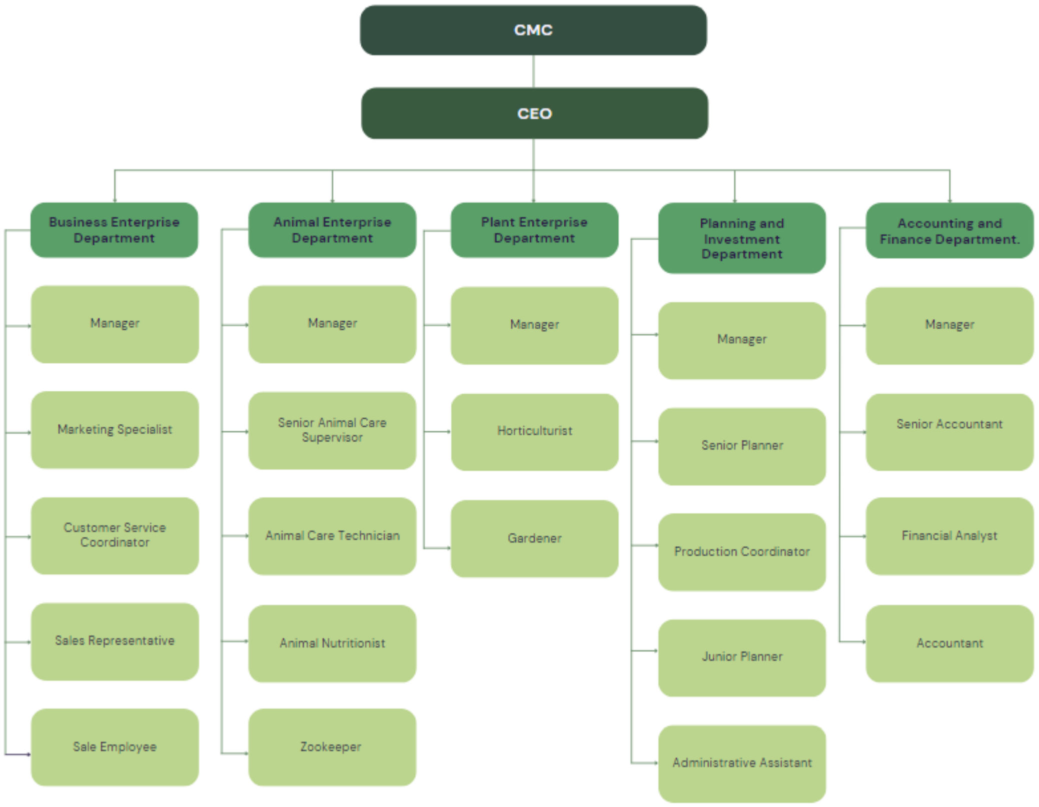


*Figure 2.2 Organizational structure of Thao Cam Vien*

***Including:***

- *Business Enterprise Department:* Research and develop plans for various types of business, and provide services to meet the needs of visitors.
- *Animal Enterprise Department:* Organize the management, care, and domestication of animal species according to approved technical procedures.
- *Plant Enterprise Department:* Manage and maintain park landscapes; implement measures for the management of flowers, ornamental plants, and greenery
- *Planning and Investment Department:* Advise on planning, production management, and effective utilization of the unit's available resources; supply materials, equipment, and food for animals.
- *Accounting and Finance Department:* Advise the Board of Directors on accounting, economic information, financial management, and accounting practices in accordance with regulations

### 2.1.3 Main business functions in the business model

**2.1.3.1 Sales process management:** We divide the sales process into 2 processes: ticket sales process, space rental process.

- Ticket sales process: Thao Cam Vien offers two main types of tickets: entrance tickets and game tickets. TCV will rely on employee work shifts for assignments.

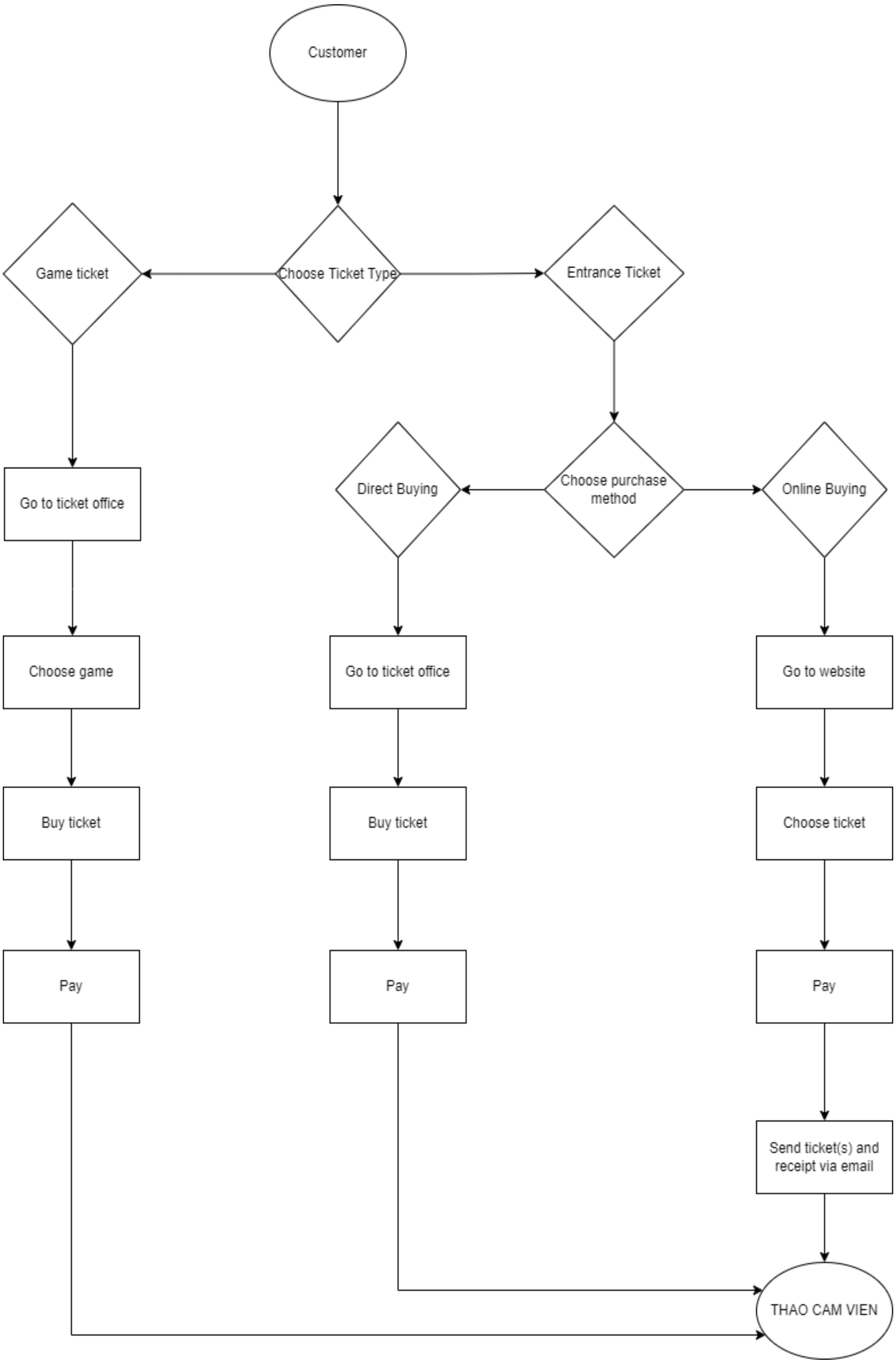


*Figure 2.3 Ticket Sale Process of Thao Cam Vien*

- If entrance tickets, customers will have 2 ways to buy (buy directly and buy online):

+ If customers buy directly at Thao Cam Vien, they will go to the ticket office and buy tickets directly through an employee. After confirming the information, customers will go to the payment step. After payment, customers will receive tickets.

+ If customers buy online on the website, fill in the information on the website and choose the ticket type. After confirming the information, customers will go to the payment step. After payment, customers will receive tickets and a receipt via email and use them to enter Thao Cam Vien.

- If game tickets: Each game will have a default price, customers who want to play will go to the ticket office and buy directly through an employee and make payment.
- All the payments will be included in Thao Cam Vien's revenue.

- Space rental process:

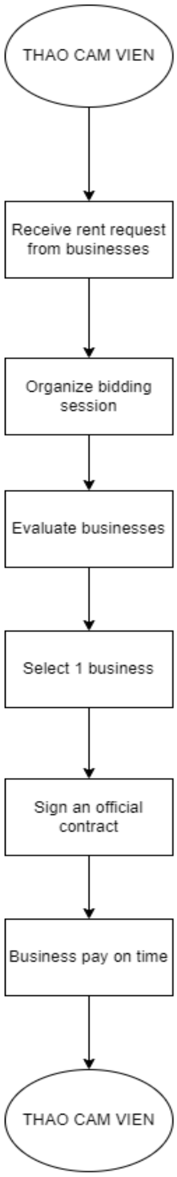


*Figure 2.4 Space Rental Process of Thao Cam Vien*

- Businesses that want to rent space from Thao Cam Vien will send rental registration information to Thao Cam Vien. Thao Cam Vien will receive it.
- Thao Cam Vien organizes a bidding session and evaluates businesses to select 1 potential business.
- If both parties reach an agreement, an official contract will be signed. The contract will have an agreement on payment conditions.
- Based on the contract, the business who rents the space will pay on time for Thao Cam Vien. The payment will be included in Thao Cam Vien's revenue.

#### 2.1.3.2 Purchase Process Management:

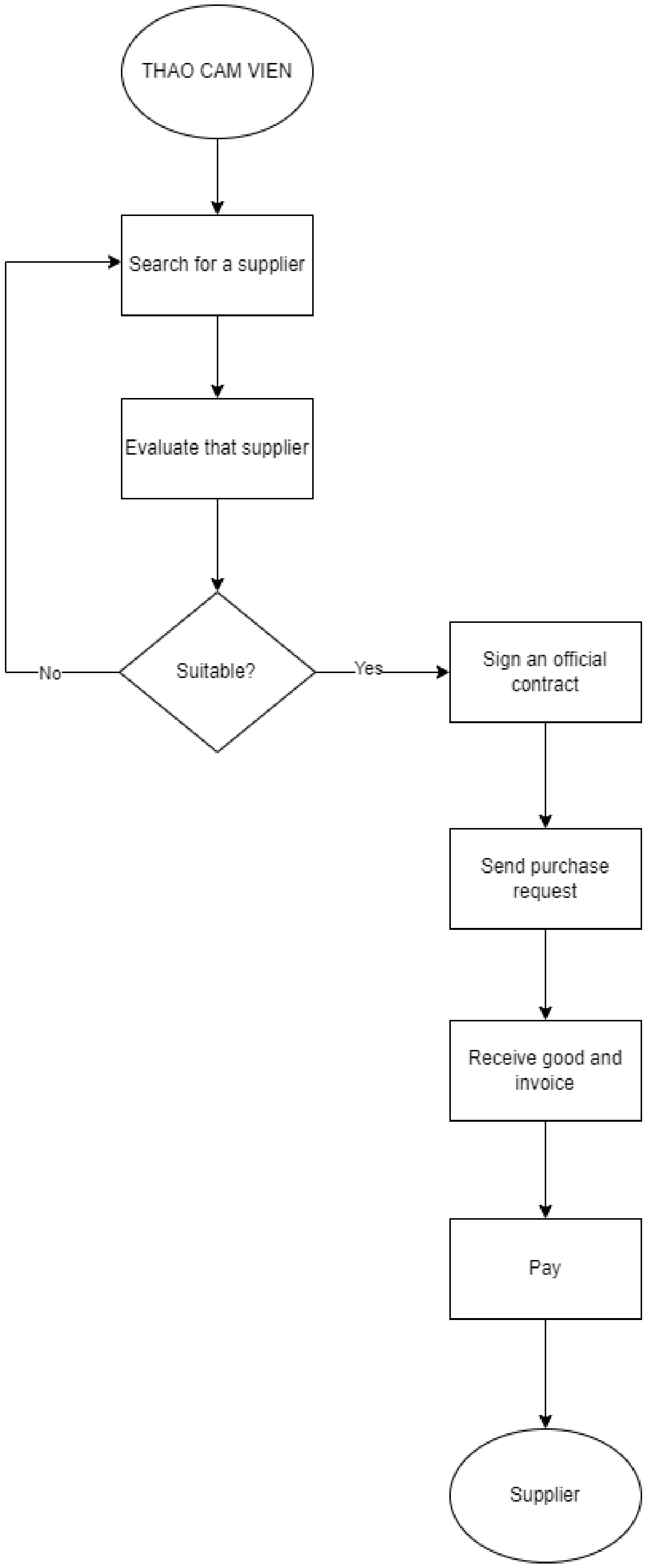


*Figure 2.5 Purchase Process of Thao Cam Vien*

- Thao Cam Vien starts to search for suppliers, evaluates suppliers based on many different criteria.
- If the supplier is not suitable, go back to search progress.

- If the supplier is suitable:

+ Suppliers will respond with information to Thao Cam Vien. If both parties reach an agreement, an official contract will be signed. Suppliers can make periodic deliveries or deliver when receiving orders from Thao Cam Vien depending on the contract.

+ Thao Cam Vien will send a purchase request to the supplier, the supplier will deliver the goods and create an invoice and send it to Thao Cam Vien.

+ After receiving the goods and invoice, Thao Cam Vien will make payment to the supplier.

### 2.2 Context DFD

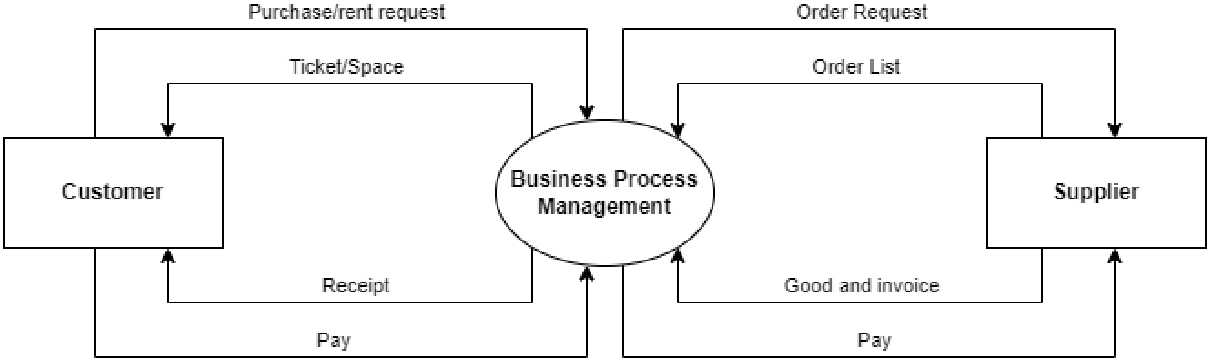


*Figure 2.6: Context DFD of Thao Cam Vien's business model*

### 2.3 Business report

Business reports show Thao Cam Vien's revenue and expenses. Besides, the report shows all items regarding Order Invoice and Receipt from Partners and Customers. In addition, the business report is full of information about employee salaries.

# CHAP 3. DESIGN ENTITY RELATIONSHIP MODEL ER (OR EER)

## 3.1 Entities and Attributes of Entities

- ***Employee:*** *People who work in the Thao Cam Vien*
  - **Attributes:** EmployeeID, FirstName, LastName, Date of Birth, Age, Sex, Hire Date, PhoneNumber, Email, Address, Salary, Department.
  - **Primary key:** EmployeeID
  - **Single value attributes:** EmployeeID, FirstName, LastName, Date of Birth, Hire Date, PhoneNumber, Email, Salary.
  - **Composite value attribute:** Address
  - **Derived attribute:** Age
  - **Not null attributes:** EmployeeID, FirstName, LastName, Date of Birth, Age, Hire Date, PhoneNumber, Email, Address, Salary.

- ***Management (Associative entity):*** *Management timeframe within the Thao Cam Vien*
  - **Attributes:** ManagementID, Workshift
  - **Primary key:** ManagementID
  - **Single value attributes:** ManagementID
  - **Multivalued attribute:** Workshift
  - **Not null attributes:** ManagementID, Workshift

- ***Services:*** *Paid services provided to customers*
  - **Attributes:** ServiceID, Name, Type, Area, Price
  - **Primary key:** ServiceID
  - **Single value attributes:** ServiceID, Name, Type, Area, Price
  - **Not null attributes:** ServiceID, Name, Type, Area, Price

- ***Activity Entertainment:*** *Entertainment activities*
  - **Attributes:** ActivityID, Name, Area
  - **Primary key:** ActivityID
  - **Single value attributes:** ActivityID, Name, Area
  - **Not null attributes:** ActivityID, Name, Area

- ***Plant:*** *Plants found in the Thao Cam Vien*
  - **Attributes:** PlantID, Name, Type, Area, Greenhouse, Origin, Description
  - **Primary key:** PlantID
  - **Single value attributes:** PlantID, Name, Type, Area, Greenhouse, Origin, Description
  - **Not null attributes:** PlantID, Name, Type, Area, Origin, Description

- ***Sale service (Associative entity):*** Manage the number of tickets and services sold over time
  - **Attributes:** SaleServiceID, Date, Quantity
  - **Primary key:** SaleServiceID
  - **Single value attributes:** Date, Quantity
  - **Not null attributes:** SaleServiceID, Date, Quantity

- ***Ticket:*** *Types of entrance tickets for customers*
  - **Attributes:** TicketID, Type, Price, Description
  - **Primary key:** TicketID
  - **Single value attributes:** TicketID, Type, Price, Description
  - **Not null attributes:** TicketID, Type, Price, Description

- ***Animal:*** *Details of animals in the Thao Cam Vien*
  - **Attributes:** AnimalID, Name, Date of Birth, Age, Day in, Area, Cage, Origin, Description.
  - **Primary key:** AnimalID
  - **Single value attributes:** AnimalID, Date of Birth, Day in, Area, Cage, Origin, Description.
  - **Derived attribute:** Age
  - **Not null attributes:** AnimalID, Day in, Area, Origin, Description.

- ***Species:*** *Species of animals in the Thao Cam Vien*
  - **Attributes:** SpeciesID, Name, Quantity
  - **Primary key:** SpeciesID
  - **Single value attributes:** SpeciesID, Name, Quantity

- **Not null attributes:** SpeciesID, Name, Quantity

- ***Medical record (Weak Entity):*** *Health status of animal species in the Thao Cam Vien*
  - **Attributes:** MedicalRecordID, History, Vaccine Status
  - **Primary key:** MedicalRecordID (Partial Identifier)
  - **Single value attributes:** MedicalRecordID, Vaccine Status
  - **Composite value attributes:** History
  - **Multi-valued attributes:** History
  - **Not null attributes:** MedicalRecordID, History, Vaccine Status

- ***Feeding Management (Associative entity):*** Management of feeding times for animals
  - **Attributes:** FManagementID, Workshift, Feeding Time
  - **Primary key:** FManagementID
  - **Single value attributes**: FManagementID, Workshift, Feeding Time
  - **Not null attributes:** FManagementID, Workshift, Feeding Time

- ***Food:*** Types of food used for animals to eat
  - **Attributes:** FoodID, Name
  - **Primary key:** FoodID
  - **Single value attributes:** FoodID, Name
  - **Not null attributes:** FoodID, Name

- ***Partner:*** Partners of the Thao Cam Vien
  - **Attributes:** PartnerID, Name, Phone, Address, Partner Type
  - **Primary key:** PartnerID
  - **Single value attributes:** PartnerID, Name, Phone, Partner Type
  - **Composite value attribute:** Address
  - **Not null attributes:** PartnerID, Name, Phone, Address, Partner Type

- ***Supplier:*** Supplier of animal food, equipment, tools, etc. at Thao Cam Vien
  - **Attributes:** SupplierID, Main Product Line
  - **Primary key:** SupplierID
  - **Single value attributes:** SupplierID, Main Product Line

- **Not null attributes:** SupplierID, Main Product Line

- ***Business:*** Partners rent business services at Thao Cam Vien
  - **Attributes:** BusinessID, Area, Business Field
  - **Primary key:** BusinessID
  - **Single value attributes:** BusinessID, Area, Business Field
  - **Not null attributes:** BusinessID, Area, Business Field

- ***Receipt:*** Invoices from business activities at Thao Cam Vien
  - **Attributes:** ReceiptID, Order Date, Price
  - **Primary key:** ReceiptID
  - **Single value attributes:** ReceiptID, Order Date, Price
  - **Not null attributes:** ReceiptID, Order Date, Price

- ***Customer (Online):*** Customers buy tickets online to visit Thao Cam Vien
  - **Attributes:** CustomerID, Name, Phone, Address
  - **Primary key:** CustomerID
  - **Single value attribute:** CustomerID, Name, Phone
  - **Composite value attribute:** Address
  - **Not null attributes:** CustomerID, Name, Phone, Address

- ***Order Invoice*:** Record business activities and purchase invoices from business activities with Supplier
  - **Attributes:** InvoiceID, Order Date, Price
  - **Primary key:** InvoiceID
  - **Single value attributes:** InvoiceID, Order Date, Price
  - **Not null attributes:** InvoiceID, Order Date, Price

### 3.2 Business rules:

- **Employee:**

- Each employee is identified by a unique ID and has information such as full name, date of birth, gender, salary, start date, phone number, email, address and department.

- Each employee is assigned many fixed shifts and can be assigned to many different tasks in Thao Cam Vien.

- **Species:**

- Each animal is identified by a unique ID and has information such as species name and quantity. In addition, each animal species also has a specific type of food and appropriate feeding time to ensure the best nutrition and health.

- Each animal is cared for, managed, and fed by staff according to staff shifts.

- **Food:**

- Each type of food needs to have a specific food supplier to ensure its quality and it is identified by a unique ID, containing information such as the name of the food and the supplier of that food.

- **Animal:**

- Each animal has information such as age, entry date, area, origin, and description saved in the database. In Thao Cam Vien, each animal has a specific cage or no cage.

- The health status and related medical issues of each animal are also monitored and the results recorded periodically to ensure their well-being.

- **Plant:**

- Each plant has a unique ID and has information such as name, type, description, area, origin. Each plant will be associated with a specific area and cared for by specific staff members according to staff shifts.

- **Activity Entertainment:**

- Each activity has a unique ID and information such as the activity name and where it takes place. These activities are performed by the animals under the management of staff to provide an enjoyable experience for customers when visiting Thao Cam Vien.

- **Services:**

- Each service provided to a customer is saved with an identification ID, name, service type, area and price. Services are also managed by staff members according to staff shifts.

- **Ticket:**

- Each ticket used to enter Thao Cam Vien has a unique ID and information about the ticket type and ticket price.

- **Partner:**

- Each business partner is identified by a unique identifier and has information such as name, address and contact phone number. A business partner can be a supplier or a collaborating enterprise:

+ Each supplier needs to have main product line attributes they offer for easier management. Suppliers submit their order invoices to Thao Cam Vien, which are managed by staff within Thao Cam Vien.

+ Each business cooperating to rent business premises must have the properties of the rental area in Thao Cam Vien and the business field of that business. Businesses receive payment invoices from Thao Cam Vien and Thao Cam Vien's staff.

- **Customer (Online):**

- Each visitor who books tickets online has a unique ID and information such as name, phone number, and address. Visitor information is used to manage their tour schedule and to record their payment.

### 3.3 Relationships:

| Name | Relationship | Cardinality Constraints | Description |
|---|---|---|---|
| R1 | Employee - Employee | **1 - n** | Each employee manages many employees or no employee, each employee is managed by another employee |
| R2 | Plant - Management | **1 - n** | Each plant has many managements, each management manages one or no plant |
| R3 | Species - Management | **1 - n** | Each species has many managements, each management manages one or no species |

| | | | |
|---|---|---|---|
| R4 | Employee - Management | **1- n** | Each employee has many or no management, each management is only managed by one employee |
| R5 | Services - Management | **1- n** | Each service has many managements, each each management manages one or no service |
| R6 | Employee - Feeding Management | **1 - n** | Each employee has many or no feeding management, each feeding management is only managed by one employee |
| R7 | Species - Feeding Management | **1 - n** | Each species is managed by many feeding managements, each feeding management only manages 1 species |
| R8 | Employee - Sale service | **1 - n** | Each employee manages many or no sale service, each sale service is managed by one and only one employee |
| R9 | Service - Sale Service | **1 - n** | Each service is |

|  |  |  |  |
|---|---|---|---|
|  |  |  | followed by many sale services, each sale service follows one or no service |
| R10 | Ticket - Sale service | **1 - n** | Each ticket is followed by many sale services, each sale service follows one or no ticket |
| R11 | Activity Entertainment - Management | **1 - n** | Each entertainment activity has many management, each management manages one or no entertainment activity. |
| R12 | Activity Entertainment - Animal | **1 - n** | Each entertainment activity is participated in by many or no animal, each animal participates in one or no entertainment activity. |
| R13 | Species - Animal | **1- n** | Each species has many animals, each animal belongs to one and only one species |
| R14 | Food - Species | **1 - n** | Each species eats only one food, each food is the food of one or many species. |
| R15 | Food - Supplier | **1 - n** | Each food is provided |

|  |  |  |  |
|---|---|---|---|
|  |  |  | by one and only one supplier, each supplier provides many or no food. |
| R16 | Medical Record - Animal | **1 - 1** | Each animal has 1 and only 1 medical record, each medical record belongs to 1 and only 1 animal |
| R17 | Partner - Supplier | **Supertype/Subtype** | Disjoint rule: A partner can be either a supplier or a business, but not both at the same time |
| R18 | Partner - Business |  |  |
| R19 | Business - Receipt | **1 - n** | Each business receives 1 or more receipts, each receipt belongs to one or no business |
| R20 | Employee - Receipt | **1 - n** | Each employee checks many or no receipt, each receipt is checked by 1 and only 1 employee |
| R21 | Customer - Receipt | **1 -1** | Each customer receives 1 and only 1 receipt, each receipt belongs to one or no customer |

| | | | |
|---|---|---|---|
| R22 | Employee - Order Invoice | **1 - n** | Each employee checks many or no order invoice, each order invoice is checked by 1 and only 1 employee |
| R23 | Supplier - Order Invoice | **1 - n** | Each supplier provides 1 or many order invoices, each order invoice is provided by 1 and only 1 supplier |

*Table 3.1 Relationship between Entities*

### 3.4 EER model

Link drawio: Link

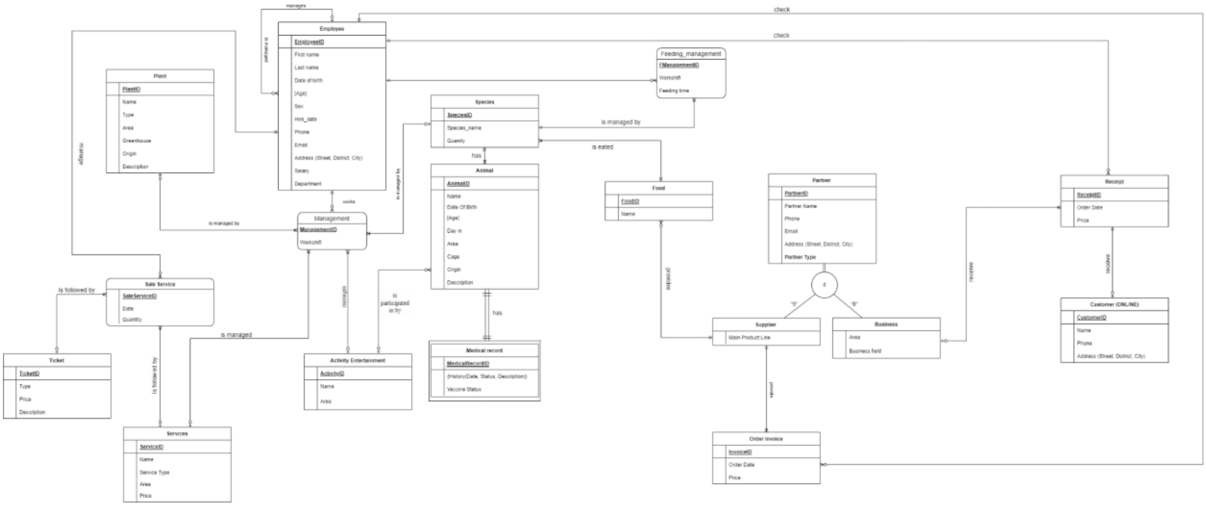


*Figure 3.1 EER Model*

**Assumption:**

- An employee has one fixed shift per day for an entire year.
- Each animal receives a health check at least once a year.

# CHAP 4. LOGICAL DATABASE DESIGN

**Link drawio (Trang 2):** [Link]()

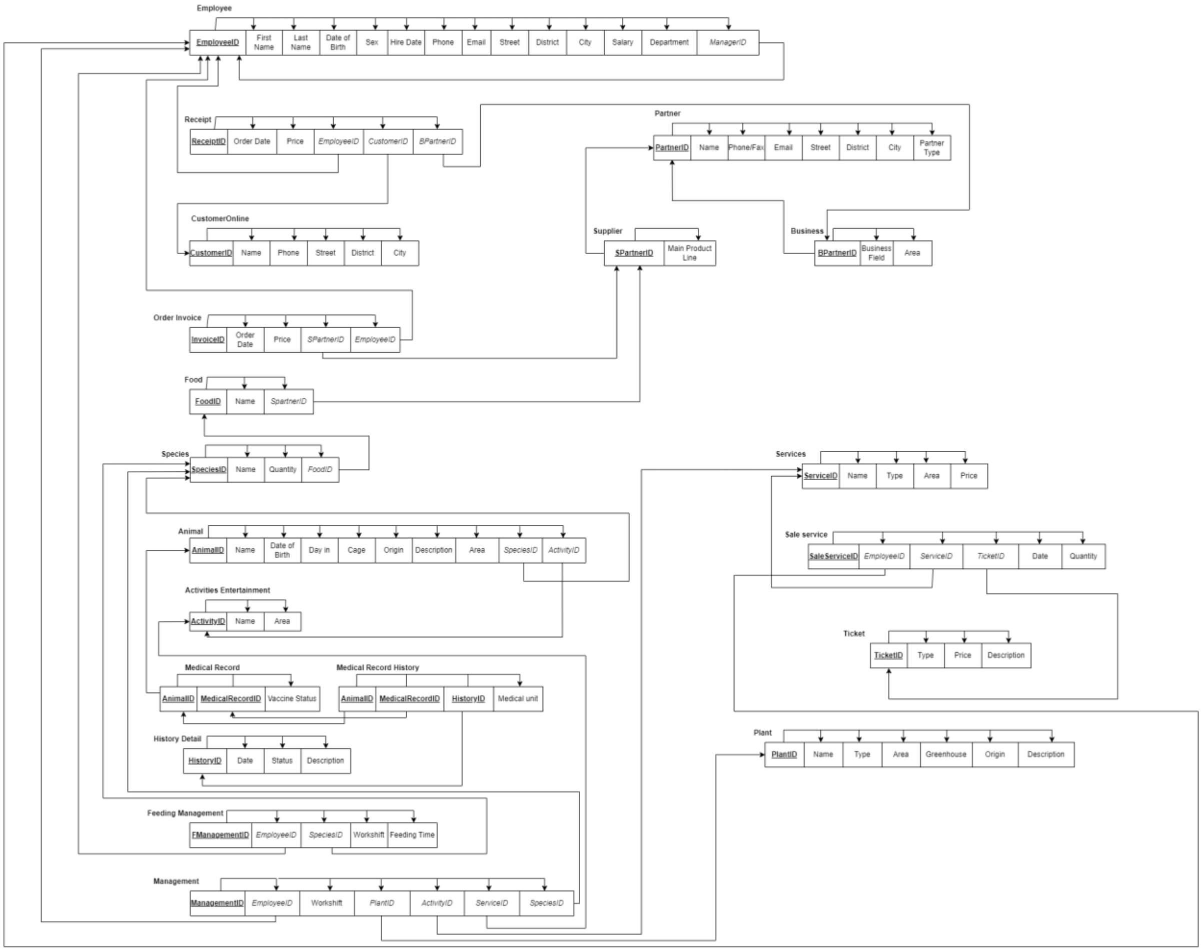


*Figure 4.1 3nf*

# CHAP 5. PHYSICAL DATABASE DESIGN

## 5.1 Data type:

| Entity | Attribute | Attribute Type | Key | Null |
|---|---|---|---|---|
| Employee | **EmployeeID** | CHAR(7) | PK | |
| | Last Name | NVARCHAR (40) | | |
| | First Name | NVARCHAR(10) | | |
| | Date of birth | DATE | | |
| | Sex | NCHAR(3) | | |
| | Hire Date | DATE | | |
| | Phone | VARCHAR(12) | | |
| | Email | NVARCHAR(30) | | |
| | Street | NVARCHAR(50) | | |
| | District | NVARCHAR(20) | | |
| | City | NVARCHAR(20) | | |
| | Department | NVARCHAR(25) | | |
| | Salary | MONEY | | |
| | *ManagerID* | CHAR(7) | FK | x |
| Management | **ManagementID** | CHAR(7) | PK | |
| | *EmployeeID* | CHAR(7) | FK | |
| | Work shift | NVARCHAR(30) | | |
| | *SpeciesID* | CHAR(5) | FK | x |
| | *ActivityID* | CHAR(5) | FK | x |

| | | | | |
|---|---|---|---|---|
| | *ServiceID* | CHAR(5) | FK | x |
| | *PlantID* | CHAR(5) | FK | x |
| Services | **ServiceID** | CHAR(5) | PK | |
| | Name | NVARCHAR(30) | | |
| | Type | NVARCHAR(5) | | |
| | Area | NVARCHAR(10) | | |
| | Price | MONEY | | |
| Activity Entertainment | **ActivityID** | CHAR(5) | PK | |
| | Name | NVARCHAR(15) | | |
| | Area | NVARCHAR(10) | | |
| Plant | **PlantID** | CHAR(5) | PK | |
| | Name | NVARCHAR(25) | | |
| | Type | NVARCHAR(15) | | |
| | Area | NVARCHAR(10) | | |
| | Greenhouse | TINYINT | | x |
| | Origin | NVARCHAR(15) | | |
| | Description | NVARCHAR(200) | | |
| Sale service | **SaleServiceID** | CHAR(5) | PK | |
| | *EmployeeID* | CHAR(7) | FK | |
| | *ServiceID* | CHAR(5) | FK | x |
| | *TicketID* | CHAR(4) | FK | x |
| | Date | DATE | | |
| | Quantity | INT | | |

|  |  |  |  |  |
|---|---|---|---|---|
| Ticket | **TicketID** | CHAR(4) | PK |  |
|  | Price | MONEY |  |  |
|  | Type | NVARCHAR(15) |  |  |
|  | Description | NVARCHAR(50) |  |  |
| Animal | **AnimalID** | CHAR(5) | PK |  |
|  | Name | NVARCHAR(15) |  | x |
|  | Date of birth | DATE |  | x |
|  | Day in | DATE |  |  |
|  | Cage | TINYINT |  | x |
|  | Origin | NVARCHAR(10) |  |  |
|  | Description | NVARCHAR(500) |  |  |
|  | Area | NVARCHAR(10) |  |  |
|  | *SpeciesID* | CHAR(5) | FK |  |
|  | *ActivityID* | CHAR(5) | FK | x |
| Species | **SpeciesID** | CHAR(5) | PK |  |
|  | Name | NVARCHAR(50) |  |  |
|  | Quantity | INT |  |  |
|  | *FoodID* | CHAR(4) | FK |  |
| Medical record | **AnimalID** | CHAR(5) | PK |  |
|  | **MedicalRecordID** | CHAR(4) | PK |  |
|  | Vaccine Status | NVARCHAR(10) |  |  |
|  | **AnimalID** | CHAR(5) | PK |  |

|  |  |  |  |  |
|---|---|---|---|---|
| Medical Record History | **MedicalRecordID** | CHAR(4) | PK |  |
|  | **HistoryID** | CHAR(4) | PK |  |
|  | Medical unit | NVARCHAR(40) |  |  |
| History Detail | **HistoryID** | CHAR(4) | PK |  |
|  | Date | DATETIME |  |  |
|  | Status | NVARCHAR(20) |  |  |
|  | Description | NVARCHAR(20) |  |  |
| Feeding Management | **FManagementID** | CHAR(5) | PK |  |
|  | Work Shift | NVARCHAR(50) |  |  |
|  | *EmployeeID* | CHAR(7) | FK |  |
|  | *SpeciesID* | CHAR(5) | FK |  |
|  | Feeding time | NVARCHAR(50) |  |  |
| Food | **FoodID** | CHAR(4) | PK |  |
|  | Name | NVARCHAR(50) |  |  |
|  | *PartnerID* | CHAR(4) | FK |  |
| Partner | **PartnerID** | CHAR(4) | PK |  |
|  | Name | NVARCHAR(50) |  |  |
|  | Phone | VARCHAR(12) |  |  |
|  | Email | NVARCHAR(30) |  |  |
|  | Street | NVARCHAR(50) |  |  |
|  | District | NVARCHAR(20) |  |  |

| | | | | |
|---|---|---|---|---|
| | City | NVARCHAR(20) | | |
| | Partner Type | NVARCHAR(10) | | |
| Supplier | **PartnerID** | CHAR(4) | PK, FK | |
| | Main Product Line | NVARCHAR(50) | | |
| Business | **PartnerID** | CHAR(4) | PK,FK | |
| | Business Field | NVARCHAR(50) | | |
| | Area | NVARCHAR(10) | | |
| Receipt | **ReceiptID** | CHAR(4) | PK | |
| | Order Date | DATETIME | | |
| | Price | MONEY | | |
| | *EmployeeID* | CHAR(7) | FK | |
| | *PartnerID* | CHAR(4) | FK | |
| | *CustomerID* | CHAR(4) | FK | |
| Customer Online | **CustomerID** | CHAR(4) | PK | |
| | Name | NVARCHAR(40) | | |
| | Phone | VARCHAR(12) | | |
| | Street | NVARCHAR(50) | | |
| | District | NVARCHAR(20) | | |
| | City | NVARCHAR(20) | | |
| | **InvoiceID** | CHAR(5) | PK | |
| | Order date | DATETIME | | |

|  |  |  |  |  |
|---|---|---|---|---|
| Order Invoice | Price | MONEY |  |  |
|  | *PartnerID* | CHAR(4) | FK |  |
|  | *EmployeeID* | CHAR(7) | FK |  |

*Table 5.1 Data Type*

### 5.2 Design THAOCAMVIEN database by SQL Server:

Link Github (có insert data): Link

Link Drive: Link

```sql
CREATE database THAOCAMVIEN

--Tạo bảng Partner
CREATE TABLE Partner(
    PartnerID CHAR(4) PRIMARY KEY,
    Name NVARCHAR(50) NOT NULL,
    Phone VARCHAR(12) NOT NULL,
    Email NVARCHAR(30) NOT NULL,
    Street NVARCHAR(50) NOT NULL,
    District NVARCHAR(20) NOT NULL,
    City NVARCHAR(20) NOT NULL,
    PartnerType NVARCHAR(10) NOT NULL);

--Tạo bảng Supplier
CREATE TABLE Supplier(
    PartnerID CHAR(4) PRIMARY KEY,
    MainProductLine NVARCHAR(50) NOT NULL,
    FOREIGN KEY (PartnerID) REFERENCES Partner(PartnerID));

--Tạo bảng Business
CREATE TABLE Business(
    PartnerID CHAR(4) PRIMARY KEY,
    BusinessField NVARCHAR(50) NOT NULL,
    Area NVARCHAR(10),
    FOREIGN KEY (PartnerID) REFERENCES Partner(PartnerID));

--Tạo bảng Food
CREATE TABLE Food(
       FoodID CHAR(4) PRIMARY KEY,
       Name NVARCHAR(50) NOT NULL,
       PartnerID CHAR(4) REFERENCES Supplier(PartnerID));


--Tạo bảng Customer Online
CREATE TABLE CustomerOnline(
       CustomerID CHAR(4) PRIMARY KEY,
       Name NVARCHAR(40) NOT NULL,
       Phone VARCHAR(12) NOT NULL,
       Street NVARCHAR(50) NOT NULL,
       District NVARCHAR(20) NOT NULL,
       City NVARCHAR(20) NOT NULL);
```

```
--Tạo bảng Employee
CREATE TABLE Employee(
       EmployeeID CHAR(7) PRIMARY KEY,
       LastName NVARCHAR(40) NOT NULL,
       FirstName NVARCHAR(10) NOT NULL,
       DateOfBirth DATE NOT NULL,
       Sex NCHAR(3) NOT NULL,
       HireDate DATE NOT NULL,
       Phone VARCHAR(12) NOT NULL,
       Email NVARCHAR(30) NOT NULL,
       Street NVARCHAR(50) NOT NULL,
       District NVARCHAR(20) NOT NULL,
       City NVARCHAR(20) NOT NULL,
       Department NVARCHAR(25) NOT NULL,
       Salary MONEY NOT NULL,
       ManagerID CHAR(7),
       CONSTRAINT FK_ManagerID FOREIGN KEY (ManagerID) REFERENCES
Employee(EmployeeID))

--Tạo bảng Receipt
CREATE TABLE Receipt(
    ReceiptID CHAR(4) PRIMARY KEY,
    OrderDate DATETIME NOT NULL,
    Price MONEY NOT NULL,
    EmployeeID CHAR(7) NOT NULL,
    PartnerID CHAR(4),
    CustomerID CHAR(4),
    FOREIGN KEY (PartnerID) REFERENCES Business(PartnerID));

ALTER TABLE Receipt
ADD CONSTRAINT FK_Receipt_Employee
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
CONSTRAINT FK_Receipt_Customer
FOREIGN KEY (CustomerID) REFERENCES CustomerOnline(CustomerID);

--Tạo bảng Order Invoice
CREATE TABLE OrderInvoice(
       InvoiceID CHAR(5) PRIMARY KEY,
       OrderDate DATETIME NOT NULL,
       Price MONEY NOT NULL,
       PartnerID CHAR(4) FOREIGN KEY
       REFERENCES Supplier(PartnerID),
       EmployeeID CHAR(7) NOT NULL,
       FOREIGN KEY (EmployeeID)
       REFERENCES Employee(EmployeeID));

-- Tạo bảng Ticket
CREATE TABLE Ticket(
       TicketID CHAR(4) PRIMARY KEY,
       Type NVARCHAR(15) NOT NULL,
       Price MONEY NOT NULL,
       Description NVARCHAR(50) NOT NULL)

--Tạo bảng Plant
CREATE TABLE Plant(
       PlantID CHAR(5) PRIMARY KEY,
       Name NVARCHAR (25) NOT NULL,
       Type NVARCHAR(15) NOT NULL,
       Area NVARCHAR(10) NOT NULL,
       GreenHouse TINYINT,
       Origin NVARCHAR(15) NOT NULL,
       Description NVARCHAR(200) NOT NULL)
```

```
--Tạo bảng Services
CREATE TABLE Services(
       ServiceID CHAR(5) PRIMARY KEY,
       Name NVARCHAR (30) NOT NULL,
       Type NVARCHAR(5) NOT NULL,
       Area NVARCHAR(10) NOT NULL,
       Price MONEY NOT NULL)

--Tạo bảng Activity Entertainment
CREATE TABLE Activity_Entertainment(
       ActivityID CHAR(5) PRIMARY KEY,
       Name NVARCHAR (15) NOT NULL,
       Area NVARCHAR(10) NOT NULL)

--Tạo bảng Species
CREATE TABLE Species(
       SpeciesID CHAR(5) PRIMARY KEY,
       Name NVARCHAR(50) NOT NULL,
       Quantity INT NOT NULL,
       FoodID CHAR(4) NOT NULL)

ALTER TABLE Species
ADD CONSTRAINT FK_Species_Food FOREIGN KEY (FoodID) REFERENCES Food(FoodID)

-- Tạo bảng Animal
CREATE TABLE Animal (
       AnimalID CHAR(5) PRIMARY KEY,
       Name NVARCHAR(15),
       DateOfBirth DATE,
       DayIn DATE NOT NULL,
       Cage TINYINT,
       Origin NVARCHAR(10) NOT NULL,
       Area NVARCHAR(10) NOT NULL,
       SpeciesID CHAR(5) NOT NULL,
       ActivityID CHAR(5),
       Description NVARCHAR(500) NOT NULL,
       FOREIGN KEY (SpeciesID) REFERENCES Species(SpeciesID),
       FOREIGN KEY (ActivityID) REFERENCES Activity_Entertainment(ActivityID))

--Tạo bảng Medical Record
CREATE TABLE Medical_Record (
       AnimalID CHAR(5),
       MedicalRecordID CHAR(4),
       VaccineStatus NVARCHAR(10) NOT NULL,
       PRIMARY KEY (AnimalID, MedicalRecordID),
       FOREIGN KEY (AnimalID)
       REFERENCES Animal (AnimalID))


--Tạo bảng History_Detail
CREATE TABLE History_Detail (
       HistoryID CHAR(4) PRIMARY KEY,
    Date DATETIME NOT NULL,
    Status NVARCHAR(20) NOT NULL,
    Description NVARCHAR(200) NOT NULL);

--Tạo bảng Medical record history
CREATE TABLE Medical_Record_History(
       AnimalID CHAR(5) NOT NULL,
    MedicalRecordID CHAR(4) NOT NULL,
       HistoryID CHAR(4) NOT NULL,
        MedicalRecordUnit NVARCHAR(40),
        PRIMARY KEY (AnimalID,MedicalRecordID,HistoryID),
        FOREIGN KEY (AnimalID, MedicalRecordID) REFERENCES Medical_Record (AnimalID,
MedicalRecordID),
        FOREIGN KEY (HistoryID) REFERENCES History_Detail (HistoryID))


--Tạo bảng Sale Service
CREATE TABLE SaleService ( SaleServiceID CHAR(5) PRIMARY KEY, EmployeeID CHAR(7) NOT
NULL,
ServiceID CHAR(5),TicketID CHAR(4), Date DATE NOT NULL, Quantity INT NOT NULL,
CONSTRAINT FK_SS_EmployeeID FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
CONSTRAINT FK_SS_ServiceID FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID),
CONSTRAINT FK_SS_TicketID FOREIGN KEY (TicketID) REFERENCES Ticket(TicketID));

--Tạo bảng Management
CREATE TABLE Management(ManagementID CHAR(7) PRIMARY KEY,EmployeeID CHAR(7) NOT NULL,
WorkShift NVARCHAR(30) NOT NULL,
SpeciesID CHAR(5), ActivityID CHAR(5), ServiceID CHAR(5), PlantID CHAR(5),
CONSTRAINT FK_EmployeeID FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
CONSTRAINT FK_ActivityID FOREIGN KEY (ActivityID) REFERENCES
Activity_Entertainment(ActivityID),
CONSTRAINT FK_ServiceID FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID),
CONSTRAINT FK_PlantID FOREIGN KEY (PlantID) REFERENCES Plant(PlantID),
CONSTRAINT FK_SpeciesID FOREIGN KEY (SpeciesID) REFERENCES Species(SpeciesID))


--Tạo bảng Feeding Management
CREATE TABLE FeedingManagement(
    FManagementID CHAR(5) PRIMARY KEY,
    EmployeeID CHAR(7) NOT NULL,
       FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
    SpeciesID CHAR(5) NOT NULL,
       FOREIGN KEY (SpeciesID) REFERENCES Species(SpeciesID),
       WorkShift NVARCHAR(50) NOT NULL,
        FeedingTime NVARCHAR(50) NOT NULL);
```

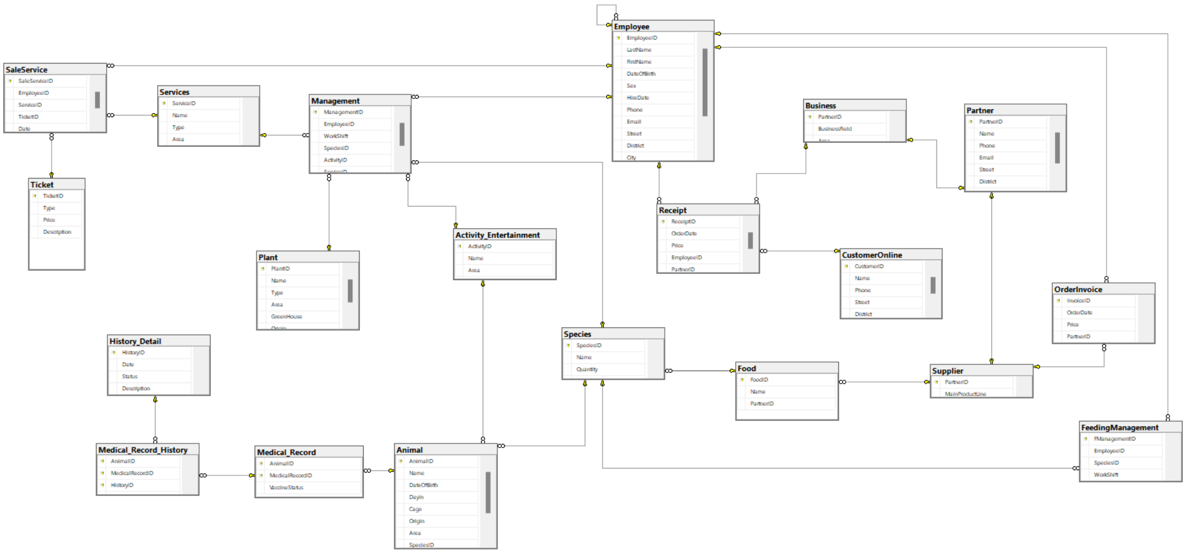


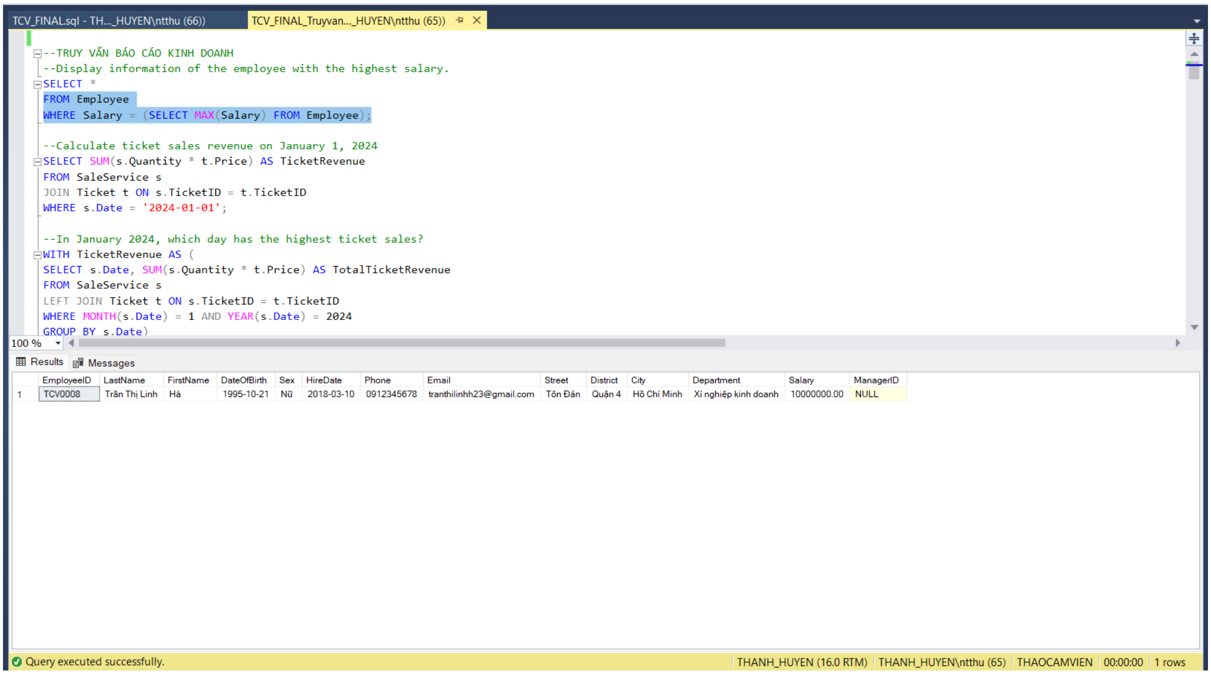
*Figure 5.1 Diagram*

# CHAP 6. QUERY COMMANDS FOR BUSINESS REPORT

Link Github: Link

Link Drive: Link

```
--TRUY VẤN BÁO CÁO KINH DOANH
--Display information of the employee with the highest salary.
SELECT *
FROM Employee
WHERE Salary = (SELECT MAX(Salary) FROM Employee);
```



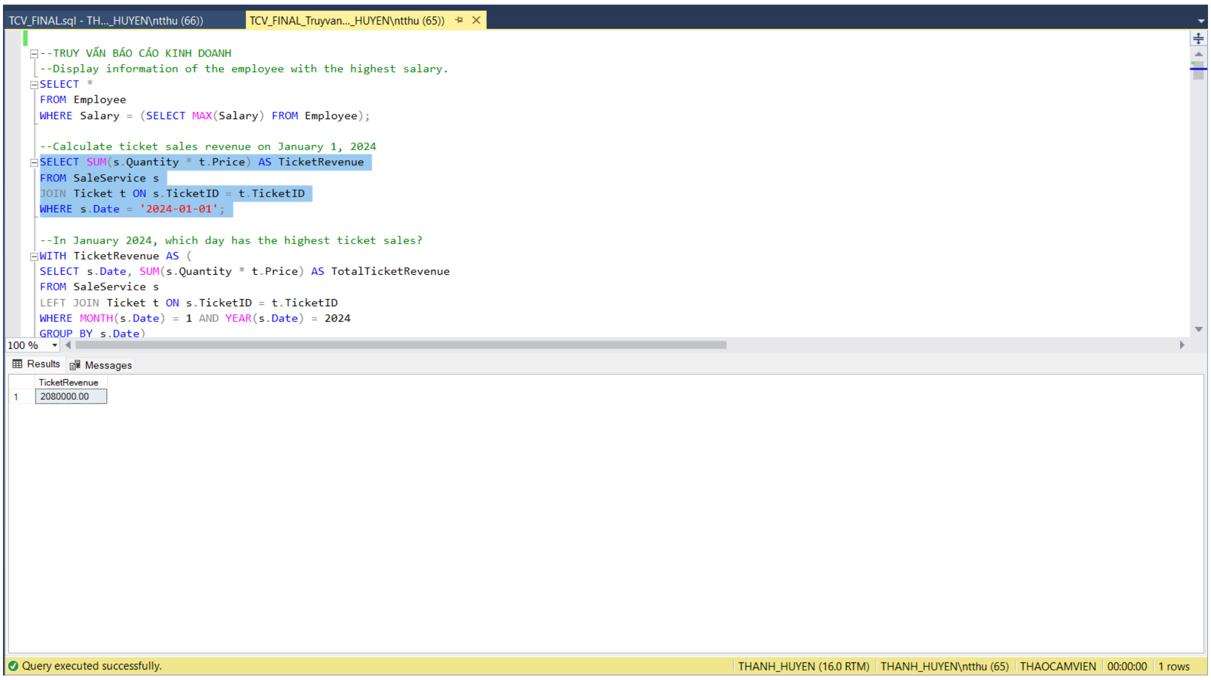
TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```
--TRUY VẤN BÁO CÁO KINH DOANH
--Display information of the employee with the highest salary.
SELECT *
FROM Employee
WHERE Salary = (SELECT MAX(Salary) FROM Employee);

--Calculate ticket sales revenue on January 1, 2024
SELECT SUM(s.Quantity * t.Price) AS TicketRevenue
FROM SaleService s
JOIN Ticket t ON s.TicketID = t.TicketID
WHERE s.Date = '2024-01-01';

--In January 2024, which day has the highest ticket sales?
WITH TicketRevenue AS (
SELECT s.Date, SUM(s.Quantity * t.Price) AS TotalTicketRevenue
FROM SaleService s
LEFT JOIN Ticket t ON s.TicketID = t.TicketID
WHERE MONTH(s.Date) = 1 AND YEAR(s.Date) = 2024
GROUP BY s.Date)
```

| | EmployeeID | LastName | FirstName | DateOfBirth | Sex | HireDate | Phone | Email | Street | District | City | Department | Salary | ManagerID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TCV0008 | Trần Thị Linh | Hà | 1995-10-21 | Nữ | 2018-03-10 | 0912345678 | tranthilinhh23@gmail.com | Tôn Đản | Quận 4 | Hồ Chí Minh | Xí nghiệp kinh doanh | 10000000.00 | NULL |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows

```
--Calculate ticket sales revenue on January 1, 2024
SELECT SUM(s.Quantity * t.Price) AS TicketRevenue
FROM SaleService s
JOIN Ticket t ON s.TicketID = t.TicketID
WHERE s.Date = '2024-01-01';
```

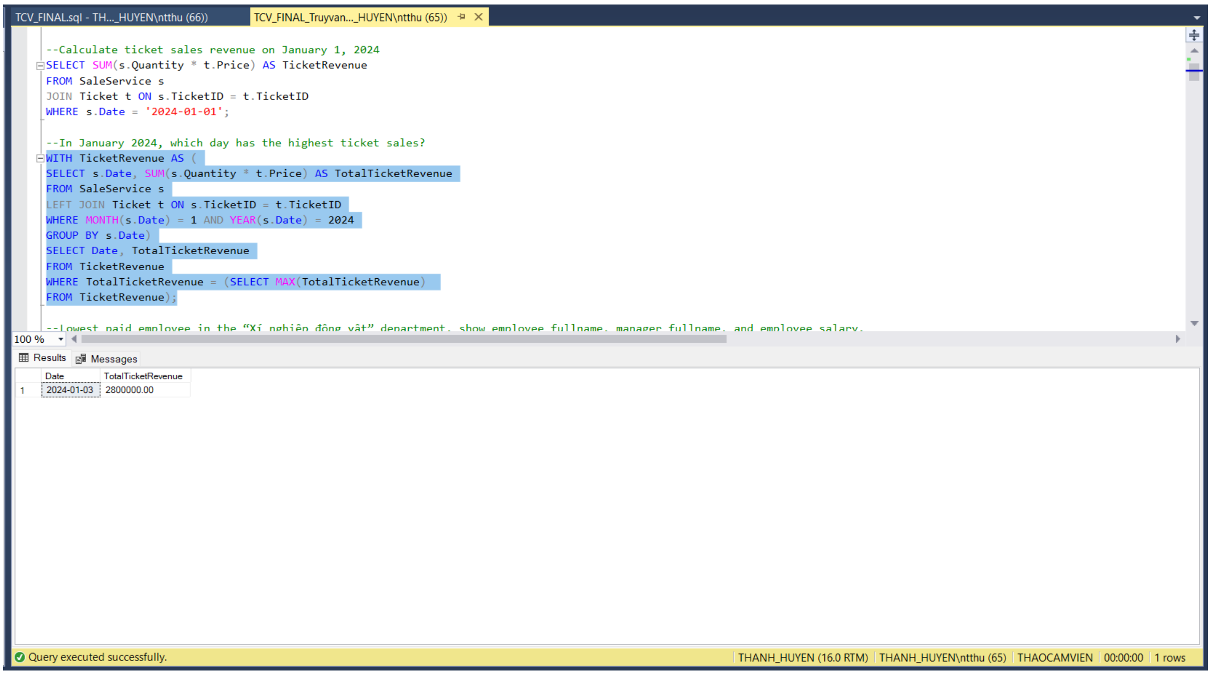

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```sql
--TRUY VẤN BÁO CÁO KINH DOANH
--Display information of the employee with the highest salary.
SELECT *
FROM Employee
WHERE Salary = (SELECT MAX(Salary) FROM Employee);

--Calculate ticket sales revenue on January 1, 2024
SELECT SUM(s.Quantity * t.Price) AS TicketRevenue
FROM SaleService s
JOIN Ticket t ON s.TicketID = t.TicketID
WHERE s.Date = '2024-01-01';

--In January 2024, which day has the highest ticket sales?
WITH TicketRevenue AS (
SELECT s.Date, SUM(s.Quantity * t.Price) AS TotalTicketRevenue
FROM SaleService s
LEFT JOIN Ticket t ON s.TicketID = t.TicketID
WHERE MONTH(s.Date) = 1 AND YEAR(s.Date) = 2024
GROUP BY s.Date)
```

|   | TicketRevenue |
|---|---|
| 1 | 2080000.00 |

Query executed successfully. THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows

```sql
--In January 2024, which day has the highest ticket sales?
WITH TicketRevenue AS (
SELECT s.Date, SUM(s.Quantity * t.Price) AS TotalTicketRevenue
FROM SaleService s
LEFT JOIN Ticket t ON s.TicketID = t.TicketID
WHERE MONTH(s.Date) = 1 AND YEAR(s.Date) = 2024
GROUP BY s.Date)
SELECT Date, TotalTicketRevenue
FROM TicketRevenue
WHERE TotalTicketRevenue = (SELECT MAX(TotalTicketRevenue)
FROM TicketRevenue);
```


TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

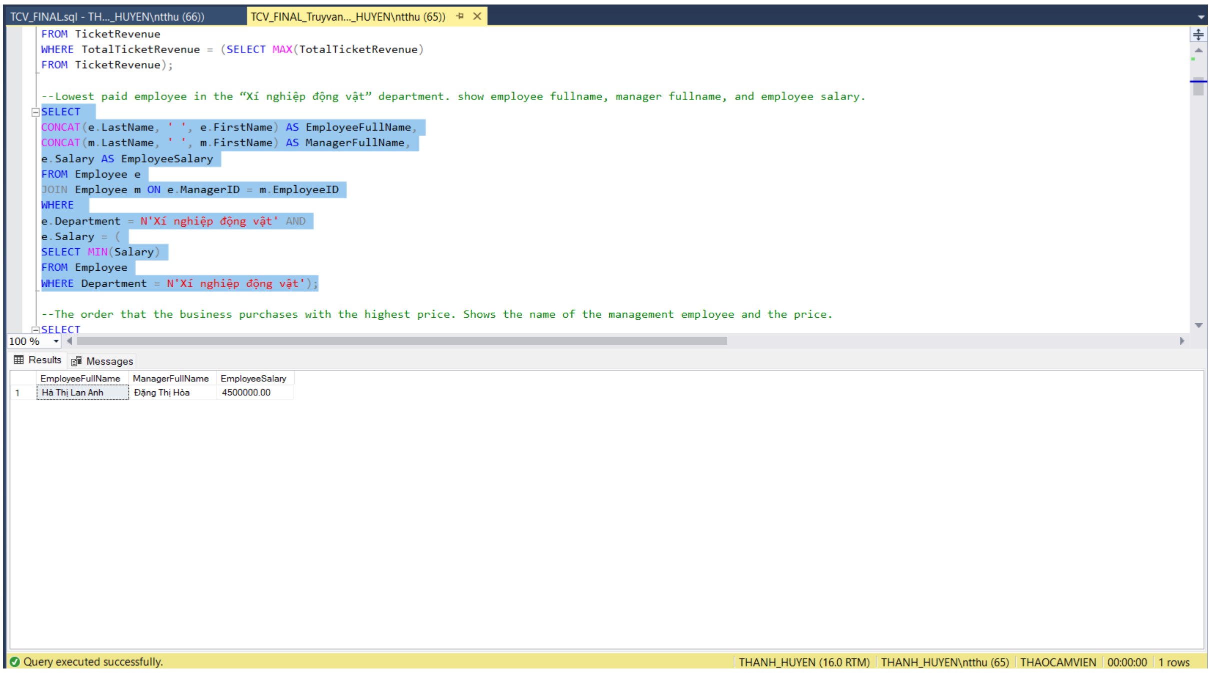
```sql
--Calculate ticket sales revenue on January 1, 2024
SELECT SUM(s.Quantity * t.Price) AS TicketRevenue
FROM SaleService s
JOIN Ticket t ON s.TicketID = t.TicketID
WHERE s.Date = '2024-01-01';

--In January 2024, which day has the highest ticket sales?
WITH TicketRevenue AS (
SELECT s.Date, SUM(s.Quantity * t.Price) AS TotalTicketRevenue
FROM SaleService s
LEFT JOIN Ticket t ON s.TicketID = t.TicketID
WHERE MONTH(s.Date) = 1 AND YEAR(s.Date) = 2024
GROUP BY s.Date)
SELECT Date, TotalTicketRevenue
FROM TicketRevenue
WHERE TotalTicketRevenue = (SELECT MAX(TotalTicketRevenue)
FROM TicketRevenue);

--Lowest paid employee in the “Xí nghiệp động vật” department. show employee fullname, manager fullname, and employee salary.
```

|   | Date | TotalTicketRevenue |
|---|---|---|
| 1 | 2024-01-03 | 2800000.00 |

Query executed successfully. THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows

```sql
--Lowest paid employee in the “Xí nghiệp động vật” department. show employee fullname, manager fullname, and employee salary.
SELECT
```

```
CONCAT(e.LastName, ' ', e.FirstName) AS EmployeeFullName,
CONCAT(m.LastName, ' ', m.FirstName) AS ManagerFullName,
e.Salary AS EmployeeSalary
FROM Employee e
JOIN Employee m ON e.ManagerID = m.EmployeeID
WHERE
e.Department = N'Xí nghiệp động vật' AND
e.Salary = (
SELECT MIN(Salary)
FROM Employee
WHERE Department = N'Xí nghiệp động vật');
```



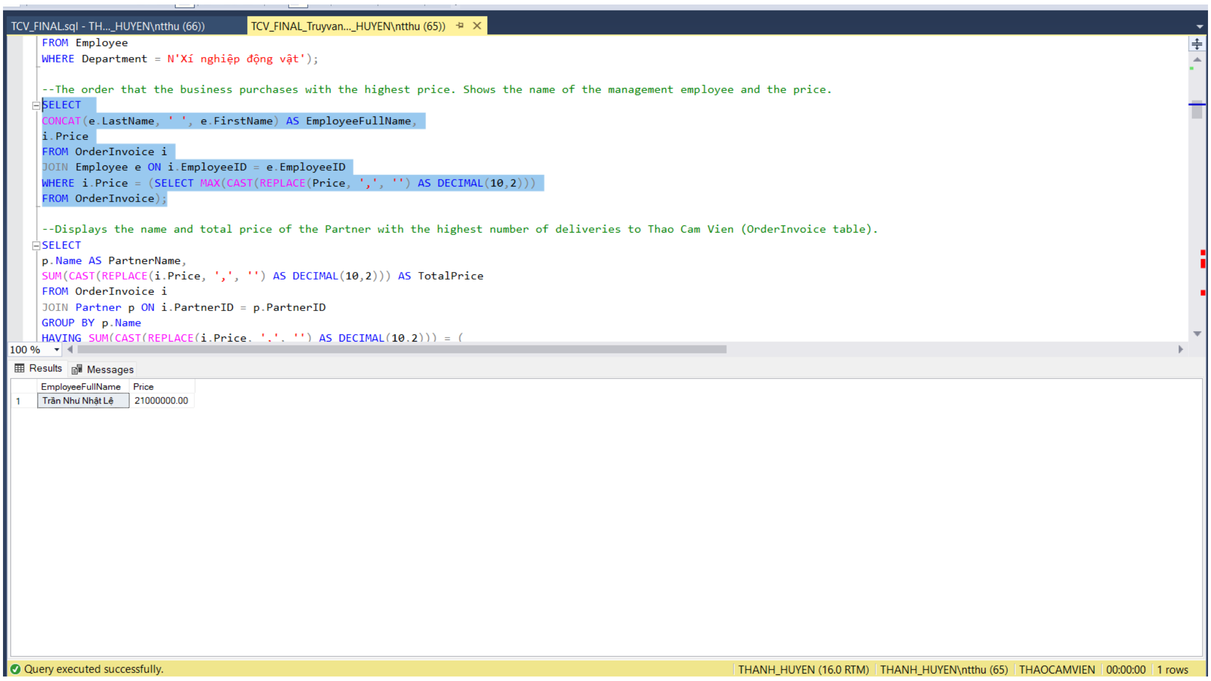
TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```
FROM TicketRevenue
WHERE TotalTicketRevenue = (SELECT MAX(TotalTicketRevenue)
FROM TicketRevenue);

--Lowest paid employee in the “Xí nghiệp động vật” department. show employee fullname, manager fullname, and employee salary.
SELECT
CONCAT(e.LastName, ' ', e.FirstName) AS EmployeeFullName,
CONCAT(m.LastName, ' ', m.FirstName) AS ManagerFullName,
e.Salary AS EmployeeSalary
FROM Employee e
JOIN Employee m ON e.ManagerID = m.EmployeeID
WHERE
e.Department = N'Xí nghiệp động vật' AND
e.Salary = (
SELECT MIN(Salary)
FROM Employee
WHERE Department = N'Xí nghiệp động vật');

--The order that the business purchases with the highest price. Shows the name of the management employee and the price.
SELECT
```

|   | EmployeeFullName | ManagerFullName | EmployeeSalary |
|---|---|---|---|
| 1 | Hà Thị Lan Anh | Đặng Thị Hòa | 4500000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows

```
--The order that the business purchases with the highest price. Shows the name of the
management employee and the price.
SELECT
CONCAT(e.LastName, ' ', e.FirstName) AS EmployeeFullName,
i.Price
FROM OrderInvoice i
JOIN Employee e ON i.EmployeeID = e.EmployeeID
WHERE i.Price = (SELECT MAX(CAST(REPLACE(Price, ',', '') AS DECIMAL(10,2)))
FROM OrderInvoice);
```

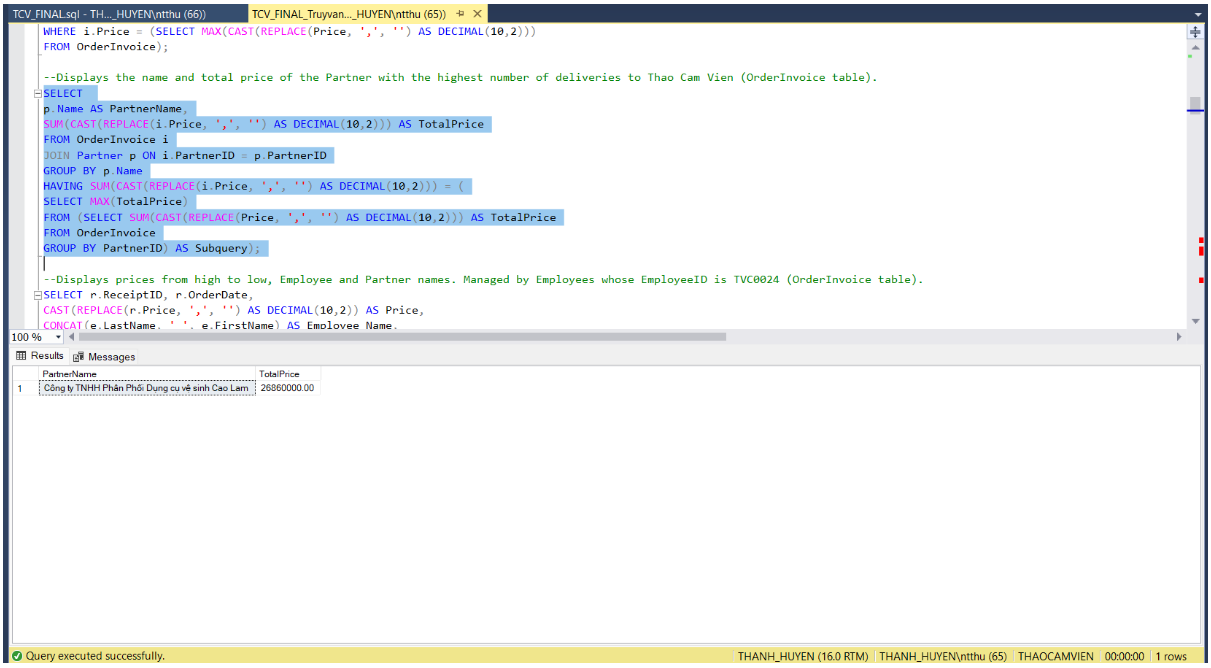

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```sql
FROM Employee
WHERE Department = N'Xí nghiệp động vật');

--The order that the business purchases with the highest price. Shows the name of the management employee and the price.
SELECT
CONCAT(e.LastName, ' ', e.FirstName) AS EmployeeFullName,
i.Price
FROM OrderInvoice i
JOIN Employee e ON i.EmployeeID = e.EmployeeID
WHERE i.Price = (SELECT MAX(CAST(REPLACE(Price, ',', '') AS DECIMAL(10,2)))
FROM OrderInvoice);

--Displays the name and total price of the Partner with the highest number of deliveries to Thao Cam Vien (OrderInvoice table).
SELECT
p.Name AS PartnerName,
SUM(CAST(REPLACE(i.Price, ',', '') AS DECIMAL(10,2))) AS TotalPrice
FROM OrderInvoice i
JOIN Partner p ON i.PartnerID = p.PartnerID
GROUP BY p.Name
HAVING SUM(CAST(REPLACE(i.Price, ',', '') AS DECIMAL(10,2))) = (
```

|  | EmployeeFullName | Price |
|---|---|---|
| 1 | Trần Như Nhật Lệ | 21000000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows

```sql
--Displays the name and total price of the Partner with the highest number of
deliveries to Thao Cam Vien (OrderInvoice table).
SELECT
p.Name AS PartnerName,
SUM(CAST(REPLACE(i.Price, ',', '') AS DECIMAL(10,2))) AS TotalPrice
FROM OrderInvoice i
JOIN Partner p ON i.PartnerID = p.PartnerID
GROUP BY p.Name
HAVING SUM(CAST(REPLACE(i.Price, ',', '') AS DECIMAL(10,2))) = (
SELECT MAX(TotalPrice)
FROM (SELECT SUM(CAST(REPLACE(Price, ',', '') AS DECIMAL(10,2))) AS TotalPrice
FROM OrderInvoice
GROUP BY PartnerID) AS Subquery);
```

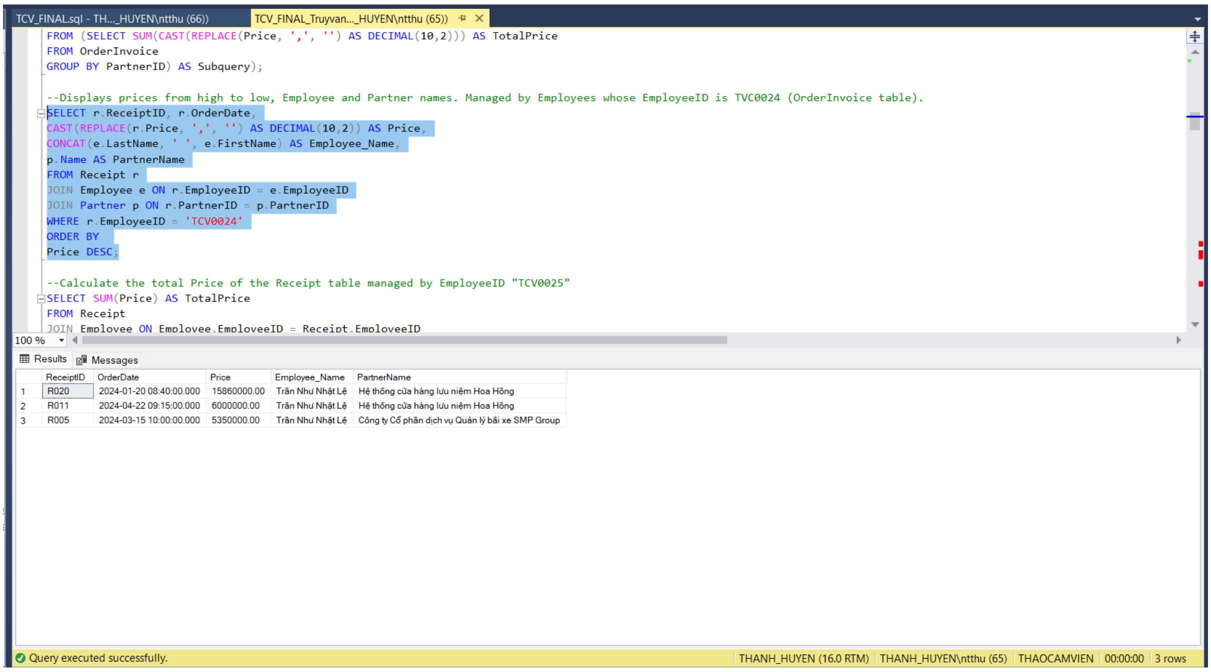

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```sql
WHERE i.Price = (SELECT MAX(CAST(REPLACE(Price, ',', '') AS DECIMAL(10,2)))
FROM OrderInvoice);

--Displays the name and total price of the Partner with the highest number of deliveries to Thao Cam Vien (OrderInvoice table).
SELECT
p.Name AS PartnerName,
SUM(CAST(REPLACE(i.Price, ',', '') AS DECIMAL(10,2))) AS TotalPrice
FROM OrderInvoice i
JOIN Partner p ON i.PartnerID = p.PartnerID
GROUP BY p.Name
HAVING SUM(CAST(REPLACE(i.Price, ',', '') AS DECIMAL(10,2))) = (
SELECT MAX(TotalPrice)
FROM (SELECT SUM(CAST(REPLACE(Price, ',', '') AS DECIMAL(10,2))) AS TotalPrice
FROM OrderInvoice
GROUP BY PartnerID) AS Subquery);

--Displays prices from high to low, Employee and Partner names. Managed by Employees whose EmployeeID is TVC0024 (OrderInvoice table).
SELECT r.ReceiptID, r.OrderDate,
CAST(REPLACE(r.Price, ',', '') AS DECIMAL(10,2)) AS Price,
CONCAT(e.LastName, ' ', e.FirstName) AS Employee_Name,
```

|  | PartnerName | TotalPrice |
|---|---|---|
| 1 | Công ty TNHH Phân Phối Dụng cụ vệ sinh Cao Lam | 26860000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows

```sql
--Displays prices from high to low, Employee and Partner names. Managed by Employees
whose EmployeeID is TVC0024 (OrderInvoice table).
SELECT r.ReceiptID, r.OrderDate,
CAST(REPLACE(r.Price, ',', '') AS DECIMAL(10,2)) AS Price,
CONCAT(e.LastName, ' ', e.FirstName) AS Employee_Name,
p.Name AS PartnerName
FROM Receipt r
JOIN Employee e ON r.EmployeeID = e.EmployeeID
JOIN Partner p ON r.PartnerID = p.PartnerID
WHERE r.EmployeeID = 'TCV0024'
ORDER BY
Price DESC;
```

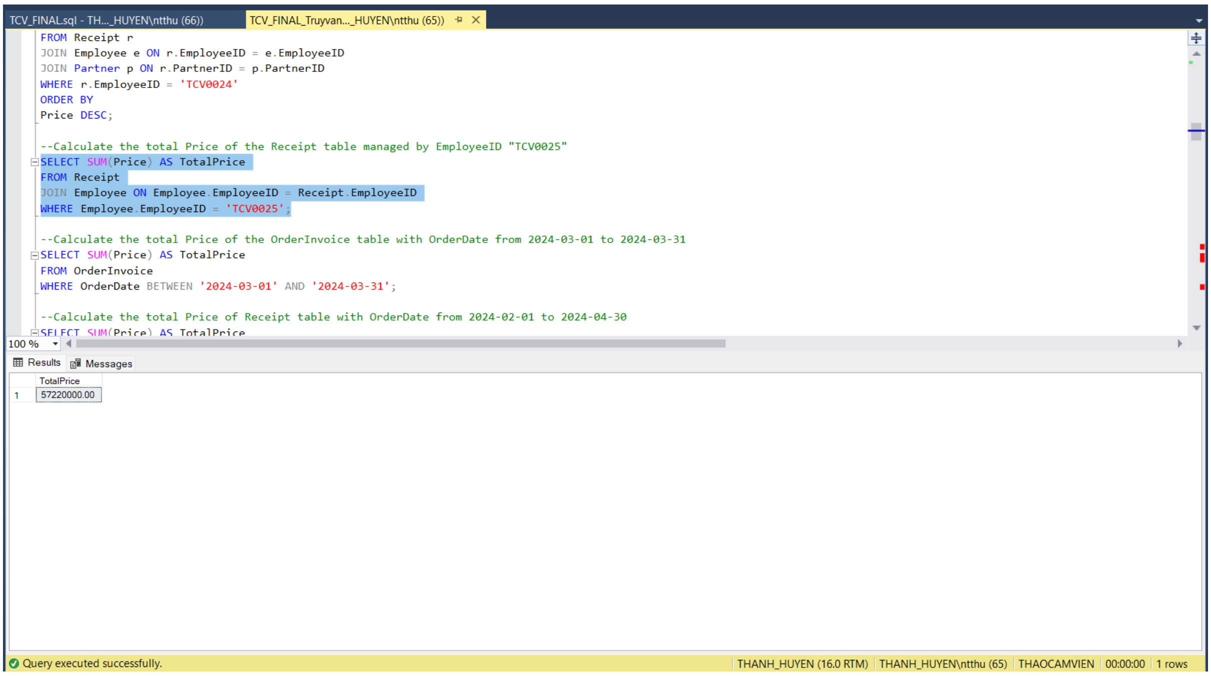

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```sql
FROM (SELECT SUM(CAST(REPLACE(Price, ',', '') AS DECIMAL(10,2))) AS TotalPrice
FROM OrderInvoice
GROUP BY PartnerID) AS Subquery);

--Displays prices from high to low, Employee and Partner names. Managed by Employees whose EmployeeID is TVC0024 (OrderInvoice table).
SELECT r.ReceiptID, r.OrderDate,
CAST(REPLACE(r.Price, ',', '') AS DECIMAL(10,2)) AS Price,
CONCAT(e.LastName, ' ', e.FirstName) AS Employee_Name,
p.Name AS PartnerName
FROM Receipt r
JOIN Employee e ON r.EmployeeID = e.EmployeeID
JOIN Partner p ON r.PartnerID = p.PartnerID
WHERE r.EmployeeID = 'TCV0024'
ORDER BY
Price DESC;

--Calculate the total Price of the Receipt table managed by EmployeeID "TCV0025"
SELECT SUM(Price) AS TotalPrice
FROM Receipt
JOIN Employee ON Employee.EmployeeID = Receipt.EmployeeID
```

| | ReceiptID | OrderDate | Price | Employee_Name | PartnerName |
|---|---|---|---|---|---|
| 1 | R020 | 2024-01-20 08:40:00.000 | 15860000.00 | Trần Như Nhật Lệ | Hệ thống cửa hàng lưu niệm Hoa Hồng |
| 2 | R011 | 2024-04-22 09:15:00.000 | 6000000.00 | Trần Như Nhật Lệ | Hệ thống cửa hàng lưu niệm Hoa Hồng |
| 3 | R005 | 2024-03-15 10:00:00.000 | 5350000.00 | Trần Như Nhật Lệ | Công ty Cổ phần dịch vụ Quản lý bãi xe SMP Group |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 3 rows

```sql
--Calculate the total Price of the Receipt table managed by EmployeeID "TCV0025"
SELECT SUM(Price) AS TotalPrice
FROM Receipt
JOIN Employee ON Employee.EmployeeID = Receipt.EmployeeID
WHERE Employee.EmployeeID = 'TCV0025';
```

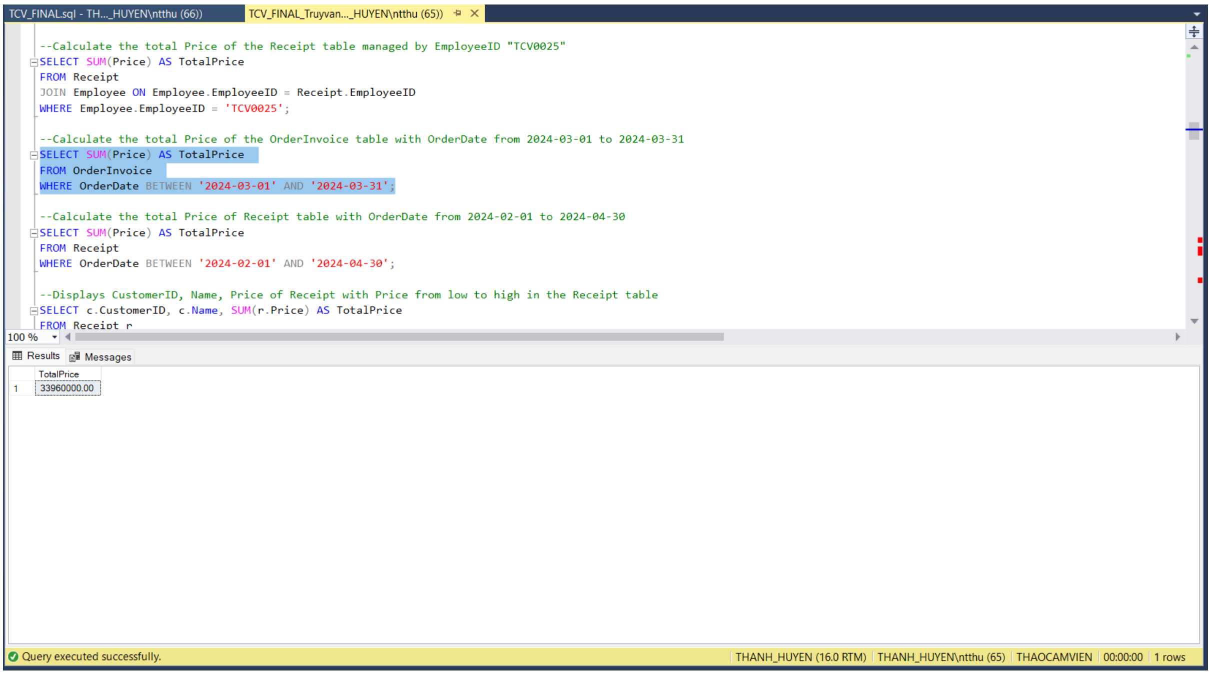

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```
FROM Receipt r
JOIN Employee e ON r.EmployeeID = e.EmployeeID
JOIN Partner p ON r.PartnerID = p.PartnerID
WHERE r.EmployeeID = 'TCV0024'
ORDER BY
Price DESC;

--Calculate the total Price of the Receipt table managed by EmployeeID "TCV0025"
SELECT SUM(Price) AS TotalPrice
FROM Receipt
JOIN Employee ON Employee.EmployeeID = Receipt.EmployeeID
WHERE Employee.EmployeeID = 'TCV0025';

--Calculate the total Price of the OrderInvoice table with OrderDate from 2024-03-01 to 2024-03-31
SELECT SUM(Price) AS TotalPrice
FROM OrderInvoice
WHERE OrderDate BETWEEN '2024-03-01' AND '2024-03-31';

--Calculate the total Price of Receipt table with OrderDate from 2024-02-01 to 2024-04-30
SELECT SUM(Price) AS TotalPrice
```

|   | TotalPrice |
|---|---|
| 1 | 57220000.00 |

```
Query executed successfully.    THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows
```

```
--Calculate the total Price of the OrderInvoice table with OrderDate from 2024-03-01
to 2024-03-31
SELECT SUM(Price) AS TotalPrice
FROM OrderInvoice
WHERE OrderDate BETWEEN '2024-03-01' AND '2024-03-31';
```

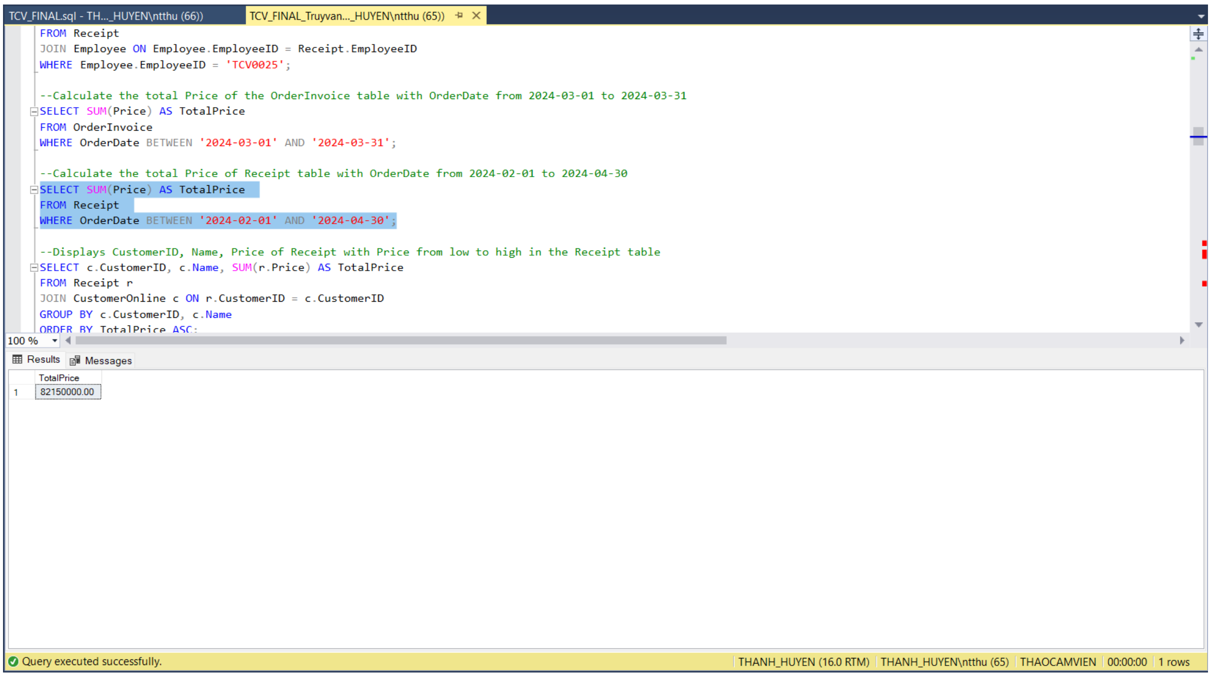

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```
--Calculate the total Price of the Receipt table managed by EmployeeID "TCV0025"
SELECT SUM(Price) AS TotalPrice
FROM Receipt
JOIN Employee ON Employee.EmployeeID = Receipt.EmployeeID
WHERE Employee.EmployeeID = 'TCV0025';

--Calculate the total Price of the OrderInvoice table with OrderDate from 2024-03-01 to 2024-03-31
SELECT SUM(Price) AS TotalPrice
FROM OrderInvoice
WHERE OrderDate BETWEEN '2024-03-01' AND '2024-03-31';

--Calculate the total Price of Receipt table with OrderDate from 2024-02-01 to 2024-04-30
SELECT SUM(Price) AS TotalPrice
FROM Receipt
WHERE OrderDate BETWEEN '2024-02-01' AND '2024-04-30';

--Displays CustomerID, Name, Price of Receipt with Price from low to high in the Receipt table
SELECT c.CustomerID, c.Name, SUM(r.Price) AS TotalPrice
FROM Receipt r
```

|   | TotalPrice |
|---|---|
| 1 | 33960000.00 |

```
Query executed successfully.    THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 1 rows
```

```
--Calculate the total Price of Receipt table with OrderDate from 2024-02-01 to 2024-
04-30
SELECT SUM(Price) AS TotalPrice
FROM Receipt
WHERE OrderDate BETWEEN '2024-02-01' AND '2024-04-30';
```



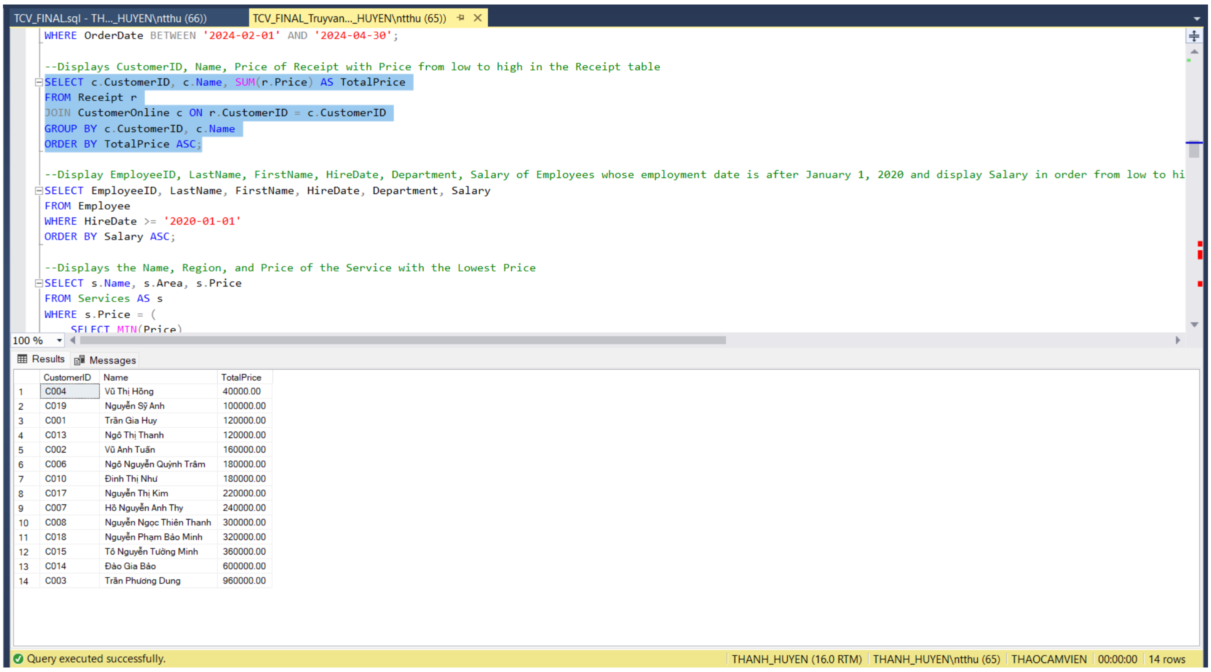
TCV_FINAL.sql - TH..._HUYEN\ntthu (66))    TCV_FINAL_Truyvan..._HUYEN\ntthu (65))

```sql
FROM Receipt
JOIN Employee ON Employee.EmployeeID = Receipt.EmployeeID
WHERE Employee.EmployeeID = 'TCV0025';

--Calculate the total Price of the OrderInvoice table with OrderDate from 2024-03-01 to 2024-03-31
SELECT SUM(Price) AS TotalPrice
FROM OrderInvoice
WHERE OrderDate BETWEEN '2024-03-01' AND '2024-03-31';

--Calculate the total Price of Receipt table with OrderDate from 2024-02-01 to 2024-04-30
SELECT SUM(Price) AS TotalPrice
FROM Receipt
WHERE OrderDate BETWEEN '2024-02-01' AND '2024-04-30';

--Displays CustomerID, Name, Price of Receipt with Price from low to high in the Receipt table
SELECT c.CustomerID, c.Name, SUM(r.Price) AS TotalPrice
FROM Receipt r
JOIN CustomerOnline c ON r.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.Name
ORDER BY TotalPrice ASC;
```

| | TotalPrice |
|---|---|
| 1 | 82150000.00 |

Query executed successfully.    THANH_HUYEN (16.0 RTM)    THANH_HUYEN\ntthu (65)    THAOCAMVIEN    00:00:00    1 rows

```sql
--Displays CustomerID, Name, Price of Receipt with Price from low to high in the
Receipt table
SELECT c.CustomerID, c.Name, SUM(r.Price) AS TotalPrice
FROM Receipt r
JOIN CustomerOnline c ON r.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.Name
ORDER BY TotalPrice ASC;
```



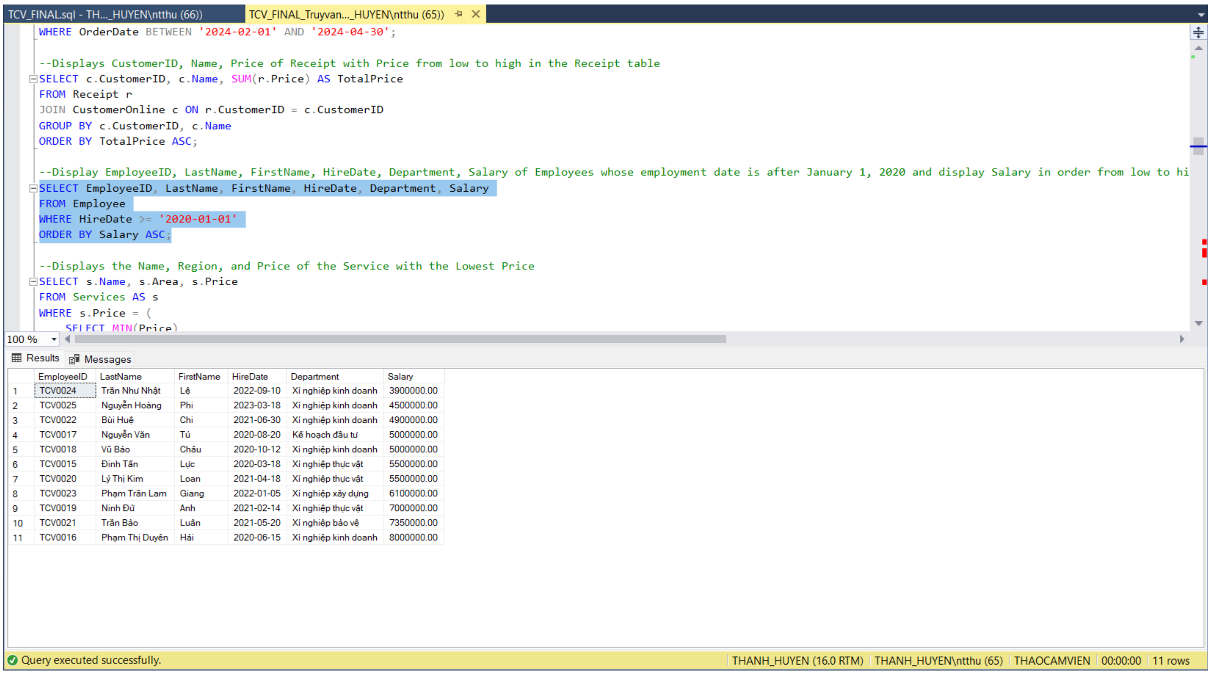
TCV_FINAL.sql - TH..._HUYEN\ntthu (66))    TCV_FINAL_Truyvan..._HUYEN\ntthu (65))

```sql
WHERE OrderDate BETWEEN '2024-02-01' AND '2024-04-30';

--Displays CustomerID, Name, Price of Receipt with Price from low to high in the Receipt table
SELECT c.CustomerID, c.Name, SUM(r.Price) AS TotalPrice
FROM Receipt r
JOIN CustomerOnline c ON r.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.Name
ORDER BY TotalPrice ASC;

--Display EmployeeID, LastName, FirstName, HireDate, Department, Salary of Employees whose employment date is after January 1, 2020 and display Salary in order from low to hi
SELECT EmployeeID, LastName, FirstName, HireDate, Department, Salary
FROM Employee
WHERE HireDate >= '2020-01-01'
ORDER BY Salary ASC;

--Displays the Name, Region, and Price of the Service with the Lowest Price
SELECT s.Name, s.Area, s.Price
FROM Services AS s
WHERE s.Price = (
    SELECT MIN(Price)
```

| | CustomerID | Name | TotalPrice |
|---|---|---|---|
| 1 | C004 | Vũ Thị Hồng | 40000.00 |
| 2 | C019 | Nguyễn Sỹ Anh | 100000.00 |
| 3 | C001 | Trần Gia Huy | 120000.00 |
| 4 | C013 | Ngô Thị Thanh | 120000.00 |
| 5 | C002 | Vũ Anh Tuấn | 160000.00 |
| 6 | C006 | Ngô Nguyễn Quỳnh Trâm | 180000.00 |
| 7 | C010 | Đinh Thị Như | 180000.00 |
| 8 | C017 | Nguyễn Thị Kim | 220000.00 |
| 9 | C007 | Hồ Nguyễn Anh Thy | 240000.00 |
| 10 | C008 | Nguyễn Ngọc Thiên Thanh | 300000.00 |
| 11 | C018 | Nguyễn Phạm Bảo Minh | 320000.00 |
| 12 | C015 | Tô Nguyễn Tường Minh | 360000.00 |
| 13 | C014 | Đào Gia Bảo | 600000.00 |
| 14 | C003 | Trần Phương Dung | 960000.00 |

Query executed successfully.    THANH_HUYEN (16.0 RTM)    THANH_HUYEN\ntthu (65)    THAOCAMVIEN    00:00:00    14 rows

```sql
--Display EmployeeID, LastName, FirstName, HireDate, Department, Salary of Employees
whose employment date is after January 1, 2020 and display Salary in order from low to
high
SELECT EmployeeID, LastName, FirstName, HireDate, Department, Salary
FROM Employee
WHERE HireDate >= '2020-01-01'
ORDER BY Salary ASC;
```

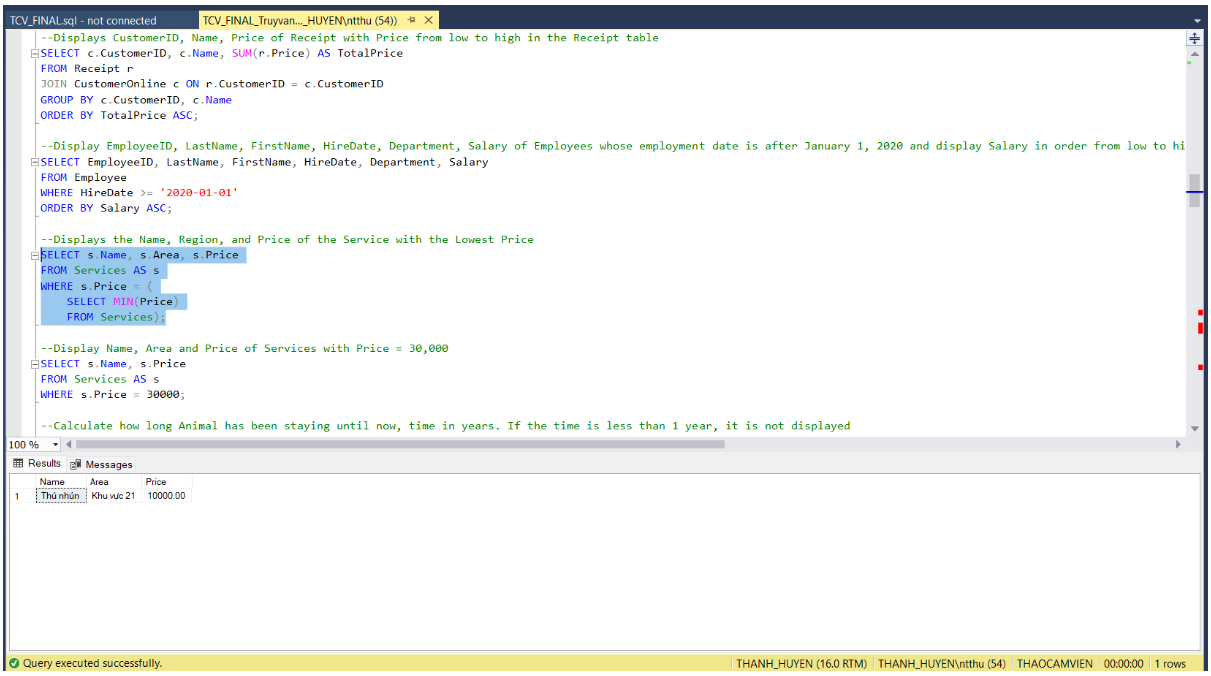

TCV_FINAL.sql - TH..._HUYEN\ntthu (66))
TCV_FINAL_Truyvan..._HUYEN\ntthu (65))
100 %
Results
Messages

```sql
WHERE OrderDate BETWEEN '2024-02-01' AND '2024-04-30';

--Displays CustomerID, Name, Price of Receipt with Price from low to high in the Receipt table
SELECT c.CustomerID, c.Name, SUM(r.Price) AS TotalPrice
FROM Receipt r
JOIN CustomerOnline c ON r.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.Name
ORDER BY TotalPrice ASC;

--Display EmployeeID, LastName, FirstName, HireDate, Department, Salary of Employees whose employment date is after January 1, 2020 and display Salary in order from low to hi
SELECT EmployeeID, LastName, FirstName, HireDate, Department, Salary
FROM Employee
WHERE HireDate >= '2020-01-01'
ORDER BY Salary ASC;

--Displays the Name, Region, and Price of the Service with the Lowest Price
SELECT s.Name, s.Area, s.Price
FROM Services AS s
WHERE s.Price = (
    SELECT MIN(Price)
```

|  | EmployeeID | LastName | FirstName | HireDate | Department | Salary |
|---|---|---|---|---|---|---|
| 1 | TCV0024 | Trần Như Nhật | Lệ | 2022-09-10 | Xí nghiệp kinh doanh | 3900000.00 |
| 2 | TCV0025 | Nguyễn Hoàng | Phi | 2023-03-18 | Xí nghiệp kinh doanh | 4500000.00 |
| 3 | TCV0022 | Bùi Huệ | Chi | 2021-06-30 | Xí nghiệp kinh doanh | 4900000.00 |
| 4 | TCV0017 | Nguyễn Văn | Tú | 2020-08-20 | Kế hoạch đầu tư | 5000000.00 |
| 5 | TCV0018 | Vũ Bảo | Châu | 2020-10-12 | Xí nghiệp kinh doanh | 5000000.00 |
| 6 | TCV0015 | Đinh Tấn | Lực | 2020-03-18 | Xí nghiệp thực vật | 5500000.00 |
| 7 | TCV0020 | Lý Thị Kim | Loan | 2021-04-18 | Xí nghiệp thực vật | 5500000.00 |
| 8 | TCV0023 | Phạm Trần Lam | Giang | 2022-01-05 | Xí nghiệp xây dựng | 6100000.00 |
| 9 | TCV0019 | Ninh Đứ | Anh | 2021-02-14 | Xí nghiệp thực vật | 7000000.00 |
| 10 | TCV0021 | Trần Bảo | Luân | 2021-05-20 | Xí nghiệp bảo vệ | 7350000.00 |
| 11 | TCV0016 | Phạm Thị Duyên | Hải | 2020-06-15 | Xí nghiệp kinh doanh | 8000000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (65) | THAOCAMVIEN | 00:00:00 | 11 rows

```sql
--Displays the Name, Region, and Price of the Service with the Lowest Price
SELECT s.Name, s.Area, s.Price
FROM Services AS s
WHERE s.Price = (
    SELECT MIN(Price)
    FROM Services);
```

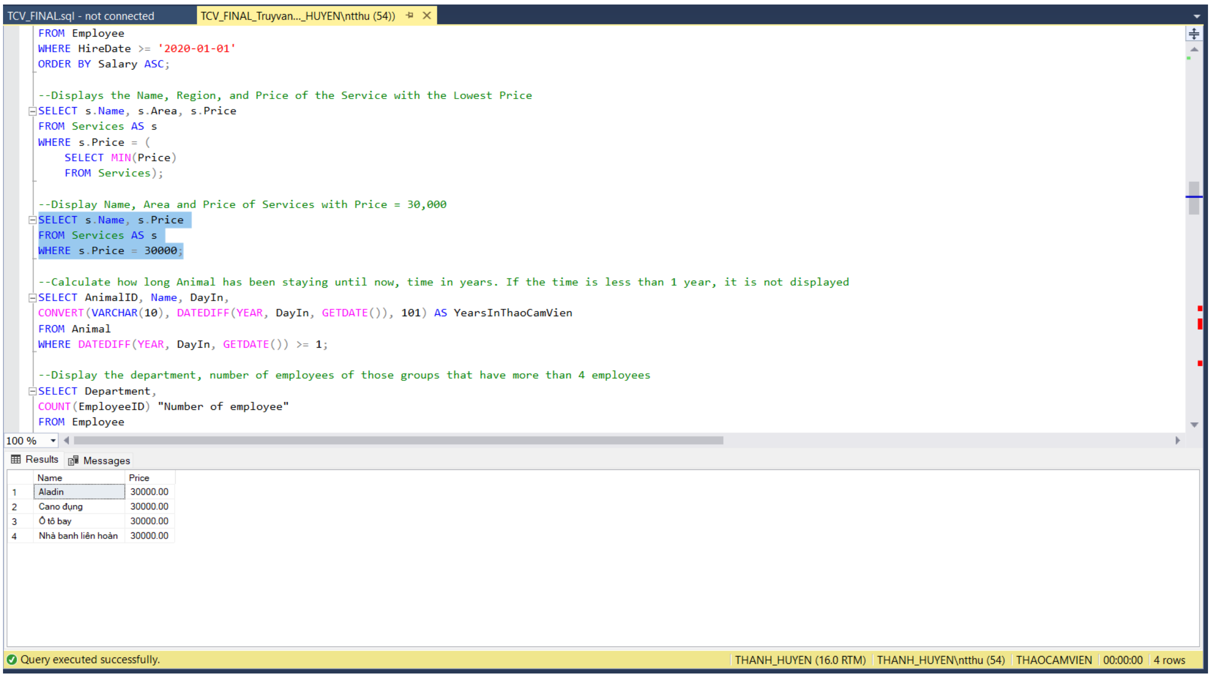

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```sql
--Displays CustomerID, Name, Price of Receipt with Price from low to high in the Receipt table
SELECT c.CustomerID, c.Name, SUM(r.Price) AS TotalPrice
FROM Receipt r
JOIN CustomerOnline c ON r.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.Name
ORDER BY TotalPrice ASC;

--Display EmployeeID, LastName, FirstName, HireDate, Department, Salary of Employees whose employment date is after January 1, 2020 and display Salary in order from low to hi
SELECT EmployeeID, LastName, FirstName, HireDate, Department, Salary
FROM Employee
WHERE HireDate >= '2020-01-01'
ORDER BY Salary ASC;

--Displays the Name, Region, and Price of the Service with the Lowest Price
SELECT s.Name, s.Area, s.Price
FROM Services AS s
WHERE s.Price = (
    SELECT MIN(Price)
    FROM Services);

--Display Name, Area and Price of Services with Price = 30,000
SELECT s.Name, s.Price
FROM Services AS s
WHERE s.Price = 30000;

--Calculate how long Animal has been staying until now, time in years. If the time is less than 1 year, it is not displayed
```

|  | Name | Area | Price |
|---|---|---|---|
| 1 | Thú nhún | Khu vực 21 | 10000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 1 rows

```sql
--Display Name, Area and Price of Services with Price = 30,000
SELECT s.Name, s.Price
FROM Services AS s
WHERE s.Price = 30000;
```

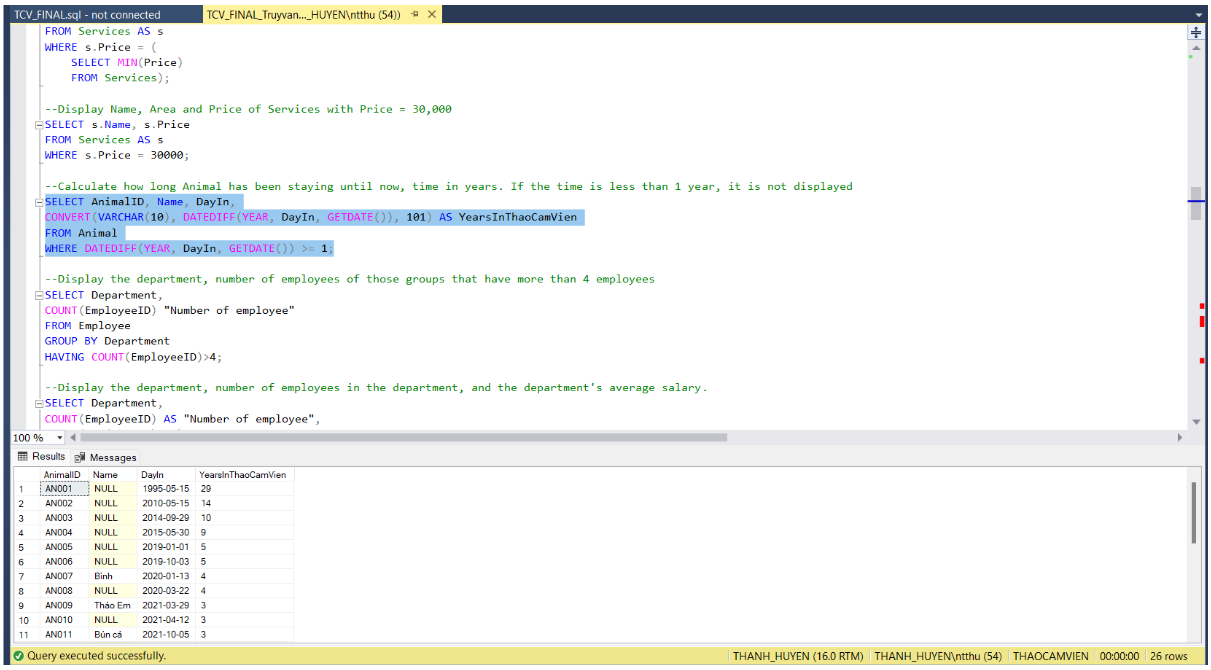

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```
FROM Employee
WHERE HireDate >= '2020-01-01'
ORDER BY Salary ASC;

--Displays the Name, Region, and Price of the Service with the Lowest Price
SELECT s.Name, s.Area, s.Price
FROM Services AS s
WHERE s.Price = (
    SELECT MIN(Price)
    FROM Services);

--Display Name, Area and Price of Services with Price = 30,000
SELECT s.Name, s.Price
FROM Services AS s
WHERE s.Price = 30000;

--Calculate how long Animal has been staying until now, time in years. If the time is less than 1 year, it is not displayed
SELECT AnimalID, Name, DayIn,
CONVERT(VARCHAR(10), DATEDIFF(YEAR, DayIn, GETDATE()), 101) AS YearsInThaoCamVien
FROM Animal
WHERE DATEDIFF(YEAR, DayIn, GETDATE()) >= 1;

--Display the department, number of employees of those groups that have more than 4 employees
SELECT Department,
COUNT(EmployeeID) "Number of employee"
FROM Employee
```

| | Name | Price |
|---|---|---|
| 1 | Aladin | 30000.00 |
| 2 | Cano đụng | 30000.00 |
| 3 | Ô tô bay | 30000.00 |
| 4 | Nhà banh liên hoàn | 30000.00 |

Query executed successfully. THANH_HUYEN (16.0 RTM) THANH_HUYEN\ntthu (54) THAOCAMVIEN 00:00:00 4 rows

```
--Calculate how long Animal has been staying until now, time in years. If the time is
less than 1 year, it is not displayed
SELECT AnimalID, Name, DayIn,
CONVERT(VARCHAR(10), DATEDIFF(YEAR, DayIn, GETDATE()), 101) AS YearsInThaoCamVien
FROM Animal
WHERE DATEDIFF(YEAR, DayIn, GETDATE()) >= 1;
```

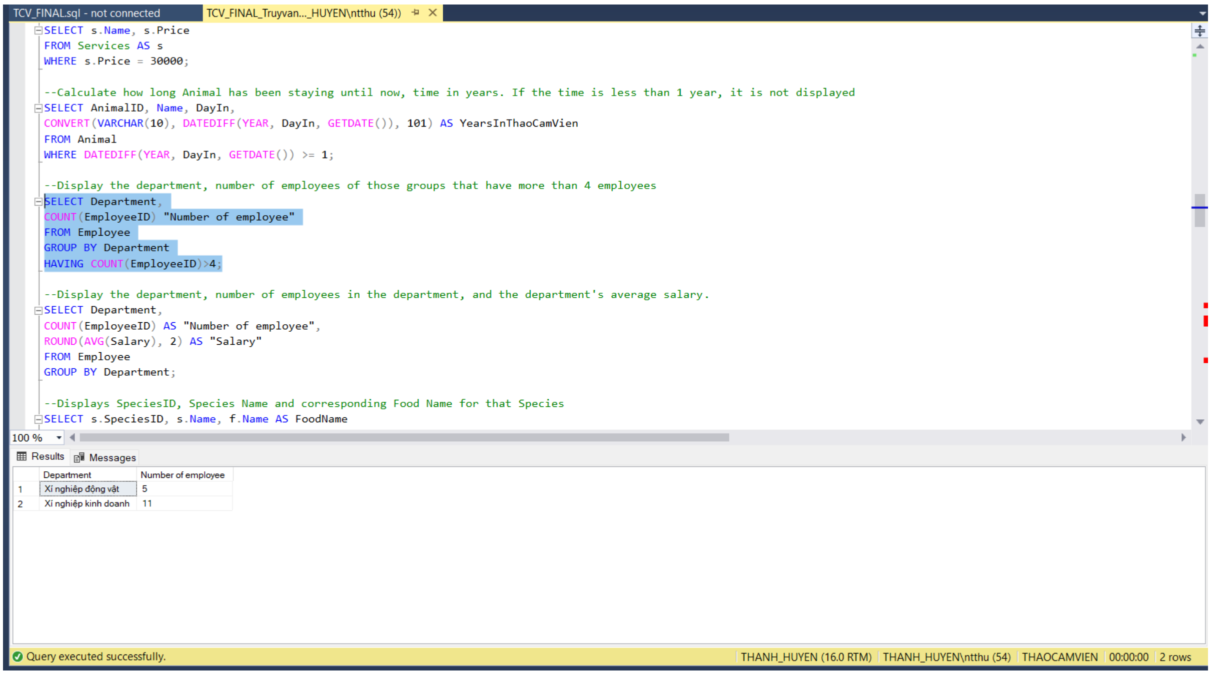

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```
FROM Services AS s
WHERE s.Price = (
    SELECT MIN(Price)
    FROM Services);

--Display Name, Area and Price of Services with Price = 30,000
SELECT s.Name, s.Price
FROM Services AS s
WHERE s.Price = 30000;

--Calculate how long Animal has been staying until now, time in years. If the time is less than 1 year, it is not displayed
SELECT AnimalID, Name, DayIn,
CONVERT(VARCHAR(10), DATEDIFF(YEAR, DayIn, GETDATE()), 101) AS YearsInThaoCamVien
FROM Animal
WHERE DATEDIFF(YEAR, DayIn, GETDATE()) >= 1;

--Display the department, number of employees of those groups that have more than 4 employees
SELECT Department,
COUNT(EmployeeID) "Number of employee"
FROM Employee
GROUP BY Department
HAVING COUNT(EmployeeID)>4;

--Display the department, number of employees in the department, and the department's average salary.
SELECT Department,
COUNT(EmployeeID) AS "Number of employee",
```

| | AnimalID | Name | DayIn | YearsInThaoCamVien |
|---|---|---|---|---|
| 1 | AN001 | NULL | 1995-05-15 | 29 |
| 2 | AN002 | NULL | 2010-05-15 | 14 |
| 3 | AN003 | NULL | 2014-09-29 | 10 |
| 4 | AN004 | NULL | 2015-05-30 | 9 |
| 5 | AN005 | NULL | 2019-01-01 | 5 |
| 6 | AN006 | NULL | 2019-10-03 | 5 |
| 7 | AN007 | Bình | 2020-01-13 | 4 |
| 8 | AN008 | NULL | 2020-03-22 | 4 |
| 9 | AN009 | Thảo Em | 2021-03-29 | 3 |
| 10 | AN010 | NULL | 2021-04-12 | 3 |
| 11 | AN011 | Bún cá | 2021-10-05 | 3 |

Query executed successfully. THANH_HUYEN (16.0 RTM) THANH_HUYEN\ntthu (54) THAOCAMVIEN 00:00:00 26 rows

```
--Display the department, number of employees of those groups that have more than 4
employees
SELECT Department,
COUNT(EmployeeID) "Number of employee"
FROM Employee
GROUP BY Department
HAVING COUNT(EmployeeID)>4;
```

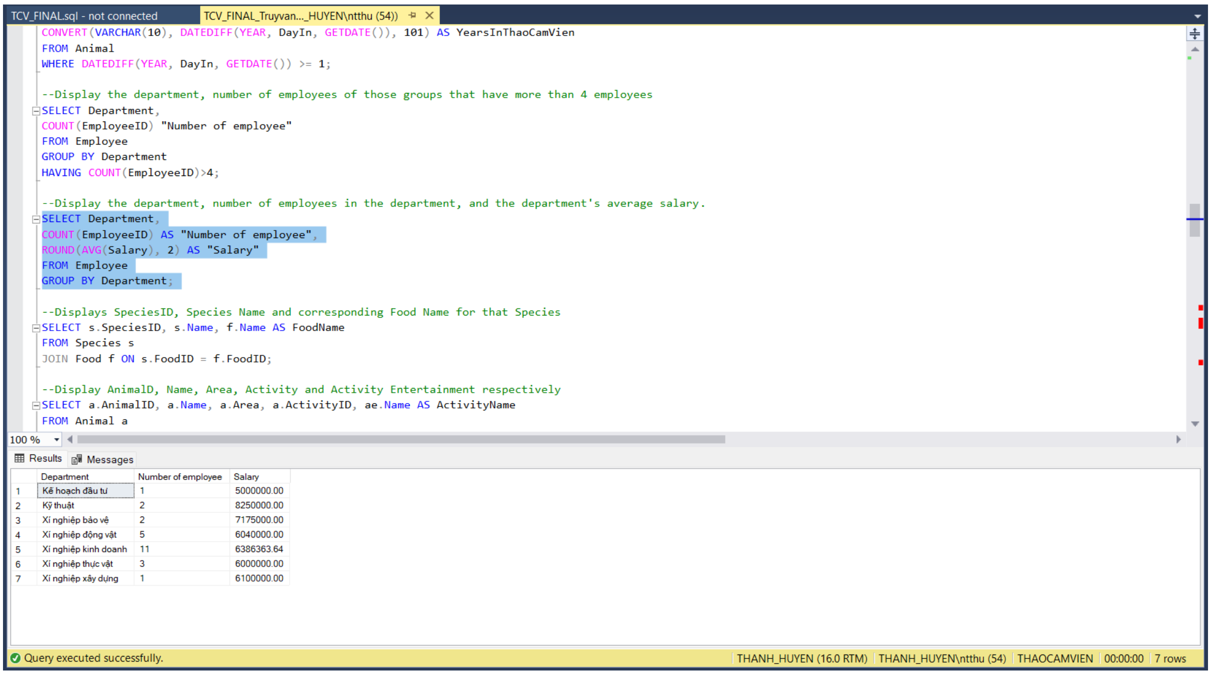

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```
SELECT s.Name, s.Price
FROM Services AS s
WHERE s.Price = 30000;

--Calculate how long Animal has been staying until now, time in years. If the time is less than 1 year, it is not displayed
SELECT AnimalID, Name, DayIn,
CONVERT(VARCHAR(10), DATEDIFF(YEAR, DayIn, GETDATE()), 101) AS YearsInThaoCamVien
FROM Animal
WHERE DATEDIFF(YEAR, DayIn, GETDATE()) >= 1;

--Display the department, number of employees of those groups that have more than 4 employees
SELECT Department,
COUNT(EmployeeID) "Number of employee"
FROM Employee
GROUP BY Department
HAVING COUNT(EmployeeID)>4;

--Display the department, number of employees in the department, and the department's average salary.
SELECT Department,
COUNT(EmployeeID) AS "Number of employee",
ROUND(AVG(Salary), 2) AS "Salary"
FROM Employee
GROUP BY Department;

--Displays SpeciesID, Species Name and corresponding Food Name for that Species
SELECT s.SpeciesID, s.Name, f.Name AS FoodName
```

| | Department | Number of employee |
|---|---|---|
| 1 | Xí nghiệp động vật | 5 |
| 2 | Xí nghiệp kinh doanh | 11 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 2 rows

```
--Display the department, number of employees in the department, and the department's
average salary.
SELECT Department,
COUNT(EmployeeID) AS "Number of employee",
ROUND(AVG(Salary), 2) AS "Salary"
FROM Employee
GROUP BY Department;
```

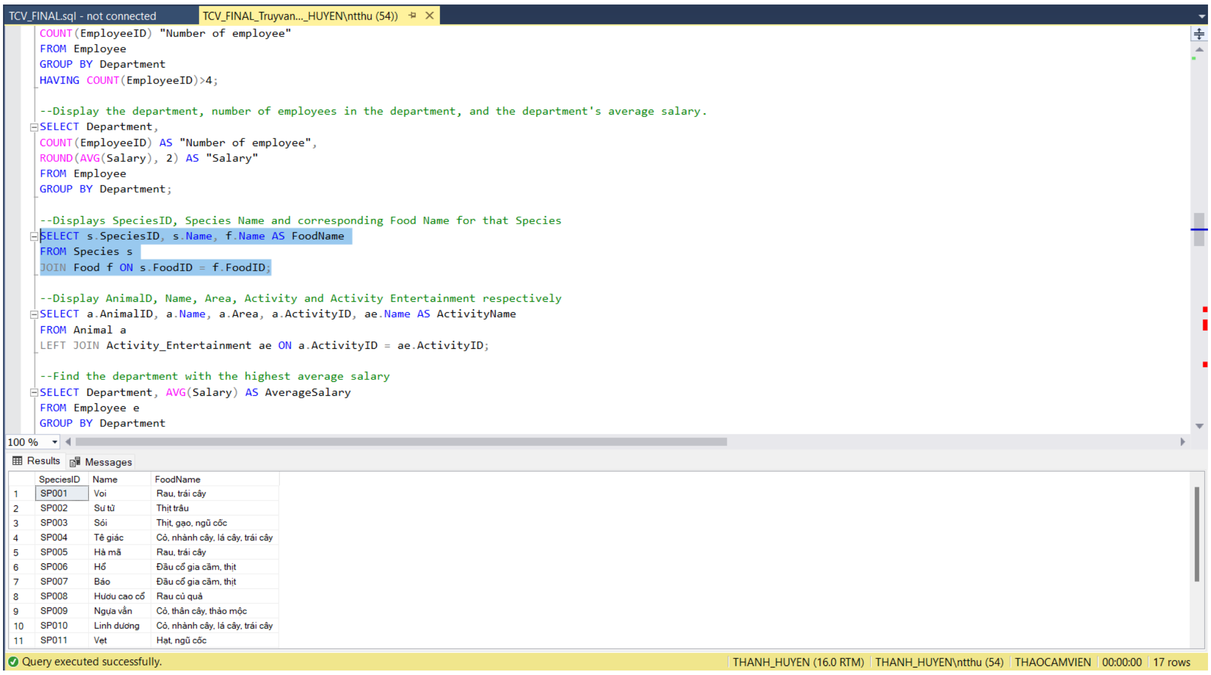

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```
CONVERT(VARCHAR(10), DATEDIFF(YEAR, DayIn, GETDATE()), 101) AS YearsInThaoCamVien
FROM Animal
WHERE DATEDIFF(YEAR, DayIn, GETDATE()) >= 1;

--Display the department, number of employees of those groups that have more than 4 employees
SELECT Department,
COUNT(EmployeeID) "Number of employee"
FROM Employee
GROUP BY Department
HAVING COUNT(EmployeeID)>4;

--Display the department, number of employees in the department, and the department's average salary.
SELECT Department,
COUNT(EmployeeID) AS "Number of employee",
ROUND(AVG(Salary), 2) AS "Salary"
FROM Employee
GROUP BY Department;

--Displays SpeciesID, Species Name and corresponding Food Name for that Species
SELECT s.SpeciesID, s.Name, f.Name AS FoodName
FROM Species s
JOIN Food f ON s.FoodID = f.FoodID;

--Display AnimalD, Name, Area, Activity and Activity Entertainment respectively
SELECT a.AnimalID, a.Name, a.Area, a.ActivityID, ae.Name AS ActivityName
FROM Animal a
```

| | Department | Number of employee | Salary |
|---|---|---|---|
| 1 | Kế hoạch đầu tư | 1 | 5000000.00 |
| 2 | Kỹ thuật | 2 | 8250000.00 |
| 3 | Xí nghiệp bảo vệ | 2 | 7175000.00 |
| 4 | Xí nghiệp động vật | 5 | 6040000.00 |
| 5 | Xí nghiệp kinh doanh | 11 | 6386363.64 |
| 6 | Xí nghiệp thực vật | 3 | 6000000.00 |
| 7 | Xí nghiệp xây dựng | 1 | 6100000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 7 rows

```
--Displays SpeciesID, Species Name and corresponding Food Name for that Species
SELECT s.SpeciesID, s.Name, f.Name AS FoodName
FROM Species s
JOIN Food f ON s.FoodID = f.FoodID;
```

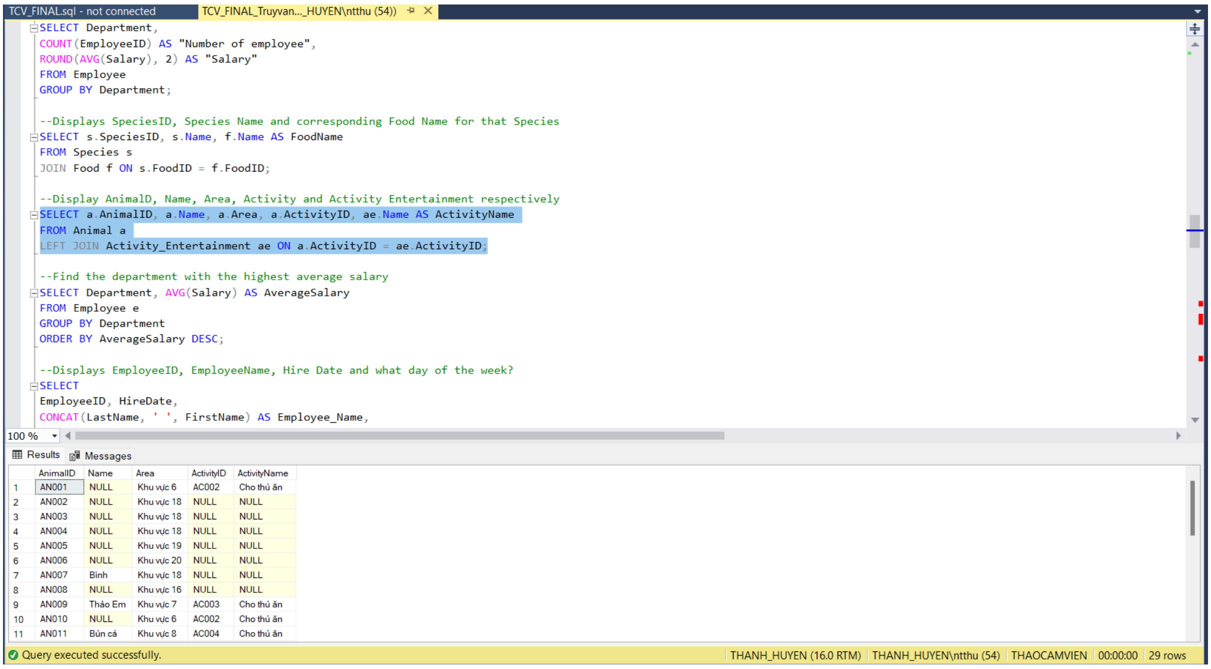

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```sql
COUNT(EmployeeID) "Number of employee"
FROM Employee
GROUP BY Department
HAVING COUNT(EmployeeID)>4;

--Display the department, number of employees in the department, and the department's average salary.
SELECT Department,
COUNT(EmployeeID) AS "Number of employee",
ROUND(AVG(Salary), 2) AS "Salary"
FROM Employee
GROUP BY Department;

--Displays SpeciesID, Species Name and corresponding Food Name for that Species
SELECT s.SpeciesID, s.Name, f.Name AS FoodName
FROM Species s
JOIN Food f ON s.FoodID = f.FoodID;

--Display AnimalD, Name, Area, Activity and Activity Entertainment respectively
SELECT a.AnimalID, a.Name, a.Area, a.ActivityID, ae.Name AS ActivityName
FROM Animal a
LEFT JOIN Activity_Entertainment ae ON a.ActivityID = ae.ActivityID;

--Find the department with the highest average salary
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employee e
GROUP BY Department
```

| | SpeciesID | Name | FoodName |
|---|---|---|---|
| 1 | SP001 | Voi | Rau, trái cây |
| 2 | SP002 | Sư tử | Thịt trâu |
| 3 | SP003 | Sói | Thịt, gạo, ngũ cốc |
| 4 | SP004 | Tê giác | Cỏ, nhành cây, lá cây, trái cây |
| 5 | SP005 | Hà mã | Rau, trái cây |
| 6 | SP006 | Hổ | Đầu cổ gia cầm, thịt |
| 7 | SP007 | Báo | Đầu cổ gia cầm, thịt |
| 8 | SP008 | Hươu cao cổ | Rau củ quả |
| 9 | SP009 | Ngựa vằn | Cỏ, thân cây, thảo mộc |
| 10 | SP010 | Linh dương | Cỏ, nhành cây, lá cây, trái cây |
| 11 | SP011 | Vẹt | Hạt, ngũ cốc |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 17 rows

```sql
--Display AnimalD, Name, Area, Activity and Activity Entertainment respectively
SELECT a.AnimalID, a.Name, a.Area, a.ActivityID, ae.Name AS ActivityName
FROM Animal a
LEFT JOIN Activity_Entertainment ae ON a.ActivityID = ae.ActivityID;
```

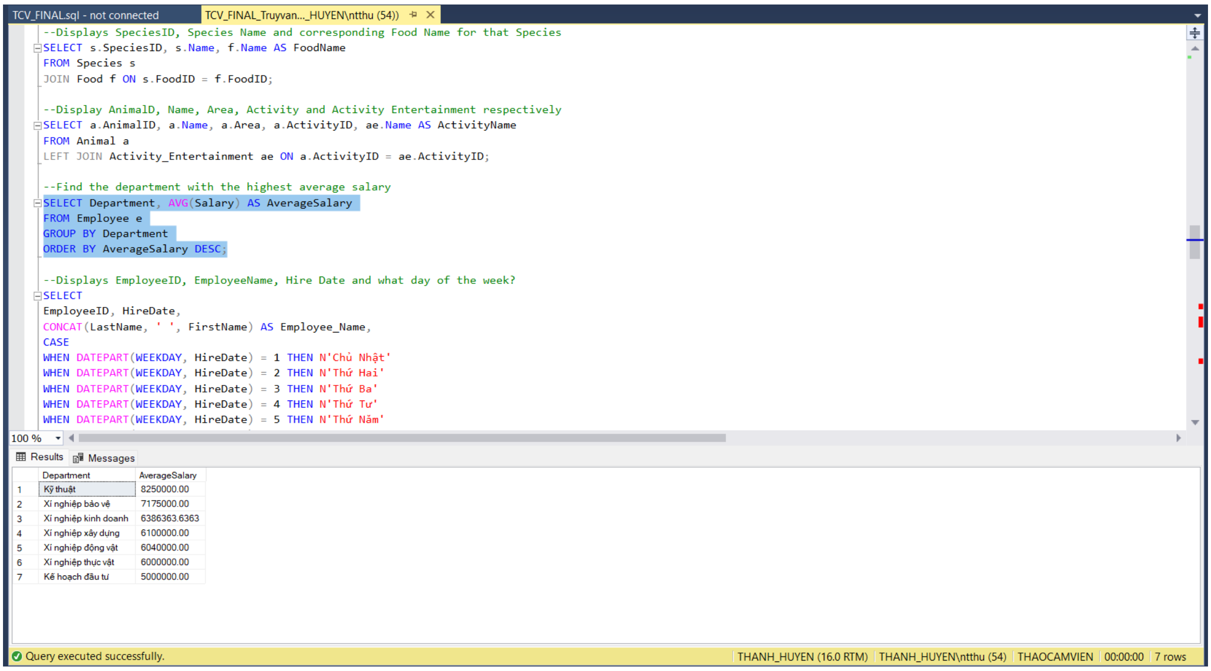

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```sql
SELECT Department,
COUNT(EmployeeID) AS "Number of employee",
ROUND(AVG(Salary), 2) AS "Salary"
FROM Employee
GROUP BY Department;

--Displays SpeciesID, Species Name and corresponding Food Name for that Species
SELECT s.SpeciesID, s.Name, f.Name AS FoodName
FROM Species s
JOIN Food f ON s.FoodID = f.FoodID;

--Display AnimalD, Name, Area, Activity and Activity Entertainment respectively
SELECT a.AnimalID, a.Name, a.Area, a.ActivityID, ae.Name AS ActivityName
FROM Animal a
LEFT JOIN Activity_Entertainment ae ON a.ActivityID = ae.ActivityID;

--Find the department with the highest average salary
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employee e
GROUP BY Department
ORDER BY AverageSalary DESC;

--Displays EmployeeID, EmployeeName, Hire Date and what day of the week?
SELECT
EmployeeID, HireDate,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
```

| | AnimalID | Name | Area | ActivityID | ActivityName |
|---|---|---|---|---|---|
| 1 | AN001 | NULL | Khu vực 6 | AC002 | Cho thú ăn |
| 2 | AN002 | NULL | Khu vực 18 | NULL | NULL |
| 3 | AN003 | NULL | Khu vực 18 | NULL | NULL |
| 4 | AN004 | NULL | Khu vực 18 | NULL | NULL |
| 5 | AN005 | NULL | Khu vực 19 | NULL | NULL |
| 6 | AN006 | NULL | Khu vực 20 | NULL | NULL |
| 7 | AN007 | Bình | Khu vực 18 | NULL | NULL |
| 8 | AN008 | NULL | Khu vực 16 | NULL | NULL |
| 9 | AN009 | Thảo Em | Khu vực 7 | AC003 | Cho thú ăn |
| 10 | AN010 | NULL | Khu vực 6 | AC002 | Cho thú ăn |
| 11 | AN011 | Bún cá | Khu vực 8 | AC004 | Cho thú ăn |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 29 rows

```sql
--Find the department with the highest average salary
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employee e
GROUP BY Department
ORDER BY AverageSalary DESC;
```



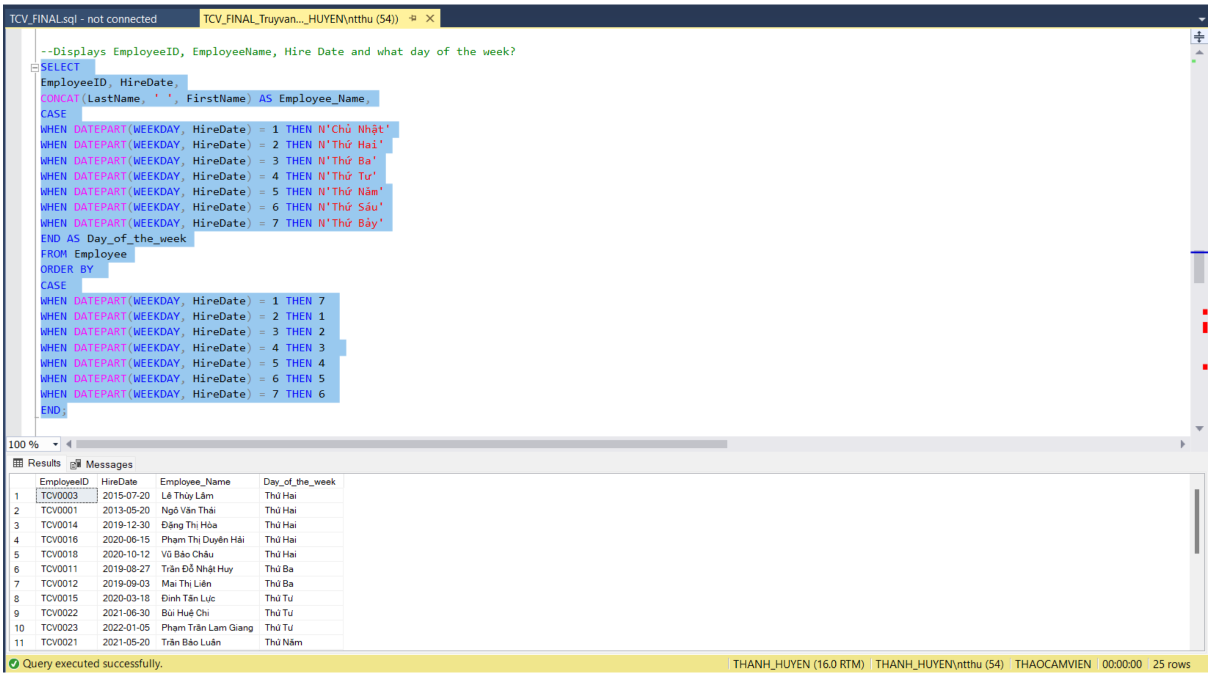
TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```sql
--Displays SpeciesID, Species Name and corresponding Food Name for that Species
SELECT s.SpeciesID, s.Name, f.Name AS FoodName
FROM Species s
JOIN Food f ON s.FoodID = f.FoodID;

--Display AnimalD, Name, Area, Activity and Activity Entertainment respectively
SELECT a.AnimalID, a.Name, a.Area, a.ActivityID, ae.Name AS ActivityName
FROM Animal a
LEFT JOIN Activity_Entertainment ae ON a.ActivityID = ae.ActivityID;

--Find the department with the highest average salary
SELECT Department, AVG(Salary) AS AverageSalary
FROM Employee e
GROUP BY Department
ORDER BY AverageSalary DESC;

--Displays EmployeeID, EmployeeName, Hire Date and what day of the week?
SELECT
EmployeeID, HireDate,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
CASE
WHEN DATEPART(WEEKDAY, HireDate) = 1 THEN N'Chủ Nhật'
WHEN DATEPART(WEEKDAY, HireDate) = 2 THEN N'Thứ Hai'
WHEN DATEPART(WEEKDAY, HireDate) = 3 THEN N'Thứ Ba'
WHEN DATEPART(WEEKDAY, HireDate) = 4 THEN N'Thứ Tư'
WHEN DATEPART(WEEKDAY, HireDate) = 5 THEN N'Thứ Năm'
```

| | Department | AverageSalary |
|---|---|---|
| 1 | Kỹ thuật | 8250000.00 |
| 2 | Xí nghiệp bảo vệ | 7175000.00 |
| 3 | Xí nghiệp kinh doanh | 6386363.6363 |
| 4 | Xí nghiệp xây dựng | 6100000.00 |
| 5 | Xí nghiệp động vật | 6040000.00 |
| 6 | Xí nghiệp thực vật | 6000000.00 |
| 7 | Kế hoạch đầu tư | 5000000.00 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 7 rows

```sql
--Displays EmployeeID, EmployeeName, Hire Date and what day of the week?
SELECT
EmployeeID, HireDate,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
CASE
WHEN DATEPART(WEEKDAY, HireDate) = 1 THEN N'Chủ Nhật'
WHEN DATEPART(WEEKDAY, HireDate) = 2 THEN N'Thứ Hai'
WHEN DATEPART(WEEKDAY, HireDate) = 3 THEN N'Thứ Ba'
WHEN DATEPART(WEEKDAY, HireDate) = 4 THEN N'Thứ Tư'
WHEN DATEPART(WEEKDAY, HireDate) = 5 THEN N'Thứ Năm'
WHEN DATEPART(WEEKDAY, HireDate) = 6 THEN N'Thứ Sáu'
WHEN DATEPART(WEEKDAY, HireDate) = 7 THEN N'Thứ Bảy'
END AS Day_of_the_week
FROM Employee
ORDER BY
CASE
WHEN DATEPART(WEEKDAY, HireDate) = 1 THEN 7
WHEN DATEPART(WEEKDAY, HireDate) = 2 THEN 1
WHEN DATEPART(WEEKDAY, HireDate) = 3 THEN 2
WHEN DATEPART(WEEKDAY, HireDate) = 4 THEN 3
WHEN DATEPART(WEEKDAY, HireDate) = 5 THEN 4
WHEN DATEPART(WEEKDAY, HireDate) = 6 THEN 5
WHEN DATEPART(WEEKDAY, HireDate) = 7 THEN 6
END;
```

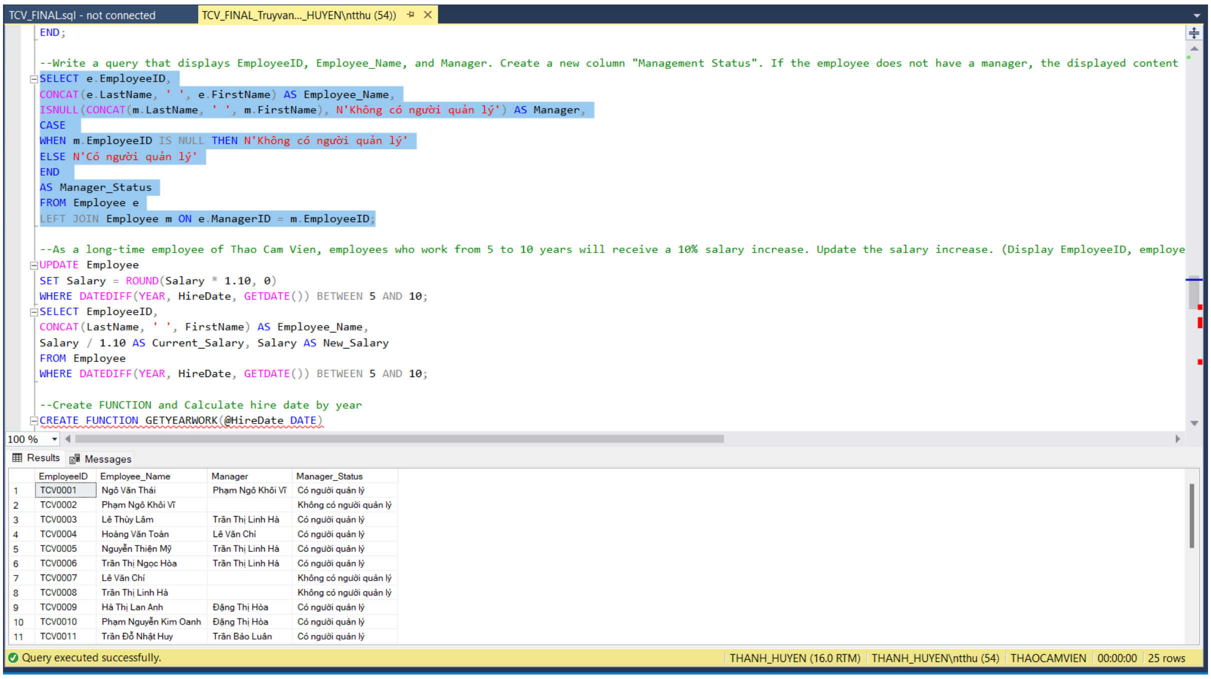

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```sql
--Displays EmployeeID, EmployeeName, Hire Date and what day of the week?
SELECT
EmployeeID, HireDate,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
CASE
WHEN DATEPART(WEEKDAY, HireDate) = 1 THEN N'Chủ Nhật'
WHEN DATEPART(WEEKDAY, HireDate) = 2 THEN N'Thứ Hai'
WHEN DATEPART(WEEKDAY, HireDate) = 3 THEN N'Thứ Ba'
WHEN DATEPART(WEEKDAY, HireDate) = 4 THEN N'Thứ Tư'
WHEN DATEPART(WEEKDAY, HireDate) = 5 THEN N'Thứ Năm'
WHEN DATEPART(WEEKDAY, HireDate) = 6 THEN N'Thứ Sáu'
WHEN DATEPART(WEEKDAY, HireDate) = 7 THEN N'Thứ Bảy'
END AS Day_of_the_week
FROM Employee
ORDER BY
CASE
WHEN DATEPART(WEEKDAY, HireDate) = 1 THEN 7
WHEN DATEPART(WEEKDAY, HireDate) = 2 THEN 1
WHEN DATEPART(WEEKDAY, HireDate) = 3 THEN 2
WHEN DATEPART(WEEKDAY, HireDate) = 4 THEN 3
WHEN DATEPART(WEEKDAY, HireDate) = 5 THEN 4
WHEN DATEPART(WEEKDAY, HireDate) = 6 THEN 5
WHEN DATEPART(WEEKDAY, HireDate) = 7 THEN 6
END;
```

| | EmployeeID | HireDate | Employee_Name | Day_of_the_week |
|---|---|---|---|---|
| 1 | TCV0003 | 2015-07-20 | Lê Thủy Lâm | Thứ Hai |
| 2 | TCV0001 | 2013-05-20 | Ngô Văn Thái | Thứ Hai |
| 3 | TCV0014 | 2019-12-30 | Đặng Thị Hòa | Thứ Hai |
| 4 | TCV0016 | 2020-06-15 | Phạm Thị Duyên Hải | Thứ Hai |
| 5 | TCV0018 | 2020-10-12 | Vũ Bảo Châu | Thứ Hai |
| 6 | TCV0011 | 2019-08-27 | Trần Đỗ Nhật Huy | Thứ Ba |
| 7 | TCV0012 | 2019-09-03 | Mai Thị Liên | Thứ Ba |
| 8 | TCV0015 | 2020-03-18 | Đinh Tấn Lực | Thứ Tư |
| 9 | TCV0022 | 2021-06-30 | Bùi Huệ Chi | Thứ Tư |
| 10 | TCV0023 | 2022-01-05 | Phạm Trần Lam Giang | Thứ Tư |
| 11 | TCV0021 | 2021-05-20 | Trần Bảo Luân | Thứ Năm |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 25 rows

```sql
--Write a query that displays EmployeeID, Employee_Name, and Manager. Create a new column "Management Status". If the employee does not have a manager, the displayed content is "Không có người quản lý", if the employee has a manager, it displays "Có người quản lý".
SELECT e.EmployeeID,
CONCAT(e.LastName, ' ', e.FirstName) AS Employee_Name,
ISNULL(CONCAT(m.LastName, ' ', m.FirstName), N'Không có người quản lý') AS Manager,
CASE
WHEN m.EmployeeID IS NULL THEN N'Không có người quản lý'
ELSE N'Có người quản lý'
END
AS Manager_Status
FROM Employee e
LEFT JOIN Employee m ON e.ManagerID = m.EmployeeID;
```

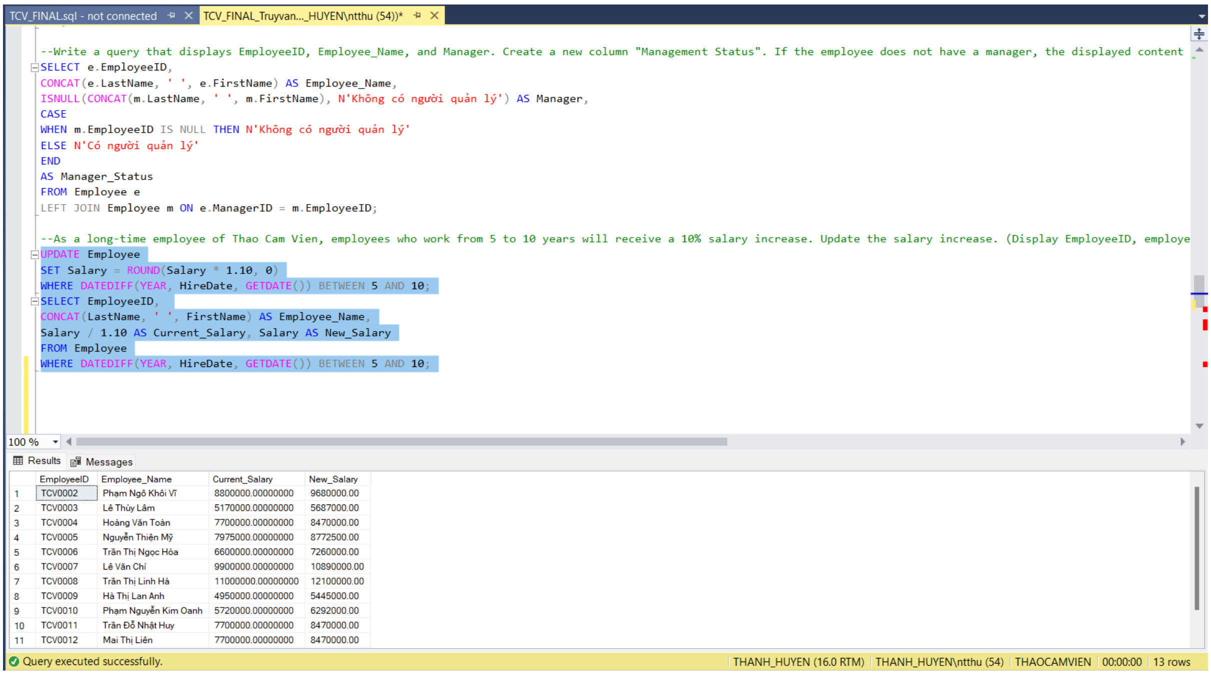

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))
100 %
Results
Messages

```sql
END;

--Write a query that displays EmployeeID, Employee_Name, and Manager. Create a new column "Management Status". If the employee does not have a manager, the displayed content
SELECT e.EmployeeID,
CONCAT(e.LastName, ' ', e.FirstName) AS Employee_Name,
ISNULL(CONCAT(m.LastName, ' ', m.FirstName), N'Không có người quản lý') AS Manager,
CASE
WHEN m.EmployeeID IS NULL THEN N'Không có người quản lý'
ELSE N'Có người quản lý'
END
AS Manager_Status
FROM Employee e
LEFT JOIN Employee m ON e.ManagerID = m.EmployeeID;

--As a long-time employee of Thao Cam Vien, employees who work from 5 to 10 years will receive a 10% salary increase. Update the salary increase. (Display EmployeeID, employe
UPDATE Employee
SET Salary = ROUND(Salary * 1.10, 0)
WHERE DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 5 AND 10;
SELECT EmployeeID,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
Salary / 1.10 AS Current_Salary, Salary AS New_Salary
FROM Employee
WHERE DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 5 AND 10;

--Create FUNCTION and Calculate hire date by year
CREATE FUNCTION GETYEARWORK(@HireDate DATE)
```

| | EmployeeID | Employee_Name | Manager | Manager_Status |
|---|---|---|---|---|
| 1 | TCV0001 | Ngô Văn Thái | Phạm Ngô Khôi Vĩ | Có người quản lý |
| 2 | TCV0002 | Phạm Ngô Khôi Vĩ | | Không có người quản lý |
| 3 | TCV0003 | Lê Thủy Lâm | Trần Thị Linh Hà | Có người quản lý |
| 4 | TCV0004 | Hoàng Văn Toản | Lê Văn Chí | Có người quản lý |
| 5 | TCV0005 | Nguyễn Thiện Mỹ | Trần Thị Linh Hà | Có người quản lý |
| 6 | TCV0006 | Trần Thị Ngọc Hòa | Trần Thị Linh Hà | Có người quản lý |
| 7 | TCV0007 | Lê Văn Chí | | Không có người quản lý |
| 8 | TCV0008 | Trần Thị Linh Hà | | Không có người quản lý |
| 9 | TCV0009 | Hà Thị Lan Anh | Đặng Thị Hòa | Có người quản lý |
| 10 | TCV0010 | Phạm Nguyễn Kim Oanh | Đặng Thị Hòa | Có người quản lý |
| 11 | TCV0011 | Trần Đỗ Nhật Huy | Trần Bảo Luân | Có người quản lý |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 25 rows

```
--As a long-time employee of Thao Cam Vien, employees who work from 5 to 10 years will
receive a 10% salary increase. Update the salary increase. (Display EmployeeID,
employee name, current salary, new salary).
UPDATE Employee
SET Salary = ROUND(Salary * 1.10, 0)
WHERE DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 5 AND 10;
SELECT EmployeeID,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
Salary / 1.10 AS Current_Salary, Salary AS New_Salary
FROM Employee
WHERE DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 5 AND 10;
```

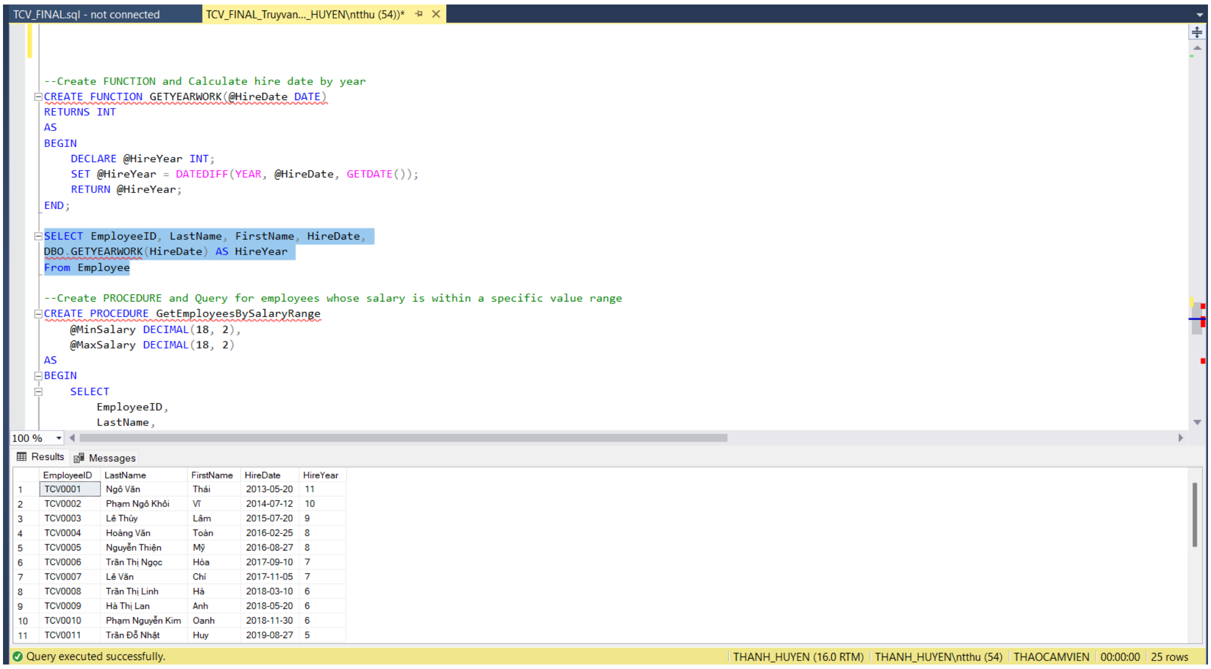

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))*
100 %
Results
Messages

```
--Write a query that displays EmployeeID, Employee_Name, and Manager. Create a new column "Management Status". If the employee does not have a manager, the displayed content
SELECT e.EmployeeID,
CONCAT(e.LastName, ' ', e.FirstName) AS Employee_Name,
ISNULL(CONCAT(m.LastName, ' ', m.FirstName), N'Không có người quản lý') AS Manager,
CASE
WHEN m.EmployeeID IS NULL THEN N'Không có người quản lý'
ELSE N'Có người quản lý'
END
AS Manager_Status
FROM Employee e
LEFT JOIN Employee m ON e.ManagerID = m.EmployeeID;

--As a long-time employee of Thao Cam Vien, employees who work from 5 to 10 years will receive a 10% salary increase. Update the salary increase. (Display EmployeeID, employe
UPDATE Employee
SET Salary = ROUND(Salary * 1.10, 0)
WHERE DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 5 AND 10;
SELECT EmployeeID,
CONCAT(LastName, ' ', FirstName) AS Employee_Name,
Salary / 1.10 AS Current_Salary, Salary AS New_Salary
FROM Employee
WHERE DATEDIFF(YEAR, HireDate, GETDATE()) BETWEEN 5 AND 10;
```

|  | EmployeeID | Employee_Name | Current_Salary | New_Salary |
|---|---|---|---|---|
| 1 | TCV0002 | Phạm Ngô Khôi Vĩ | 8800000.00000000 | 9680000.00 |
| 2 | TCV0003 | Lê Thủy Lâm | 5170000.00000000 | 5687000.00 |
| 3 | TCV0004 | Hoàng Văn Toàn | 7700000.00000000 | 8470000.00 |
| 4 | TCV0005 | Nguyễn Thiện Mỹ | 7975000.00000000 | 8772500.00 |
| 5 | TCV0006 | Trần Thị Ngọc Hòa | 6600000.00000000 | 7260000.00 |
| 6 | TCV0007 | Lê Văn Chí | 9900000.00000000 | 10890000.00 |
| 7 | TCV0008 | Trần Thị Linh Hà | 11000000.00000000 | 12100000.00 |
| 8 | TCV0009 | Hà Thị Lan Anh | 4950000.00000000 | 5445000.00 |
| 9 | TCV0010 | Phạm Nguyễn Kim Oanh | 5720000.00000000 | 6292000.00 |
| 10 | TCV0011 | Trần Đỗ Nhật Huy | 7700000.00000000 | 8470000.00 |
| 11 | TCV0012 | Mai Thị Liên | 7700000.00000000 | 8470000.00 |

```
Query executed successfully.    THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 13 rows
```

```
--Create FUNCTION and Calculate hire date by year
CREATE FUNCTION GETYEARWORK(@HireDate DATE)
RETURNS INT
AS
BEGIN
        DECLARE @HireYear INT;
        SET @HireYear = DATEDIFF(YEAR, @HireDate, GETDATE());
        RETURN @HireYear;
END;

SELECT EmployeeID, LastName, FirstName, HireDate,
DBO.GETYEARWORK(HireDate) AS HireYear
From Employee
```



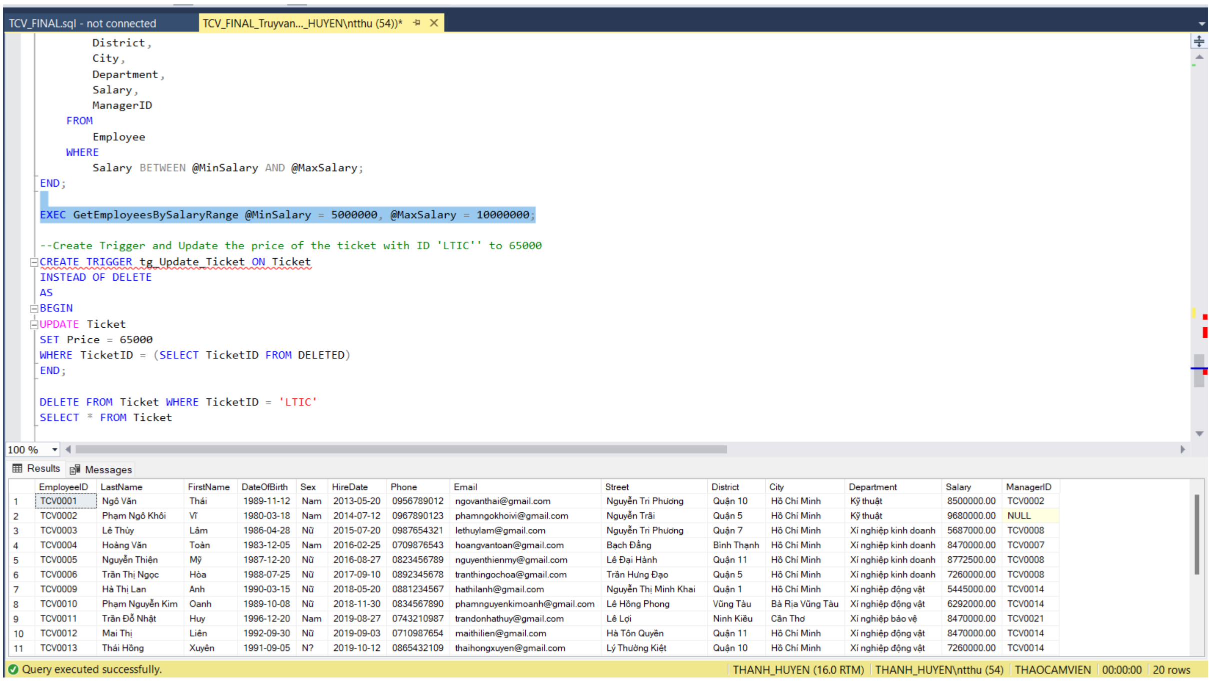
TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))*
100 %
Results
Messages

```sql
--Create FUNCTION and Calculate hire date by year
CREATE FUNCTION GETYEARWORK(@HireDate DATE)
RETURNS INT
AS
BEGIN
    DECLARE @HireYear INT;
    SET @HireYear = DATEDIFF(YEAR, @HireDate, GETDATE());
    RETURN @HireYear;
END;

SELECT EmployeeID, LastName, FirstName, HireDate,
DBO.GETYEARWORK(HireDate) AS HireYear
From Employee

--Create PROCEDURE and Query for employees whose salary is within a specific value range
CREATE PROCEDURE GetEmployeesBySalaryRange
    @MinSalary DECIMAL(18, 2),
    @MaxSalary DECIMAL(18, 2)
AS
BEGIN
    SELECT
        EmployeeID,
        LastName,
```

|  | EmployeeID | LastName | FirstName | HireDate | HireYear |
|---|---|---|---|---|---|
| 1 | TCV0001 | Ngô Văn | Thái | 2013-05-20 | 11 |
| 2 | TCV0002 | Phạm Ngô Khôi | Vĩ | 2014-07-12 | 10 |
| 3 | TCV0003 | Lê Thùy | Lâm | 2015-07-20 | 9 |
| 4 | TCV0004 | Hoàng Văn | Toàn | 2016-02-25 | 8 |
| 5 | TCV0005 | Nguyễn Thiện | Mỹ | 2016-08-27 | 8 |
| 6 | TCV0006 | Trần Thị Ngọc | Hòa | 2017-09-10 | 7 |
| 7 | TCV0007 | Lê Văn | Chí | 2017-11-05 | 7 |
| 8 | TCV0008 | Trần Thị Linh | Hà | 2018-03-10 | 6 |
| 9 | TCV0009 | Hà Thị Lan | Anh | 2018-05-20 | 6 |
| 10 | TCV0010 | Phạm Nguyễn Kim | Oanh | 2018-11-30 | 6 |
| 11 | TCV0011 | Trần Đỗ Nhật | Huy | 2019-08-27 | 5 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 25 rows

```sql
--Create PROCEDURE and Query for employees whose salary is within a specific value
range
CREATE PROCEDURE GetEmployeesBySalaryRange
    @MinSalary DECIMAL(18, 2),
    @MaxSalary DECIMAL(18, 2)
AS
BEGIN
    SELECT
        EmployeeID,
        LastName,
        FirstName,
        DateOfBirth,
        Sex,
        HireDate,
        Phone,
        Email,
        Street,
        District,
        City,
        Department,
        Salary,
        ManagerID
    FROM
        Employee
    WHERE
        Salary BETWEEN @MinSalary AND @MaxSalary;
END;

EXEC GetEmployeesBySalaryRange @MinSalary = 5000000, @MaxSalary = 10000000;
```

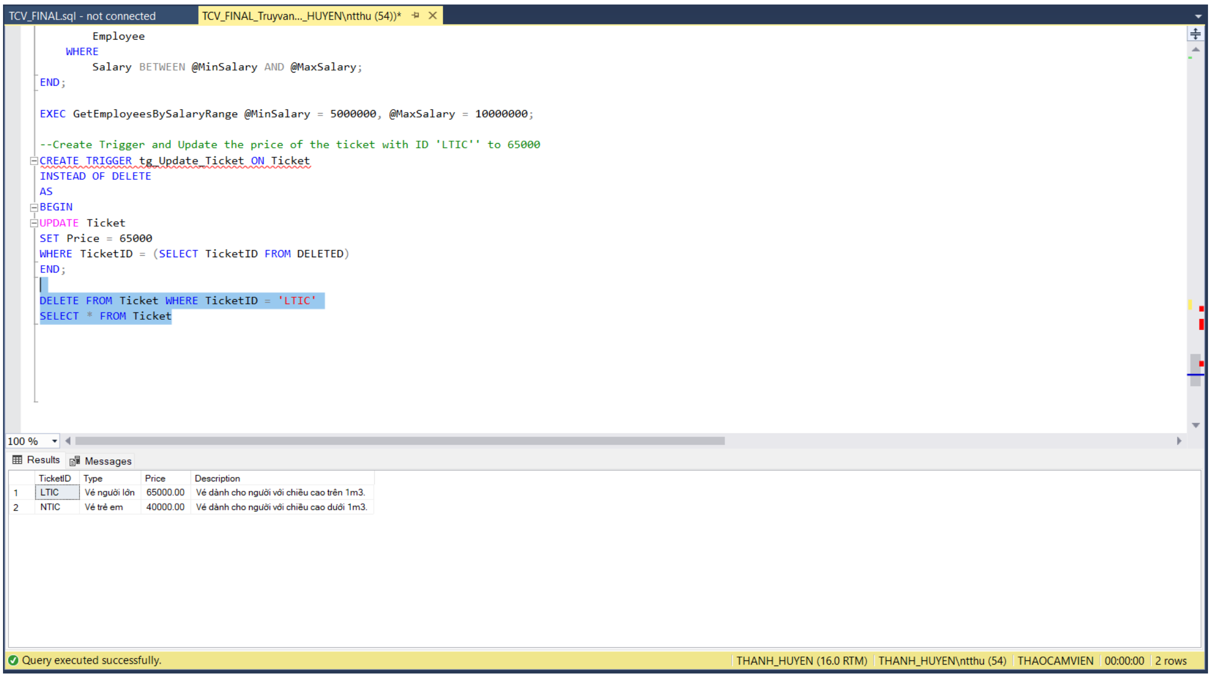

TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))*
100 %
Results
Messages

```
            District,
            City,
            Department,
            Salary,
            ManagerID
        FROM
            Employee
        WHERE
            Salary BETWEEN @MinSalary AND @MaxSalary;
END;

EXEC GetEmployeesBySalaryRange @MinSalary = 5000000, @MaxSalary = 10000000;

--Create Trigger and Update the price of the ticket with ID 'LTIC'' to 65000
CREATE TRIGGER tg_Update_Ticket ON Ticket
INSTEAD OF DELETE
AS
BEGIN
UPDATE Ticket
SET Price = 65000
WHERE TicketID = (SELECT TicketID FROM DELETED)
END;

DELETE FROM Ticket WHERE TicketID = 'LTIC'
SELECT * FROM Ticket
```

| | EmployeeID | LastName | FirstName | DateOfBirth | Sex | HireDate | Phone | Email | Street | District | City | Department | Salary | ManagerID |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | TCV0001 | Ngô Văn | Thái | 1989-11-12 | Nam | 2013-05-20 | 0956789012 | ngovanthai@gmail.com | Nguyễn Tri Phương | Quận 10 | Hồ Chí Minh | Kỹ thuật | 8500000.00 | TCV0002 |
| 2 | TCV0002 | Phạm Ngô Khôi | Vĩ | 1980-03-18 | Nam | 2014-07-12 | 0967890123 | phamngokhoivi@gmail.com | Nguyễn Trãi | Quận 5 | Hồ Chí Minh | Kỹ thuật | 9680000.00 | NULL |
| 3 | TCV0003 | Lê Thùy | Lâm | 1986-04-28 | Nữ | 2015-07-20 | 0987654321 | lethuylam@gmail.com | Nguyễn Tri Phương | Quận 7 | Hồ Chí Minh | Xí nghiệp kinh doanh | 5687000.00 | TCV0008 |
| 4 | TCV0004 | Hoàng Văn | Toàn | 1983-12-05 | Nam | 2016-02-25 | 0709876543 | hoangvantoan@gmail.com | Bạch Đằng | Bình Thạnh | Hồ Chí Minh | Xí nghiệp kinh doanh | 8470000.00 | TCV0007 |
| 5 | TCV0005 | Nguyễn Thiện | Mỹ | 1987-12-20 | Nữ | 2016-08-27 | 0823456789 | nguyenthienmy@gmail.com | Lê Đại Hành | Quận 11 | Hồ Chí Minh | Xí nghiệp kinh doanh | 8772500.00 | TCV0008 |
| 6 | TCV0006 | Trần Thị Ngọc | Hòa | 1988-07-25 | Nữ | 2017-09-10 | 0892345678 | tranthingochoa@gmail.com | Trần Hưng Đạo | Quận 5 | Hồ Chí Minh | Xí nghiệp kinh doanh | 7260000.00 | TCV0008 |
| 7 | TCV0009 | Hà Thị Lan | Anh | 1990-03-15 | Nữ | 2018-05-20 | 0881234567 | hathilanh@gmail.com | Nguyễn Thị Minh Khai | Quận 1 | Hồ Chí Minh | Xí nghiệp động vật | 5445000.00 | TCV0014 |
| 8 | TCV0010 | Phạm Nguyễn Kim | Oanh | 1989-10-08 | Nữ | 2018-11-30 | 0834567890 | phamnguyenkimoanh@gmail.com | Lê Hồng Phong | Vũng Tàu | Bà Rịa Vũng Tàu | Xí nghiệp động vật | 6292000.00 | TCV0014 |
| 9 | TCV0011 | Trần Đỗ Nhật | Huy | 1996-12-20 | Nam | 2019-08-27 | 0743210987 | trandonhathuy@gmail.com | Lê Lợi | Ninh Kiều | Cần Thơ | Xí nghiệp bảo vệ | 8470000.00 | TCV0021 |
| 10 | TCV0012 | Mai Thị | Liên | 1992-09-30 | Nữ | 2019-09-03 | 0710987654 | maithilien@gmail.com | Hà Tôn Quyền | Quận 11 | Hồ Chí Minh | Xí nghiệp động vật | 8470000.00 | TCV0014 |
| 11 | TCV0013 | Thái Hồng | Xuyên | 1991-09-05 | N? | 2019-10-12 | 0865432109 | thaihongxuyen@gmail.com | Lý Thường Kiệt | Quận 10 | Hồ Chí Minh | Xí nghiệp động vật | 7260000.00 | TCV0014 |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 20 rows

```
--Create Trigger and Update the price of the ticket with ID 'LTIC'' to 65000
CREATE TRIGGER tg_Update_Ticket ON Ticket
INSTEAD OF DELETE
AS
BEGIN
UPDATE Ticket
SET Price = 65000
WHERE TicketID = (SELECT TicketID FROM DELETED)
END;

DELETE FROM Ticket WHERE TicketID = 'LTIC'
SELECT * FROM Ticket
```


TCV_FINAL.sql - not connected
TCV_FINAL_Truyvan..._HUYEN\ntthu (54))*
100 %
Results
Messages

```
            Employee
        WHERE
            Salary BETWEEN @MinSalary AND @MaxSalary;
END;

EXEC GetEmployeesBySalaryRange @MinSalary = 5000000, @MaxSalary = 10000000;

--Create Trigger and Update the price of the ticket with ID 'LTIC'' to 65000
CREATE TRIGGER tg_Update_Ticket ON Ticket
INSTEAD OF DELETE
AS
BEGIN
UPDATE Ticket
SET Price = 65000
WHERE TicketID = (SELECT TicketID FROM DELETED)
END;

DELETE FROM Ticket WHERE TicketID = 'LTIC'
SELECT * FROM Ticket
```

| | TicketID | Type | Price | Description |
|---|---|---|---|---|
| 1 | LTIC | Vé người lớn | 65000.00 | Vé dành cho người với chiều cao trên 1m3. |
| 2 | NTIC | Vé trẻ em | 40000.00 | Vé dành cho người với chiều cao dưới 1m3. |

Query executed successfully. | THANH_HUYEN (16.0 RTM) | THANH_HUYEN\ntthu (54) | THAOCAMVIEN | 00:00:00 | 2 rows

## EVALUATE MEMBERS

| No. | Full Name | Student ID | Contribution |
|---|---|---|---|
| 1 | Nguyễn Trần Thanh Huyền | K224111450 | 100% |
| 2 | Phạm Tuyết Nhung | K224111460 | 100% |
| 3 | Vũ Quỳnh Như | K224111461 | 100% |
| 4 | Lê Nguyễn Minh Thảo | K224111462 | 100% |
| 5 | Nguyễn Ngọc Diểm Thúy | K224111466 | 100% |

# THANK YOU

GROUP 3