

Lecture 2 – Relational Models

CPE241 – Database Systems

20 January 2026

Dr. Jaturon Harnsomburana

Dr. Piyanit Ua-areemitr

Department of Computer Engineering

King Mongkut's University of Technology Thonburi

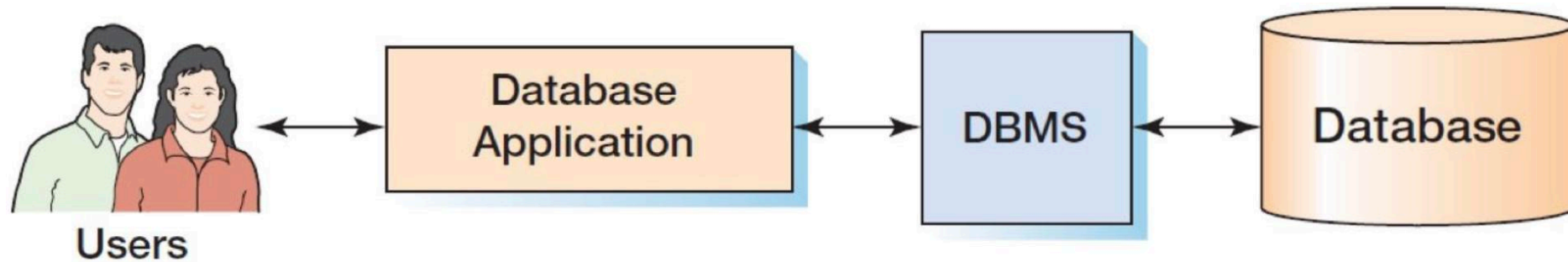


Recap

- Evolution of ways to store data
 - Database history
- What are database and DBMS?
 - Tasks
 - Storage
 - Database types
- A short introduction
 - Relational databases
 - SQL

Recap

- DBMS - a collection of programs that enables users to create and maintain a database.



- Create database
- Create tables
- Create supporting structures (e.g., indexes)
- Read database data
- Modify (insert, update, or delete) database data
- Maintain database structures
- Enforce rules
- Control concurrency
- Provide security
- Perform backup and recovery

- **Create**
- **Process**
- **Administer**

Recap

- Database - a form of data storage that can **manipulate data** to make it more **manageable, efficient, and secure**.
 - Store data in a structured way that is can be easily retrieved.
 - Quickly and accurately provide data.

Today's Goals

- Data models
- Database schemas, instances, and states
- The basic relational model
 - Data
 - Relationship
 - Constraints
- Create database using SQL

Data Models

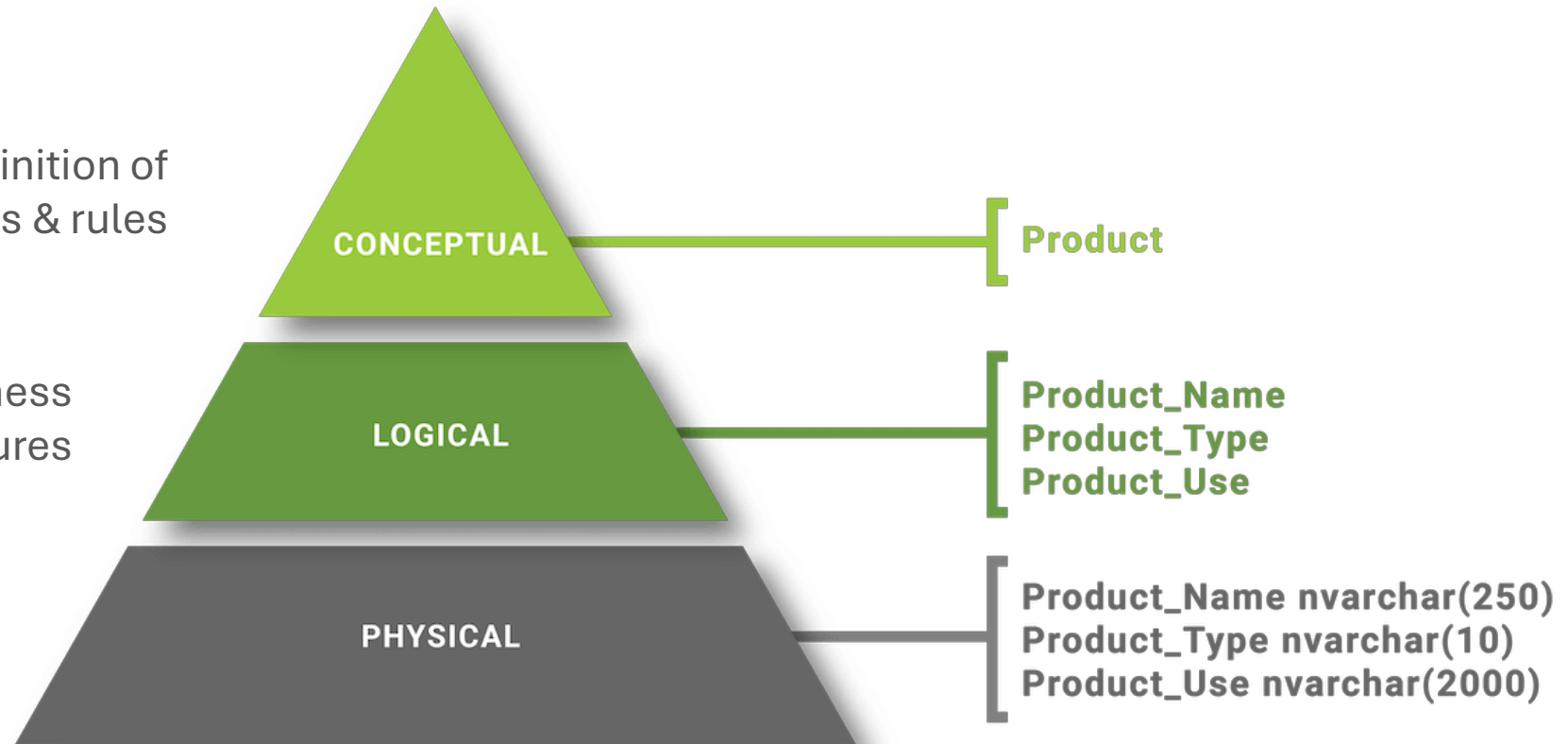
- A collection of concepts that can be used to describe **how data should be stored, organized, and utilized**.
 - Structure of a database i.e. data types, relationships, and constraints.
- Types of data models
 - High-level or **conceptual data models** - concepts that are close to the way many users perceive data.
 - **Logical data models** - concepts used by DBMS implementations, focusing on how data is logically organized.
 - Low-level or **physical data models** - concepts that describe the details of how and where data is physically stored on the computer storage and managed by DBMS.

Data Models

Business Concept – definition of business terms & rules

Data Entities – details of business rules & data structures

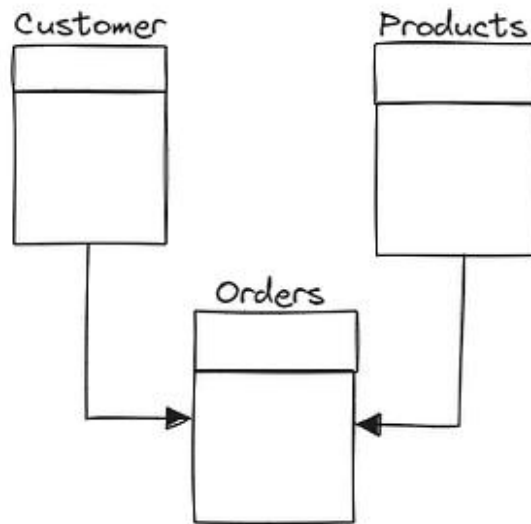
Physical Tables – technical implementation on a physical database



Source: <https://mulianaraut.medium.com/how-we-create-a-reliable-database-through-data-modelling-84b9f74ffbbb>

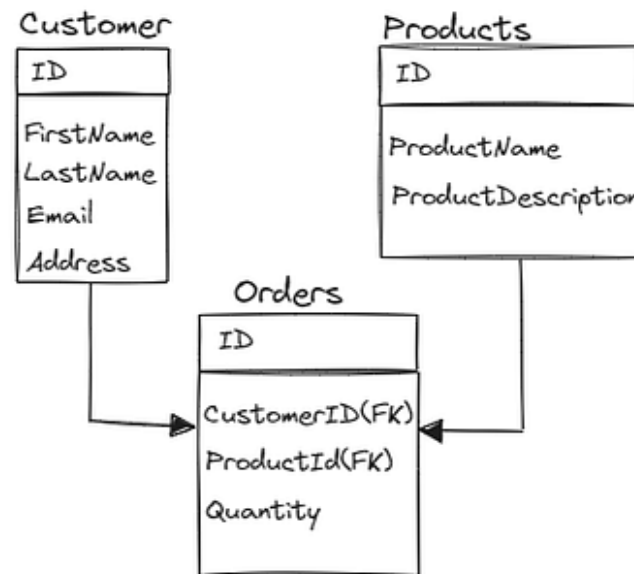
Data Models

CONCEPTUAL MODEL



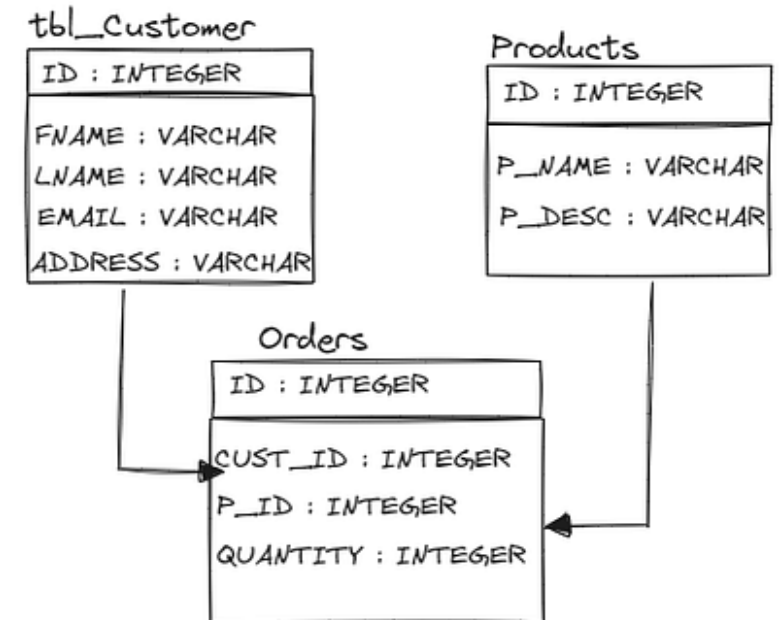
**Entities
Relation**

LOGICAL MODEL



**Attributes
Keys**

PHYSICAL MODEL



**Data Types
Constraints**

Database Schemas

- Description of a database
 - *how data is organized in the database and how it's related to other data.*
- Specified during database design.
- Schema diagram
 - *illustrative display of a database schema.*

An example of relational database schema

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

Database Schemas with Instances

Student

Student ID	Name	Surname	Email
100001	Alice	Bond	Alice@kmutt.ac.th
100002	Bob	Holland	Bob@kmutt.ac.th
100003	Carol	Brown	Carol@kmutt.ac.th

Course

Course ID	Name
CPE100	Computer Programming for Engineers
CPE112	Programming with Data Structures
CPE241	Database Systems

Database Instances and States

- Database instance: actual data (content) stored in a database at a *particular moment* in time.
- Database state
 - **Initial state** - when database is loaded.
 - **Current state** - the state of database at any point of time.
 - **Valid state** - a state that satisfies the structures and constraints of the database (job of DBMS to ensure valid entities).
- The **database schema** changes very infrequently, preferably never.
- The **database state** changes every time the database is updated.

Relational Model Main Concepts

- Data
- Relationships
- Constraints

Relational Model – Data

- An **entity** is some identifiable thing that users want to track.
- A **relation** is a table with columns and rows.
- An **attribute** is a named column of a relation.
- A **tuple** is a row of a relation

- The **degree** of relation is the number of **attributes** it contains.
- The **cardinality** of relation is the number of **tuples** it contains.

Entity, Relation / Table / File, Disk

Attribute / Column / Field

	A	B	C	D	E	F
1						
2						
3						
4						
5	Tuple / Row / Record					
6						
7						
8						
9						
10						

Domain & Data Types

- **Domain** - set of **legal values** that can be assigned to an attribute.
- **Data type** – defines what type of value a table column can hold
 - SQL knows expected data type in each column.

Read more about SQL data types: <https://www.w3resource.com/mysql/mysql-data-types.php>

Student

Student ID	Name	Email	GPAX
100001	Alice	Alice@kmutt.ac.th	3.12
100002	Bob	Bob@kmutt.ac.th	2.85
100003	Carol	Carol@kmutt.ac.th	3.08

Types of Attributes

- **Simple vs Composite** attributes

- **Simple/Atomic** attributes are not divisible.

Each simple attribute is associated with a value set.

- **Composite** attributes can be divided into smaller subparts.

Address -> Street_address, City, State, Zip_code

Types of Attributes

- **Single-valued vs Multi-valued** attributes

- **Multi-valued** attributes have a set of values for the same entity.

Entity – Attribute

PERSON – Age

PERSON – College_degree

CAR - Color

- **Stored vs Derived** attributes

- **Derived** attributes can be determined from other attributes.

Age can be derived from Birth_date.

Keys

- Each tuple has **one or more attributes** that can uniquely identify a specific row.

Student

Student ID	Name	Surname	Email
100001	Alice	Bond	Alice@kmutt.ac.th
100002	Bob	Holland	Bob@kmutt.ac.th
100003	Carol	Brown	Carol@kmutt.ac.th

Course

Course ID	Name
CPE100	Computer Programming for Engineers
CPE112	Programming with Data Structures
CPE241	Database Systems

More Keys...

- **Candidate key** attribute – any attribute that contains a unique value in each row of the table.
- **Composite key** attribute – a key composed of two or more attributes to form a unique value in each new row (consider when a single attribute key cannot be identified).
- **Primary key (PK)** - a selected candidate key.
- **Alternate key** – a candidate key not selected as the PK.

Vehicle

Vehicle ID	Engine ID	Owner ID	Car plate no.
V00001	E100A38	OW2493	AA9439
V00002	E1200F2	OW4387	BC6731
V00003	E200971	OW1104	C888

Relational Model – Relationship

- A relationship is an association among two or more entities, for example, *Bob studies at CPE*.
- Types of relationship
 - One-to-one (1:1)
 - One-to-many (1:N)
 - Many-to-many (M:N)

Types of Relationships

- One-to-one (1:1) - each record in Table A relates to one, and only one, record in Table B.

Table A - Student

Student ID	Name	Surname	Email

Table B – Birth Certificate

Certificate ID	Name	Surname	Country

Types of Relationships

- One-to-many (1:N)
 - A record in Table A can relate to zero, one, or many records in Table B.
 - Many records in Table B can relate to one record in Table A.

Table A - Department

Department ID	Department Name

Table B - Student

Student ID	Name	Surname	Email

Types of Relationships

- Many-to-many (M:N)
 - Many records in Table A can relate to many records in Table B.
 - Many records in Table B can relate to many records in Table A.

Table A - Student

Student ID	Name	Surname	Email

Table B - Supervisor

Supervisor ID	Name	Surname	Email

Types of Relationships

- Check point

Student

Student ID	Name	Surname	Email
100001	Alice	Bond	Alice@kmutt.ac.th
100002	Bob	Holland	Bob@kmutt.ac.th
100003	Carol	Brown	Carol@kmutt.ac.th

Course

Course ID	Name
CPE100	Computer Programming for Engineers
CPE112	Programming with Data Structures
CPE241	Database Systems

Foreign Key

- A set of fields in one relation that is used to ‘refer’ to a tuple in another relation. Must correspond to primary key of the second relation.
- Connect tables, like a logical pointer.

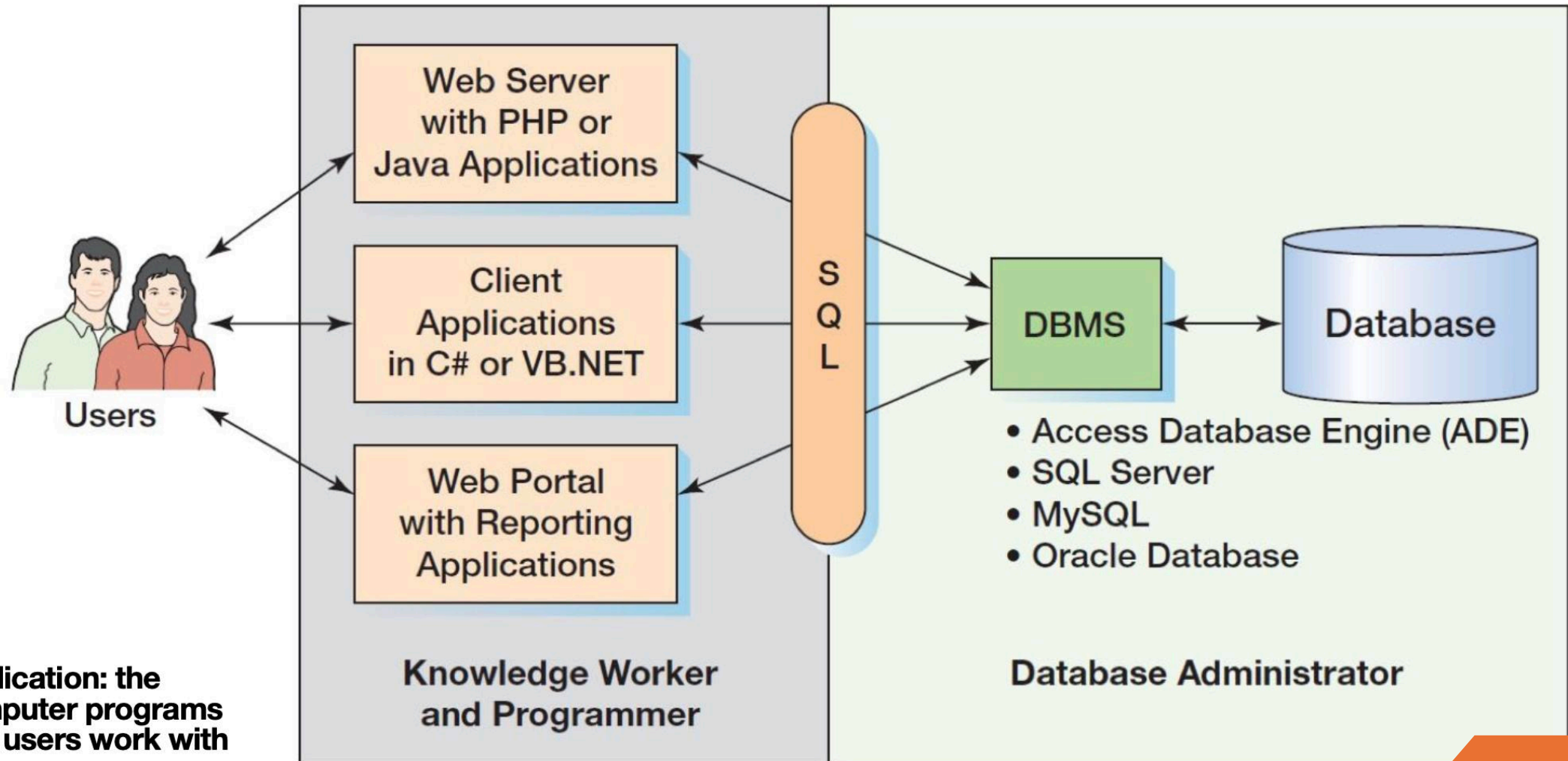
Student

Student ID	Name	Email
100001	Alice	Alice@kmutt.ac.th
100002	Bob	Bob@kmutt.ac.th
100003	Carol	Carol@kmutt.ac.th

Enrollment

Student ID	Course ID	Grade
100001	CPE100	A
100001	CPE112	B
100001	CPE241	B+
100002	CPE100	A
100002	CPE112	C
100003	CPE100	B

Enterprise-class DBMS



SQL

- Standard language that can be interact with structured data on databases.
- DBMS changes SQL instructions into a form understood by the database.
- Categories of SQL commands
 - Data Definition Language (DDL)
 - Data Manipulation Language (DML)
 - Data Query Language (DQL)
 - Data Control Language (DCL)
 - Transaction Control Language (TCL)

Creating a Database

- Step 1 – create a database
- Step 2 – create table(s)
- Step 3 – add data to each table
- What's else?
 - Update tables/data
 - Delete tables/data
 - Query data
 - Etc.

Data Definition Language

- Define data in a database.
 - **CREATE** command – create storage objects in a database i.e. tables
 - **ALTER** command – modify the structure of a table object in a database
 - **DROP** command - remove an existing object from a database
- STEP 1 – create a database
`CREATE DATABASE database_name;`
`CREATE DATABASE cpe_students;`

Data Definition Language

- Step 2 – create table(s)

```
CREATE TABLE table_name (column1_name DATATYPE, ...);
```

```
CREATE TABLE Student (StudentID CHAR(6), Name  
VARCHAR(50), Email VARCHAR(50));
```

Student

StudentID	Name	Email

Note: SQL data types tell a DBMS how to

- Interpret the value of a column.
- Maintain data in the right format.
- Make sure the value of each column is as expected.

Data Manipulation Language

- Manipulate data in a database (most CRUD).
 - **INSERT** command – insert records of data into a database table
 - **UPDATE** command – edit data that already exists in a database table
 - **DELETE** command – delete one or more rows of data from a table
- Step 3 – add data to each table

```
INSERT INTO table_name (column1_name,  
column2_name, ...) VALUES (value1, value2,  
...);
```

```
INSERT INTO Student (StudentID, Name,  
Email) VALUES ("100001", "Alice",  
"Alice@kmutt.ac.th");
```

StudentID	Name	Email
100001	Alice	Alice@kmutt.ac.th

Other SQLs

- Data query language (DQL) – read data stored in a database.
 - **SELECT** command – retrieve data based on preferred filter criteria.
- Data control language (DCL) – control access to the database e.g. give user privilege to data by **GRANT** and **REVOKE** command.
- Transaction Control Language (TCL) - group a set of tasks into a single execution unit.

Wrap up

- Data models: conceptual, logical, and physical models
- Database schemas
- The basic relational model
 - Data: domain, types of attributes, keys
 - Relationship: types of relationships
 - Constraints
- Create database using SQL
 - DDL & DML