PROGRAM TITLE: BTEC in Computing (Software Engineering)

UNIT TITLE: Database Design & Development

ASSIGNMENT NUMBER: assignment 2

ASSIGNMENT NAME: Database Design & Development

SUBMISSION DATE: 4/25/2022

DATE RECEIVED: 4/25/2022

TUTORIAL LECTURER: Nguyen Duc Giang

WORD COUNT: 4325

STUDENT NAME: Duong Duc Manh

STUDENT ID: BKC12128

MOBILE NUMBER: 0348121868

LO1. Use an appropriate design tool to design a relational database system for a substantial problem.

problem statement

Air ticket sales management problem

1. Staff

   Employee information includes: name, age, date of birth, position, department, employee code
2. Manage

   Management information includes: name, age, date of birth, position, management code
3. Flight

   Flight information includes: flight code, aircraft parking location, date_tine, number, boarding gate
4. Customer

   Customer information includes: customer code, name, class of customer
5. Plane

   Information about the plane includes: airline, number, number of seats, number of passengers, popularity
6. Ticket

   Ticket information includes: fare, seat code, ticket code, date_time, plane number, plane code

P1. Design a relational database system using appropriate design tools and techniques, containing at least four interrelated tables, with clear statements of user and system requirements.

1. User and system requirements

Requirement is a thing demanded or obligatory. Comprehensive requirement document will help to decide if the benefits of new application or upgrade justify the cost.

User requirements are statements, in a natural language plus diagrams, of what services the system is expected to provide to system users and the constraints under which it must operate. The user requirements may vary from broad statements of the system features required to detailed, precise descriptions of the system functionality.

System requirements are more detailed descriptions of the software system's functions, services, and operational constraints. The system requirements document (sometimes called a functional specification) should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.

Different kinds of requirement are needed to communicate information about a system to different types of reader. Figure illustrates the distinction between user and system requirements.

User requirements definition

1. The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System requirements specification

1.1 On the last working day of each month, a summary of the drugs prescribed, their cost and the prescribing clinics shall be generated.
1.2 The system shall generate the report for printing after 17.30 on the last working day of the month.
1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of prescriptions, the number of doses prescribed and the total cost of the prescribed drugs.
1.4 If drugs are available in different dose units (e.g. 10mg, 20mg, etc.) separate reports shall be created for each dose unit.
1.5 Access to drug cost reports shall be restricted to authorized users as listed on a management access control list.

2. Describe the binding of the objects of interest: object names, properties, binding between objects

a. Use Case Diagram

Theoretically: Use Case is a technique that describes the interaction between the user and the system (in a specific environment, for a specific purpose).

This interaction could be:

- The way in which the user interacts with the system;
- The way in which a system interacts with other systems.

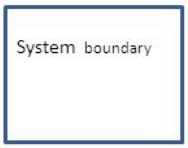
Purpose: Like a contract between a player software developers and customers.

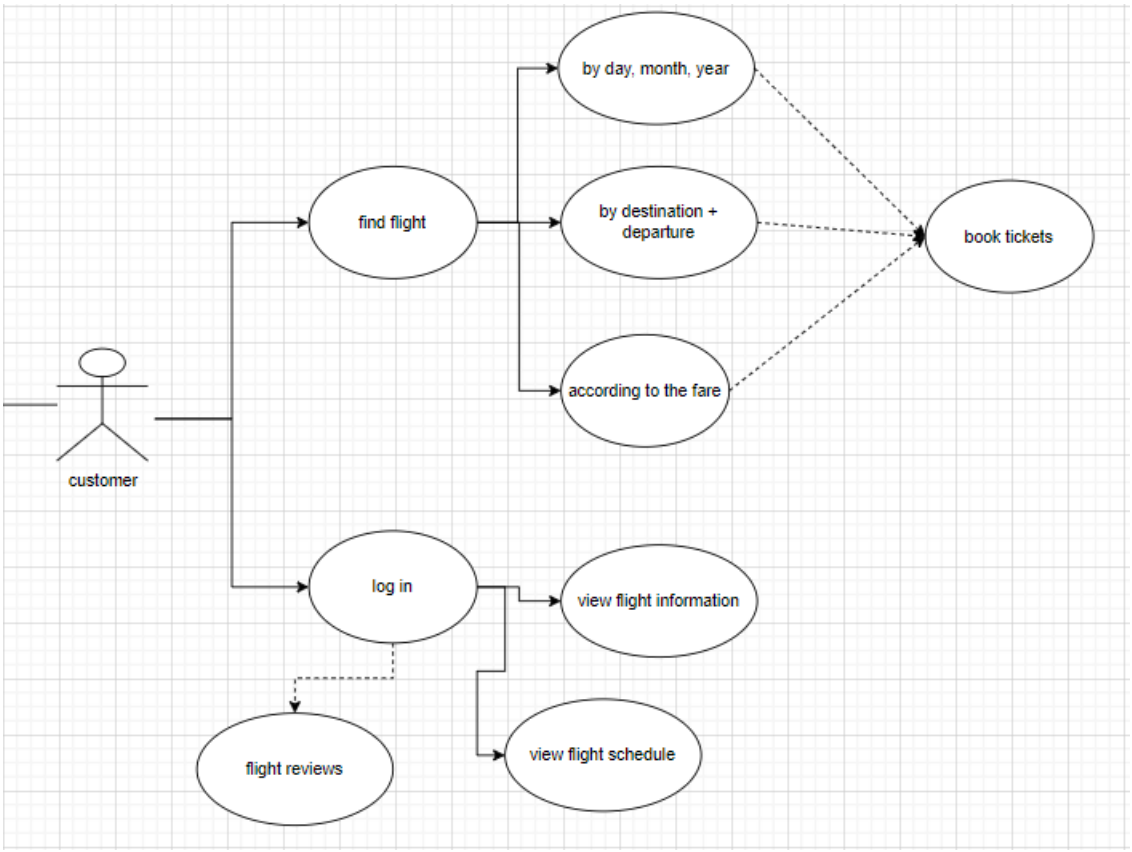
A powerful tool for planning Used in all stages of the process system developer

- Customer approves use-case diagram
- Use use case diagrams to discuss with customers.
- The members involved in the project, use this model to better understand the system

Use Case Components:

Components of a Use Case Diagram

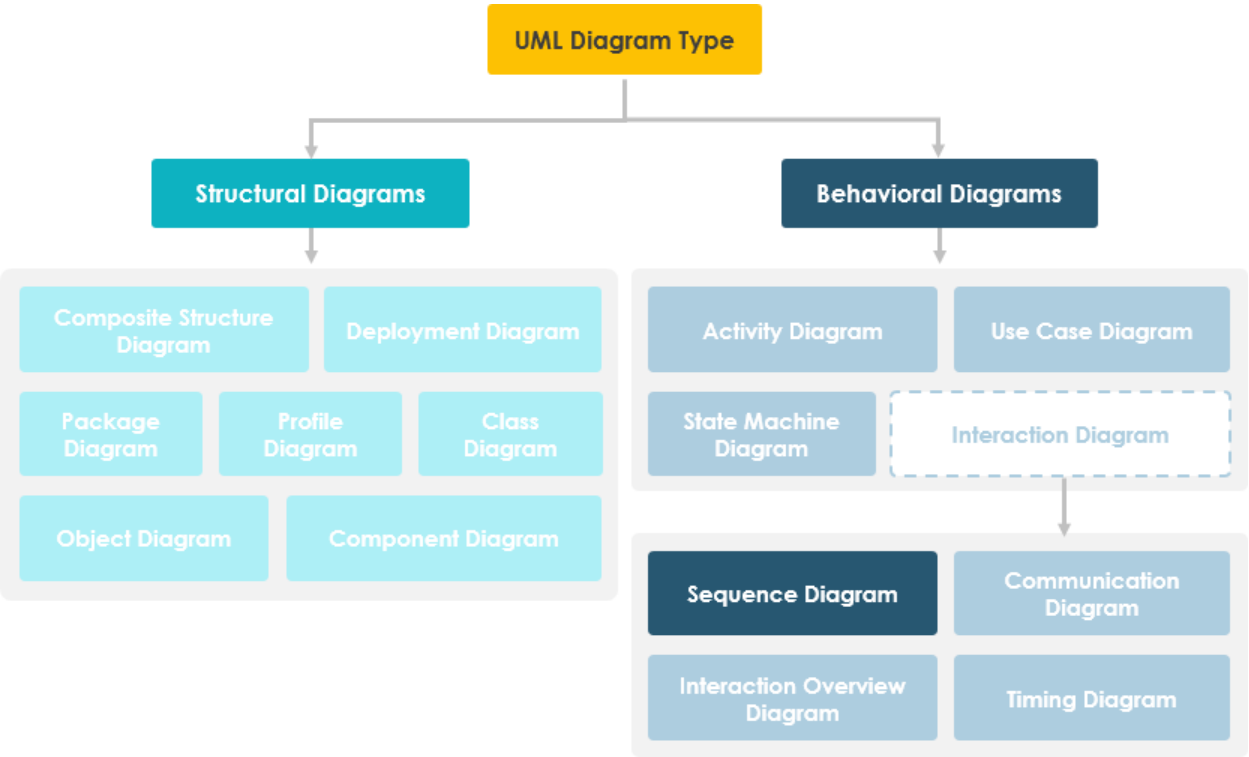
Notation	Meaning
	Use case: An Ellipse represents a single Use Case. The name of the use case is written inside ellipse.
 Actor	Actor: Represents every element which interact with the system. It may be a user a another system which interact with the system described in the use case.
	Ordinary Relationship: Represents a connection that is a channel for the transfer of information between a use case and an actor or between use cases.
	System Boundary: A square represents the boundary of the system. Inside the system are the use cases and outside the square are the actors who interact with the system.



Img-1. Use case diagram of customer

b. Sequence diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.



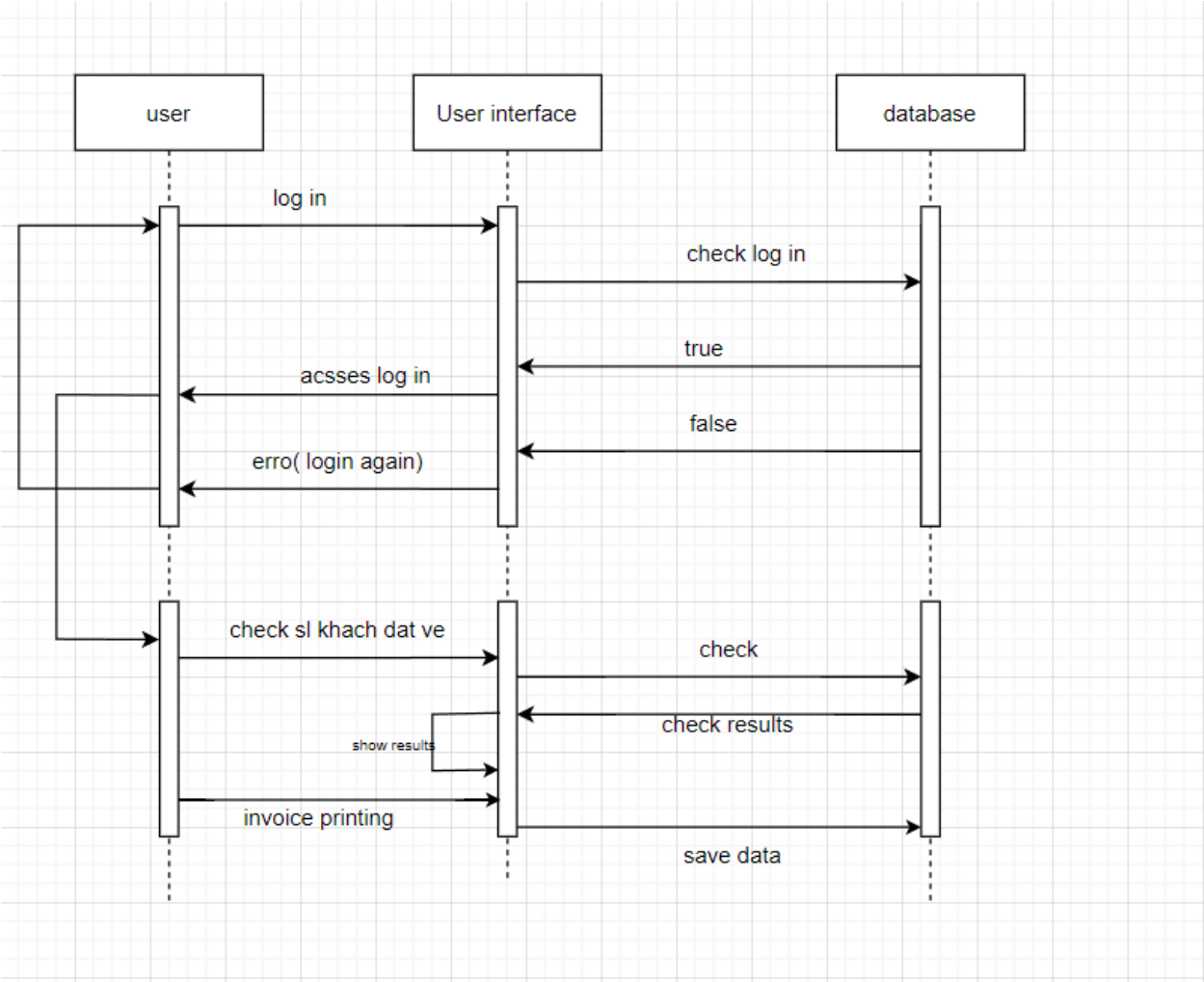
Sequence Diagrams captures:

- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation

- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)



Img-2. Sequence diagram of user

a. Class diagram

Class diagram (CD) shows the existence of classes and their relationships in the version logical design of a system

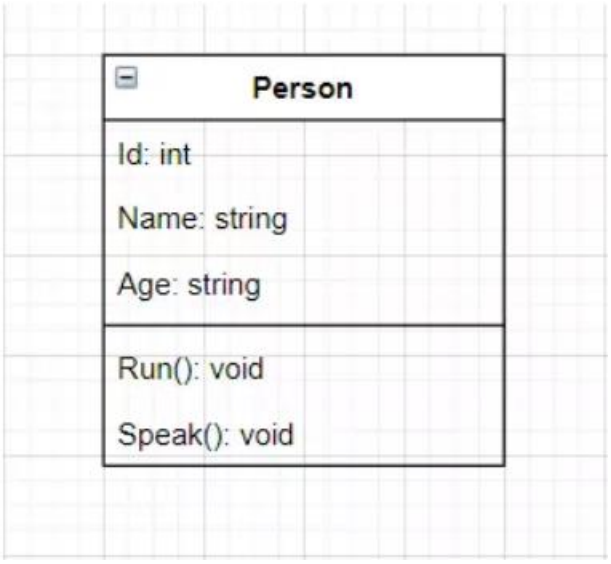
- Indicate the static structure of the model such as class, side structure within them and their relationships with other classes.
- Indicate all or part of the class structure of a system.
- Do not provide provisional information.

Static view of a primary system that supports functional requirements of the system.

Basic properties of class diagrams

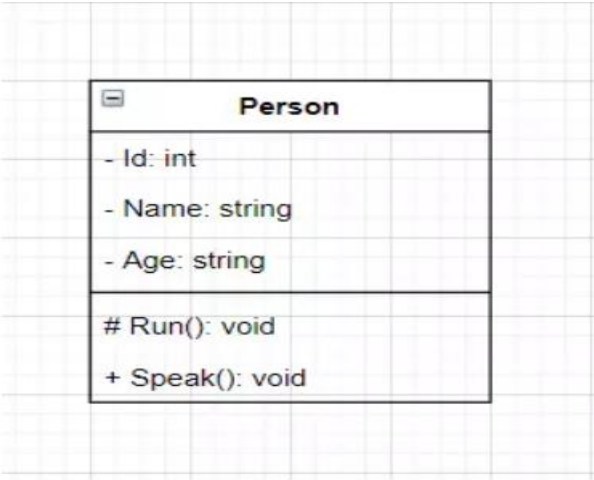
- Class name
- Attribute (field, property)
- Operation (method, function)

Example declaration of name, attribute, operation with return type of a class:



Access Modifier in class diagram:

- Used to specify the scope of access for the Attribute and Operation of a class (Grant permission to other classes to use the Attribute and Operation of this class).
- 4 access range selection
    - Private (-): Only the objects created from this class can be used.
    - Public (+): Any object can use it.
    - Protected (#): Only objects created from this class and classes that inherit from this class can be used.
    - Package/Default: Objects created from classes in the same package can be used.

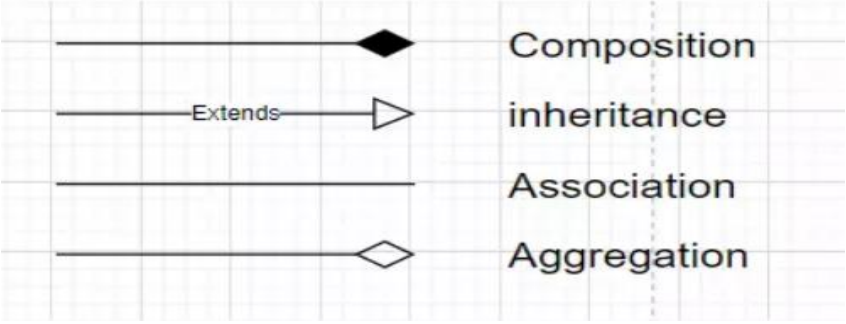


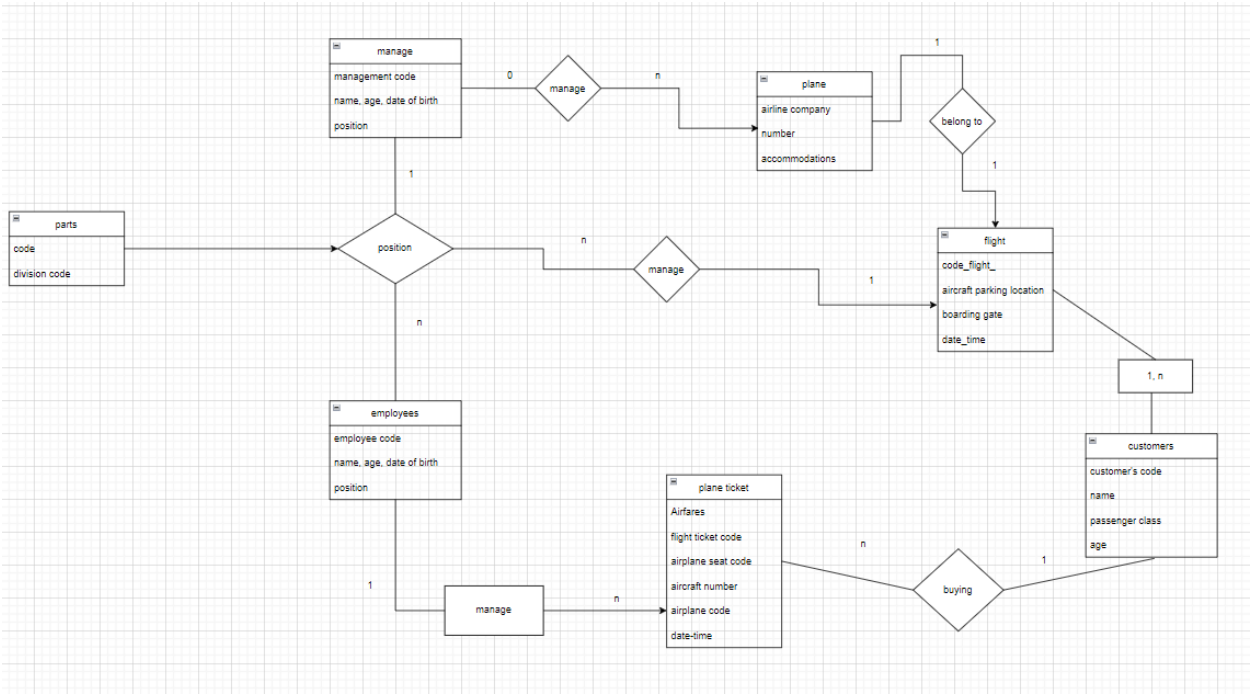
Relationship in class diagram: Used to represent each relationship between objects created from one class and objects created from other classes in the class diagram.

4 types of Relationships:

- Inheritance: A class inherits from another class.
- Association: 2 classes that are related but do not specify the relationship.
- Composition: If the object created from class A is lost, the object created from class B will be lost.

- Agreegation: If the object created from class A is lost, the object created from class B will still exist independently.





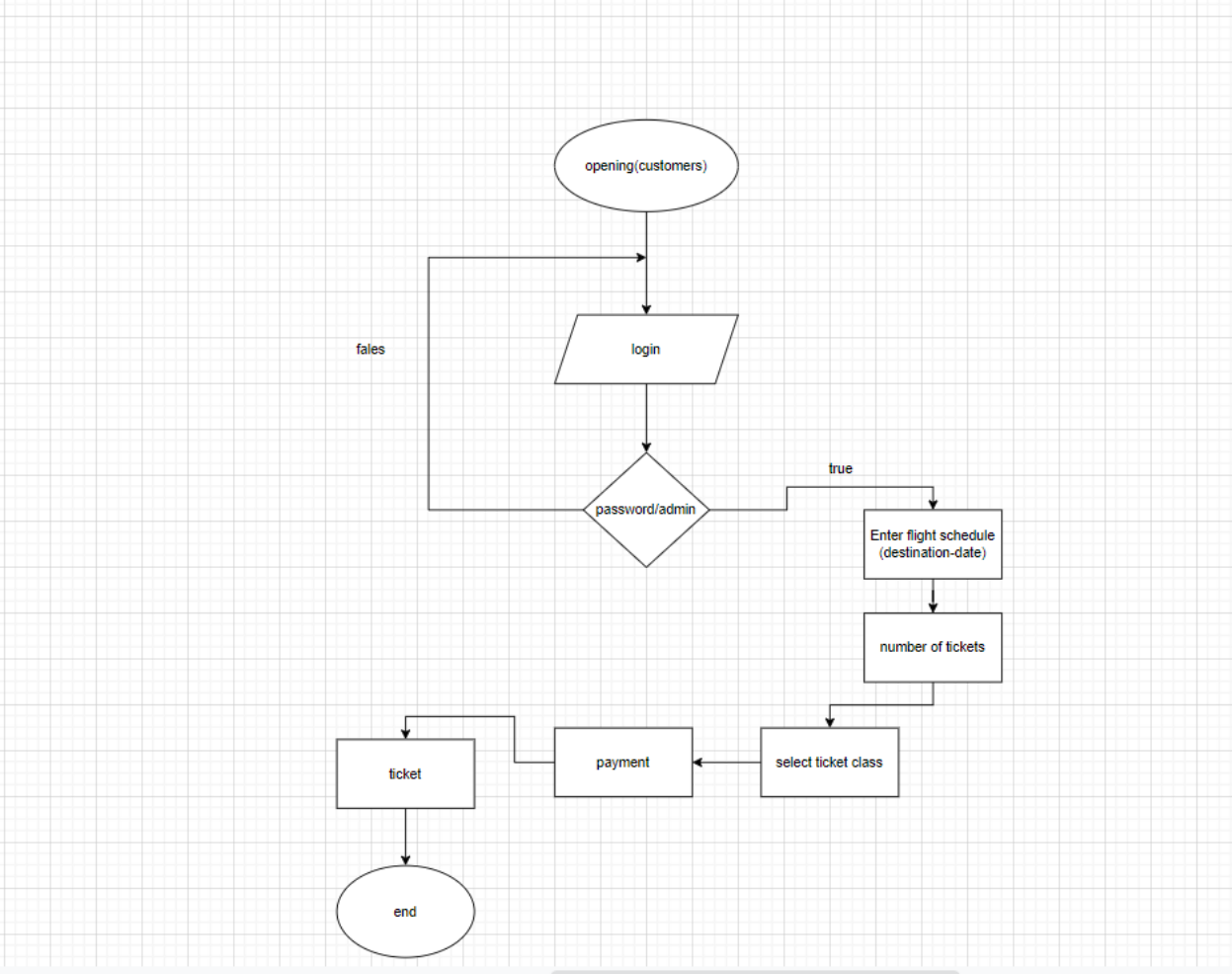
Img-3. Class diagram of flight ticket management

a. Flowchar

Indicates the flow of control from one activity/action to another.

Flow of Events: this use case starts when the Registrar requests that the system close registration.

- The system checks to see if registration is in progress. If it is, then a message is displayed to the Registrar and the use case terminates. The Close Registration processing cannot be performed if registration is in progress.
- For each course offering, the system checks if a professor has signed up to teach the course offering and at least three students have registered. If so, the system commits the course offering for each schedule that contains it



Img-4.Class flowchar of customers

3. Produce a comprehensive design for a fully functional system which includes interface and output designs, data validations and data normalization.

a. Explain data normalization with examples. Check whether the provided logical design in task 1.1 is normalised. If not, normalize the database by removing the anomalies.

Normalization of data

Normalization is a technique of arranging data into many related tables to minimize data redundancy. Normalization is the process of decomposing relations from anomalies to produce smaller and well-structured relations.

Normalization is used for:

- Eliminate redundant data
- Make sure data dependencies make sense

What is the purpose of normalization?

If redundant data is found in the table, it will take up more memory space and will also make it difficult for users to process and update the database. In other words, repetition of data increases the size of the database.

What does Anomalies mean?

Anomalies are problems that occur in un normalized databases. Where all data is stored in a table.

Types of anomalies

- Insertion Anomaly - Add or insert new data to existing existing data.

  For example, new admission to a school.

- Abnormal Update- Update on data to existing data.

  Example: Mr. Adam left school? Or no longer the school's HOD? This is Update anomaly.

- Abnormal deletion- Delete data from existing data.

  Example: In the students table, both the student data and the branch data are together. If student records are deleted we will lose data on the branch because we

Types of Normalization

- First normal form
- second normal form
- third normal form
- BC normal form

First Normal Form

First normal form sets fundamental rule for database normalization and relates to a single table within a relational database system.

In a first normal form table the table should contain atomic values and there are no repeating groups.

An atomic value is a value which cannot be divided.

A repeating group means the tables containing two or more columns which are closely related

EX:

Employee table:

Employee code	Name	Age	Department	Date of birth
100AB	Duong	26	Sale	1996/7/4
100AC	Thi	27	Flight operator	1993/6/5
100AD	Linh	25	Sale	1995/5/6
100AE	Chi	25	Sale	1995/4/7

100AF	Na	24	Air traffic control	1996/3/8

Manage table:

Manage code	Name	Age	Department	Date of birth
1A	Dep	30	Sale	1991/3/11
1B	Thang	36	Flight operator	1985/3/12
1C	Lam	35	Air traffic control	1986/6/11

Second Normal Form Second normal form (2NF)

is the second step in normalizing a database. A table is in 2nd Normal Form if:

The table is in First normal form, and

All the non-key columns are dependent on the table's primary key.

Second normal form and third normal form handle with functional dependency and transitive dependency.

A functional dependency means that it meets the requirements of First Normal Form(1NF), and all non-key attributes are fully functionally dependent on the primary key. A database to be fully functional dependency it must be in first normal form. Each attributes holds a single atomic value.

e.g.: If attribute A determines the value of B, we write this A -> B — meaning that B is functionally dependent on A. In this relationship, A determines the value of B, while B depends on A.

Employee code	Name
100AB	Duong
100AC	Thi
100AD	Linh

Manage code	Name
1A	Dep
1B	Thang
1C	Lam

Third normal form

Third normal form is the third step of normalization. A table is in third normal form if: A table is in 2nd normal form. It contains only columns that are non- transitively dependent on the primary key.

Transitive dependency

Transitive dependency means if we have a primary key A and a non-key domain B and C where C is more dependent on B than A and B is directly dependent on A, then C can be considered transitively dependent on A.

E.g.: As the below table is 2nd normal form fails to meet 3rd normal form

Manage code	Name	Department	Employee code	Name
1A	Dep	Sale	100AB	Duong
1B	Thang	Flight operator	100AC	Thi
1C	Lam	Air traffic control	100AD	Linh

Table1. 9

Manage code	Name	Employee code	Name
1A	Dep	100AB	Duong
1B	Thang	100AC	Thi
1C	Lam	100AD	Linh

Table1. 10 Club winner

Manage code	Employee code	Department
1A	100AB	Sale
1B	100AC	Flight operator
1C	100AD	Air traffic control

Boyce and Codd Normal Form

Boyce and codd normal form is the latest version of normalization. This form handles certain anomalies that cannot be handled by 3rd normal form.

For a table to be in BCNF, following conditions must be satisfied:

- R must be in 3rd Normal Form
- And, for each functional dependency (X → Y), X should be a super Key

b. Design set of simple interfaces to input and output for the above scenario using Wireframe or any interface-designing tool.

Wireframe is a design tool to create or develop web. It helps to create the basic structure of a website. Wireframe is a site's blue print and visual guide the provides details of website navigation, content category and organization, content layout and structure, functionality items, user interaction items, methods of displaying the content.

The benefits of using a wireframe are:

- Visualize the structure clearly.
- Clarify the features of interface.
- Push usability to the forefront.
- Help to refine navigation.
- Make the design process iterative.

According to the scenario the following are the functionalities of the system

- Customer details
- Job details
- Transport details
- Payment details
- Deport details

- Product details
- Load details

c. Explain the validation mechanisms to validate data in the tables with examples.

Validation mechanism

Validation is a process that ensures the delivery of a clear data to the program. It checks for the integrity and validity of data that is being input to different software and its components. Data validation is also known as input validation.

Constraint Validation is a type of data validation. Constraint Validation is an algorithm browsers run when a form is submitted to determine validity.

There are methods of validations which are named below

Check digit - A redundancy check used for error detection. It is used to check ifthe range of numbers are entered correctly

Example, the ISBN-10 numbering system for books uses 'Modulo-11' division, where it outputs the remainder of the division as the result of the operation

[CITATION tea20 \l 1033].

- Type Check – this is a way to confirm that the correct data type is inputted.

  Example, in an application form age may range from 0 to 100. A number data type would be an appropriate choice for this data. By defining the data type as number ,only numbers are allowed in the field (e.g. 18, 20, 25) and it would prevent people from inputting verbal data, like 'eighteen'[CITATION tea201 \l 1033].

- Length check- Allows checking if the data entered is not short or long. this is used to make sure that the correct number of characters are entered into the field

Example, consider a password that needs to be 8 characters long. The length check will ensure that exactly 8 characters are entered into the field [CITATION tea20 \l1033].

- Format check- Checks if the entered data is in the right and appropriate format.

Example, a National Insurance number is in the form XX 99 99 99 XX where X is any letter and 9 is any number[CITATION tea20 \l 1033]

- Constraints

Constraints ensure the data entered into a relational database within a limit. Purpose of constraints is to maintain data integrity during an update, insert and delete into a relation.

Types of constraints:

- Not null

Not null ensures that a column does not hold any null or empty value.

e.g:

```
CREATE TABLE manage (
maquanly NCHAR(100) not null ,
namemanage NCHAR(100)not null,
age NCHAR(100)not null,
Dateofbirth CHAR(100)not null,
department NCHAR(100)not null
);
```

- Unique

Unique constraint ensures the column in a table contain unique values. If a column contains unique constraints then the column cannot have duplicate values in the table.

e.g:

```
CREATE TABLE manage (
maquanly NCHAR(100) not null unique,
namemanage NCHAR(100)not null,
age NCHAR(100)not null,
Dateofbirth CHAR(100)not null,
department NCHAR(100)not null
);
```

- Check

e.g:

```
CREATE TABLE plane (
airlinecompany NCHAR(100) not null,
numberair NCHAR(100)not null primary key ,
seats NCHAR(100)not null,
accommodations int  check(accommodations >=500) ,
popular char not null
);
```

- Primary key

Primary key is a unique identification in a relation. Whereas it should have unique values and should be not null.

e.g:

```
CREATE TABLE plane (
airlinecompany NCHAR(100) not null,
numberair NCHAR(100)not null primary key ,
seats NCHAR(100)not null,
accommodations int  check(accommodations >=500) ,
popular char not null
);
```

- Foreign key

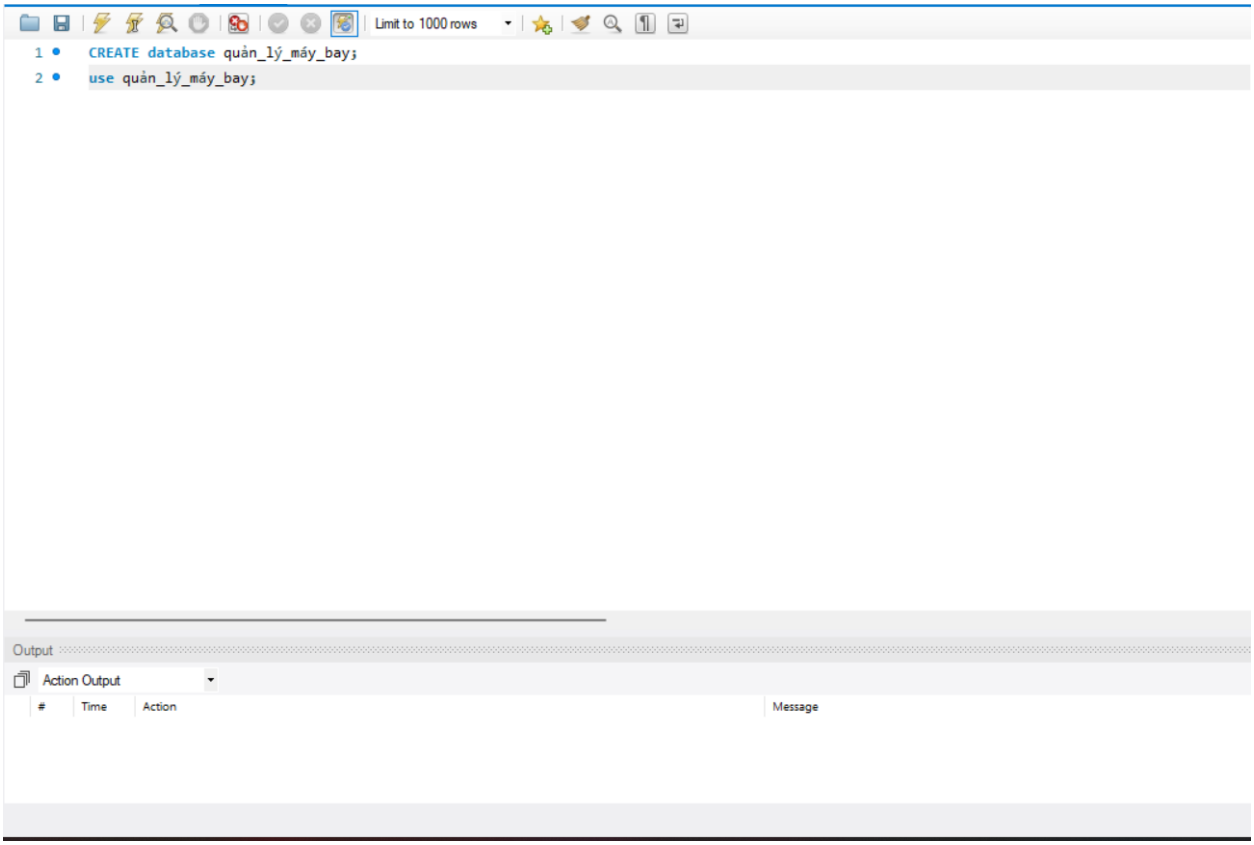
E.g:

```
CREATE TABLE plane (
airlinecompany NCHAR(100) not null,
numberair NCHAR(100)not null primary key ,
seats NCHAR(100)not null ,
accommodations int  check(accommodations >=500) ,
popular char not null,
    FOREIGN KEY (seats)
    REFERENCES airlinecompany(seats)
);
```

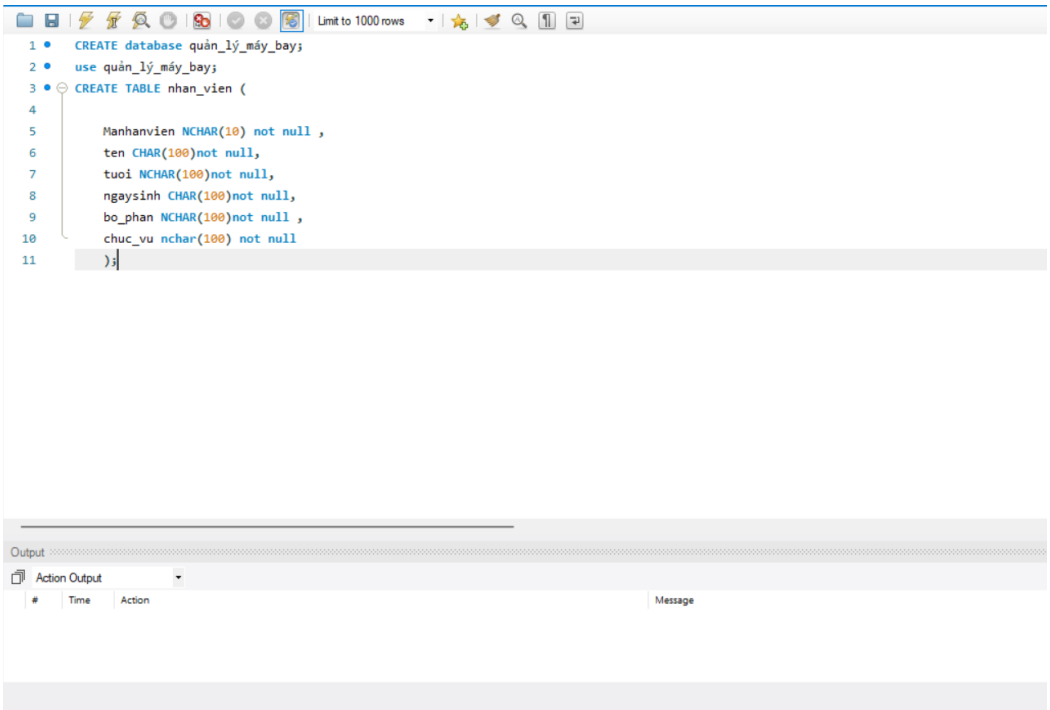
LO2. Develop a fully functional relational database system, based on an existing system design.

P2 Develop the database system with evidence of user interface, output, and data validations, and querying across multiple tables.

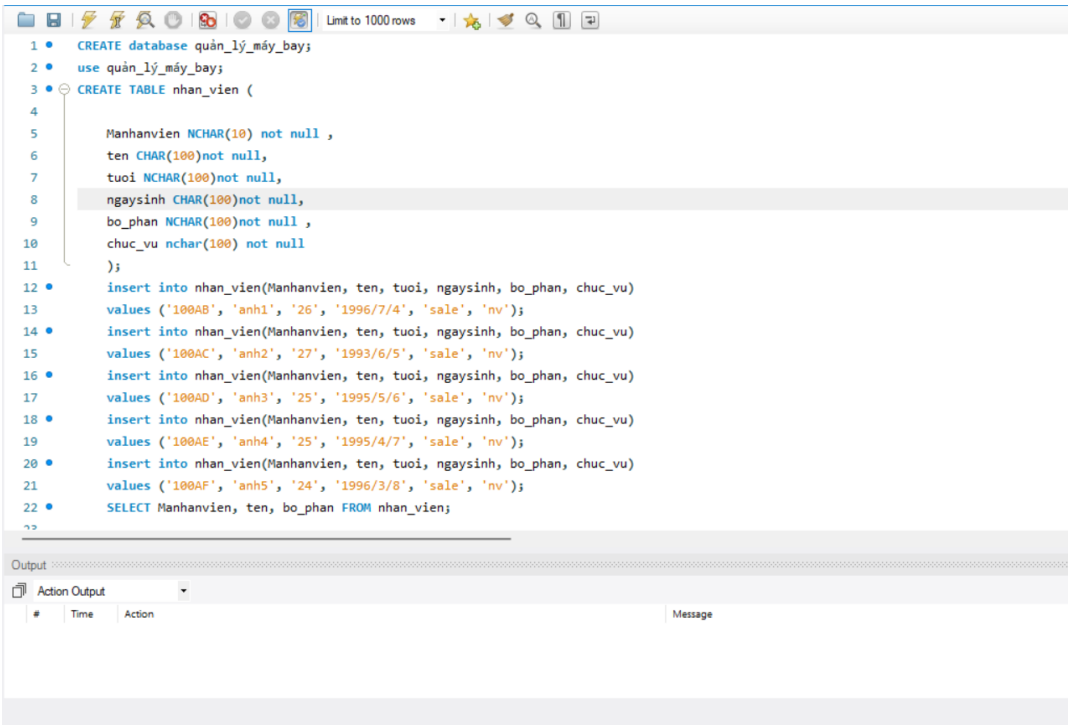
1. Develop a relational database system according to the ER diagram you have created (Use SQL DDL statements).



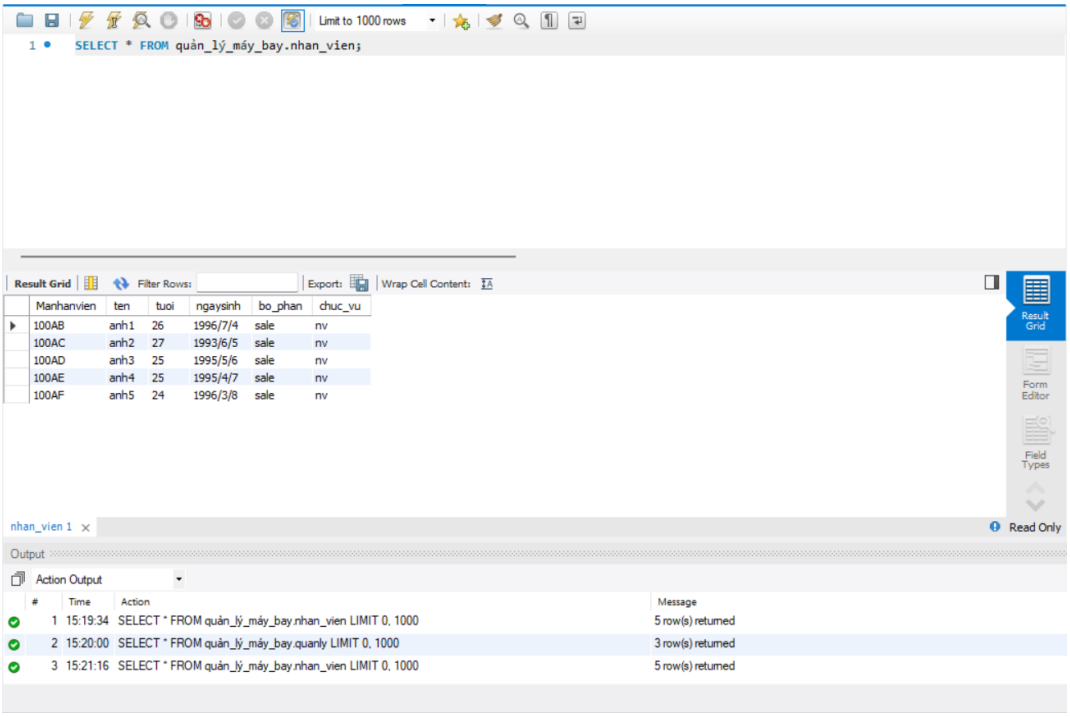
Img-1. Create and Use the database



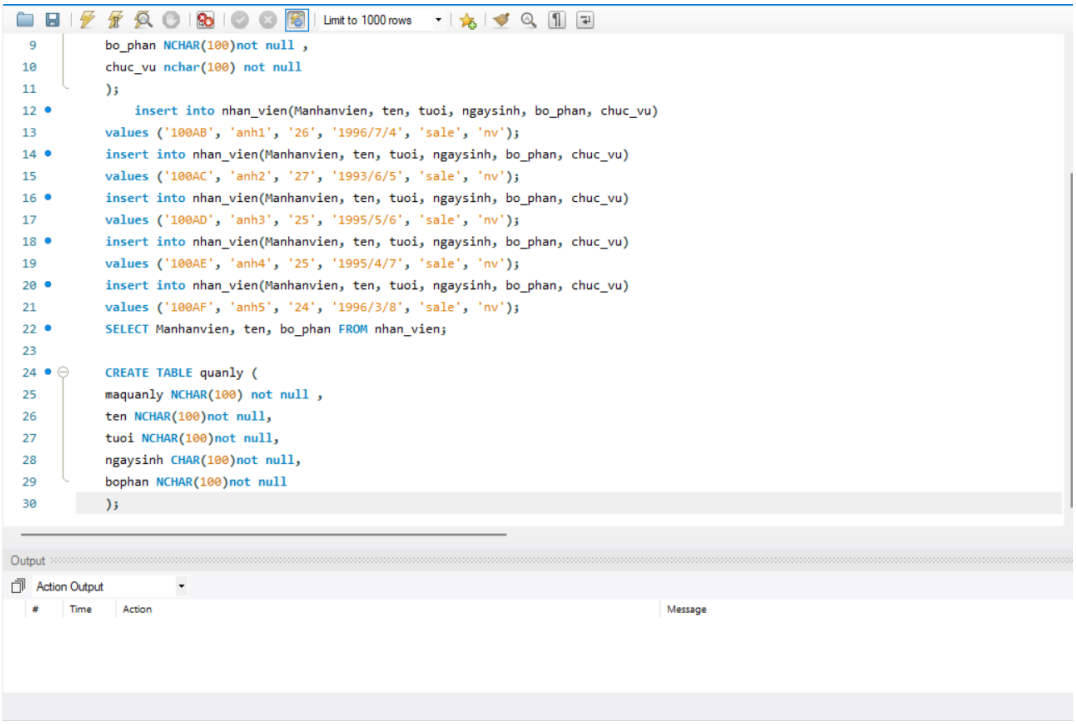
Img-2. Create Tables nhan_vien



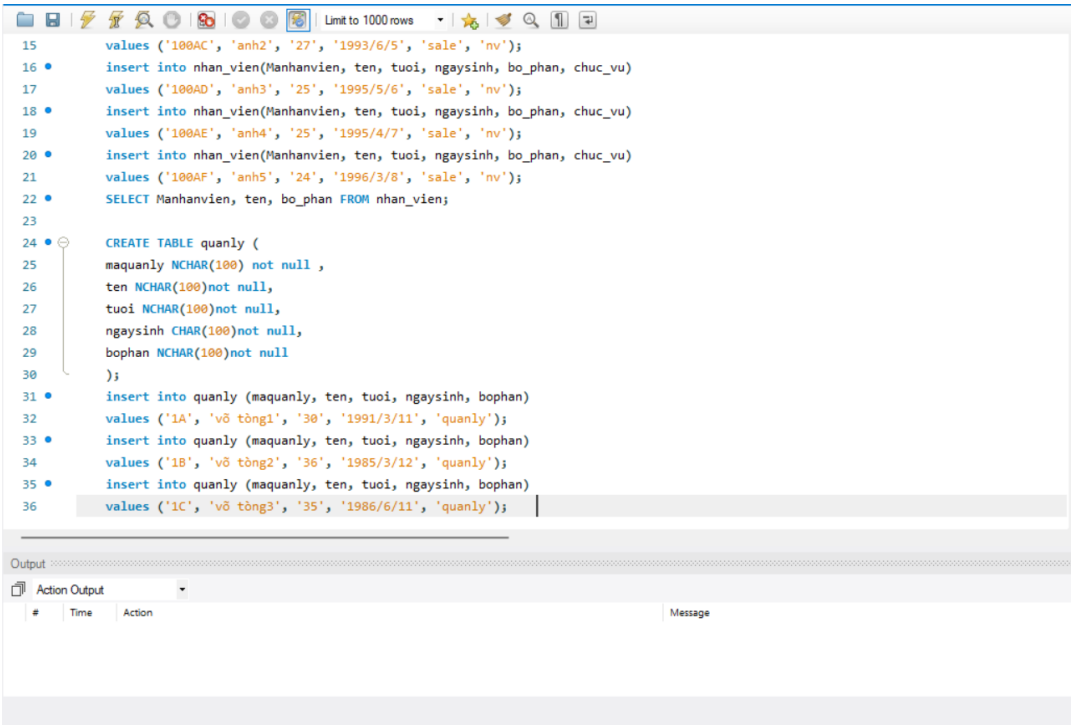
Img-3. Insert nhan_vien values



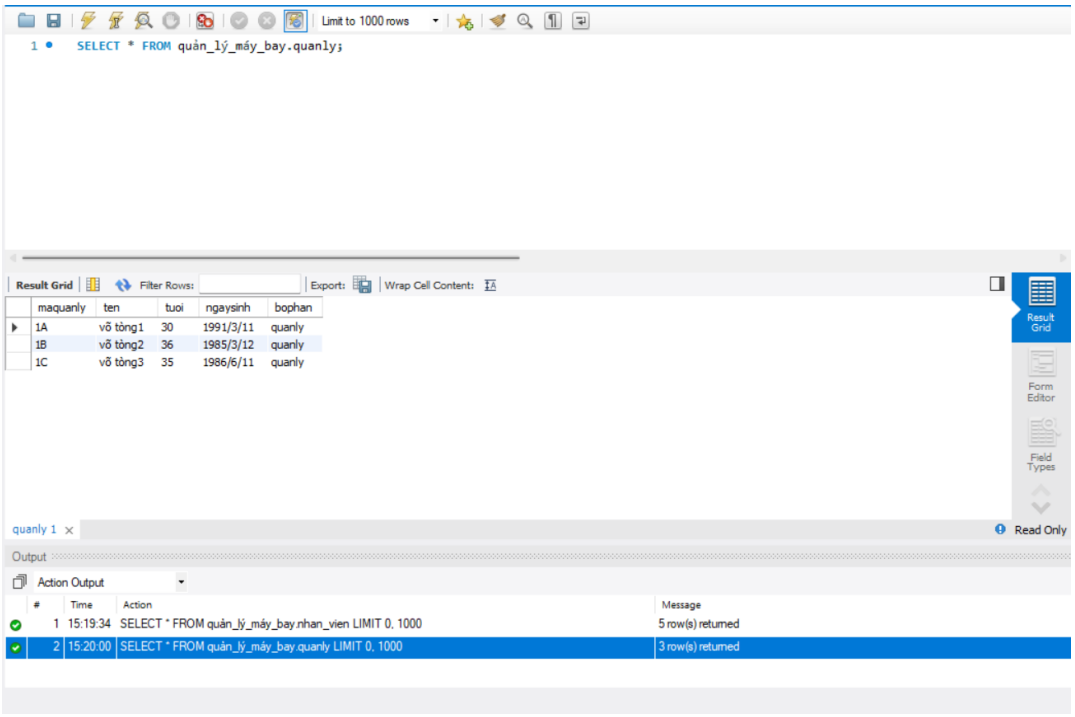
Img-4. Select all from product



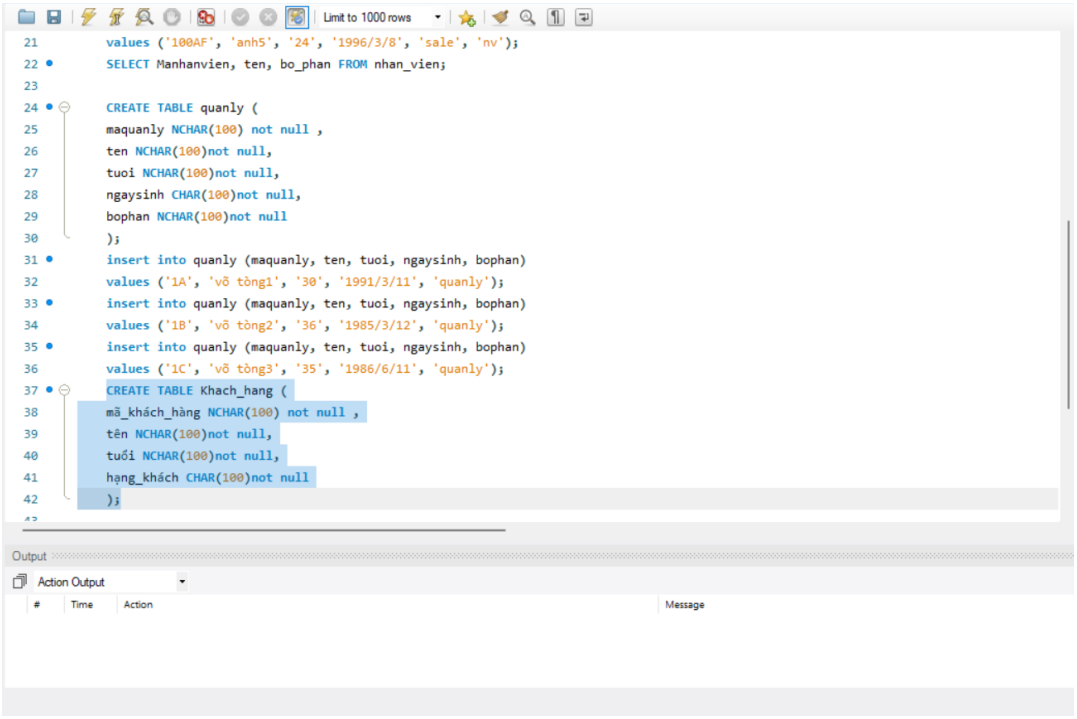
Img-5. Create Tables quanly



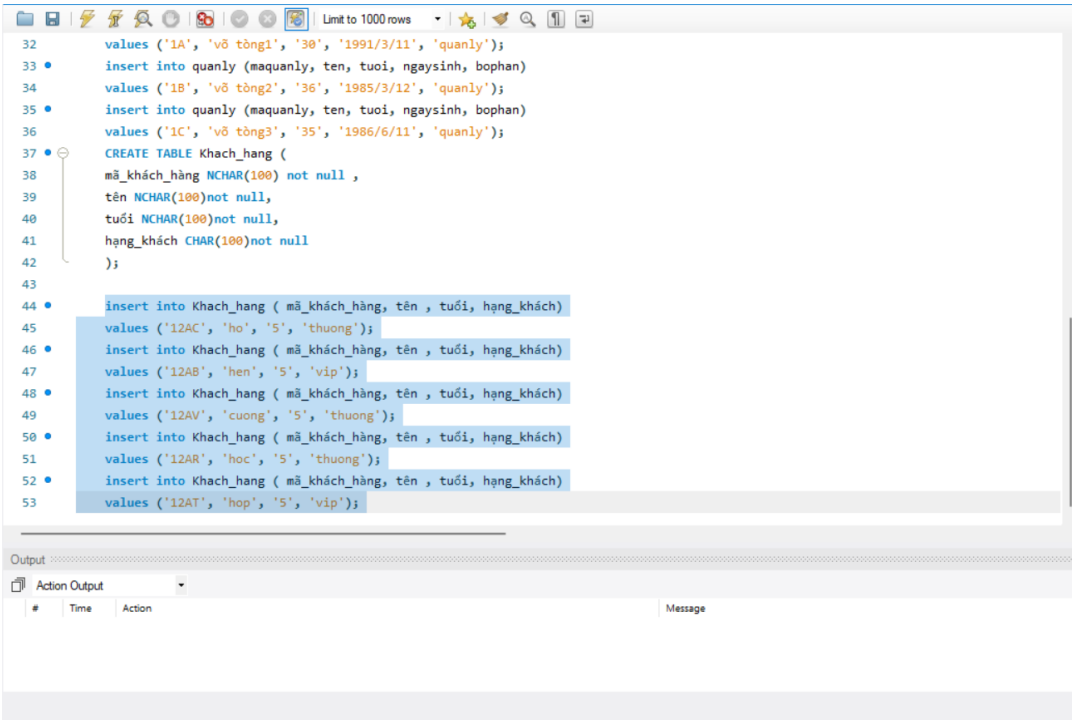
Img-6. Insert quanly values



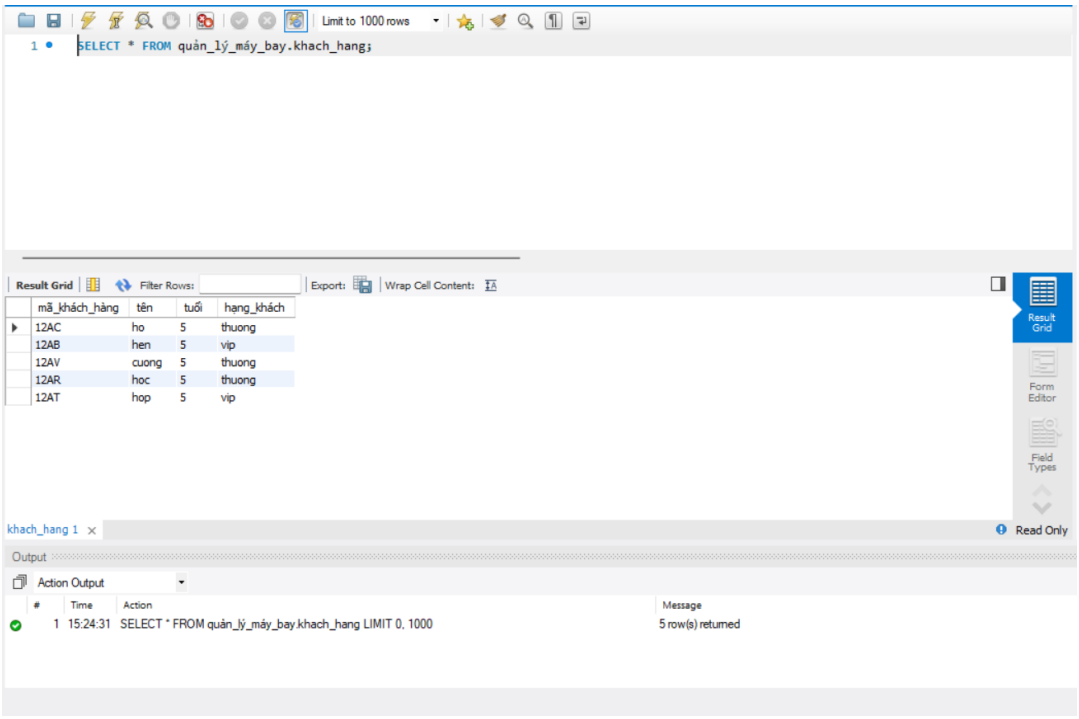
Img-7. Select all from product



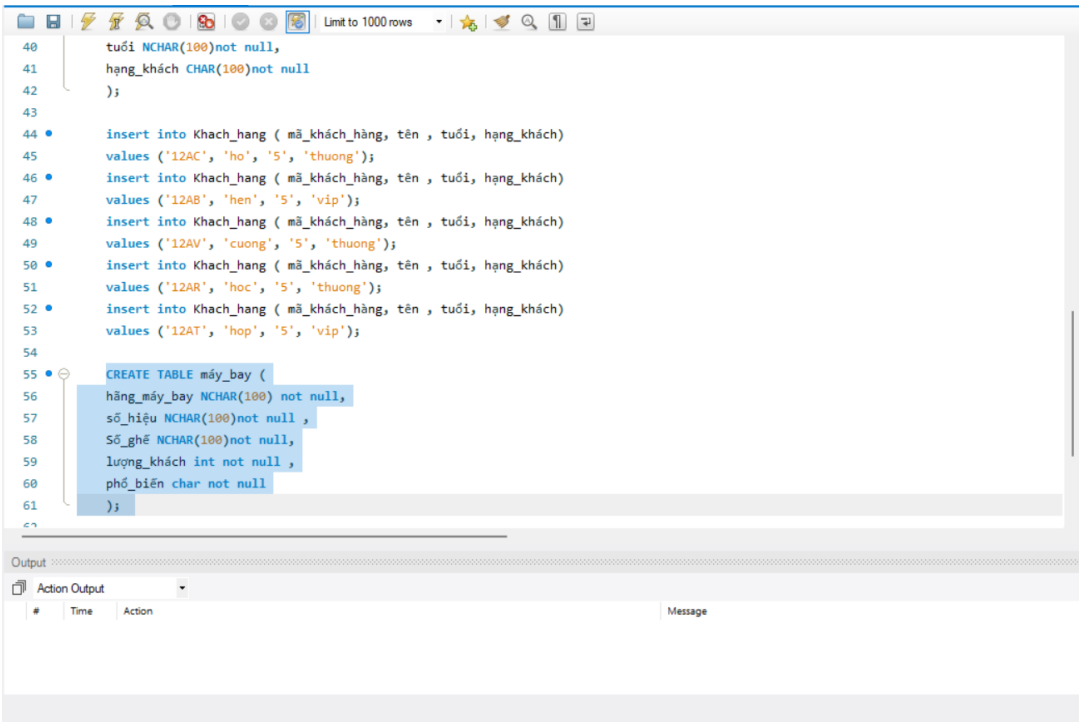
Img-8. Create Tables Khach_hang



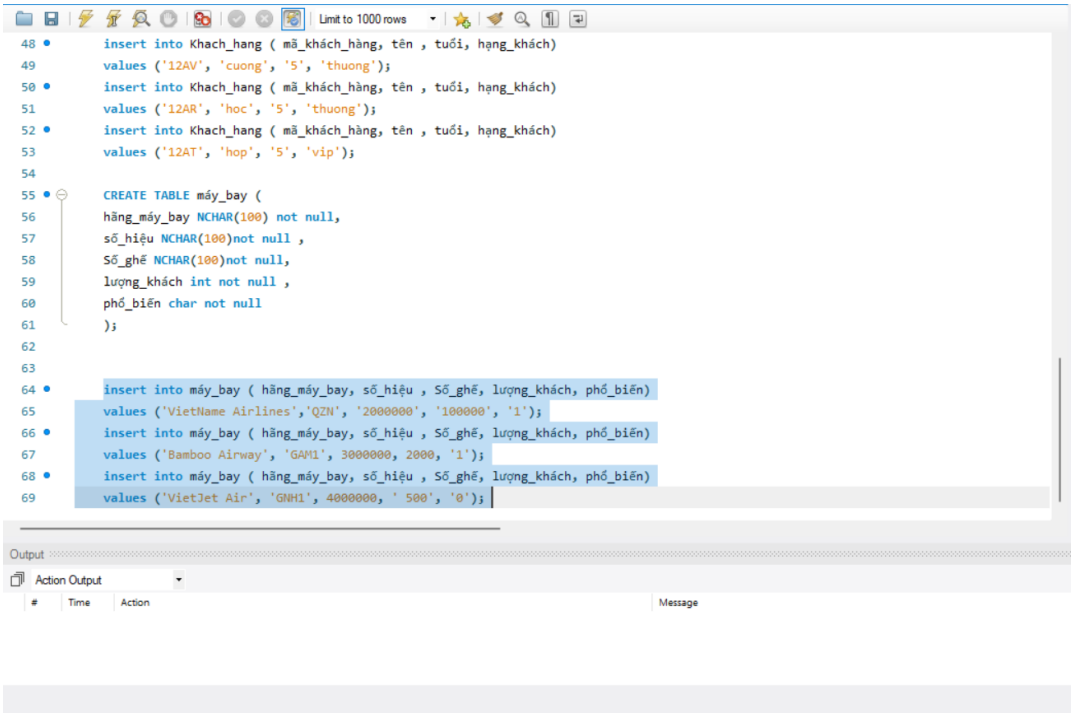
Img-9. Insert Khach_hang values



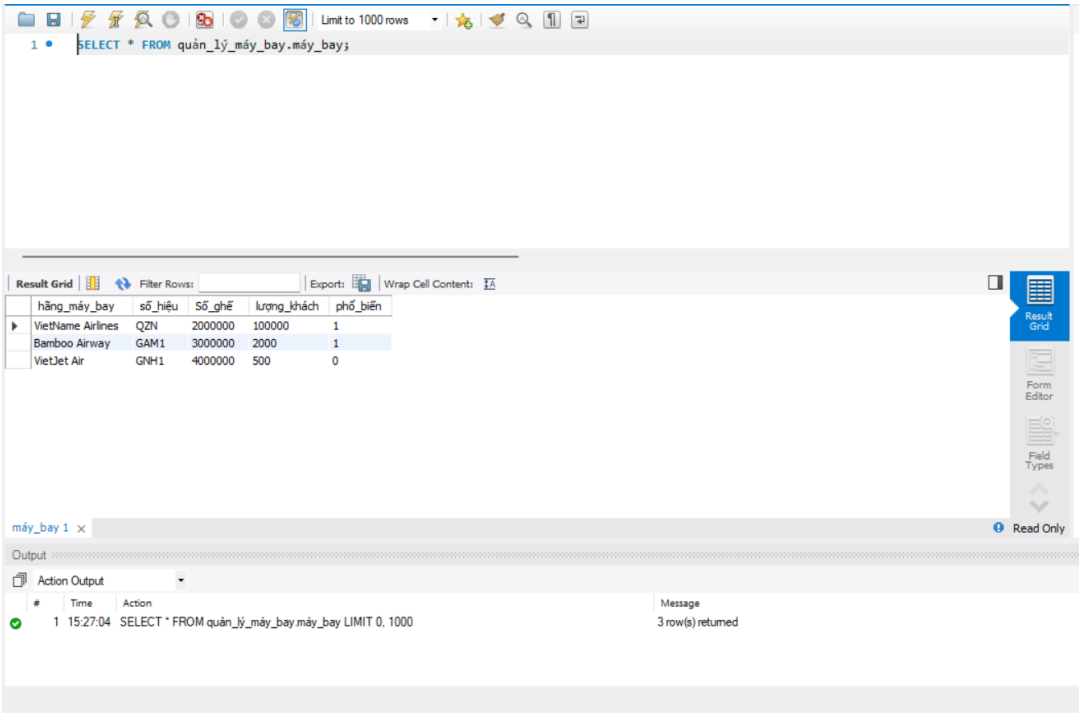
Img-10. Select all from product



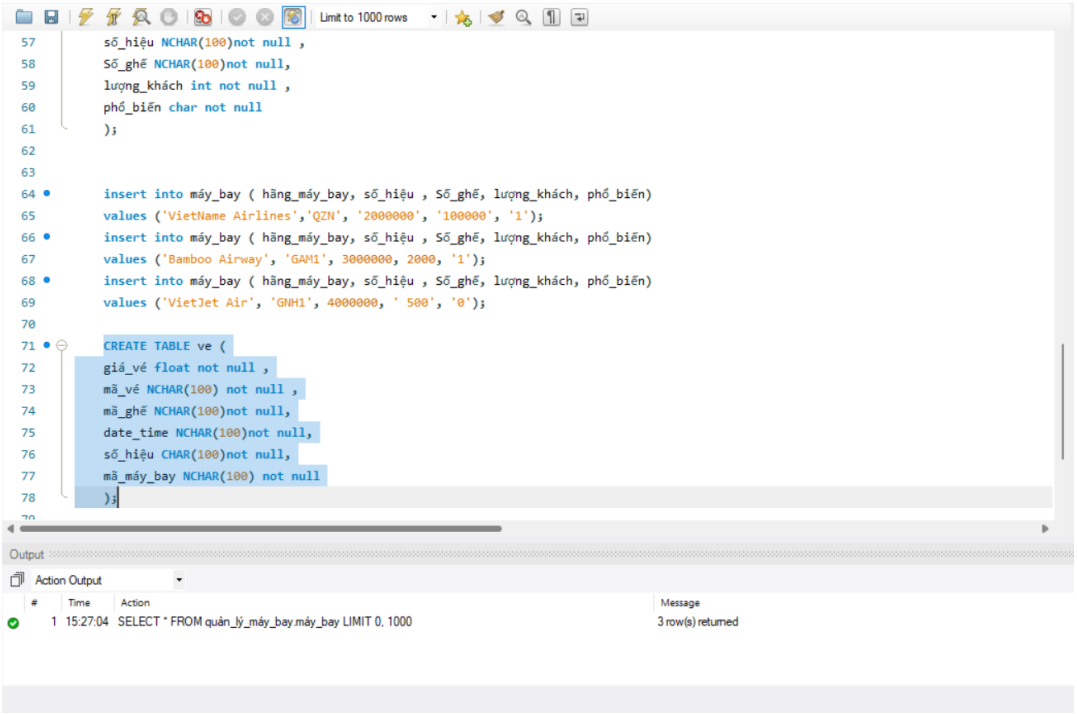
Img-11. Create Tables máy_bay



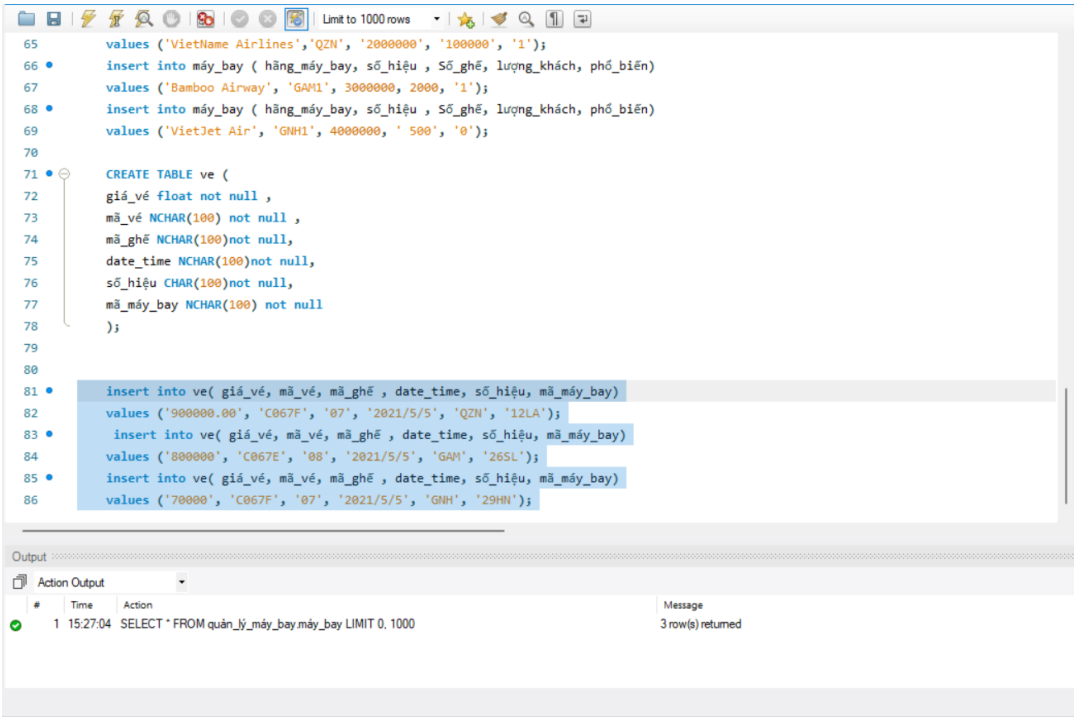
Img-12. Insert máy_bay values



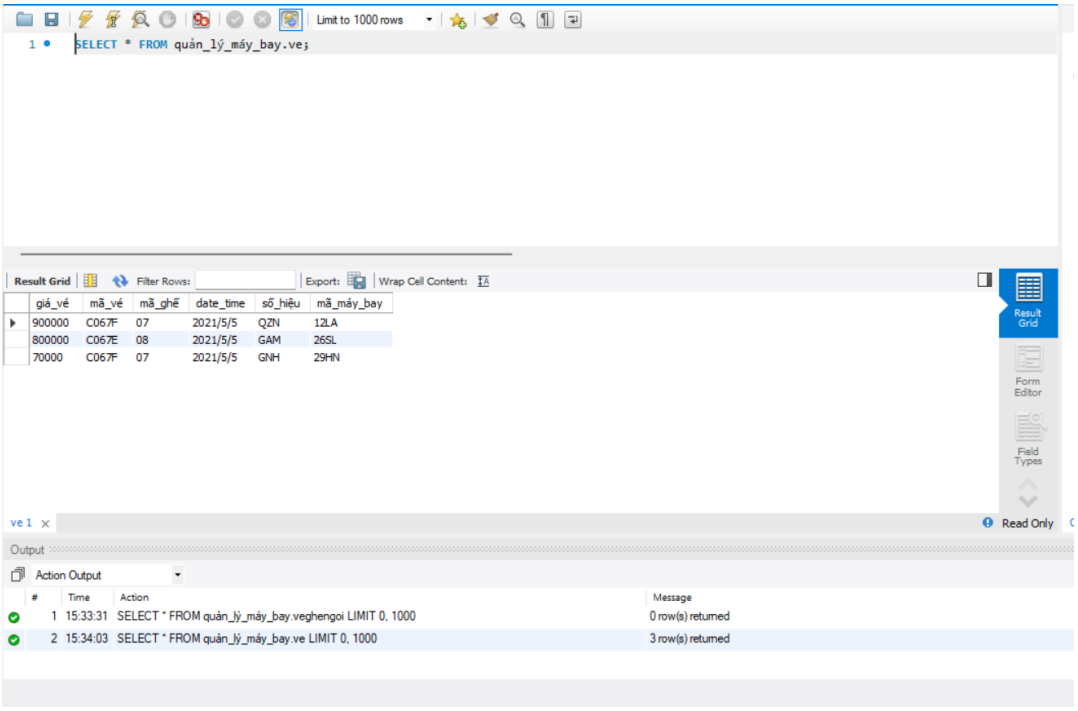
Img-13. Select all from product



Img-14. Create Tables ve



Img-15. Insert ve values



Img-16. Select all from product

2. Provide evidence of the use of a suitable IDE to create a simple interface to insert, update and delete data in the database

Microsoft Visual Studio is a integrated development environment IDE used to develop computer programs, websites, web apps and mobile applications. Visual studio supports 36 different programming languages and allow code editors and debugger to support any programming language such as C++, C#, JavaScript, Python, etc. Visual studio 2017 is the latest version of visual basic launched by Microsoft.

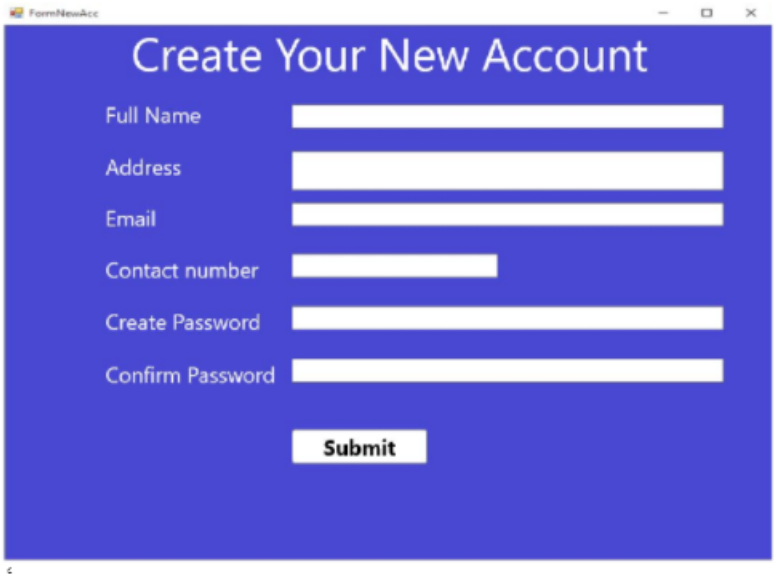
Visual Studio 2017 consists of many new features

- Designer
- Code Editor
- Debugging
- Extensibility
- Other tools

The edition in Visual Studio:

- Professional
- Enterprises
- Community

-

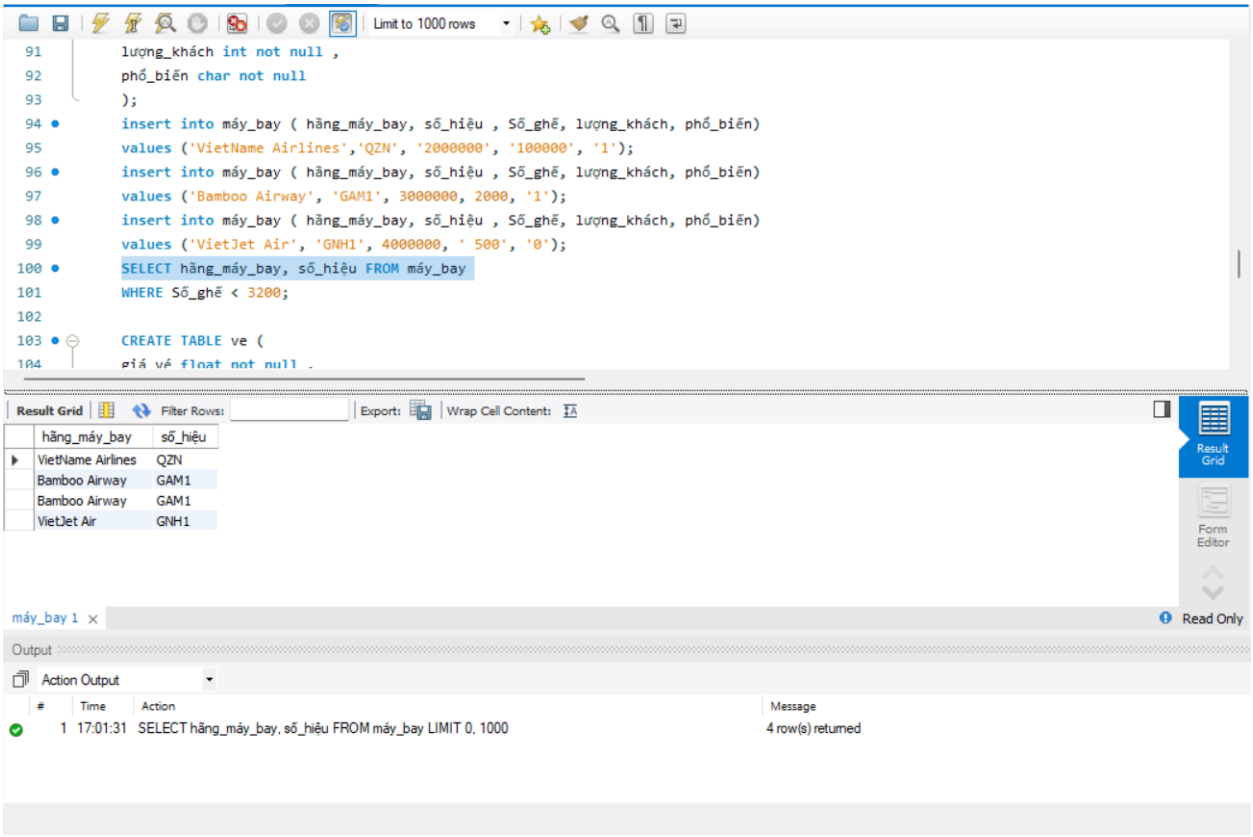

FormNewAcc
Create Your New Account
Full Name
Address
Email
Contact number
Create Password
Confirm Password
Submit

(P3) Implement a query language into the relational database system

A data manipulation language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

The functional capability of DML is organized in manipulation commands like SELECT, UPDATE, INSERT INTO and DELETE FROM, as described below:

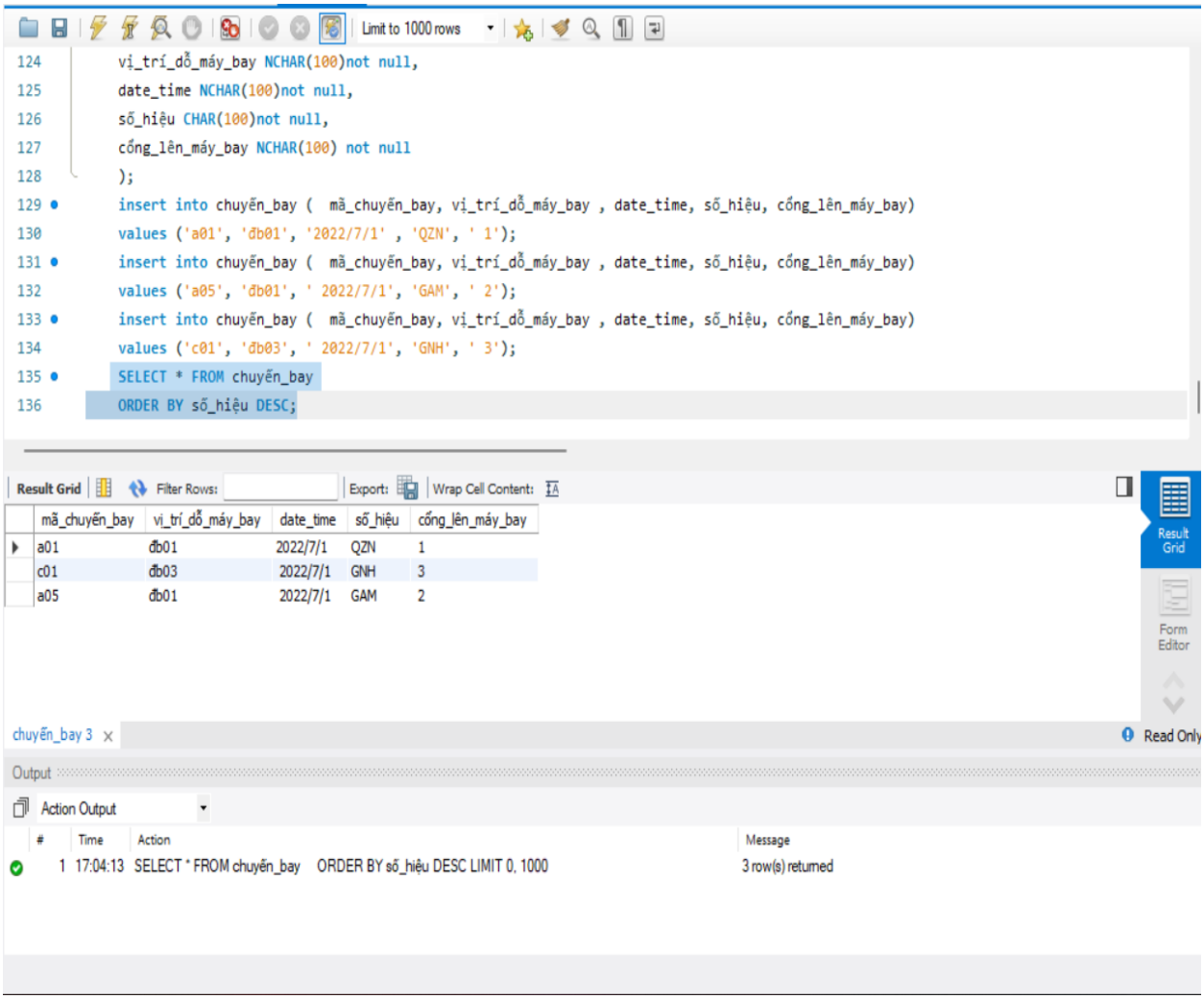
- SELECT: This command is used to retrieve rows from a table. The syntax is SELECT [column name(s)] from [table name] where [conditions]. SELECT is the most widely used DML command in SQL.



Img-1. Select column hãng_máy_bay, số_hiệu from table máy_bay

- The ORDER BY keyword is used to sort the result-set in ascending or descending order.

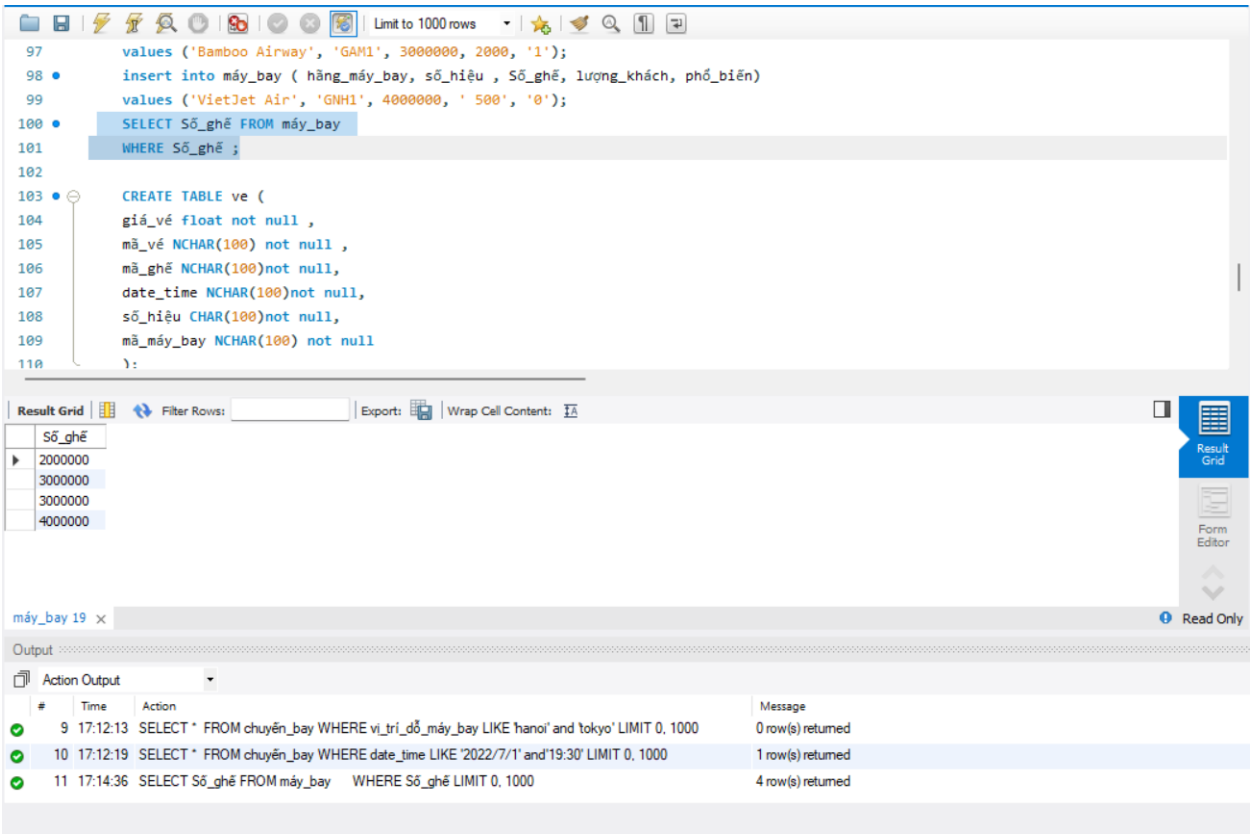
  The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.



Img-2. selects all customers from the " chuyến_bay" table, sorted DESCENDING by the " số_hiệu" column

- The WHERE clause is used to filter records.

  It is used to extract only those records that fulfill a specified condition.



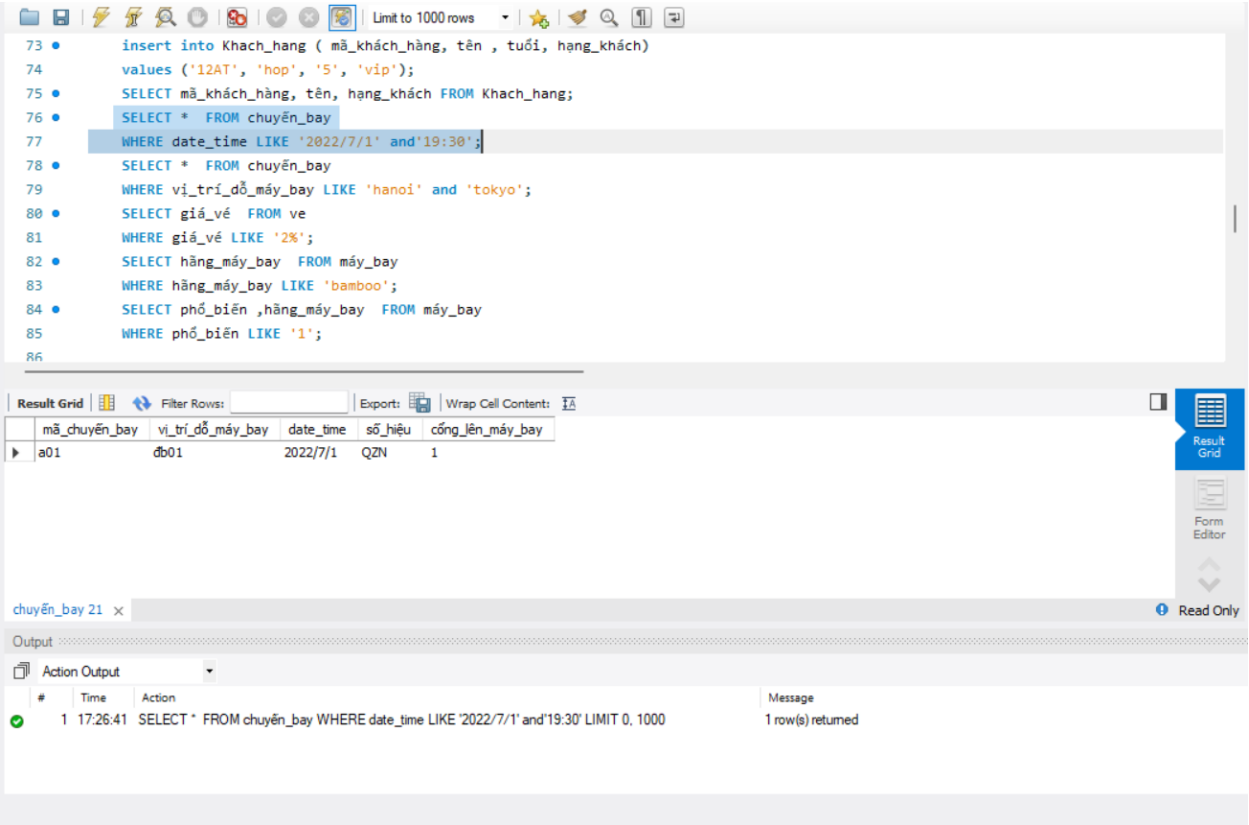
Img-3. select all seat numbers from the table máy_bay

- The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

  There are two wildcards often used in conjunction with the LIKE operator:

  - The percent sign (%) represents zero, one, or multiple characters
  - The underscore sign (_) represents one, single character

  The percent sign and the underscore can also be used in combinations!



Img-4. Select all flights with a date_time stating with 2022/7/1 and 19:30

(M2) Implement a fully functional database system, which includes system security and database maintenance.

Security mechanisms

Database security is a top priority for today's organizations. Database security concerns the use of a broad range of information security controls to protect databases against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/administrative and physical. Database security is a specialist topic within the broader realms of computer security, information security and risk management.

Types of security control in databases:

- Access control
- Auditing
- Authentication
- Encryption
- Integrity controls
- Backups
- Application security

Access control (**AC**) is the selective restriction of access to a place or other resource. The act of accessing may mean consuming, entering, or using. Permission to access a resource is called authorization.

Database auditing involves observing a database so as to be aware of the actions of database users. Database administrators and consultants often set up auditing for security purposes, for example, to ensure that those without the permission to access information do not access it.

Authentication is the first step in access control, and there are three common factors used for authentication. It is the process or act of confirming that a user who is attempting to log in to a database is authorized to do so, and is only

accorded the rights to perform activities that he or she has been authorized to do. There are three common factors used for authentication:

- Something you know (such as a password)
- Something you have (such as a smart card)
- Something you are (such as a fingerprint or other biometric method)

Encryption is the process that uses an algorithm to transform data stored in a database into "cipher text" that is incomprehensible without first being decrypted. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm.

Data integrity is the overall completeness, accuracy and consistency of data. Data integrity is usually imposed during the database design phase through the use of standard procedures and rules.

Three integrity constraints are used in a relational database

- Entity Integrity
- Referential Integrity
- Domain Integrity

Backups are copies of data from your database that can be used to reconstruct that data.

Backups can be divided into physical backups and logical backups.

- Logical backups contain logical data exported from a database with an Oracle export utility and stored in a binary file, for later re-importing into a database using the corresponding Oracle import utility.
- Physical backups are the foundation of any sound backup and recovery strategy. Logical backups are a useful supplement to physical backups in

many circumstances but are not sufficient protection against data loss without physical backups.

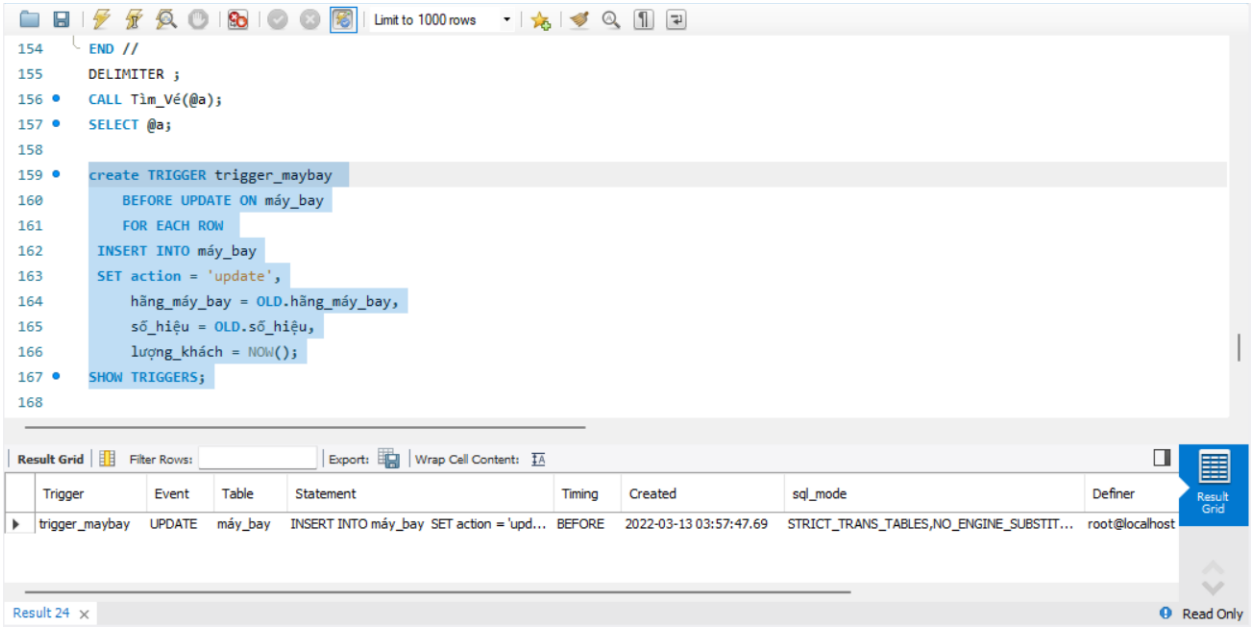
Create Trigger command in MySQL

In this tutorial you will learn how to use the Create Trigger command in MySQL, with which you can use this command to create any new trigger.

If you do not know what the concept of Trigger is, I can briefly say as follows: Trigger is a trigger, which will be activated when a certain impact on it.

In MySQL Trigger means a piece of SQL code that handles a certain function, it works in the background and is run when one of the actions such as Insert / Update / Delete into the table.

Ex:



Img-1. create and test a trigger in MySQL

MySQL Stored Procedures

The stored procedure is SQL statements wrapped within the CREATE PROCEDURE statement. The stored procedure may contain a conditional statement like IF or CASE or the Loops. The stored procedure can also execute another stored procedure or a function that modularizes the code.

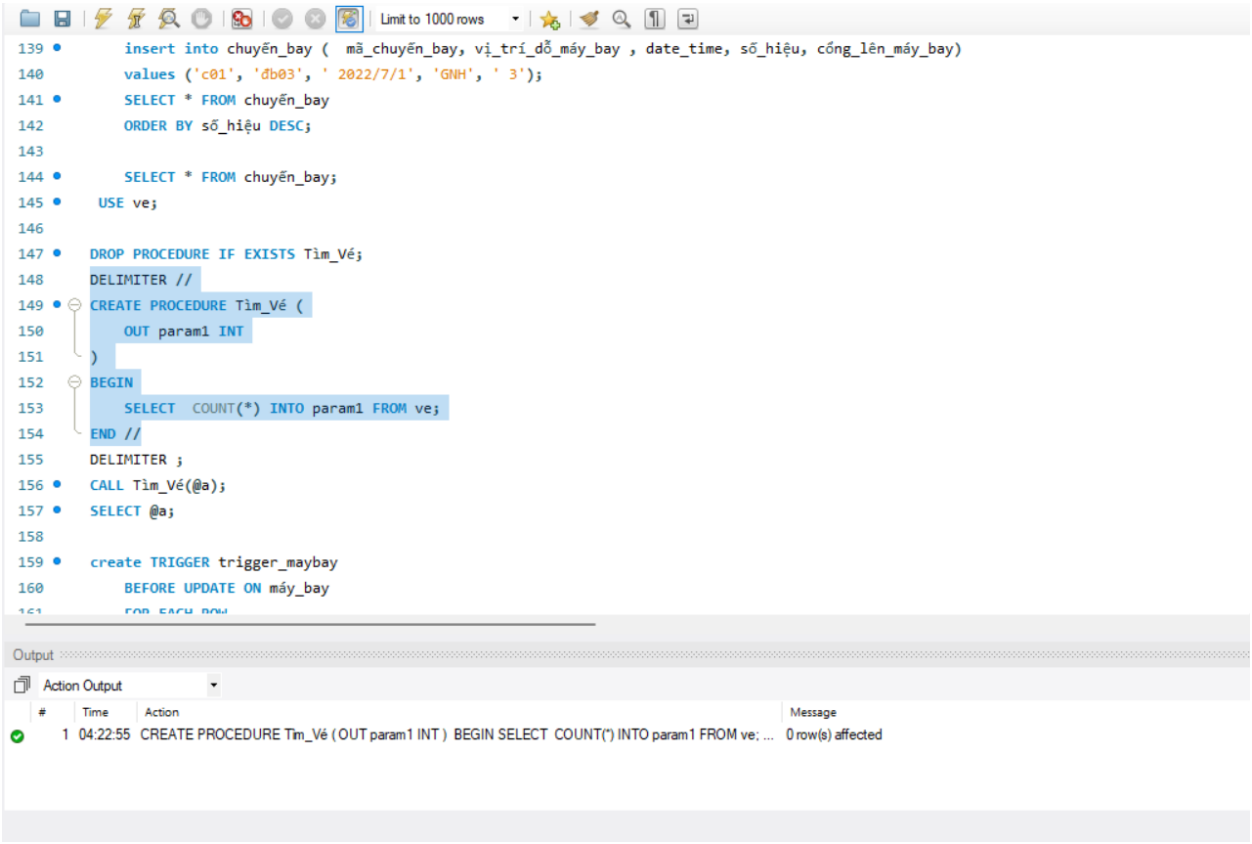
Following are the benefits of a stored procedure:

Reduce the Network Traffic: Multiple SQL Statements are encapsulated in a stored procedure. When you execute it, instead of sending multiple queries, we are sending only the name and the parameters of the stored procedure

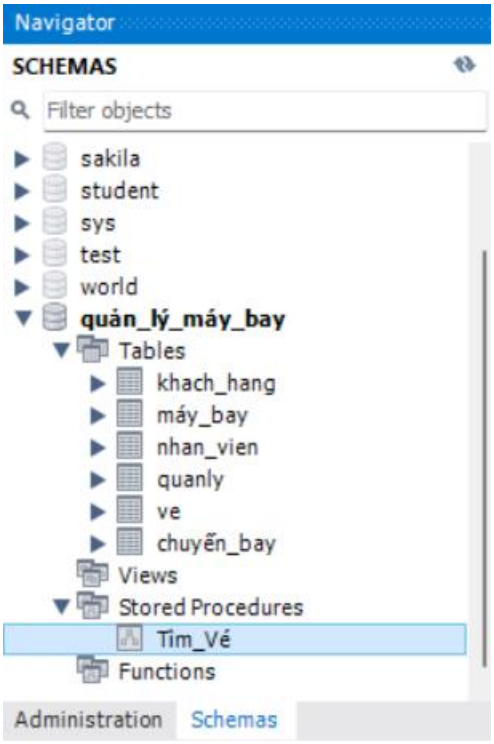
Easy to maintain: The stored procedure are reusable. We can implement the business logic within an SP, and it can be used by applications multiple times, or different modules of an application can use the same procedure. This way, a stored procedure makes the database more consistent. If any change is required, you need to make a change in the stored procedure only

Secure: The stored procedures are more secure than the AdHoc queries. The permission can be granted to the user to execute the stored procedure without giving permission to the tables used in the stored procedure. The stored procedure helps to prevent the database from SQL Injection

EX:



Img-2. create and test Stored Procedures in MySQL



Img-3. stored procedure under the ***Stored Procedures*** folder of the quản_lý_máy_bay schema.

LO3. Test the system against user and system requirements

Test: create a table to test

Description	Step	Expected Result
Check the layout of the account registration screen.	Check UI.	Display the same screen.
Check the registration screen processing.	1. Click the registration button. 2.Enter the full field information correctly. 3.Click button Submit.	Display screen login.
	1. Click the registration button. 2.Enter missing a field information. 3.Click button Submit.	Show error request enter the full field information.
Check the layout of login screen	Check UI.	Show the same screen .
Check the layout login.	3.Click button Submit.	Displaying an error asking to enter full-field information.
Check the layout of home screen.	1.Log in to the system. 2.Check UI.	Display the same screen.
show flight schedule.	1.Log in to the system. 2.Click the search bar.	Show products in the menu .

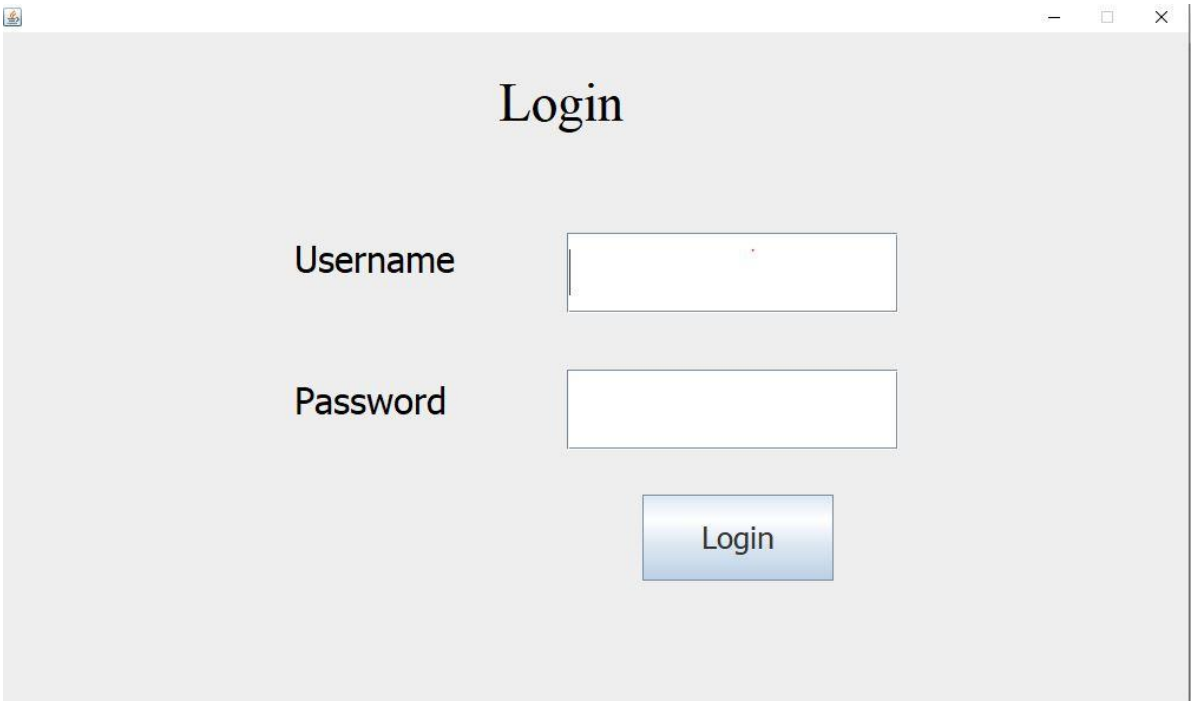
	3.Enter the product name in the menu. 4. Press Enter.	
	1.Log in to the system. 2.Click the search bar. 3.Enter a product name that is not on the menu. 4. Press Enter.	-No products found in the menu.--The screen does not show the Product.
Check processing button SAVE.	1.Log in to the system. 2. choose 1 flight schedule. 3.Click the button ADD TO CARD.	-added to the schedule.
Check cart processing.	1. Log in to the system. 2. choose 1 flight schedule. 3 Click the button search.	- Display the products in the shopping cart - Show the total flight schedule in the cart
Check the layout of booking record.	1.Log 1m to the system. 2. choose 1 flight schedule. 3.Click the button SAVE	Show the screen same.

LO4. Produce technical and user documentation

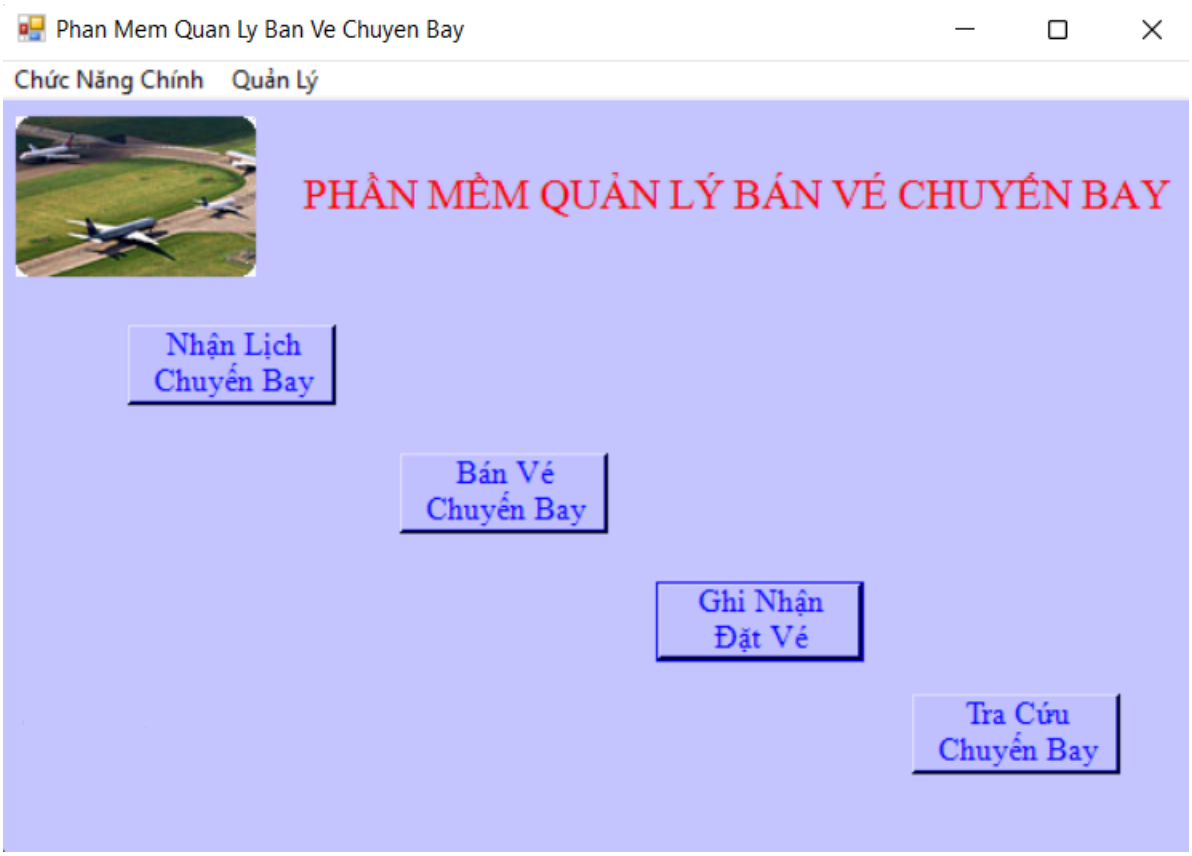
(P5) Produce technical and user documentation.

a. Prepare a simple users' guide and a technical documentation for the support and maintenance of the software.

User's Guide



Img-1. Sign In

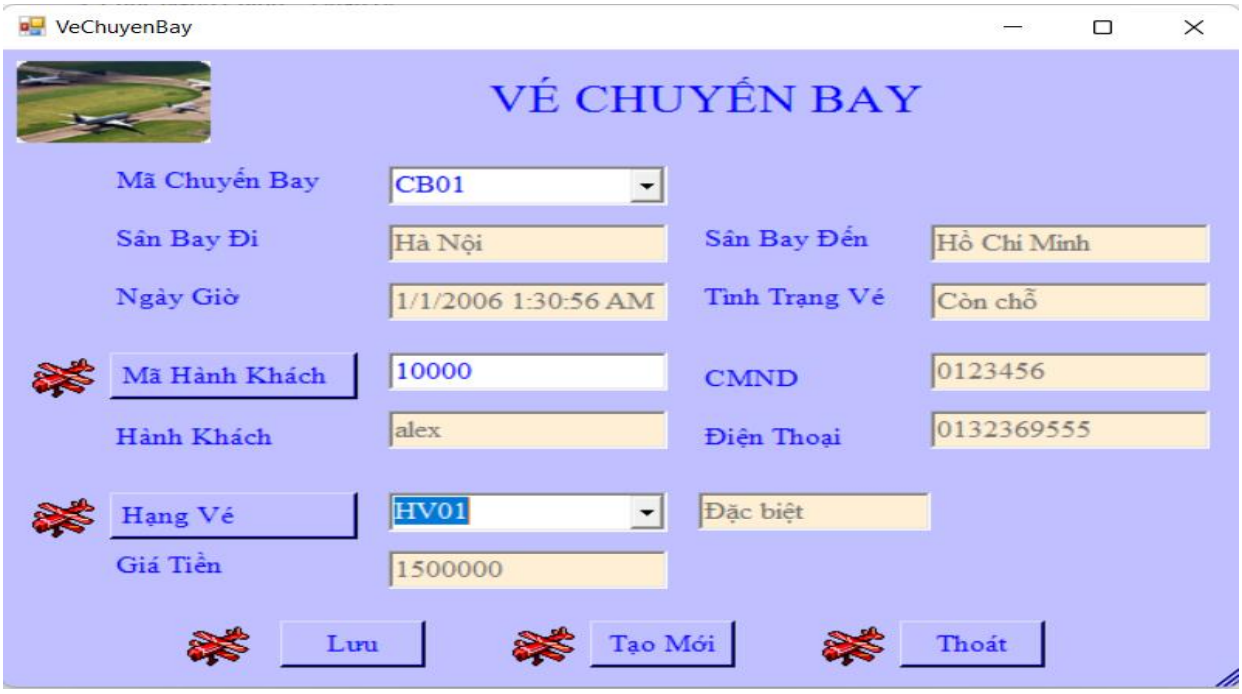


Img-2. Profile

This is the Flight Ticket Manager's Profile. After logging into the system, you will get here.

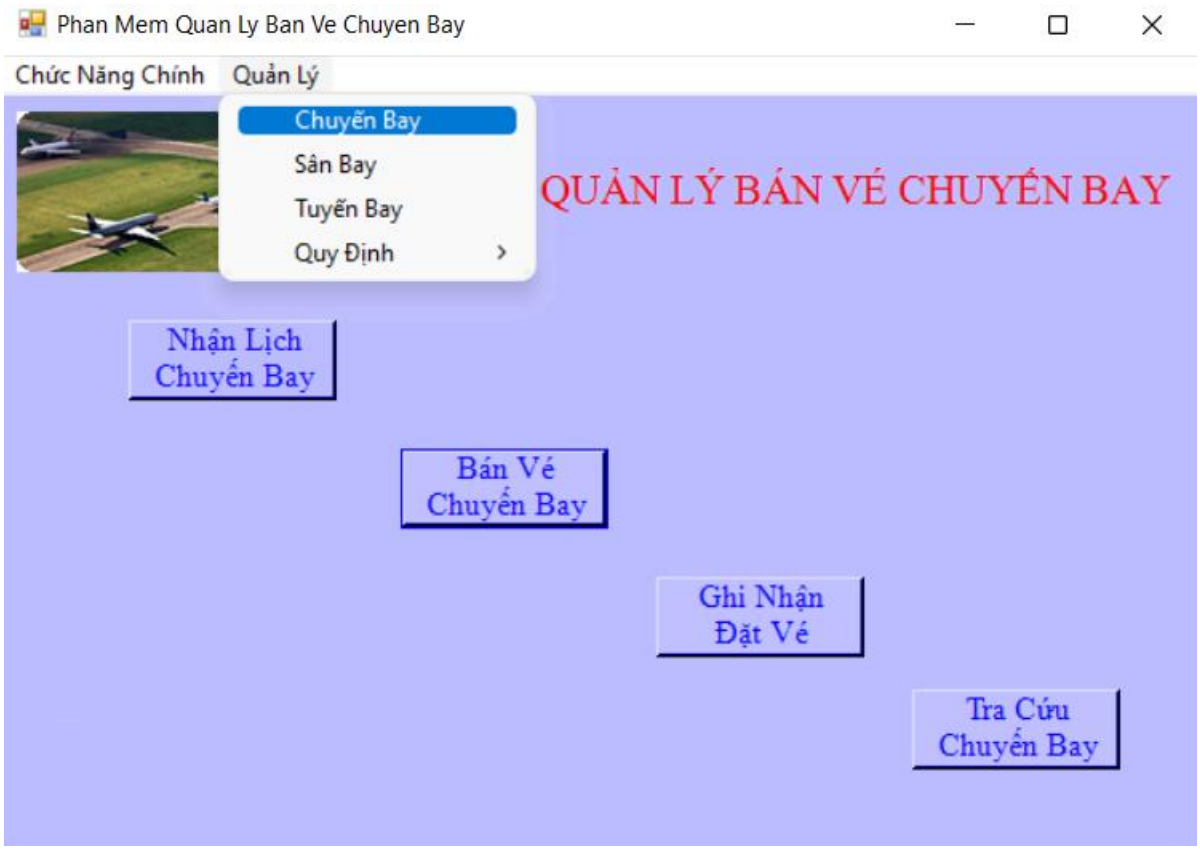
In this page, you can make air ticket sales by clicking on “bán vé chuyến bay” button

Img-3. ticket booking interface

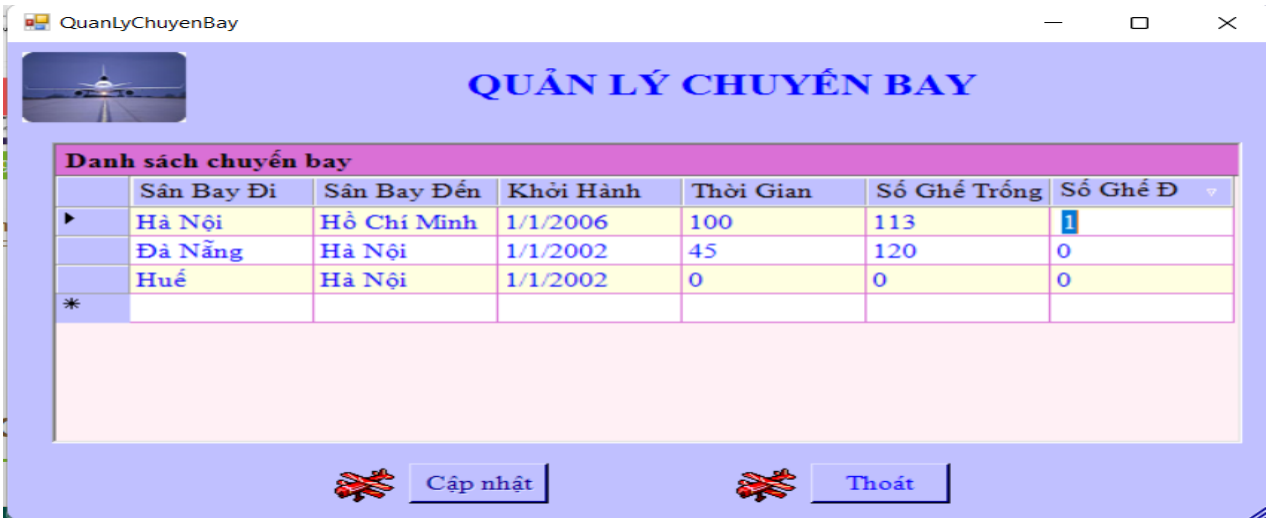


Img-4. book flight tickets

Look up your current flight by clicking “lưu” and view your booking history by clicking on “quản lý” (in the main menu) next clicking on “chuyến bay”.



Img-5. Click “quản lý” next click “chuyến bay”



Img-6. Profile “chuyến bay”

Technical Documentation

To run the program smoothly and for better performance and maintenance following requirements is needed.

For Visual Studio 2017

Table4. 1.Visual Studio 2017

Hardware	• 1.8 GHz or faster processor. Dual-core or better recommended • 2 GB of RAM; 4 GB of RAM recommended (2.5 GB minimum if running on a virtual machine) • Hard disk space: up to 130 GB of available space, depending on features installed; typical installations require 20-50 GB of free space. • Hard disk speed: to improve performance, install Windows and Visual Studio on a solid state drive (SSD). • Video card that supports a minimum display resolution of 720p (1280 by 720); Visual Studio will work best at a resolution of WXGA (1366 by 768) or higher.
Supported Operating Systems	Visual Studio 2017 will install and run on the following operating systems: • Windows 10 version 1507 or higher: Home, Professional, Education, and Enterprise (LTSB and S are not supported) • Windows Server 2016: Standard and Data center

	• Windows 8.1 (with Update 2919355): Core, Professional, and Enterprise • Windows Server 2012 R2 (with Update 2919355): Essentials, Standard, Data center • Windows 7 SP1 (with latest Windows Updates): Home Premium, Professional, Enterprise, Ultimate

Table4. 2. MS SQL Server 2012

Component	**Requirement**
Memory [1]	Minimum: Express Editions: 512 MB All other editions: 1 GB Recommended: Express Editions: 1 GB All other editions: At least 4 GB and should be increased as database size increases to ensure optimal performance.
Processor Speed	Minimum: x86 Processor: 1.0 GHz x64 Processor: 1.4 GHz Recommended: 2.0 GHz or faster
Processor Type	x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support x86 Processor: Pentium III-compatible p

References:

https://www.studocu.com/vn/document/esoft-metro-campus/higher-national-diploma/database-assignment/11737722

https://users.soict.hust.edu.vn/trungtt/uploads/slides/OOP_Bai13(vi).pdf

https://www.cloudflare.com/learning/dns/what-is-dns/

https://www.cloudflare.com/learning/dns/dns-server-types/#:~:text=All%20DNS%20servers%20fall%20into,TLD%20nameservers%2C%20and%20authoritative%20nameservers.