



LAB 2

DATA DEFINITION LANGUAGE (DDL) #2

By Kanokwan Atchariyachanvanich

Faculty of Information Technology

KMITL

Database System Concepts

2/2565

OUTLINE ก่อนสอบกลางภาค

Date	SQL
9, 11 JAN 2023	<ul style="list-style-type: none">• Lab Introduction• Introduction to DBLearn (SQL tool)
16, 18 JAN 2023	LAB 1 - Creating and Managing Tables (DDL)
23, 25 JAN 2023	LAB 2 - Including Constraints (DDL)
30 JAN, 1 FEB 2023	LAB 3 - Manipulating Data (DML)
6, 8 FEB 2023	LAB 4 - SQL SELECT Statements <ul style="list-style-type: none">• Writing Basic
13, 15 FEB 2023	LAB 5 - SQL SELECT Statements <ul style="list-style-type: none">• Restricting and Sorting Data
20, 22 FEB 2023	- ทวนก่อนสอบ Quiz 1
27 FEB, 1 MAR 2023	Quiz 1: LAB 1 – LAB 5

OBJECTIVES

After completing this lesson, you should be able to do the following:

- Describe constraints**
- Create and maintain constraints**

WHAT ARE CONSTRAINTS?

- **Constraints enforce rules at the table level.**
- **Constraints prevent the deletion of a table if there are dependencies.**
- **The following constraint types are valid:**
 - **NOT NULL**
 - **UNIQUE**
 - **PRIMARY KEY**
 - **FOREIGN KEY**

DATA INTEGRITY CONSTRAINTS

Constraint	Description
NOT NULL	กำหนดต้องมีค่าในคอลัมน์ที่กำหนด คือไม่สามารถมีค่า NULL ในคอลัมน์นี้ได้
UNIQUE	กำหนด 1 คอลัมน์หรือรวมกันมากกว่า 1 คอลัมน์โดยค่าในคอลัมน์เหล่านี้ต้องแตกต่างกันจากทุกแถวในตารางเดียวกัน
PRIMARY KEY	ระบุว่าแต่ละแถวของตารางมีความแตกต่างกัน
FOREIGN KEY	กำหนดและบังคับใช้ความสัมพันธ์ foreign key relationship ระหว่างคอลัมน์ในตารางใดๆ และคอลัมน์ของตารางที่ถูกอ้างอิง

CONSTRAINT GUIDELINES

- สามารถกำหนด Constraint ด้วย 2 วิธีคือ
 1. ในขณะที่สร้างตาราง
 2. หลังจากสร้างตารางแล้ว
- สามารถนิยาม constraint ได้ที่ระดับคอลัมน์หรือระดับตาราง
- สามารถดูการกำหนด constraint ได้จาก

```
SELECT    TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME  
  
FROM      INFORMATION_SCHEMA.KEY_COLUMN_USAGE ;
```

ชนิด CONSTRAINTS

1. Column Constraint level ระดับคอลัมน์ :

- References a single column and is defined within a specification for the owning column;
- can define any type of integrity constraint.

2. Table Constraint level ระดับตาราง :

- References one or more column and is defined separated from the definitions of the column in the table;
- can define any constraints except NOT NULL.

DEFINING CONSTRAINTS

```
CREATE TABLE table_name (  
    column datatype [column_constraint] ,  
    column datatype [column_constraint] ,  
    ... ,  
    [table_constraint] [, ...]  
);
```

- Table_name is the name of the table
- Column is the name of the column
- Column_constraint is an integrity constraint as part of the column definition
- Table_constraint is an integrity constraint as part of the table definition

1. COLUMN CONSTRAINT

- can define any type of integrity constraint.

Column datatype **constraint_type**,

```
CREATE TABLE employ (  
    employee_id INT(6),  
    first_name   VARCHAR(20),  
    ...  
    hire_date    DATE NOT NULL,  
    PRIMARY KEY (employee_id )  
);
```

Column
constraint

2. TABLE CONSTRAINT

- can define any constraints except NOT NULL.

Last Column, . . .

constraint_type **`constraint_name`** (**column**),

CREATE TABLE **employ** (

employee_id **INT(6)**,

first_name **VARCHAR(20)**,

...

hire_date **DATE** **NOT NULL**,

PRIMARY KEY **`employ_employee_id`** (**employee_id**)

);

Table
constraint

Column_name ที่เป็น
primary key

THE **NOT NULL** CONSTRAINT

- Ensures that null values are not permitted for the column:

EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10



NOT NULL constraint
(ไม่มีแถวไหนที่มีค่าเป็น Null
สำหรับคอลัมน์ last_name)



NOT NULL constraint
(ไม่มีแถวไหนที่มีค่าเป็น Null สำหรับ
คอลัมน์ hire_date)



**ABSENCE OF NOT
NULL constraint**
(มีบางแถวที่มีค่าเป็น Null สำหรับ
คอลัมน์ department_id)

THE NOT NULL CONSTRAINT

- Is defined only at the column level:


```
CREATE TABLE employees(  
    employee_id    INT,  
    last_name      VARCHAR(25) NOT NULL ,  
    salary         INT ,  
    commission_pct decimal (4, 2) ,  
    hire_date      DATE NOT NULL  
);
```

THE **UNIQUE** CONSTRAINT


- A UNIQUE key integrity constraint requires that every value in a column or set of column (key) be unique— that is, no two rows of a table can have duplicate values in a specified column or set of column.
- UNIQUE constraints allow the input of nulls unless you also define NOT NULL constraints for the same columns.
- UNIQUE constraints can be defined at the column or table level.
- A composite unique key is created by using the table level definition.

THE **UNIQUE** CONSTRAINT

**UNIQUE
constraint**



EMPLOYEE_ID	LAST_NAME	EMAIL
100	King	SKING
101	Kochhar	NKOCHHAR
102	De Haan	LDEHAAN
103	Hunold	AHUNOLD
104	Ernst	BERNST



EMPLOYEE_ID	LAST_NAME	EMAIL
208	Smith	JSMITH
209	Smith	JSMITH



Allowed
Not allowed;
Already exists

EXAMPLE: THE **UNIQUE** CONSTRAINT

- Defined at either the table level or the column level:

```
CREATE TABLE employ_uu(  
    employee_id      INT,  
    last_name        VARCHAR(25) NOT NULL ,  
    email            VARCHAR(25) ,  
    salary            INT ,  
    commission_pct   decimal (4,2) ,  
    hire_date        DATE  NOT NULL,  
    UNIQUE employ_uu_email_uk (email) );
```

THE PRIMARY KEY CONSTRAINT

- แต่ละตารางสามารถมีได้เพียง 1 primary key
- Primary key constraint คือคอลัมน์หรือกลุ่มของคอลัมน์ที่ระบุว่าแต่ละแถวในตารางจะไม่ซ้ำกัน
- Primary key constraint บังคับใช้เอกลักษณ์ของคอลัมน์หรือการรวมกันของคอลัมน์
- Primary key constraint ทำให้แน่ใจว่า ไม่มี คอลัมน์ที่เป็นส่วนหนึ่งของคีย์หลักสามารถมีค่า Null ได้

THE PRIMARY KEY CONSTRAINT

DEPARTMENT

PRIMARY KEY



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

Not allowed
(NULL value)



INSERT INTO

	Public Accounting		1400
50	Finance	124	1500

Not allowed
(50 already exists)



EXAMPLE: THE **PRIMARY KEY** CONSTRAINT

- Defined at either the table level or the column level:

```
CREATE TABLE depart(  
    department_id INT(4) PRIMARY KEY,  
    department_name VARCHAR(30) NOT NULL,  
    manager_id INT(6),  
    location_id INT(4)  
);
```

Column-level
constraint

NOTE: A table can have only one **PRIMARY KEY** constraint but can have several **UNIQUE** constraints.

EXAMPLE: THE **PRIMARY KEY** CONSTRAINT

- Defined at either the table level or the column level:

```
CREATE TABLE deps(  
    department_id    INT(4),  
    department_name  VARCHAR(30) NOT NULL,  
    manager_id       INT(6),  
    location_id      INT(4),  
    PRIMARY KEY `dept_id_pk` (department_id) );
```

Table-level constraint

NOTE: A table can have only one PRIMARY KEY constraint but can have several UNIQUE constraints.

THE **FOREIGN KEY** CONSTRAINT (REFERENTIAL INTEGRITY CONSTRAINT)

DEPARTMENT (parent table)

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

**PRIMARY
KEY**

EMPLOYEES (child table)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60

FOREIGN KEY

INSERT INTO

200	Ford	9
201	Ford	60

**Not allowed
(9 does not exist)**

Allowed



FOREIGN KEY CONSTRAINT **KEYWORDS**

- **FOREIGN KEY**: นิยามคอลัมน์ในตารางลูกที่ระดับ table constraint
- **REFERENCES**: กำหนดตารางและคอลัมน์ในตารางแม่
- **ON DELETE CASCADE**: ลบ dependent rows ในตารางลูก เมื่อแถวในตารางแม่ถูกลบ
- **ON DELETE SET NULL**: แปลงค่าที่อยู่ใน foreign key เป็นค่า null
- ถ้าไม่มีคีย์เวิร์ด ON DELETE CASCADE หรือ ON DELETE SET NULL จะไม่สามารถลบแถวใน parent table ได้ถ้ามันถูกอ้างอิงในตารางลูก

EXAMPLE: THE **FOREIGN KEY** CONSTRAINT

- Defined at the table level :

```
CREATE TABLE Persons
(
    PersonID    int Primary Key,
    LastName    varchar(255),
    FirstName    varchar(255),
    Address      varchar(255),
    City         varchar(255)
);
```

```
CREATE TABLE Orders
(
    OrderID      int NOT NULL,
    OrderNumber  int NOT NULL,
    PersonID      int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES
    Persons(PersonID)
    ON DELETE CASCADE
);
```

คำสั่งเรียกดู CONSTRAINTS

```
SELECT constraint_name,table_name,constraint_type  
FROM    information_schema.table_constraints  
WHERE   table_name='ชื่อตารางที่ต้องการแสดง constraint'
```

Example:

```
SELECT constraint_name, table_name, constraint_type  
FROM    information_schema.table_constraints  
WHERE   table_name='countries'
```

SUMMARY

- In this lesson, you should have learned how to create constraints when creating table.
- Types of constraints:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
- You can query the `INFORMATION_SCHEMA.KEY_COLUMN_USAGE` table to view all constraint definitions and names.

EXERCISE 1: จงสร้างตารางชื่อ MEMBER ซึ่งมีโครงสร้างดังต่อไปนี้ (COLUMN-LEVEL CONSTRAINTS)

Column name	MEMBER_ID	LAST_NAME	FIRST_NAME	ADDRESS	CITY
Key Type	Primary key				
NULL/ UNIQUE	NN, U	NN			
Data type	INT	VARCHAR	VARCHAR	VARCHAR	VARCHAR
Length	10	25	25	100	30

EXERCISE 2: จงสร้างตารางชื่อ MEMBER2 ซึ่งมีโครงสร้างดังต่อไปนี้ (TABLE-LEVEL CONSTRAINT)

Column name	MEMBER_ID	LAST_NAME	FIRST_NAME	ADDRESS	CITY
Key Type	Primary key				
NULL/ UNIQUE	NN, U	NN			
Data type	INT	VARCHAR	VARCHAR	VARCHAR	VARCHAR
Length	10	25	25	100	30

CONSTRAINT GUIDELINES

- สามารถกำหนด Constraint ด้วย 2 วิธีคือ

1. ในขณะที่สร้างตาราง

2. หลังจากสร้างตารางแล้ว

- สามารถดูการกำหนด constraint ได้จาก

```
SELECT    TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME  
  
FROM      INFORMATION_SCHEMA.KEY_COLUMN_USAGE ;
```

CONSTRAINTS FOR EXISTING TABLES

Use the **ALTER TABLE** statement to:

- Add or drop a constraint, but not modify its structure
- Add a **NOT NULL** constraint by using the **MODIFY** clause

ADDING A CONSTRAINT SYNTAX

- คำสั่งเพิ่ม Constraint
- Note: สามารถกำหนด NOT NULLให้กับคอลัมน์ได้ ก็ต่อเมื่อคอลัมน์นั้นว่าง
ไม่มีข้อมูล หรือ ทุกแถวในคอลัมน์นั้นมีค่า

```
ALTER TABLE table_name  
ADD [CONSTRAINT constraint_name]  
constraint_type (column) ;
```

Exercise 3 : Adding Primary Key Constraint

- จงเพิ่ม PRIMARY KEY constraint ให้กับตาราง lab_emp

Exercise 4 : Adding Primary Key Constraint

- จงเพิ่ม PRIMARY KEY constraint ให้กับตาราง lab_location

ADDING A FOREIGN KEY CONSTRAINT

- ตัวอย่าง : เพิ่ม FOREIGN KEY constraint ชื่อ constraint_name ให้กับตาราง table_name ซึ่งมี Parent_name เป็น primary key ใน parent_table
- หมายเหตุ ต้องกำหนด parent_column เป็น primary ก่อนถึงจะสามารถเพิ่ม foreign key เชื่อมไปหาได้

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name
FOREIGN KEY (column_name)
REFERENCES parent_table(parent_column);
```


Exercise 5 : Adding Foreign Key Constraint

- จงเพิ่ม FOREIGN KEY constraint ชื่อ lab_emp_fk ให้กับตาราง lab_emp โดยมี Parent table คือ lab_location

ADDING A UNIQUE CONSTRAINT

- ตัวอย่าง : เพิ่ม UNIQUE constraint ชื่อ constraint_name ให้กับ column_name ใน ตาราง table_name

```
ALTER TABLE table_name
ADD CONSTRAINT constraint_name
UNIQUE (column_name) ;
```

Exercise 6 : Adding UNIQUE Constraint

- จงเพิ่ม UNIQUE constraint ชื่อ loc_name_un ให้กับ location_name ในตาราง lab_location

ADDING A NOT NULL CONSTRAINT

- คำสั่งเพิ่ม NOT NULL Constraint ให้กับคอลัมน์ได้ ก็ต่อเมื่อคอลัมน์นั้นมีค่า
 - 1. ตรวจสอบค่าปัจจุบันที่อยู่ในคอลัมน์ โดยใช้คำสั่ง
SELECT * FROM table_name WHERE column_name IS NULL ;
 - 2. อัปเดตค่าที่เป็น NULL ให้เป็นค่าที่ไม่ใช่ NOT NULL
UPDATE table_name SET column_name = "new_value"
WHERE column_name IS NULL ;
 - 3. เพิ่ม NOT NULL constraint

```
ALTER TABLE table_name  
MODIFY column_name datatype NOT NULL ;
```

Example 7: A NOT NULL CONSTRAINT

- จงเพิ่ม NOT NULL constraint ให้กับ last_name และ first_name ในตาราง lab_emp

DROPPING A CONSTRAINT SYNTAX

○ คำสั่ง ลบ Constraint

```
ALTER TABLE      table_name  
DROP PRIMARY KEY ;
```

```
ALTER TABLE      table_name  
DROP FOREIGN KEY constraint_name ;
```

DROPPING A PK/FK CONSTRAINT

- ตัวอย่าง : ลบ PRIMARY KEY constraint จากตาราง lab_emp และลบ FOREIGN KEY constraint ที่เกี่ยวข้อง (location_id) ในตาราง

```
ALTER TABLE      lab_emp
DROP PRIMARY KEY ;
```

- ตัวอย่าง : ลบ FOREIGN KEY Constraint จากตาราง lab_emp

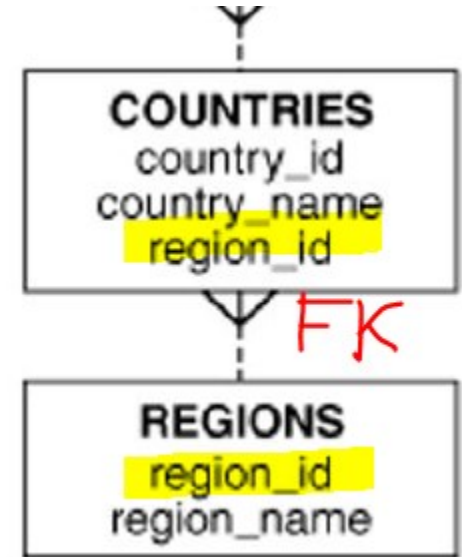
```
ALTER TABLE      lab_emp
DROP FOREIGN KEY lab_emp_fk;
```

ตรวจสอบชื่อ constraint ด้วยคำสั่ง

```
SELECT  TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME
FROM    INFORMATION_SCHEMA.KEY_COLUMN_USAGE ;
```

Exercise 8 : Drop **Primary Key constraint**

- จงลบ primary key constraint ของตาราง regions



Exercise 9 : Drop **Foreign Key constraint**

- จงลบ foreign key constraint ของตาราง locations

DROPPING A UNIQUE CONSTRAINT

- ตัวอย่าง : ลบ UNIQUE constraint จากตาราง lab_location จะลบด้วย ชื่อ Constraint ที่ตั้งไว้

```
ALTER TABLE      table_name  
DROP INDEX        constraint_name;
```

```
ALTER TABLE      lab_location  
DROP INDEX        loc_name_un;
```

ไม่มีคำสั่ง **DROP NOT NULL CONSTRAINT**

- แต่จะใช้ คำสั่งแก้ไข เป็น NULL

```
ALTER TABLE    table_name  
MODIFY          column_name datatype NULL;
```

- จงแก้ไข constraint ของ first_name ในตาราง lab_emp จาก NOT NULL เป็น NULL

```
ALTER TABLE    lab_emp  
MODIFY          first_name varchar(25) NULL;
```

VIEWING CONSTRAINTS

- Query the key_column_usage table to view all constraint definitions and names.

```
Select  table_name, column_name, constraint_name,  
        referenced_column_name, referenced_table_name  
  
From  
  
        information_schema.key_column_usage  
  
Where  table_name = 'lab_emp'
```

VIEWING THE COLUMNS ASSOCIATED WITH CONSTRAINTS

- View the columns associated with the constraint names in the `information_schema.key_column_usage`.

```
SELECT  constraint_name, column_name
FROM    information_schema.key_column_usage
WHERE    table_name = 'lab_location' ;
```

DROP TABLE with CONSTRAINT ลบตารางที่กำหนด FK CONSTRAINT ไว้

1. ไม่ลบ foreign key constraint

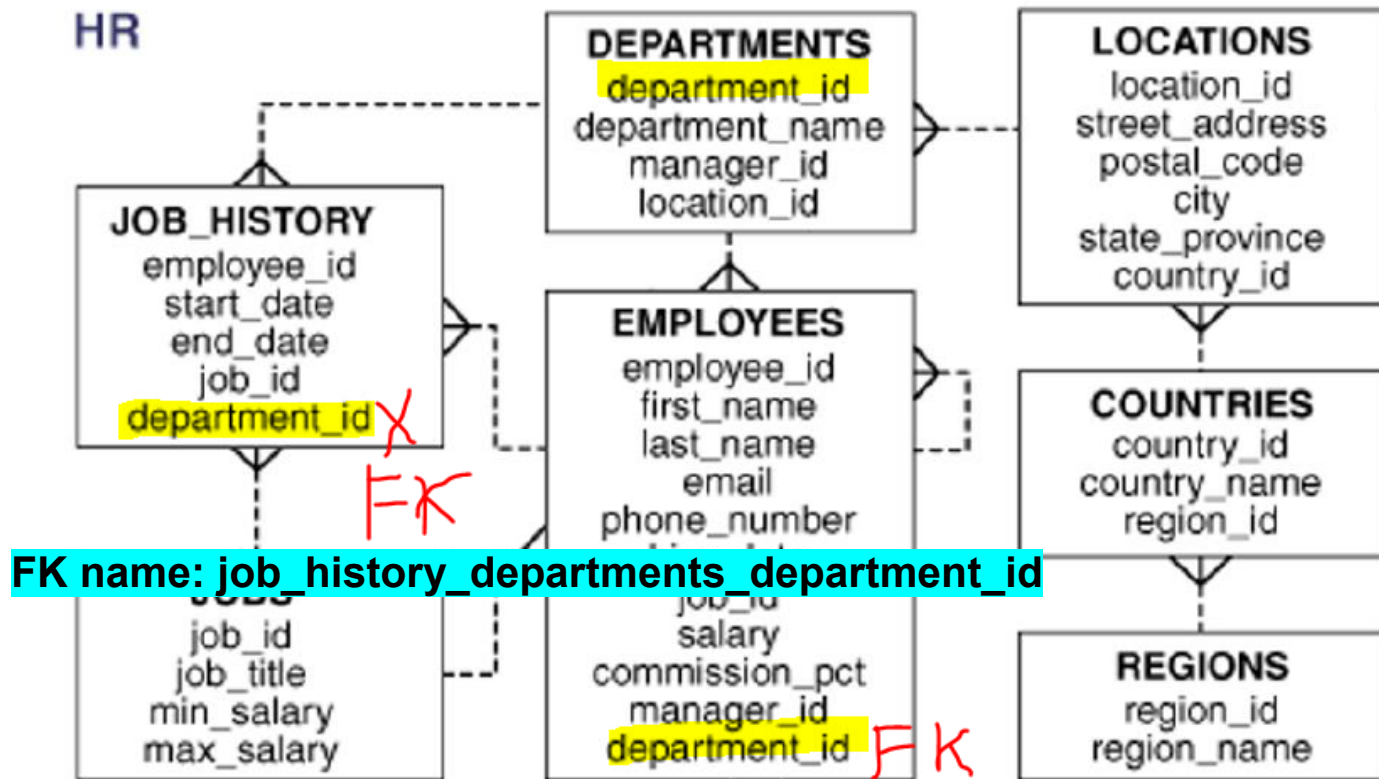
```
SET      FOREIGN_KEY_CHECKS = 0;  
  
DROP     TABLE  table_name;  
  
SET      FOREIGN_KEY_CHECKS = 1;
```

2. ลบ foreign key constraint (ต้องลบทั้งหมดที่เชื่อมโยงไปยังตารางอื่น)

```
ALTER TABLE  table_name  
DROP FOREIGN KEY constraint_name;  
DROP TABLE  table_name;
```

DROP TABLE with CONSTRAINT

- ลบตาราง departments ด้วยวิธีการที่ต้องลบ foreign key constraint



FK name: job_history_departments_department_id

FK name : employees_departments_department_id

Exercise 10 : Drop Table with constraint

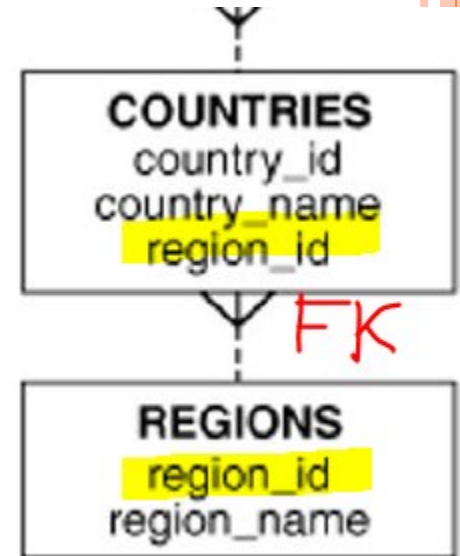
- จงลบตาราง departments ด้วยการลบ foreign key constraint

Exercise 11 : Drop Table with constraint

- จงลบตาราง regions ด้วยการไม่ลบ foreign key constraint

Exercise 12 : Drop Table with constraint

- จงลบตาราง regions ด้วยการลบ foreign key constraint



DROPPING A COLUMN with Constraint

1. ตรวจสอบ constraint name ด้วยคำสั่ง

```
SELECT  TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME  
FROM    INFORMATION_SCHEMA.KEY_COLUMN_USAGE ;
```

2. ลบ constraint นั้นออกจากตารางที่มีคอลัมน์ที่ต้องการลบ เช่น คอลัมน์นั้นเป็น foreign key จำเป็นต้องลบ constraint ของคอลัมน์นี้ ด้วยคำสั่ง

```
ALTER TABLE    table_name  
DROP foreign key constraint_name ;
```

3. ลบคอลัมน์นั้นออกจากตาราง ด้วยคำสั่ง

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

VIEWING CONSTRAINS

```
SELECT  TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME
FROM    INFORMATION_SCHEMA.KEY_COLUMN_USAGE ;
```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME
countries	country_id	PRIMARY
countries	region_id	countries_regions_region_id
departments	department_id	PRIMARY
departments	manager_id	departments_ibfk_1
departments	location_id	departments_locations_location_id
employees	employee_id	PRIMARY
employees	department_id	employees_departments_department_id
employees	manager_id	employees_employees_employee_id
employees	job_id	employees_jobs_job_id
job_history	employee_id	employee_id
job_history	start_date	employee_id
job_history	department_id	job_history_departments_department_id
job_history	employee_id	job_history_employees_employee_id
job_history	job_id	job_history_jobs_job_id
jobs	job_id	PRIMARY
locations	location_id	PRIMARY
locations	country_id	locations_countries_country_id
regions	region_id	PRIMARY

EXERCISE 13 : DROPPING A COLUMN with Constraint

จงลบคอลัมน์ `job_id` ในตาราง `employees` และแสดงคำสั่งเพื่อดูโครงสร้างหลังจากการลบคอลัมน์