ITCS393 Database Systems Lab

Database Definition Language (DDL)

Dr. Petch Sajjacholapunt
Dr. Wudhichart Sawangphol
Dr. Jidapa Kraisangka

Wudhichart.saw@mahidol.edu

Learning Outcomes

- To learn how to modify columns in existing table
  - Add new column, Alter column and Delete column
- To learn how to create primary key, foreign key
- To learn how to create and delete constraint

RECAP from Lab #1

- Write and run an SQL script that perform the following tasks:
  - Create Database
  - Create table, for example:

```
USE ITCS393DB;
CREATE TABLE Staff (
     StaffID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

2.0 To show detail of columns

- Show all tables in a current database.

```
SHOW TABLES;
```

- Show the columns of a table.

```
DESCRIBE table_name;
```

- Show the information of a column in a table

```
DESCRIBE table_name column_name;
```

Learning Outcomes

- To learn how to modify columns in existing table
  - Add new column, Alter column and Delete column
- To learn how to create primary key, foreign key
- To learn how to create and delete constraint

2.1 Alter Columns in Existing Table

- There are many way that you can alter columns in the existing table without dropping table and creating a new one
- The command that we will use is ALTER.
- The full syntax can be found here. (Link: https://dev.mysql.com/doc/refman/8.0/en/alter-table.html)
- In this lab, we will focus on some common modification on tables.

2.1 Alter Columns in Existing Table (cont.)

- We are going to
  - Add a new column
  - Modify a column
  - Rename columns
  - Drop a column

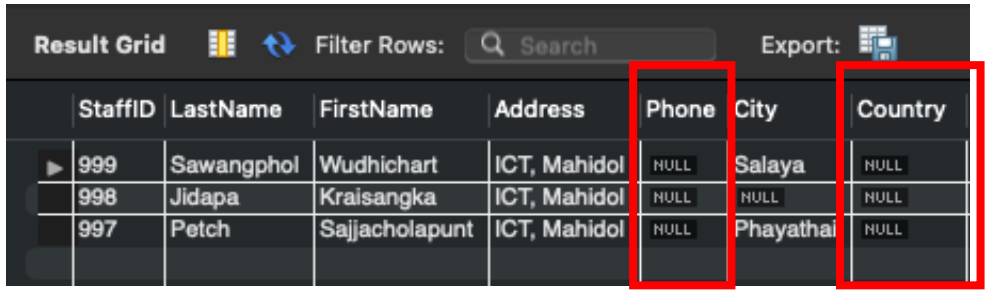
2.1 Alter Columns in Existing Table (cont.)

- Adding a new column
- Syntax

```
ALTER TABLE table
ADD [COLUMN] column_name_1 column_1_definition [FIRST|AFTER
existing_column],
ADD [COLUMN] column_name_2 column_2_definition [FIRST|AFTER
existing_column], ...;
```

- Example

```
ALTER TABLE Staff
ADD COLUMN Phone VARCHAR(15) AFTER Address,
ADD COLUMN Country VARCHAR(20) AFTER City;
```



2.1 Alter Columns in Existing Table (cont.)

- Modify a column
- Syntax

```
ALTER TABLE table_name
MODIFY column_name column_definition
[ FIRST | AFTER column_name],
MODIFY column_name column_definition
[ FIRST | AFTER column_name], ...;
```

2.1 Alter Columns in Existing Table (cont.)

- Modify a column
- Example

```
ALTER TABLE Staff
MODIFY FirstName VARCHAR(100) NOT NULL,
MODIFY LastName VARCHAR(100) NOT NULL AFTER FirstName;
```

Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
LastName	varchar(255)	YES		NULL	
FirstName	varchar(255)	YES		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	
Country	varchar(20)	YES		NULL	

Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
FirstName	varchar(100)	NO		NULL	
LastName	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	
Country	varchar(20)	YES		NULL	

2.1 Alter Columns in Existing Table (cont.)

- Rename columns
- Syntax

```
ALTER TABLE table_name
CHANGE COLUMN original_name new_name column_definition
[FIRST | AFTER column_name],
CHANGE COLUMN original_name new_name column_definition
[FIRST | AFTER column_name], …;
```

2.1 Alter Columns in Existing Table (cont.)

- Rename columns
- Example

```
ALTER TABLE Staff
CHANGE COLUMN FirstName Staff_FN VARCHAR(100) NOT NULL,
CHANGE COLUMN LastName Staff_LN VARCHAR(100) NOT NULL;
```

Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
FirstName	varchar(100)	NO		NULL	
LastName	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	
Country	varchar(20)	YES		NULL	

Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
Staff_FN	varchar(100)	NO		NULL	
Staff_LN	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	
Country	varchar(20)	YES		NULL	

2.1 Alter Columns in Existing Table (cont.)

- Drop a column
- Syntax

```
ALTER TABLE table_name DROP COLUMN column_name;
```

- Example

```
ALTER TABLE Staff DROP COLUMN Country;
```

Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
Staff_FN	varchar(100)	NO		NULL	
Staff_LN	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	
Country	varchar(20)	YES		NULL	

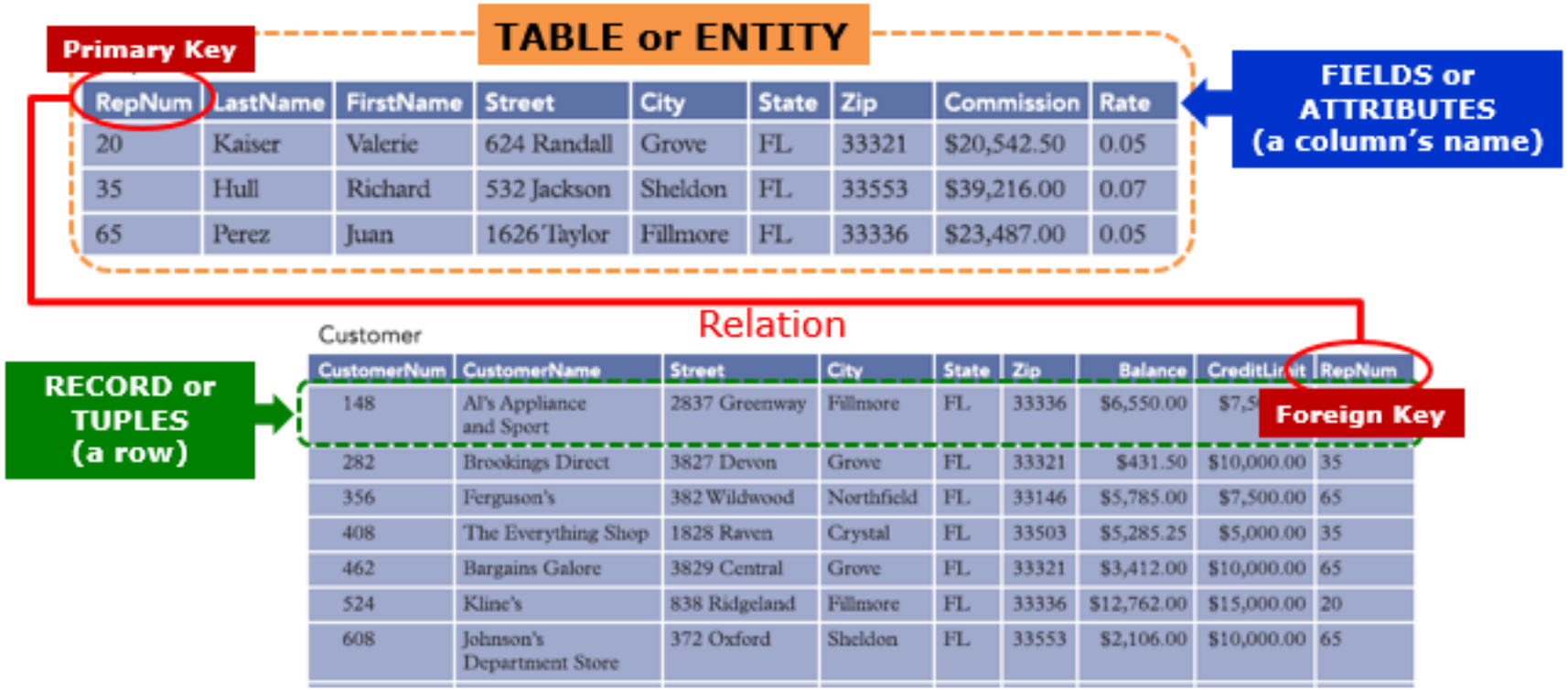
Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
Staff_FN	varchar(100)	NO		NULL	
Staff_LN	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	

Learning Outcomes

- To learn how to modify columns in existing table
  - Add new column, Alter column and Delete column
- To learn how to create primary key, foreign key
- To learn how to create and delete constraint

2.2 Primary Key & Foreign Key

- What is Primary Key?
- What is Foreign Key?



RepNum	LastName	FirstName	Street	City	State	Zip	Commission	Rate
20	Kaiser	Valerie	624 Randall	Grove	FL	33321	$20,542.50	0.05
35	Hull	Richard	532 Jackson	Sheldon	FL	33553	$39,216.00	0.07
65	Perez	Juan	1626 Taylor	Fillmore	FL	33336	$23,487.00	0.05

CustomerNum	CustomerName	Street	City	State	Zip	Balance	CreditLimit	RepNum
148	Al's Appliance and Sport	2837 Greenway	Fillmore	FL	33336	$6,550.00	$7,5	
282	Brookings Direct	3827 Devon	Grove	FL	33321	$431.50	$10,000.00	35
356	Ferguson's	382 Wildwood	Northfield	FL	33146	$5,785.00	$7,500.00	65
408	The Everything Shop	1828 Raven	Crystal	FL	33503	$5,285.25	$5,000.00	35
462	Bargains Galore	3829 Central	Grove	FL	33321	$3,412.00	$10,000.00	65
524	Kline's	838 Ridgeland	Fillmore	FL	33336	$12,762.00	$15,000.00	20
608	Johnson's Department Store	372 Oxford	Sheldon	FL	33553	$2,106.00	$10,000.00	65

2.2 Primary Key & Foreign Key (cont.)

- Create Primary Key with table creation
- Recall table creation syntax

```
CREATE TABLE [table name]  (
  ColumnName1 data type [constraint]
  [, ColumnName2 data type [constraint]]
  [PRIMARY KEY  (column2 [,column2]) ]  REFERENCES
  tablename]  [, CONSTRAINT ConstraintName ]
);
```

2.2 Primary Key & Foreign Key (cont.)

- Example

```
CREATE TABLE Department (
      DeptID INT PRIMARY KEY,
     DeptName VARCHAR(50) NOT NULL,
     DeptCity VARCHAR(50)
);
```

Field	Type	Null	Key	Default	Extra
DeptID	int	NO	PRI	NULL	
DeptName	varchar(50)	NO		NULL	
DeptCity	varchar(50)	YES		NULL	

2.2 Primary Key & Foreign Key (cont.)

- The effect of primary key

```
INSERT INTO Department
VALUES(NULL, 'ICT', 'Salaya');
```

```
Error Code: 1048. Column
'DeptID' cannot be null
```

```
INSERT INTO Department
VALUES(1, 'ICT', 'Salaya'),
(2, 'ICT', 'Phayathai');
```

```
2 row(s) affected Records: 2
Duplicates: 0  Warnings: 0
```

```
INSERT INTO Department
VALUES (2, 'Engineer',
'Salaya');
```

```
Error Code: 1062. Duplicate
entry '2' for key
'department.PRIMARY'
```

2.2 Primary Key & Foreign Key (cont.)

- You can also create PRIMARY KEY using ALTER command.
- Syntax

```
ALTER TABLE table_name
ADD PRIMARY KEY (column_name)
```

2.2 Primary Key & Foreign Key (cont.)

- Example

```
ALTER TABLE Staff
ADD PRIMARY KEY (StaffID);
```

Field	Type	Null	Key	Default	Extra
StaffID	int	YES		NULL	
Staff_FN	varchar(100)	NO		NULL	
Staff_LN	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	

Field	Type	Null	Key	Default	Extra
StaffID	int	NO	PRI	NULL	
Staff_FN	varchar(100)	NO		NULL	
Staff_LN	varchar(100)	NO		NULL	
Address	varchar(255)	YES		NULL	
Phone	varchar(15)	YES		NULL	
City	varchar(255)	YES		NULL	

2.2 Primary Key & Foreign Key (cont.)

- Cautions:
  - In order to ALTER column to be PRIMARY KEY or NOT NULL, you need to ensure that the data have been filled in that column.

2.2 Primary Key & Foreign Key (cont.)

- A FOREIGN KEY in one table points to a PRIMARY KEY in another table.
- You can create FOREIGN KEY with table creation.
- Note that you need to consider the order of table creation as well.
- Example

```
CREATE TABLE Course(
      CourseID INT PRIMARY KEY,
    CourseName VARCHAR(50) NOT NULL,
    StaffID INT NOT NULL,
    FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);
```

Field	Type	Null	Key
CourseID	int	NO	PRI
CourseName	varchar(50)	NO	
StaffID	int	NO	MUL

2.2 Primary Key & Foreign Key (cont.)

- Let's add a new column 'StaffDeptID' into the Staff table
- Then we are going to add FOREIGN KEY between the Staff table and Department table

```
ALTER TABLE Staff ADD COLUMN StaffDeptID INT;
ALTER TABLE Staff
ADD FOREIGN KEY (StaffDeptID) REFERENCES Department(DeptID);
```

Field	Type	Null	Key
StaffID	int	NO	PRI
Staff_FN	varchar(100)	NO	
Staff_LN	varchar(100)	NO	
Address	varchar(255)	YES	
Phone	varchar(15)	YES	
City	varchar(255)	YES	
StaffDeptID	int	YES	MUL

2.2 Primary Key & Foreign Key (cont.)

- When you add the FOREIGN KEY to the table, you can add some action constraint to it.
- Syntax

```
[CONSTRAINT [symbol]] FOREIGN KEY [index_name]
(col_name, ...)
REFERENCES tbl_name (col_name,...)
[ON DELETE reference_option]
[ON UPDATE reference_option]

reference_option:
RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
```

2.2 Primary Key & Foreign Key (cont.)

- Example

```
CREATE TABLE Course(
      CourseID INT PRIMARY KEY,
    CourseName VARCHAR(50) NOT NULL,
    StaffID INT NOT NULL,
    FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

2.2 Primary Key & Foreign Key (cont.)

Source: https://dev.mysql.com/doc/refman/8.0/en/create-table-foreign-keys.html

Referential Actions	Description
CASCADE	Delete or update the row from the parent table and automatically delete or update the matching rows in the child table. Both ON DELETE CASCADE and ON UPDATE CASCADE are supported.
SET NULL	Delete or update the row from the parent table and set the foreign key column or columns in the child table to NULL. Both ON DELETE SET NULL and ON UPDATE SET NULL clauses are supported.
RESTRICT	Rejects the delete or update operation for the parent table. Specifying RESTRICT (or NO ACTION) is the same as omitting the ON DELETE or ON UPDATE clause.
NO ACTION	The MySQL Server rejects the delete or update operation for the parent table if there is a related foreign key value in the referenced table. In MySQL, foreign key constraints are checked immediately, so NO ACTION is the same as RESTRICT.
SET DEFAULT	This action is recognized by the MySQL parser, but both InnoDB and NDB reject table definitions containing ON DELETE SET DEFAULT or ON UPDATE SET DEFAULT clauses.

Learning Outcomes

- To learn how to modify columns in existing table
  - Add new column, Alter column and Delete column
- To learn how to create primary key, foreign key
- To learn how to create and delete constraint

2.3 Constraints

- What are Constraints?
  - Rules for the data in a table
  - The action is terminated, if there is violation of any constraints.
  - They can be specified when creating a table (CREATE TABLE statement) or modifying a table (ALTER TABLE statement)
- How many types?
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK
  - DEFAULT

2.3 Constraints (cont.)

Source: https://www.w3resource.com/mysql/creating-table-advance/constraint.php

CONSTRAINT	DESCRIPTION
NOT NULL	In MySQL NOT NULL constraint allows to specify that a column can not contain any NULL value. MySQL NOT NULL can be used to CREATE and ALTER a table.
UNIQUE	The UNIQUE constraint in MySQL does not allow to insert a duplicate value in a column. The UNIQUE constraint maintains the uniqueness of a column in a table. More than one UNIQUE column can be used in a table.
PRIMARY KEY	A PRIMARY KEY constraint for a table enforces the table to accept unique data for a specific column and this constraint creates a unique index for accessing the table faster.
FOREIGN KEY	A FOREIGN KEY in MySQL creates a link between two tables by one specific column of both tables. The specified column in one table must be a PRIMARY KEY and referred by the column of another table known as FOREIGN KEY.
CHECK	A CHECK constraint controls the values in the associated column. The CHECK constraint determines whether the value is valid or not from a logical expression.
DEFAULT	In a MySQL table, each column must contain a value (including a NULL). While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT.

2.3 Constraints (cont.)

- You can add constraint with table creation
- Example

```
CREATE TABLE tempStaff
(
   StaffID INT CHECK (StaffID > 0), -- StaffID has to be greater than 0
   FirstName VARCHAR(50) NOT NULL,  -- FirstName cannot be null value
   Email VARCHAR(50) UNIQUE,        -- Email has to be unique
   StaffDeptID INT DEFAULT 0,       -- Default value of StaffDeptID is 0
      -- Set PK
   CONSTRAINT PK_tempStaff_StaffID PRIMARY KEY (StaffID),
      -- Set FK with table Department
   CONSTRAINT FK_tempStaff_DeptID FOREIGN KEY (StaffDeptID)
      REFERENCES Department(DeptID)
);
```

2.3 Constraints (cont.)

- In addition, you can add constraint using ALTER command.
- We will add DEFAULT to FirstName of the tempStaff table.

```
ALTER TABLE tempStaff
ALTER FirstName SET DEFAULT 'unknown';
```

- We will add CHECK to StaffDeptID of the tempStaff table stating that it need to be more than or equat to 0.

```
ALTER TABLE tempStaff
ADD CONSTRAINT CHK_StaffDeptID CHECK
(StaffDeptID >=0);
```

2.3 Constraints (cont.)

- You can show constraints on a particular table using the following SQL:

```
SELECT TABLE_NAME, CONSTRAINT_TYPE,
CONSTRAINT_NAME
FROM information_schema.table_constraints
WHERE table_name='tempStaff';
```

	TABLE_NAME	CONSTRAINT_TYPE	CONSTRAINT_NAME
▶	tempStaff	UNIQUE	Email
	tempStaff	PRIMARY KEY	PRIMARY
	tempStaff	FOREIGN KEY	FK_tempStaff_DeptID
	tempStaff	CHECK	CHK_StaffDeptID
	tempStaff	CHECK	tempstaff_chk_1

2.3 Constraints (cont.)

- Drop constraints
- For DEFAULT constraint,

```
ALTER TABLE tempStaff
ALTER FirstName DROP DEFAULT;
```

- For the constraints that have name, you can drop them using:

```
ALTER TABLE <table_name>
DROP CONSTRAINT <constraint_name>;
```

- Example

```
ALTER TABLE tempStaff
DROP CONSTRAINT CHK_StaffDeptID;
```

THANKS FOR YOUR ATTENTION ANY QUESTIONS?