TRƯỜNG ĐẠI HỌC KINH TẾ

**KHOA THỐNG KÊ – TIN HỌC**

–––––––––––––––––––––––––––––––

Ảnh có chứa văn bản

Mô tả được tạo tự động

**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ**

**CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN**

**KIỂM THỬ PHẦN MỀM QUẢN LÝ DƯỢC TẠI BỆNH VIỆN ĐA KHOA GIA ĐÌNH ĐÀ NẴNG**

Sinh viên thực hiện : Nguyễn Tống Khánh Linh

Lớp : 45K21.1

Đơn vị thực tập : Bệnh Viện Đa Khoa Gia Đình ĐN

Cán bộ hướng dẫn : Nguyễn Thị Thanh Mai

Giảng viên hướng dẫn : Ths. Cao Thị Nhâm

**Đà Nẵng, 8/2022**

**NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP**

Họ và tên sinh viên:

Lớp: Khoa: Trường:

Thực tập từ ngày: …./……/ 2022 đến ngày: ........./ .……./ 2022

Tại:

Địa chỉ:

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét, đánh giá như sau:

**1. Về thái độ, ý thức, đạo đức, kỷ luật**

**2. Kiến thức chuyên môn**

**3. Khả năng hòa nhập và thích nghi với công việc**

**4. Trách nhiệm, sáng tạo trong công việc**

**5. Các nhận xét khác**

**Đánh giá chung:**

**Điểm:**

……….., ngày .......tháng ......năm 2022

**Xác nhận của đơn vị thực tập**

# LỜI CẢM ƠN

# LỜI CAM ĐOAN

Quy định:

Soạn thảo trên trang A4 (trang dọc, lề trái: 3.5cm; trên, phải, dưới: 2.5 cm), Font Times New Roman, canh đều 2 bên, size 13, cách dòng 1.5, cách đoạn trên 6pt, cách đoạn dưới 3pt, hàng đâu tiên lùi vào 1.27 cm; hình và bảng soạn theo caption, chèn trích dẫn chéo (Cross-reference) cho bảng và hình; các danh mục hình, bảng, mục lục làm tự động; đánh số trang như file mẫu (bìa không có số trang, danh mục + mục lục số trang theo i, ii, iii…, nội dung chính theo 1,2,3…)

# MỤC LỤC

[NHẬN XÉT CỦA DOANH NGHIỆP THỰC TẬP ii](#_Toc98339732)

[LỜI CẢM ƠN iii](#_Toc98339733)

[LỜI CAM ĐOAN iv](#_Toc98339734)

[MỤC LỤC vi](#_Toc98339735)

[DANH MỤC HÌNH ẢNH viii](#_Toc98339736)

[DANH MỤC BẢNG BIỂU ix](#_Toc98339737)

[DANH MỤC CÁC TỪ VIẾT TẮT x](#_Toc98339738)

[LỜI MỞ ĐẦU 1](#_Toc98339739)

[CHƯƠNG 1. TỔNG QUAN … 2](#_Toc98339740)

[1.1. Mục 1.1 2](#_Toc98339741)

[1.1.1. Mục 1.1.1 2](#_Toc98339742)

[1.1.2. Mục 1.1.2 2](#_Toc98339743)

[1.2. Mục 1.2 2](#_Toc98339744)

[CHƯƠNG 2. LÝ THUYẾT 3](#_Toc98339745)

[2.1. Mục 2.1 3](#_Toc98339746)

[2.1.1. Mục 2.1.1 3](#_Toc98339747)

[2.1.2. Mục 2.1.2 3](#_Toc98339748)

[2.2. Mục 2.2 3](#_Toc98339749)

[CHƯƠNG 3. TRIỂN KHAI … 4](#_Toc98339750)

[3.1. Mục 3.1 4](#_Toc98339751)

[3.1.1. Mục 3.1.1 4](#_Toc98339752)

[3.1.2. Mục 3.1.2 4](#_Toc98339753)

[3.2. Mục 3.2 4](#_Toc98339754)

[CHƯƠNG 4. KẾT QUẢ 5](#_Toc98339755)

[4.1. Mục 4.1… 5](#_Toc98339756)

[4.2. Mục 4.2… 5](#_Toc98339757)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 6](#_Toc98339758)

[TÀI LIỆU THAM KHẢO 7](#_Toc98339759)

[PHỤ LỤC 8](#_Toc98339760)

*(Mục lục này chỉ là ví dụ)*

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Ngôn ngữ lập trình Python 2](#_Toc98336120)

[Hình 4.1 Kiến trúc của mô hình 5](#_Toc98336121)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1 Kiến trúc 5](#_Toc74235471)

# DANH MỤC CÁC TỪ VIẾT TẮT

**AI** : Artificial Intelligence

**…**

# LỜI MỞ ĐẦU

1. **Mục tiêu nghiên cứu của đề tài**

* Đề tài này nghiên cứu …

1. **Nhiệm vụ của đề tài**

* Nghiên cứu ...
* ...

1. **Phương pháp nghiên cứu**

* ...

1. **Đối tượng và phạm vi nghiên cứu**

* ...

1. **Kết cấu của đề tài**

Đề tài được tổ chức gồm phần mở đầu, x chương nội dung và phần kết luận...

* Mở đầu
* **Chương 1**:
* **Chương 2**:
* **Chương 3**:
* **Chương 4**:
* Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT

## Tổng quan các khái niệm cơ bản về kiểm thử phần mềm

### Kiểm thử phần mềm là gì?

Kiểm thử phần mềm là một cuộc kiểm tra được tiến hành để cung cấp cho các bên liên quan thông tin về chất lượng của sản phẩm hoặc dịch vụ được kiểm thử. Cung cấp cho doanh nghiệp một quan điểm, cách nhìn độc lập về phần mềm để từ đó cho phép đánh giá và thấu hiểu rủi ro trong quá trình triển khai phần mềm.

### Mục đích của kiểm thử phần mềm

* Tìm các bug phát sinh do dev tạo ra khi code và ngăn ngừa lỗi
* Phê chuẩn và xác minh một ứng dụng/ sản phẩm:
  + Hoàn thành công việc đúng như kì vọng
  + Đáp ứng được nhu cầu của các bên liên quan:
    - Yêu cầu của doanh nghiệp và người dùng
    - Đáp ứng BRS ( Business Requirement Specification) – đặc tả yêu cầu kinh doanh và SRS( System Requirement Specification) – đặc tả yêu cầu hệ thống.
    - Đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ một sản phẩm chất lượng.

### Vai trò của kiểm thử phần mềm

Kiểm thử phần mềm có vai trò quan trọng trong sự phát triển phần mềm, Kiểm thử bao phủ các lĩnh vực khác nhau như: chức năng các ứng dụng, khả năng tương thích của các ứng dụng với các hệ điều hành, phần cứng và các loại khác nhau của trình duyệt để đảm bảo rằng sẽ không có bất kì vấn đề trục trặc nào.

Kiểm tra xác nhận rằng hệ thống đáp ứng các yêu cầu khác nhau bao gồm: chức năng, hiệu suất, độ tin cậy, an toàn, khả năng sử dụng để có thể tiết kiệm thời gian, tiết kiệm chi phí và đem lại sự hài lòng của khách hàng.

### Người thực hiện kiểm thử (test)

Điều này sẽ phụ thuộc vào quy trình của các bên liên quan đến dự án. Trong ngành công nghiệp phần mềm, thì sẽ có 1 team chuyên chịu trách nhiệm về việc đánh giá phần mềm phát triển với yêu cầu đã được chỉ định – gọi là Tester. Tại bệnh viện thì đội ngũ Tester ấy sẽ được Test trong một chức năng của phần mềm tại Bệnh viện. Vì thế, đối với vấn đề Tester thì có thể là:

* Solfware Tester – nhân viên kiểm thử phần mềm
* Solfware Developer – nhân viên phát triển phần mềm
* Leader hoặc manager dự án
* Product Owner – người sở hữu sản phẩm
* User – người dùng cuối

### Thời điểm kiểm thử phần mềm

Tuỳ vào từng mô hình phát triển phần mềm mà thời gian thực hiện test là khác nhau. Thực hiện kiểm thử thì càng sớm càng tốt. Code xong thì Dev sẽ build và bàn giao cho Tester thực hiện test.

Thực tế thì tester tham gia sớm hơn, chỉ cần có tài liệu Đặc tả Yêu cầu/ Nghiệp vụ là tester thực hiện tìm hiểu nghiệp vụ dự án và thực hiện viết Testcase

## Quy trình kiểm thử phần mềm

### Lập kế hoạch kiểm thử (test plan)

Kế hoạch kiểm thử là một tài liệu mô tả các mục tiêu, phạm vị, phương pháp, tiếp cận và tập trung vào nỗ lực kiểm thử phần mềm.

* Cấu trúc chung của một test plan:
  + Tên Project
  + Danh sách module cần test
  + Ngày bắt đầu, ngày kết thúc
  + Nhân sự tham gia ( Tham gia cuộc họp với khách hàng)
  + Kế hoạch thực hiện ( sử dụng Excel lập kế hoạch)

Khi viết test plan thì nó sẽ giúp chúng ta định hướng suy nghĩ, là phương tiện giao tiếp của nhóm test với đội dự án và công cụ giao việc trong nhóm test.

### Viết test case

Testcase là một tập hợp các thông số đầu vào kiểm thử, điều kiện thực thi và kết quả mong đợi được phát triển cho một mục tiêu cụ thể. Một trường hợp kiểm thử có thể chỉ đơn giản là một câu hỏi cho chương trình. ***(Theo: Wikipedia.vn)***

Một trường hợp kiểm thử có thể có các phần đặc thù khác nhau như: Mã Testcase, tên test case, Sub-items (mục con của mục mình test), Steps to excute (các bước thực hiện), Expected out put (KQ mong muốn) và kết quả thực tế.

Nó bao gồm 3 bước cơ bản:

* Mô tả: Đặc tả các điều kiện cần có để tiến hành kiểm tra
* Nhập: Đặc tả đối tượng hoặc dữ liệu cần thiết để thực hiện kiểm tra
* Kết quả mong muốn: Trả về từ đối tượng kiểm tra.

### Thực hành test

Việc test plan và viết testcase đã xong thì thực hành test sẽ được diễn ra dựa trên testcase đã viết. Đánh giá kết quả kiểm thử cho các trường hợp kiểm thử. So sánh kết quả thực tế khi thực thi với kết quả mong muốn.

### Viết báo cáo kiểm thử (test report)

Báo cáo kiếm thử thể hiện tiến độ kiểm thử, tiến độ sửa lỗi và số lượng lỗi được tìm thấy hay còn tồn của dự án. Nó còn đánh giá và giám sát xem dự án có kịp tiến độ để bàn giao cho khách hay không.

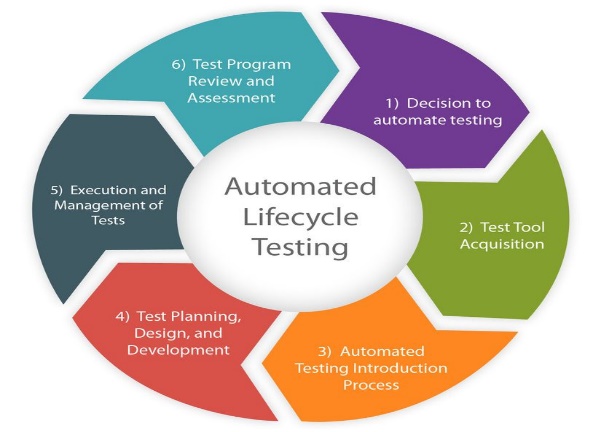
## Các loại kiểm thử phần mềm ( Software Testing)

### Kiểm thử thủ công: Manual testing



* Manual Testing không sử dụng bất kì công cụ tự động hoặc script
* Đảm nhận vai trò của người dùng cuối, xác định hành vi và lỗi không mong muốn

### Kiểm thử tự động: Automation testing



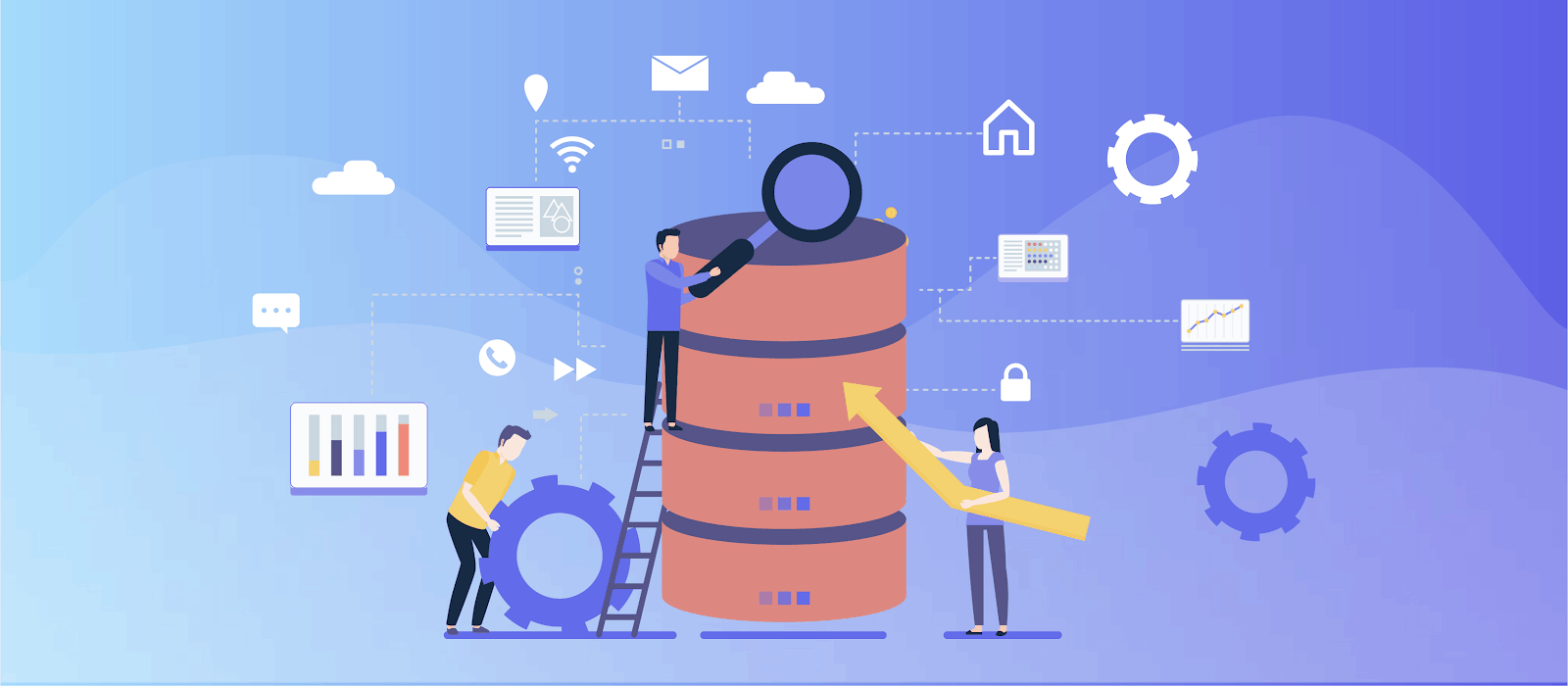
* Automation testing liên quan đến tự động hoá quy trình thủ công, thực hiện nhanh chóng hơn
* Nó làm tăng phạm vi kiểm tra, cải thiện độ chính xác và tiết kiệm thời gian và tiền bạc so với kiểm tra thủ công

## Giới thiệu về SQL

### Tổng quan về Database và sự ra đời của SQL

#### Tổng quan về Database

Data là thông tin dữ liệu gắn kèm, xung quanh 1 đối tượng nào đó, ví dụ như: data cá nhân bao gồm: họ tên, giới tính,….. Hoặc đối với data của sinh viên bao gồm: họ tên, quê quán, mã sinh viên



Database (cơ sở dữ liệu) là một tập hợp những data (dữ liệu) có liên quan với nhau. Nó được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành được lưu trữ trong các hệ quản trị cơ sở dữ liệu ***(Theo Wikipedia.vn)***

Database là kho lưu trữ dữ liệu, sẽ hỗ trợ truy xuất và quản lý dữ liệu được một cách dễ dàng, cách thức phụ thuộc vào module quan hệ.

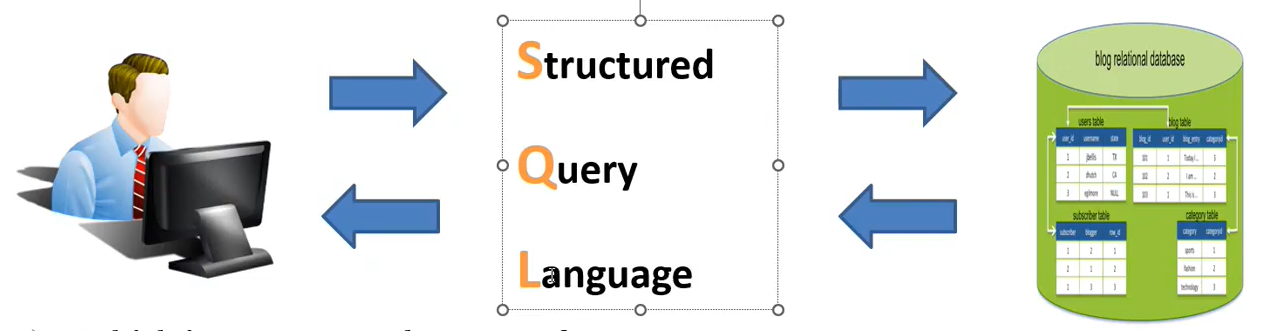
Relational Database (RDB) được lưu trữ ở dạng bảng, giữa các bảng thì sẽ có quan hệ tham chiếu với nhau. Đối với cách thức lưu trữ dữ liệu thì sẽ có 4 cách: Phân cấp, network, quan hệ, hướng đối tượng.

* Quan hệ: Cách thức lưu trữ dữ liệu này bao gồm 12 rules nó sẽ đưa ra khái niệm sơ khai mô hình cách thức lưu trữ database theo dạng quan hệ.
* Hướng đối tượng: Thường sẽ dùng NoSQL để truy cập và quản lý dữ liệu.

Relational Database Management System(RDBMS) được gọi là hệ quản trị cơ sở dữ liệu, là các công cụ, các tool hỗ trợ quản lý và lưu trữ RDB: SQL Server, MS Access, MySQL, Oracle DB, IBM DB2,…

#### Sự ra đời của SQL

Vậy Database làm như thế nào để truy xuất và quản lý dữ liệu thì SQL ra đời phục vụ điều đó. SQL hay còn gọi là Structured Query Language đơn thuần là ngôn ngữ giao tiếp giữa user và RDB.



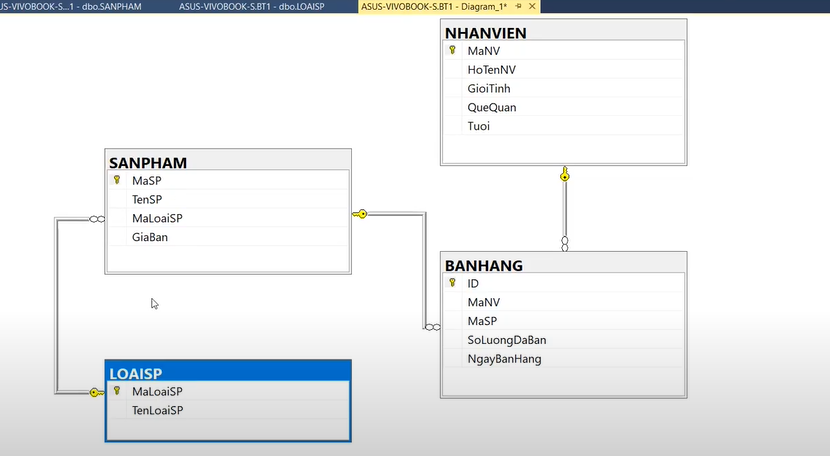
#### Các nhóm lệnh cơ bản

|  |  |  |
| --- | --- | --- |
| **Nhóm lệnh** | **Lệnh** | **Định nghĩa** |
| Data Control Language  DCL | Grant  Revole | Ngôn ngữ điều khiển dữ liệu cấp quyền hoặc huỷ quyền của người dùng trên đối tượng của cơ sở dữ liệu |
| Data Manipulation Language  DML | Delete  Insert  Select  Update | Ngôn ngữ thao tác dữ liệu dùng để thay đổi dữ liệu có trong các bảng của cơ sở dữ liệu |
| Data Definition Language  DDL | Create  Alter  Drop | Ngôn ngữ định nghĩa dữ liệu tạo cấu trúc, tạo mối quan hệ ràng buộc giữa các bảng với nhau. |

#### Ràng buộc dữ liệu

Các ràng buộc dữ liệu dùng để kiểm tra sự đúng đắn và tính toàn vẹn của dữ liệu. Ràng buộc gồm các loại sau:

* PK = Primary Key = Khoá định danh = Khoá chính:
  + Không được phép trùng nhau
  + Không được phép NULL
  + Có thể tạo thành từ 1 hoặc nhiều cột (sẽ có trường hợp trong 1 bảng có nhiều cột kết hợp với nhau để tạo thành khoá chính)
  + Một bảng chỉ có duy nhất 1 khoá chính
* FK = Foreign Key = Khoá tham chiếu = Khoá ngoại:
  + Nhập data có tồn tại ở khoá chính mà nó đang tham chiếu
  + Có thể trùng lặp
  + Được phép NULL (nếu cột đó đang khai báo là Allow Null)
  + Một bảng chỉ có thể có nhiều khoá ngoại



* CHECK: Dùng để kiểm tra dữ liệu đầu vào của 1 cột xác định

#### Các kiểu dữ liệu cơ bản

Trong SQL thì có 3 loại dữ liệu chính bao gồm: Loại kiểu số (dùng trong mục đích tính toán), ký tự (biểu diễn thông tin) và thời gian (ngày, giờ)

##### String data

* char(length): kiểu ký tự với độ dài cố đinh, lưu trữ kí tự dưới dạng ASCII
* varchar(length): kiểu ký tự với độ dài thay đổi, có thể lấy dữ liệu được nằm bên trong nó, lưu trữ kí tự dưới dạng ASCII
* nchar(length): kiểu ký tự với độ dài cố định và có thể lưu trữ ở các ký tự Unicode
* nvarchar(length): kiểu ký tự với độ dài thay đổi và có thể lưu trữ ở các ký tự Unicode

\*Với length là số lượng ký tự tối đa.

* Trường hợp là ký tự bình thường(char,varchar) thì phải nằm trong khoảng(1,8000)
* Trường hợp là ký tự Unicode(nchar,nvarchar) thì phải nằm trong khoảng (1,4000)

##### Numeric data

* Kiểu số nguyên:

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Size** | **Range of values** |
| Bigint | 8 bytes | -2^63 to 2^63-1 |
| Int | 4 bytes | -2^31 to 2^31-1 |
| Smallint | 2 bytes | -2^15 to 2^15-1 |
| Tinyint | 1 bytes | 0 -> 255 |
| Bit | 1 bytes | 1. và 1 |

* Kiểu số thực(Exact Numbers/số chính xác):

|  |  |
| --- | --- |
| **Data Type** | **Size** |
| Decimal(p,s) | 5 -> 17 bytes |
| Numeric(p,s) | 5 -> 17 bytes |
| Smallmoney | 4 bytes |
| Money | 8 bytes |

* + p: số chữ số tối đa của cả phần nguyên và phần thập phân
  + s: số chữ số tối đa của phần thập phân
* Kiểu số thực (gần đúng):

|  |  |
| --- | --- |
| **Data Type** | **Size** |
| Float | 8 bytes |
| Real | 4 bytes |

##### DateTime data

|  |  |
| --- | --- |
| **Data Type** | **Mô tả** |
| Date | Ngày – tháng – năm |
| Datetime | Ngày – tháng – năm – giờ - phút - giây. Có thêm 3 kí tự thập phân đằng sau giây |
| Datetime2 | (tương ứng với Timestamp) Ngày – tháng – năm – giờ - phút – giây. Giây của nó tới nanos |
| Datetimeoffset | Ngoài ngày giờ ra còn có múi giờ |
| Smalldate time | Ngày – tháng – năm – giờ - phút – giây và không có kí tự thập phân |
| Time | Chỉ có giờ tới nanos |

#### Các phép toán tử

##### Toán tử tính toán

Các phép toán học bao gồm: +, -, \*, /. Các phép toán này thường được dùng ở mệnh đề SELECT, WHERE, HAVING.

##### Toán tử logic

|  |  |
| --- | --- |
| **Toán tử** | **Mô tả** |
| AND | Toán tử AND cho phép nhiều điều kiện cùng tồn tại |
| ALL | Toán tử ALL được sử dụng để so sánh một giá trị với tất cả các giá trị trong tập hợp giá trị khác. |
| OR | Toán tử OR thường được sử dụng để nối nhiều điều kiện |

##### Toán tử so sánh

Phép so sánh gồm: >, >=, <, <=, = và <>(!=). Các phép toán này thường được dùng ở mệnh đề WHERE hoặc HAVING, dùng để giới hạn kết quả trả về. Kết quả trả về phép so sánh là True/False.

### Các lệnh DDL

#### Lệnh tạo bảng (CREATE TABLE)

##### Quy tắc đặt tên bảng:

* Không quá 50 ký tự
* Tên bảng bắt đầu bằng chữ cái
* Tên bảng/cột không dùng những cụm từ đã có sẵn trong SQL Server
* Tên cột phải duy nhất trong bảng

##### Cú pháp tạo Database:

Create Database mydb5

Use mydb5;

##### Cú pháp tạo Table: Trực tiếp lưu trữ thông tin

Create Table tintuc(

tieude varchar(50),

noidung varchar(500))

#### Lệnh xoá bảng (DROP TABLE)

Drop Table tintuc

#### Lệnh sửa bảng (ALTER TABLE)

##### Sửa lại và thêm cột

Alter Table tintuc

Add nguoidang varchar(30);

##### Sửa lại và xoá cột

Alter Table tintuc

Drop column nguoidang varchar(30);

### Các lệnh DML

#### Lệnh Insert

Cú pháp:

Insert into tên\_bảng (cột1, cột2, ...) values (gt1, gt2, ...)

#### Lệnh Update

Cú pháp:

Update tên\_bảng

Set cột 1=gt1, cột 2=gt2

Where điều\_kiện

#### Lệnh Delete

* Cú pháp

Delete From tên\_bảng

Where điều\_kiện

#### Lệnh Select

# TRIỂN KHAI …

## Mục 3.1

### Mục 3.1.1

### Mục 3.1.2

## Mục 3.2

# KẾT QUẢ

## Mục 4.1…

## Mục 4.2…

Kết quả được xây dựng dựa trên ngôn ngữ lập trình thể hiện như (Bảng 4.1).

Bảng 4.1 Kiến trúc

|  |  |  |
| --- | --- | --- |
| Lớp | Ý nghĩa | Tham số |
|  |  |  |
|  |  |  |
|  | | |

Hình 4.1 Kiến trúc của mô hình

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Đề tài đã thực hiện được …

# TÀI LIỆU THAM KHẢO

* + - 1. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang
      2. Tên tác giả, Tên tác giả… *Tên công trình*. Nơi xuất bản, năm, trang

# PHỤ LỤC