

ITCS393 Database Systems Lab

Relational Databases and Structured Query Language

Dr. Petch Sajjacholapunt

Dr. Wudhichart Sawangphol

Dr. Jidapa Kraisangka

Wudhichart.saw@mahidol.edu

Recap

- Lab 2
 - Add new column, Alter column and Delete column
 - Define Primary Key (PK) and Foreign Key (FK)
 - Create and Delete constraints

Learning Outcomes

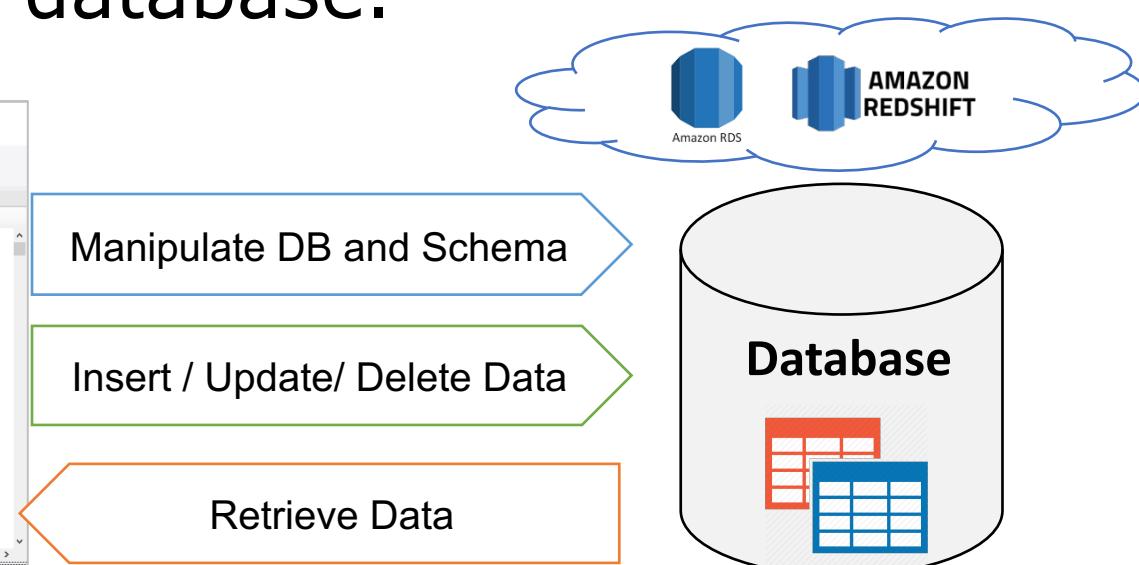
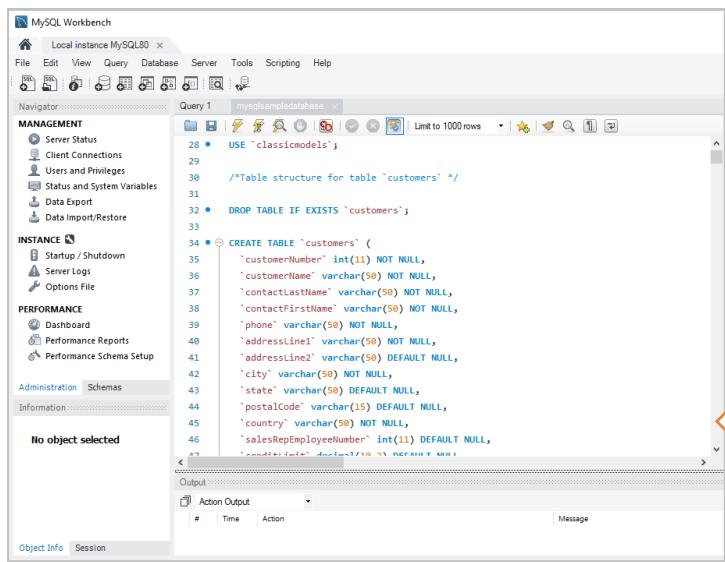
- To learn how to **retrieve** data in a relational database by using *Structured Query Language* (SQL).

Background Knowledge

- **Queries** are simply questions represented in a way that the DBMS can recognize and process.
- Normally, queries are represented by a **Structured Query language (SQL)**.

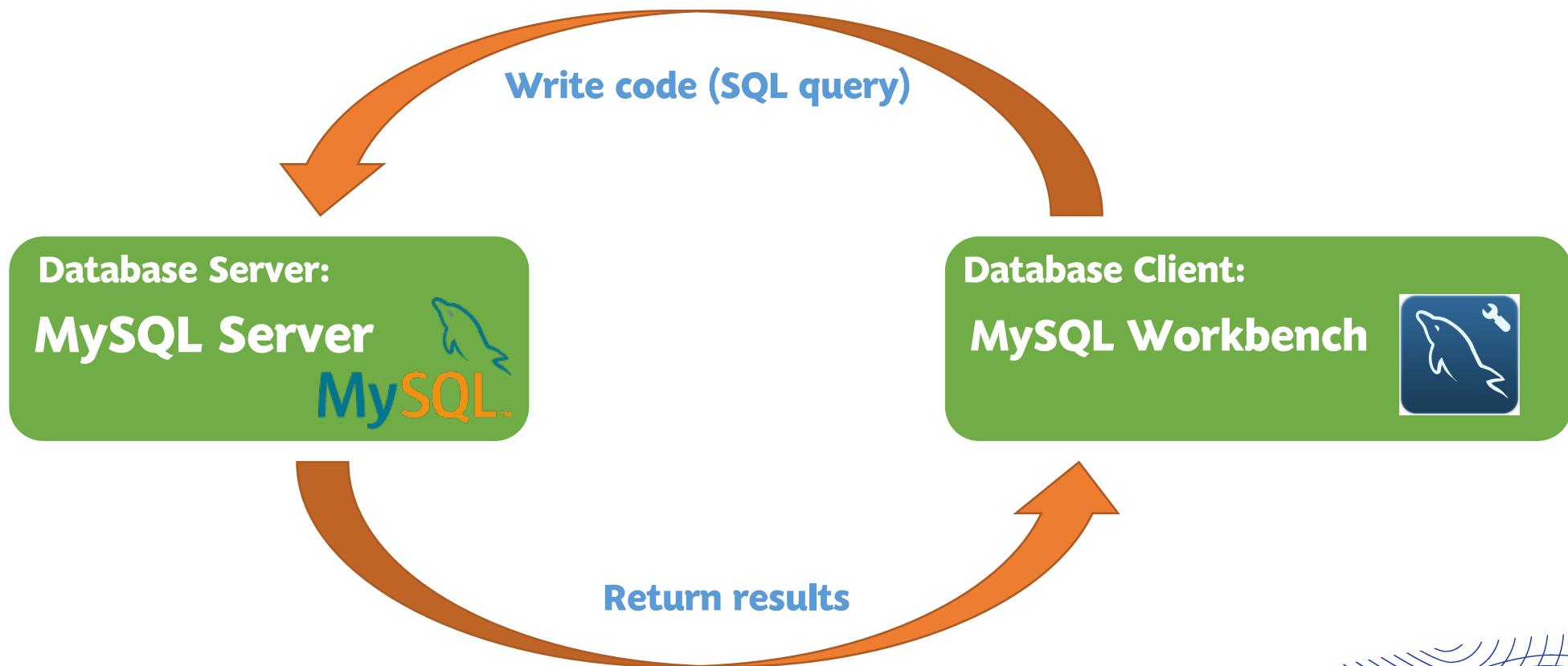
Background Knowledge (cont.)

- “S-Q-L”, often pronounced as “sequel”
- SQL is a **standard** language for querying and manipulating data in database.



Background Knowledge (cont.)

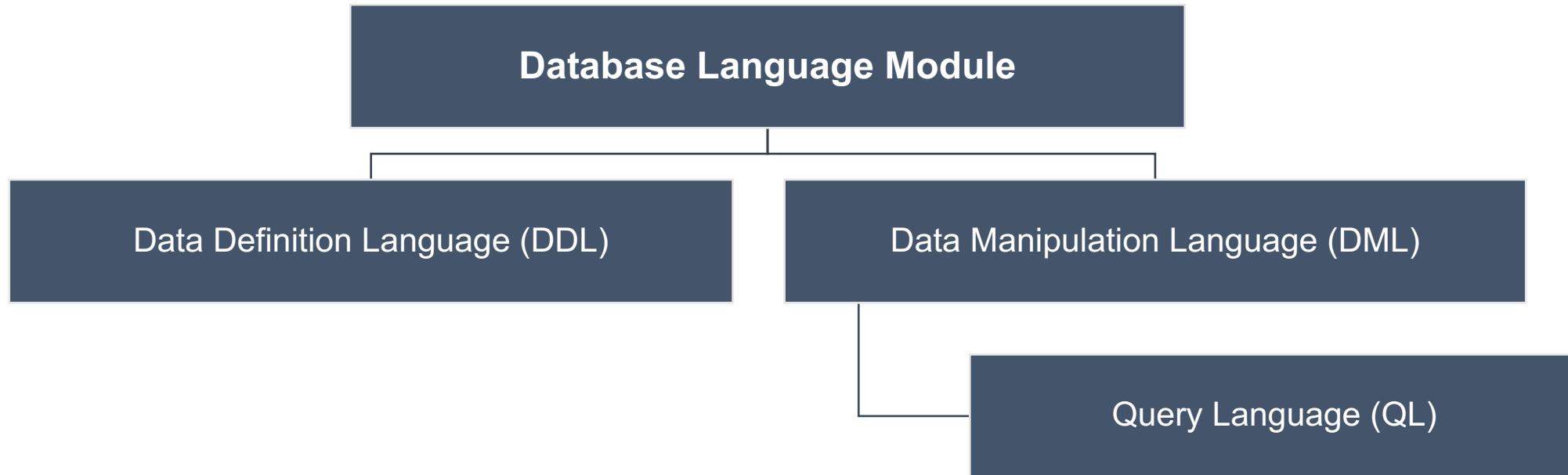
- DBMS: The client-server model



Background Knowledge (cont.)

- There are 2 main categories of SQL
 - Data Definition Language (DDL)
 - Define relational schema
 - Create/Alter/Delete structure of tables
 - Data Manipulation Language (DML)
 - Query one or more tables
 - Insert/Delete/Modify data in tables

Background Knowledge (cont.)



Background Knowledge (cont.)

- Data Manipulation Language



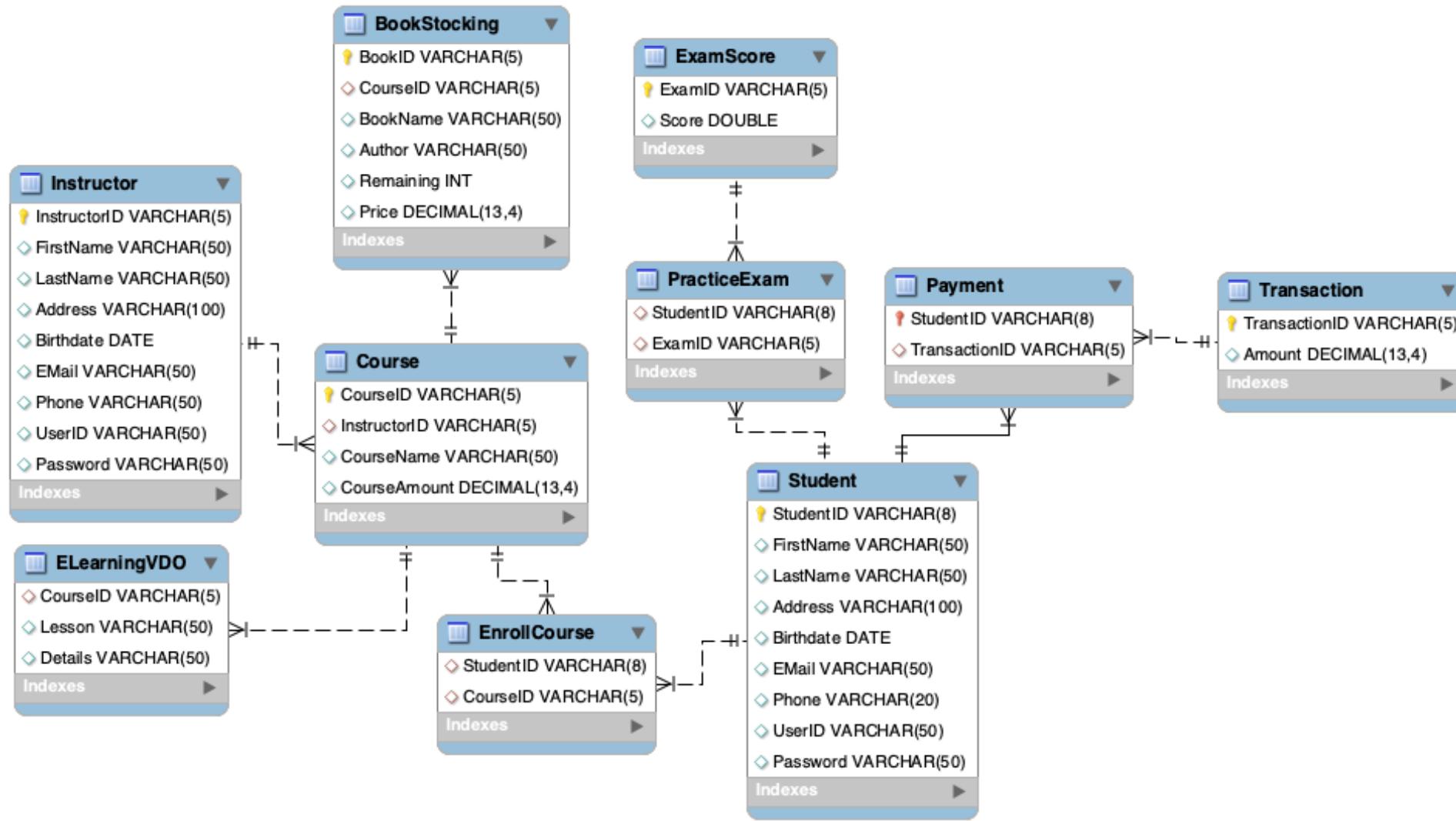
In this lab

- You will learn
 - How to use **Structured Query Language (SQL)** as a query language to access a relational database.
 - We will issue queries as SQL statements to RDBMS which will give the corresponding query results.

Outline of Lab #3

- 3.1 ICT Tutorial Database
- 3.2 SELECT Statement
- 3.3 Computed Field
- 3.4 Search Condition
- 3.5 Search Condition using Variable Declaration
- 3.6 Intro to Aggregation Functions
- 3.7 Sorting function
- 3.8 Intro to Joining

3.1 ICT Tutorial Database

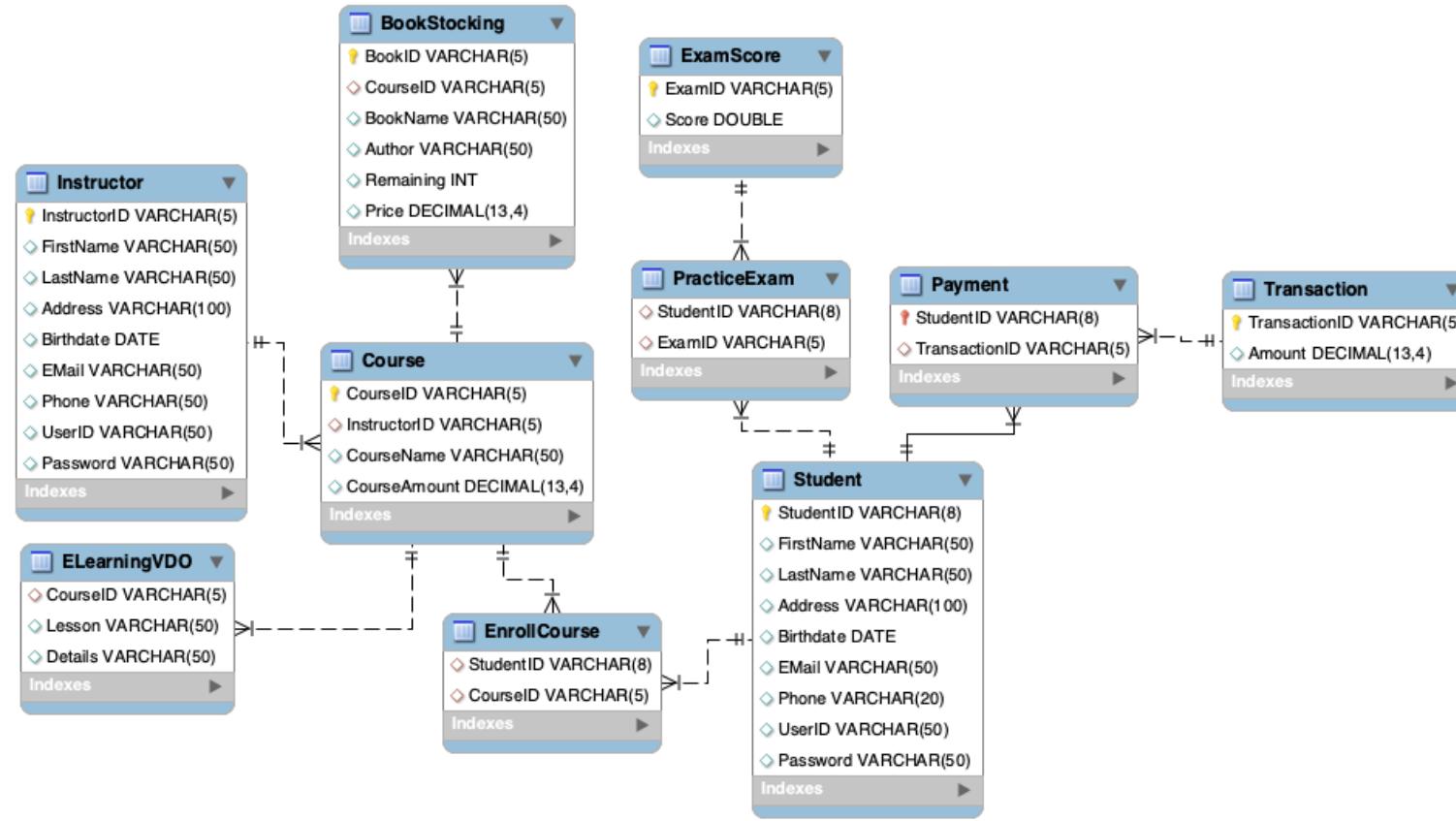
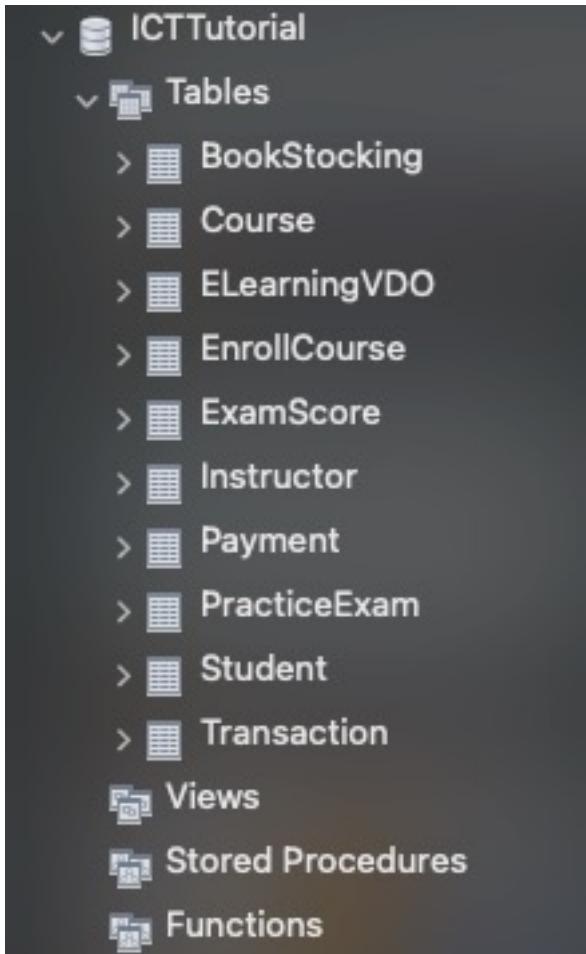


3.1 ICT Tutorial Database (cont.)

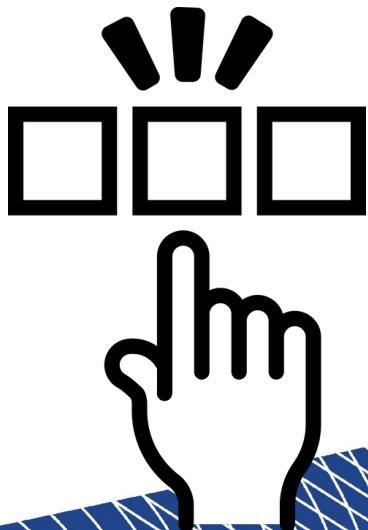
- You will start by execute the SQL file “Assignment1” to
 - Create the **ICTTutorial** Database
 - Create all **relation schemas** in the **ICTTutorial** Database; specify appropriate data type for each attribute; and specify the **PK and FK constraints**
 - Insert the instance data records in each table

3.1 ICT Tutorial Database (cont.)

- Initially, **ICTTutorial** Database consists of 10 tables:



3.2 SELECT Statement



3.2 SELECT Statement

- Retrieve data from a database
- Syntax

```
SELECT    column list
FROM      Source
[WHERE    condition]
[GROUP BY expression]
[HAVING   condition]
[ORDER BY expression ASC|DESC]
```

- Column list = one or more columns separated by commas
- * (asterisk) = list all columns

3.2 SELECT Statement (cont.)

- Query 1 List all students

```
SELECT * FROM Student;
```

	StudentID	FirstName	LastName	Address	Birthdate	EMail	Phone	UserID	Password
▶	u0001	Sukit	Lertsuk	154/2 Soi Rang-nam	1997-08-04	sukit@hotmail.com	mobile:0932133484	sukit.ler	sk9dkc
	u0002	Chatree	Meekerd	254 Salaya	2001-01-14	chatree@gmail.com	mobile:0852354821	chatree	28dc8ds
	u0003	Chujai	Sae-Tung	124/225 Klongluang	1999-06-21	chujai@gmail.com	mobile:0825212410	chujai	15d8ds5
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.2 SELECT Statement (cont.)

- Query 2 List the **ID**, **Firstname**, **Lastname**, and **Email** of all students.

```
SELECT StudentID, Firstname, Lastname, Email  
FROM Student;
```

	StudentID	Firstname	Lastname	Email
▶	u0001	Sukit	Lertsuk	sukit@hotmail.com
	u0002	Chatree	Meekerd	chatree@gmail.com
	u0003	Chujai	Sae-Tung	chujai@gmail.com
	NULL	NULL	NULL	NULL

3.3 Computed Field

3.3 Computed Field

- Query 3 List the **ID**, **Fullname**, **Age**, and **Email** of all students.

```
SELECT StudentID, CONCAT(Firstname, ' ', Lastname) as  
'Fullname', Year(current_date()) - Year(birthdate) as  
'Age', Email  
FROM Student;
```

	StudentID	Fullname	Age	Email
▶	u0001	Sukit Lertsuk	25	sukit@hotmail.com
	u0002	Chatree Meekerd	21	chatree@gmail.com
	u0003	Chujai Sae-Tung	23	chujai@gmail.com

3.3 Computed Field (cont.)

- A **computed field** (or **Derived Attribute**) is a field that is the result of calculating using one or more existing fields.
- **Arithmetic operators** (e.g., +, -, *, /) and **parentheses** can be used for integer domain.

3.3 Computed Field (cont.)

- Query 4 List the book id, book name, remaining, and total price of remaining for all books.

```
SELECT BookID, BookName, Remaining,  
Remaining*Price as TotalPrice  
FROM BookStocking;
```

	BookID	BookName	Remaining	TotalPrice
▶	B001	DBMS in action	39	48360.0000
	B002	Database Design	78	144300.0000
	B003	Expert System Design	64	60800.0000

3.4 Search Condition



3.4 Search Condition

- Query 5 Find the Name of Instructor who lives in 'Bangkae'.

```
SELECT CONCAT(Firstname, ' ', Lastname) as 'Fullname', Address
FROM Instructor
WHERE Address LIKE '%Bangkae%';
```

Fullname	Address
Nikorn Sawanglap	1125 Bangkae

3.5 Search Condition using Variable Declaration



3.5 Search Condition using Variable Declaration

- Query 6 Find the Name and Age of Instructor who are older than 40 years.

```
SET @MyAge = 40;  
SELECT CONCAT(Firstname, ' ', Lastname) as 'Fullname',  
Year(current_date()) - Year(birthdate) as 'Age', Email  
FROM Instructor  
WHERE Year(current_date()) - Year(birthdate) > @MyAge;
```

	Fullname	Age	Email
▶	Nikorn Sawanglap	47	nikorn@hotmail.com
◀	Nongkran Poonpol	51	nongkran@hotmail.com

3.6 Intro to Aggregation function

3.6 Intro to Aggregation function

- **Aggregation functions** are built-in, e.g., COUNT, SUM, AVG, MAX and MIN.
- Query 7 How many students enroll for course "ICT01"?

```
SELECT COUNT(StudentID) as 'TotalStudent'  
FROM EnrollCourse  
WHERE courseID='ICT01';
```

TotalStudent
2

3.6 Intro to Aggregation function (cont.)

- Query 8 What is the **average AGE** of all students?

```
SELECT AVG(Year(current_date()) - Year(birthdate))  
as 'avg_age'  
FROM Student;
```

avg_age
23.0000

3.6 Intro to Aggregation function (cont.)

- **Grouping** means creating groups of records that share some common characteristics.
- We can use aggregation functions in combination with grouping, where calculations affect groups of records.

3.6 Intro to Aggregation function (cont.)

- Query 9 How many books provide for each course?

```
SELECT courseid, COUNT(bookid) as 'Book4Course'  
FROM BookStocking  
GROUP BY courseid;
```

	courseid	Book4Course
	ICT01	2
	ICT02	1

3.7 Sorting function

3.7 Sorting function

- To list the records obtained from query results in a particular way, we need to **sort** the records.
- The field on which records are sorted is called the **sort key**.
- Sorting records can be performed more than one field:
 - The first sort field is called the **major sort key**.
 - The second sort field is called the **minor sort key**.

3.7 Sorting function (cont.)

- Query 10 List the Fullname and StudentID for each **Student**. Sort the output alphabetically by Student's Last name.

```
SELECT StudentID, CONCAT(Firstname, ' ',  
Lastname) as 'Fullname'  
FROM Student  
ORDER BY Lastname ASC;
```

	StudentID	Fullname
▶	u0001	Sukit Lertsuk
	u0002	Chatree Meekerd
	u0003	Chujai Sae-Tung

3.8 Join Operations

3.8 Join Operations

- In many cases, we will need to create queries to choose data from more than one tables.
- It is necessary to **join the tables based on matching fields** in corresponding columns.

3.8 Join Operations (cont.)

- Query 11 Show the course name taught by each instructor.

```
SELECT i.Firstname, i.Lastname, c.Coursename  
FROM Instructor i INNER JOIN Course c ON  
i.InstructorID = c.InstructorID;
```

	Firstname	Lastname	Coursename
▶	Nikorn	Sawanglap	Introduction to Database
	Nongkran	Poonpol	Expert System
	Chokchai	Sakooldee	Computational Thinking

3.8 Join Operations (cont.)

- Query 12 Show course name, book name, remaining amount, and price of all courses that may or may not provide the books.

```
SELECT c.Coursename, b.Bookname, Remaining, Price  
FROM Course c LEFT JOIN BookStocking b on  
c.CourseID = b.CourseID;
```

Coursename	Bookname	Remaining	Price
▶ Introduction to Database	DBMS in action	39	1240.0000
Introduction to Database	Database Design	78	1850.0000
Expert System	Expert System Design	64	950.0000
Computational Thinking	NULL	NULL	NULL



THANKS
FOR YOUR
ATTENTION
ANY
QUESTIONS?