

DATA BASE

P' Anfield



P' N.I.



Purpose

- ความหมายของฐานข้อมูล (Database)
- ประโยชน์ของฐานข้อมูล
- ส่วนประกอบของ Table
- การอ่านแบบ ER Diagram เป็นต้น

What is Data ?

- ข้อมูล คือ ข้อเท็จจริงของสิ่งที่เราสนใจ ข้อเท็จจริงที่เป็นตัวเลข ข้อความ หรือรายละเอียด ซึ่งอาจอยู่ในรูปแบบต่าง ๆ เช่น ภาพ เสียง วิดีโอ ไม่ว่าจะเป็นคน สัตว์ สิ่งของ หรือเหตุการณ์ที่เกี่ยวข้องกับสิ่งต่าง ๆ
- ข้อมูลเป็นเรื่องเกี่ยวกับเหตุการณ์ที่เกิดขึ้นอย่างต่อเนื่อง และต้องถูกต้อง แม่นยำ ครบถ้วน
- ข้อมูลจึงหมายถึงตัวแทนของข้อเท็จจริง หรือความเป็นไปของสิ่งของ ที่เราสนใจ

What is a Database ?

- ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ ซึ่งกันและกัน ไม่ได้บังคับว่าข้อมูลก็ต้องหมุนเวียนกัน แต่สามารถเพิ่มข้อมูลเดียว ก็สามารถเพิ่มข้อมูลอื่นๆ ได้โดยอัตโนมัติ
- ฐานข้อมูลสามารถทำงานร่วมกันกับภาษาคอมพิวเตอร์ได้หลายภาษา อาทิ C, C++, Python, Java และอื่นๆ อีกมาก many



DBMS

- DBMS (Database Management System) หรือ ระบบจัดการฐานข้อมูล
- มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่าย สะดวก และมีประสิทธิภาพ
- ตัวอย่าง MySQL, Oracle



ประโยชน์จากการจัดเก็บข้อมูลให้เป็นแบบ Database

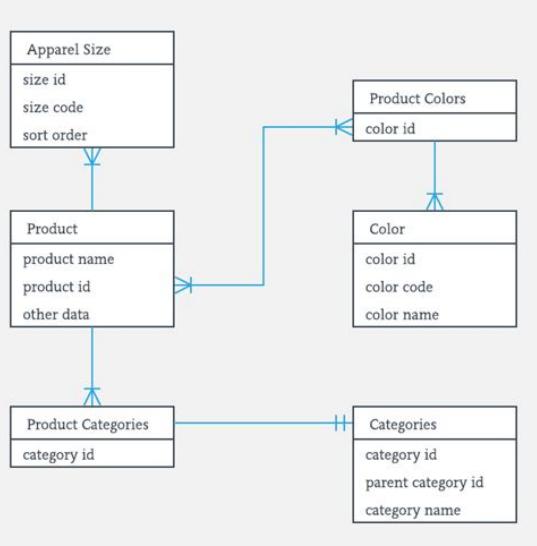
- สามารถลดความซ้ำซ้อนของข้อมูลได้
- หลักเลี้ยงความขัดแย้งของข้อมูลได้
- สามารถใช้ข้อมูลร่วมกันได้
- สามารถรักษาความถูกต้อง และความน่าเชื่อถือของข้อมูลได้
- สามารถกำหนดความเป็นมาตรฐานเดียวกันของข้อมูลได้
- สามารถกำหนดระบบความปลอดภัยของข้อมูลได้

ตาราง (Table)

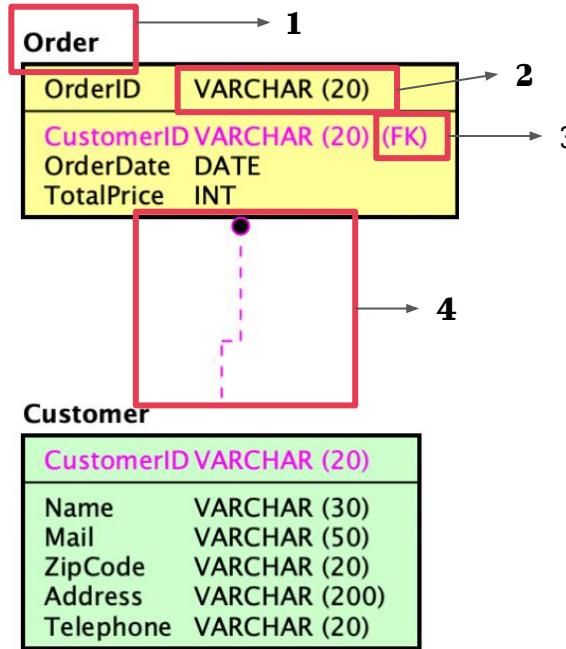
เป็นส่วนประกอบหลักของระบบเก็บข้อมูล โดยข้อมูลจะถูกจัดเก็บในรูปแบบตาราง ประกอบไปด้วยแຄวแนวตั้ง เรียกว่า **Columns (Fields)** และ แຄวแนวนอน เรียกว่า **Rows (records)**

	Columns (Fields)			
Rows(Records)	STUDENTID	FIRSTNAME	LASTNAME	NICKNAME
	63130599999	John	Samit	John
	64130588888	Muukrop	Khaoniao	Muu

ER DIAGRAM



ER Diagram (Entity Relationship Diagram)
เป็นไดอะแกรมหรือแผนภาพที่แสดงถึง^{รายละเอียดต่าง ๆ ภายในตาราง (Table)}
และแสดงถึงการรวมของ Database



(1) ชื่อตาราง

(2) Datatype

(3) ตัวระบุ Key จะเรียกว่า Constraint

(4) เส้นระบุความสัมพันธ์ของตาราง

DATA TYPES

P' Anfield



P' Nitt

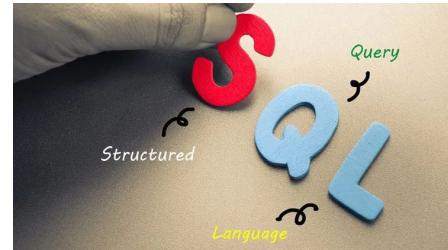


Purpose

- ความหมายของ SQL
- ชนิดข้อมูลประเภทต่าง ๆ
 - String, Number, Date & Time
- Constraints

SQL (Structured Query Language)

- เป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล (Database) เราสามารถใช้ภาษา SQL ได้จากโปรแกรมต่าง ๆ ที่ต้องทำงานร่วมกับระบบฐานข้อมูล



Data Types

String - ตัวอักษร

- CHAR(n)
- VARCHAR(n)

Number - ตัวเลข

- INT, INTEGER
- DECIMAL(n, d)
- NUMERIC

Date & Time

- DATE
- TIME
- DATETIME
- TIMESTAMP

NOTE : (n) = ขนาดของข้อมูลที่ต้องการกำหนด

(n, d) n = ขนาดหรือจำนวนหลัก, d = จำนวนตำแหน่งของกศนิยม

String - ຕັວອັກບົດ

- **CHAR(n)** - ສໍາຮັບເກີບຂ້ອນມູລປະເກດຕັວອັກບົດ ແລະຕັວເລຂທີ່ໄມ້ໄດ້ ເວົາໄວ້ໃຊ້ຄຳນວນ ໂດຍສາມາດເກີບຂ້ອນມູລທີ່ມີຄວາມຍາວສູງສຸດ 255 ຕັວອັກບົດ ເປັນຂ້ອນມູລແບບທີ່ຄູກຈຳກັດຄວາມກວ້າງເຈາໄວ້ ຜຶ່ງມີບັນດາດທີ່ແນ່ນອນເຫັນ
 - **ຮັສບັດປະຈຳຕັວປະເຈົນ** ທີ່ເປັນຕັວເລຂທີ່ມີກາຣກຳຫຼັດຂັດໄວ້ 13 ກລັກອຍ່າງໜັດເຈນ ແລະໄມ້ໄດ້ນຳໄປໃຊ້ໃນກາຣຄຳນວນ

String - ຕັວອັກບົດ

CHAR(n)

ຂ້ອດີ : ທຳໄໝມໍຄວາມມັນຄງໃນການເກີບຂ້ອມຸລ

ຂ້ອເສີຍ : ໄມຍືດຫຍຸ່ນ

ຕັວຢ່າງ ຂ້ອມຸລ ຮັບສັດຕະປະຈຳຕັວປະຊາບ ກີ່ຖຸກຄນຕ້ອງມີຄຣບກັ້ງ

13 ພລັກ ສາມາດເບີຍນອກມາໄດ້ເປັນ **CHAR(13)**

String - ຕັວອັກບົຣ

- **VARCHAR(n)** - ສໍາຮັບເກີບຂ້ອມຸລປະເກດຕັວອັກບົຣ ແລະຕັວເລກທີ່
ໄຟໄດ້ເອົາໄວ້ໃຊ້ຄຳນວນ ໂດຍສາມາດຄຳກຳຫຼັດຄ່າໄດ້ຕັ້ງແຕ່ 1 – 255 ຕັວອັກບົຣ
ຊື່ເໝາະກັບການເກີບຂ້ອມຸລທີ່ມີຂະໜາດສັ້ນ-ຍາວໄຟເກົ່າກັນ ເຊັ່ນ
 - ຜົ່ອ, ນາມສກຸລ ເພຣະ ຖຸກຄນມີຄວາມຍາວຂອງຜົ່ອແລະນາມສກຸລໄຟເກົ່າກັນ
 - ຜົ່ອສຄານທີ່

String - ຕັວອັກເປດ

VARCHAR(n)

ຂ້ອດ : ມີຄວາມຍືດຫຍຸນ

ຂ້ອເສີຍ : ອາກກຳແນດຂນາດໄວ້ເກີນຄວາມຈຳເປັນ ຈະກຳໃຫ້ເປົ້າລືອງພື້ນທີ່ຈັດເກີບ

ຕັວຢ່າງ ຂ້ອມູລ ທີ່ເນື້ອງກລວງ ເຊັ່ນ BANGKOK, PARIS, SEOUL

ຄວຣໃຊ້ໜົດຂ້ອມູລເປັນ **VARCHAR(7)** ມີມາກກວ່າ ເພຣະ ຂນາດ
ຂ້ອມູລສູງສຸດຍູ້ກໍ່ 7 ຕັວອັກເປດ

Question ?



แล้ว CHAR(n) ต่างกับ VARCHAR(n) อย่างไร?





Answer

CHAR(n) – เหมาะสำหรับข้อมูลที่มีขนาดและความยาวที่ชัดเจน

VARCHAR(n) – จะยืดหยุ่นกว่า เพราะสามารถเก็บข้อมูลได้
ตามที่กำหนดไว้หรือน้อยกว่าก็ได้

Number - ตัวเลข

- **INT | INTEGER** - สำหรับเก็บข้อมูลตัวเลข ที่มีขนาด 32 bits
ไว้สำหรับการคำนวณ และใช้ในการดำเนินการทางตรรกศาสตร์
(ตรวจสอบ จริง หรือ เก็จ) เช่น
 - ตัวแปร
 - อายุ

Number - ตัวเลข

- **DEC(n, d) | DECIMAL(n, d)** - สำหรับเก็บข้อมูลตัวเลขที่เป็นทั้งจำนวนเต็ม และกศนิยม เช่น
 - เกรดเฉลี่ย (GPA)
 - ห้าหนักตัว, ส่วนสูง
 - ราคาสินค้า

Number - ตัวเลข

ตัวอย่าง

เกรดเฉลี่ย 3.75

g = ขนาดข้อมูล จะมีค่าเท่ากับ 3

d = จำนวนตำแหน่งของคันบัญชี จะมีค่าเท่ากับ 2

จะเขียน Data Type ของเกรดเฉลี่ยออกมาได้เป็น **DECIMAL(3, 2)**

หรือ **DEC(3, 2)**

Question



มีข้อมูล ส่วนสูง = 150.5 cm. และ น้ำหนัก 40.5 kg.
ข้อใดเลือกใช้ *Data Types* ได้ถูกต้อง ตามลำดับ ?

a) DECIMAL(4, 1), DECIMAL(3, 1)

b) DEC(1, 4), DEC(1, 3)

c) DEC(4, 1), DEC(3, 1)

d) ถูกก็งข้อ a และ c



Answer

a) DECIMAL(4, 1), DECIMAL(3, 1)

b) DEC(1, 4), DEC(1, 3)

c) DEC(4, 1), DEC(3, 1)

d) ถูกกึ่งข้อ a และ c

Date & Time

- **DATE** - สำหรับเก็บข้อมูลประเภทวันที่ โดยจะมีรูปแบบดังนี้ 'YYYY-MM-DD'
- **TIME** - สำหรับเก็บข้อมูลประเภทเวลา โดยจะมีรูปแบบดังนี้ 'hh:mm:ss'

NOTE : YYYY = ปี(ค.ศ.), MM = เดือน, DD = วันที่
hh = ชั่วโมง, mm = นาที, ss = วินาที

Date & Time

- **DATETIME** - สำหรับเก็บข้อมูล วันที่และเวลา โดยจะมีรูปแบบเป็น 'YYYY-MM-DD hh:mm:ss'
 - ช่วงวัน-เวลาที่เก็บจะอยู่ในช่วง **1000-01-01 00:00:00** ถึง **9999-12-31 23:59:59**

Date & Time

- **TIMESTAMP** - สำหรับเก็บ วันที่และเวลา ซึ่งจะเก็บอยู่ในรูปแบบของ 'YYYY-MM-DD hh:mm:ss'
 - ช่วงวัน-เวลาที่เก็บจะอยู่ในช่วง **1970-01-01 00:00:00** ถึง **2037-12-31 23:59:59**

Constraints

คือ คุณสมบัติที่ Column ในแต่ละตารางนั้นต้องมีเพื่อให้เวลาจัดการข้อมูลมีความถูกต้องและน่าเชื่อถือในการใช้งาน ซึ่งคุณสมบัติมีดังนี้

- Primary Key
- Foreign Key
- Unique Key
- Not Null

1. Primary Key

คือ การเลือกให้ Column ใด Column หนึ่งมีการเก็บข้อมูลที่ไม่ซ้ำกัน ในแต่ละแถว เช่น Column รหัสสินค้า, รหัสบัตรประจำตัวประชาชน เป็นต้น



วัตถุประสงค์ของ Primary key (PK)

ทำให้ข้อมูลไม่ซ้ำกันในแต่ละแถวที่จัดเก็บในตาราง สามารถแยกข้อมูลออกจากกันได้เพื่อประโยชน์ในการบริหารจัดการฐานข้อมูล และการสืบค้น รวมไปถึงการเขียนโปรแกรมร่วมกับฐานข้อมูล

Constraints

คุณสมบัติของ Primary key (PK)

- ห้ามมีข้อมูลซ้ำกัน
(UNIQUE)



Student Table

StudentId (PK)	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500108	Josh	Disney	Josh
65130500082	Palo	Escobar	Palo
65130500082	Elon	Musk	Reeve

*ตามตารางเส้นขอบสีแดง แสดงถึงค่าที่ซ้ำ 2 record ขึ้นไป

Constraints

คุณสมบัติของ Primary key (PK)

- ห้ามมีค่าว่างในข้อมูล
(NOT NULL)

Student Table

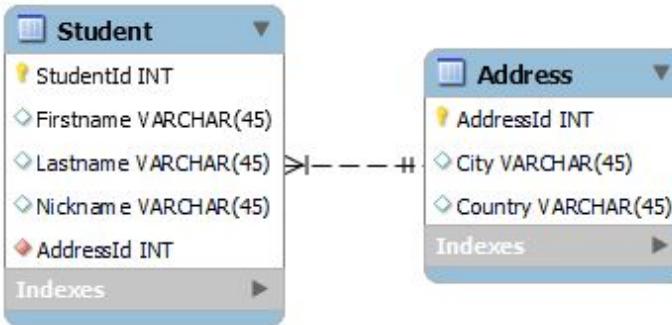
StudentId (PK)	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500108	Josh	Disney	Josh
65130500082	Palo	Escobar	Palo
	Elon	Musk	Reeve

*ตามตารางเส้นขอบสีแดง แสดงถึงค่าที่ไม่ได้ใส่ในช่อง StudentId

2. Foreign key (FK)

คือ การกำหนด Column เพื่อสร้างความสัมพันธ์กับตารางอื่น ๆ ซึ่งจะทำให้เราสามารถ เชื่อมข้อมูลกับตารางอื่น ๆ ได้ และทำให้ข้อมูลในฐานข้อมูลมีความสอดคล้องกัน

ตัวอย่าง



ตารางของ **Student** และ **Address** มีความสัมพันธ์ได้ว่า **ข้อมูล Student** นี่ อยู่ที่ **Address** ไหน เพราะบางคนก็อาศัยอยู่ที่เดียวกัน เพื่อให้ง่ายต่อการอ่านแบบตาราง จึงใช้ **Foreign key**

2. Foreign key (FK)

ตารางที่ไม่ใช้ Foreign key ในการหาข้อมูล

Student table

StudentId (PK)	Firstname	Lastname	Nickname	City	Country
65130500065	Michael	Bane	Mickie	Thung Khru	Thailand
65130500108	Josh	Disney	Josh	Bangkuntian	Thailand
65130500082	Palo	Escobar	Palo	Thung Khru	Thailand
65130500999	Elon	Musk	Reeve	Bang Bon	Thailand

2. Foreign key

ถ้าหากแยกข้อมูลออกมาเป็น 2 ตาราง ก็จะใช้ **Foreign key** ของตาราง Student ในการเชื่อมต่อกับ Primary key ของตาราง Address

Student table

StudentId (PK)	Firstname	Lastname	Nickname	Address Id (FK)
65130500065	Michael	Bane	Mickie	1
65130500108	Josh	Disney	Josh	2
65130500082	Palo	Escobar	Palo	1
65130500999	Elon	Musk	Reeve	3

Address table

Address Id (PK)	City	Country
1	Thung Khru	Thailand
2	Bangkuntian	Thailand
3	Bang Bon	Thailand

3. Unique Key (UK)

Student Table

คือ คำสั่งที่กำหนดให้ Field
 ใน ๑ ก็แล้วแต่ ไม่สามารถมีค่าซ้ำ
 กันได้ แต่สามารถเป็น ค่าว่าง
 เปล่าได้ ซึ่งต่างจาก PK หรือ
 Primary Key ที่ห้ามมีค่าซ้ำ และ
 ห้ามเป็นค่าว่างเปล่า

ตัวอย่างเช่น Record ของ
 Student Id 65130500082
 ซ้ำกับ 2 Record ทำให้ข้อมูลไม่
 Unique กัน

StudentId (UK)	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500108	Josh	Disney	Josh
65130500082	Palo	Escobar	Palo
65130500082	Elon	Musk	Reeve

*ตามตารางเส้นขอบสีแดง แสดงถึงค่าที่ซ้ำ 2 record ขึ้นไป



4. Not Null (NN)

Student Table

คือ ห้ามเป็น Null (ค่าว่าง
เปล่าหรือค่าที่หาไม่เจอนใน
Record)

ตัวอย่างเช่น Record ของ
Student Id 65130500999 ไม่
มีชื่อเล่น ดังนั้น ค่า NULL ก็จะอยู่
บนชื่อเล่นของ 65130500999

StudentId (UK)	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500108	Josh	Disney	Josh
65130500082	Palo	Escobar	Palo
65130500999	Rick	Astley	



*ตามตารางเส้นขอบสีแดง แสดงถึงค่าที่ไม่ได้ใส่ในช่อง Nickname ของ 63130500999

SQL SYNTAX

P' Anfield



P' N.I.



SQL Commands

- **DDL (Data Definition Language)**

Commands

CREATE

ALTER

DROP

SQL Commands

- **DQL (Data Query Language)**

Command

SELECT

SQL Commands

- DML (Data Manipulation Language)

Commands

INSERT

UPDATE

DELETE

SQL Commands

- **TCL (Transaction Control Language)**

Commands

COMMIT

ROLLBACK

SAVEPOINT

SET
TRANSACTION

SQL Commands

- **DCL (Data Control Language)**

Commands

GRANT

REVOKE

NOTE : TCL และ DCL น้องๆ จะได้เรียนกันตอนปี 2 น้า :D

DDL

P' Anfield



P' Nai



DDL (Data Definition Language)

DDL หรือภาษาสำหรับจัดการ และนิยามโครงสร้างของฐานข้อมูล เป็นภาษาที่มีไว้สำหรับจัดการฐานข้อมูลโดยเฉพาะ ไม่ว่าจะเป็นการสร้าง, แก้ไข หรือลบฐานข้อมูล โดยในภาษา DDL นั้นประกอบไปด้วยคำสั่งต่าง ๆ ดังนี้

- **CREATE**
- **ALTER**
- **DROP**

1. CREATE

เป็นคำสั่งที่ใช้ในการสร้างตารางด้วย Column ที่สามารถกำหนดได้โดย ชื่อของ *Column, Data Type และ Constraint*

CREATE Syntax

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

1. CREATE

CREATE TABLE Persons (

PersonID int,

Firstname varchar(255),

Lastname varchar(255),

Address varchar(255)

);

Persons table

PersonID	Firstname	Lastname	Address

2. ALTER

เป็นคำสั่งที่ใช้ในการแก้ไข Column ในตารางด้วยการลด (*DROP*) Column หรือเพิ่ม (*ADD*) Column ด้วย Data Type ที่กำหนด

DROP Column และ ADD Column Syntax

ALTER TABLE table_name

DROP COLUMN column_name;

ALTER TABLE table_name

ADD COLUMN column_name datatype;

2. ALTER with sample table

Student Table

StudentId	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500108	Josh	Disney	Josh
65130500082	Palo	Escobar	Palo

2. ALTER WITH DROP COLUMN

ALTER TABLE Student

DROP COLUMN StudentId;

Student Table

Firstname	Lastname	Nickname
Michael	Bane	Mickie
Josh	Disney	Josh
Palo	Escobar	Palo

2. ALTER WITH ADD COLUMN

ALTER TABLE Student

ADD COLUMN Age INT;

Student Table

StudentId	Firstname	Lastname	Nickname	Age
65130500065	Michael	Bane	Mickie	
65130500108	Josh	Disney	Josh	
65130500082	Palo	Escobar	Palo	

*จากในเส้นขอบสีแดงจะเห็นได้ว่า Age ยังคงมีค่าเป็น NULL อยู่ก็ต่อเมื่อเพิ่ม column นี้เข้าไปแต่ยังไม่มีการ INSERT ข้อมูล

3. DROP

เป็นคำสั่งที่ใช้ในการลบตาราง ในฐานข้อมูลนั้น

DROP Syntax

DROP TABLE table_name;

ตัวอย่างเช่น

DROP TABLE Student;

เมื่อกำการรันคำสั่งนี้แล้วตาราง Student จะหายไปจาก Database



School of Information Technology
King Mongkut's University of Technology Thonburi

DML

P' Anfield



P' Nai



DML (Data Manipulation Language)

DML គឺ ភាគមាត្រាកំណត់ការបង្កើតការលើកឱ្យការកិច្ចការនៃព័ត៌មាន ដោយប្រើប្រាស់ការសម្រេចការណ៍ទូទៅ និងការសម្រេចការណ៍ក្នុងព័ត៌មាន។ DML មានប្រភេទប្រចាំខែ ប្រចាំឆ្នាំ និងប្រចាំឆ្នាំរបស់វាទំនាក់។

- **INSERT**
- **DELETE**
- **UPDATE**

INSERT

เพิ่มข้อมูลตาม column ที่ต้องการ:

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

เพิ่มข้อมูลตาม column ที่มีอยู่ในตาราง:

```
INSERT INTO table_name VALUES (value1, value2, value3, ...);
```

*เน้นย้ำว่าลำดับ value ต้องเรียงตามตารางเพื่อให้ข้อมูลถูกต้อง



ข้อควรระวังของ INSERT



ตอน INSERT ให้เช็คด้วยว่า Data Type ที่กำลัง INSERT นั้นคืออะไร เช่น

- NUMBER & DECIMAL ไม่ต้องใส่ quote EX. (1, 1.2)
- CHAR & VARCHAR ต้องใส่ quote EX. ('hello world', '123456789')

INSERT INTO Student (StudentId,Firstname) **VALUES** ('65130500109','Luke');

StudentId	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500082	Josh	Disney	Josh
65130500108	Pablo	Escobar	Pablo
65130500109	Luke	null	null

INSERT INTO Student VALUES ('65130500109', 'Samuel', 'SQL', 'Sam');

StudentId	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500082	Josh	Disney	Josh
65130500108	Pablo	Escobar	Pablo
65130500109	Samuel	SQL	Sam

DELETE

คือการลบข้อมูลทั้งหมดในตาราง แต่สามารถกรองข้อมูลที่ต้องการลบได้โดยการใส่เงื่อนไข

DELETE Syntax

DELETE FROM table_name WHERE condition;



ข้อควรระวังของ DELETE



ถ้าไม่ได้ใส่เงื่อนไขขึ้นมาทุกข้อมูลที่อยู่ในในตารางจะหายหมด เพราะฉะนั้น
ถ้าเราจะลบข้อมูลแบบเจาะจงต้องมี WHERE เสมอ

B
E
F
O
R
E

A
F
T
E
R

Student Table

StudentId	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500108	Pablo	Escobar	Pablo

DELETE FROM Student WHERE Nickname = 'Mickie' ;

StudentId	Firstname	Lastname	Nickname
65130500108	Pablo	Escobar	Pablo

UPDATE

เปลี่ยนแปลงหรือแก้ไขข้อมูลทั้งหมดในตาราง และสามารถกรองข้อมูลที่ต้องการจะเปลี่ยนแปลงได้โดยการใส่เงื่อนไข

UPDATE Syntax

UPDATE table_name

SET column1 = value1, column2 = value 2, ...

WHERE condition;



ข้อควรระวังของ UPDATE



ถ้าเราไม่ได้ใส่เงื่อนไข ข้อมูลที่อยู่ในตารางทั้งหมดจะเปลี่ยนแปลงไปตามที่เรากำหนด และควรเช็คว่าข้อมูลที่เราจะทำการ Update นั้น ถูกคลอลัมบ์ และเงื่อนไขถูกต้องหรือไม่

BEFORE - UPDATE

Student Table

StudentId	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500082	Josh	Disney	Josh
65130500108	Pablo	Escobar	Pablo

AFTER -

UPDATE Student SET Nickname = 'Bob' WHERE Lastname = 'Escobar';

StudentId	Firstname	Lastname	Nickname
65130500065	Michael	Bane	Mickie
65130500082	Josh	Disney	Josh
65130500108	Pablo	Escobar	Bob

Question



ถ้า nous ต้องการที่จะเพิ่มข้อมูลในตาราง Student โดยใช้ข้อมูลที่มีให้จะสามารถเขียน命令มาในรูปแบบ SQL ได้อย่างไร?

ข้อมูลมีดังนี้

FIRSTNAME: Mark, LASTNAME: Zuckerberg, ID:38 และ SCORE:99

Example of Student Table

ID	Firstname	Lastname	Score
1	Pablo	Escobar	50



Answer

คำตอบมีอยู่ 2 รูปแบบ:

1. INSERT INTO Student VALUES (38, 'Mark', 'Zuckerberg', 99);
2. INSERT INTO Student (ID, Firstname, Lastname, Score)
VALUES (38, 'Mark', 'Zuckerberg', 99);

*ถ้าน้องเขียนอว客家แบบได้แบบหนึ่งในนี้ก็ถือว่าถูกต้องเหมือนกันนะค้าบบ



School of Information Technology
King Mongkut's University of Technology Thonburi

DOL

P' Anfield



P' NIA



DQL (Data Query Language)

เป็นคำสั่งที่ใช้สำหรับดำเนินการค้นหาข้อมูลภายใน Schema หรือ Database โดยใช้คำสั่งดังต่อไปนี้

- **SELECT**
- **WHERE**
- **ORDER BY**
- **Aggregation Query**
- **GROUP BY**
- **HAVING**

1. SELECT

เป็นคำสั่งที่ใช้สำหรับการเรียกดูข้อมูลใน Table โดยพื้นฐานคำสั่ง SELECT จะมี Syntax ดังนี้

SELECT Syntax

SELECT column1, column2, ...

FROM table_name;

1. SELECT

SELECT * FROM student;

Student Table

Id	Name	City	Country	Age
1	John	New York	USA	15
2	Tom	London	England	16
3	Clare	Liverpool	England	16
4	Watson	Berlin	Germany	17

1.1. AS

เป็นคำสั่งที่ใช้ในการเปลี่ยนชื่อ Column หรือ Table จากการ SELECT

AS Syntax (เปลี่ยนชื่อ Column)

```
SELECT column1 AS column_one, column2 AS column_two,  
...  
FROM table_name;
```

AS Syntax (เปลี่ยนชื่อ Table)

```
SELECT column1 FROM table_name AS name;
```

1.1. AS

SELECT Id **AS** num , Name , City FROM student;

num	Name	City
1	John	New York
2	Tom	London
3	Clare	Liverpool
4	Watson	Berlin

1.2. DISTINCT

เป็นคำสั่งที่ใช้ระบุเงื่อนไขการเลือกข้อมูลในตาราง (Table) โดยทำการเลือกข้อมูลที่ซ้ำกันมาเพียง Record เดียว

DISTINCT Syntax

```
SELECT DISTINCT column1, column2, ...
```

```
FROM table_name;
```

1.2. DISTINCT

SELECT **DISTINCT** Country FROM student;

NO DISTINCT

Country
USA
England
England
Germany

DISTINCT

Country
USA
England
Germany

2. WHERE

WHERE นั้น ใช้ระบุเงื่อนไขการเลือกข้อมูลใน Table (Filter) โดย WHERE สามารถระบุเงื่อนไขในการเลือกข้อมูลได้ 1 เงื่อนไข หรือ มากกว่า 1 เงื่อนไข ก็ได้ โดยแต่ละเงื่อนไขจะเชื่อมด้วย AND (และ) หรือ OR (หรือ)

SELECT column1, column2, ...

FROM table_name

WHERE condition;

2. WHERE

SELECT * FROM student **WHERE** Country = 'England';

Id	Name	City	Country	Age
2	Tom	London	England	16
3	Clare	Liverpool	England	16

2.1. AND Operators

เป็นคำสั่งที่ใช้เชื่อมต่อเงื่อนไข หลาย ๆ เงื่อนไข ภายในตัวคำสั่ง WHERE Clause

AND Syntax

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE condition1 AND condition2
```

```
AND condition3, ... ;
```

2.1. OR Operators

SELECT column1, column2, ...

FROM table_name

WHERE condition1 **OR** condition2

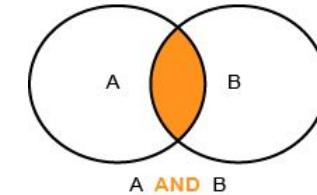
OR condition3, ...;

2.1.1 AND

SELECT * FROM student

WHERE Country = 'England'

AND Name = 'Tom';

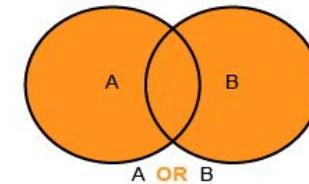


Id	Name	City	Country	Age
2	Tom	London	England	16

2.1.2 OR

SELECT * FROM student

WHERE Country = 'England' **OR** Country = 'USA';



Id	Name	City	Country	Age
1	John	New York	USA	15
2	Tom	London	England	16
3	Clare	Liverpool	England	16

3. ORDER BY

เป็นคำสั่งที่ใช้สำหรับการระบุเงื่อนไขการเลือกข้อมูลที่ต้องการ SELECT

ใน Table สามารถจัดเรียงข้อมูลได้ตามความต้องการ โดย ASC (Ascending)
เป็นการเรียงจากน้อยไปมาก ส่วน DESC (Descending) จะเรียงจากมากไปน้อย

ORDER BY Syntax

SELECT column1, column2, ...

FROM table_name

ORDER BY column1, column2, ... ASC | DESC;

3. ORDER BY

SELECT * FROM student

ORDER BY Id ASC;

Student Table

Id	Name	City	Country	Age
1	John	New York	USA	15
2	Tom	London	England	16
3	Clare	Liverpool	England	16
4	Watson	Berlin	Germany	17

3. ORDER BY

SELECT * FROM student

ORDER BY Id DESC;

Student Table

Id	Name	City	Country	Age
4	Watson	Berlin	Germany	17
3	Clare	Liverpool	England	16
2	Tom	London	England	16
1	John	New York	USA	15

4. Aggregation Query

คือฟังก์ชันคำนวณค่าสถิติ (ค่าที่ประมวลได้จากข้อมูลตัวอย่างโดยวิธีการทางสถิติ อย่างเช่น ค่าเฉลี่ย (AVG), ผลรวม (SUM) เป็นต้น) มาใช้กับการ Query หรือ SELECT ข้อมูลใน Database โดยหลัก ๆ จะแบ่งเป็น 5 ตัวคือ

Command				
COUNT	AVG	SUM	MIN	MAX

4.1 COUNT

คือการนับ Row หรือนับแ Kawว่ามีจำนวนกั้งหมดเท่าไหร่

COUNT Syntax

```
SELECT COUNT(column_name)
```

```
FROM table_name;
```

4.1 COUNT

SELECT **COUNT**(Name) FROM student;

COUNT(Name)
4

4.2 AVG

คือการคำนวณค่าเฉลี่ยของ Column ที่เราเลือก

AVG Syntax

```
SELECT AVG(column_name)
```

```
FROM table_name;
```

4.2 AVG

SELECT **AVG(age)** FROM student;

AVG(age)
16.0000

4.3 SUM

คือผลรวมของ Column ที่เราเลือก

SUM Syntax

```
SELECT SUM(column_name)
```

```
FROM table_name;
```

4.3 SUM

SELECT **SUM**(age) FROM student;

SUM(age)
64

4.4 MIN

คือการหาค่าที่น้อยที่สุดจาก Column ที่เราเลือก

MIN Syntax

```
SELECT MIN(column_name)
```

```
FROM table_name;
```

4.4 MIN

SELECT **MIN(age)** FROM student;

MIN(age)

15

4.5 MAX

คือการหาค่าที่มากที่สุดจาก Column ที่เราเลือก

MAX Syntax

```
SELECT MAX(column_name)
```

```
FROM table_name;
```

4.5 MAX

SELECT MAX(age) FROM student;

MAX(age)
17

5. GROUP BY

ใช้ในการจัดกลุ่มของข้อมูลที่มีค่าเดียวกันหรือเหมือนกันมารวมกัน
แล้วนำมาใช้คำนวณหรือวิเคราะห์ข้อมูลควบคู่กับการใช้ *Aggregate Functions*

GROUP BY Syntax

SELECT column_name(s)

FROM table_name

GROUP BY column_name(s)



ข้อควรระวังของ GROUP BY



หากมี Column กี่ไม่ใช่ Aggregate Functions อยู่ในการ SELECT
จำเป็นที่จะต้องระบุ GROUP BY Column กี่ไม่ใช่ Aggregate Functions ด้วย

5. GROUP BY

SELECT Country , SUM(age)

FROM student

GROUP BY Country;

Country	Age
USA	15
England	32
Germany	17

6. HAVING

ใช้เมื่อเราต้องการเพิ่ม Condition (เงื่อนไข) ของ Aggregate Functions

HAVING Syntax

```
SELECT column_name(s) , Aggregation  
Functions(column_name)
```

```
FROM table_name
```

```
GROUP BY column_name(s)
```

```
HAVING condition
```

6. HAVING

SELECT country, SUM(age)

FROM student

GROUP BY country

HAVING SUM(age) > 20;

Country	Age
England	32



ข้อควรระวังของ DQL



ลำดับของคำสั่งในตัวของ DQL นั้นมีความสำคัญมากถ้าเราใส่ลำดับผิดจะทำให้เกิดการ **ERROR** ได้

SELECT column_name(s)

FROM table_name

WHERE condition

GROUP BY column_name(s)

HAVING condition

ORDER BY column_name(s);

Question



Syntax ในข้อใดไม่ถูกต้อง

- a) SELECT * FROM student WHERE Country = 'England' ;
- b) SELECT SUM(age) FROM student HAVING SUM(age) > 16 ;
- c) SELECT * FROM student HAVING Age = 16 ;
- d) SELECT Name FROM student ORDER BY Name ASC ;



Answer

คำตอบคือ ข้อ c) SELECT * FROM student HAVING Age = 16 ;

เนื่องจากว่า HAVING ใช้เป็น Condition ของ Aggregate Functions
เท่านั้น จึงต้องใช้ WHERE แทน HAVING

Quest???

ມີ quest ເລີກ່ານ້ອຍໆອັກແວ້ວວ
ນ້ອງໆລອງກຳ DQL ເລຍໄດຍມີເຈື່ອນໃຫຕາມນີ້ໃນແຕ່ລະ: query

1. select
2. as
3. where
4. and ມີວິວ or ກີດໄດ້
5. aggregation query
6. order by
7. group by, having

