

# 05.CRUD-Query

---

# SQL ALTER TABLE Statement

- คำสั่ง ALTER เป็นคำสั่งในการแก้ไขและเปลี่ยนแปลงโครงสร้างตาราง โดยมีรูปแบบดังต่อไปนี้
  - สร้าง แก้ไข และลบคอลัมน์ในตารางที่มีอยู่แล้ว
  - เพิ่ม และลบ เงื่อนไข (constraints) ในการควบคุมค่าในคอลัมน์ของตารางที่มีอยู่แล้ว

Edit Type	Syntax	Example
ADD COLUMN	<b>ALTER TABLE</b> table_name <b>ADD</b> column_name datatype;	<b>ALTER TABLE</b> Customers <b>ADD</b> Email varchar(255);
DROP COLUMN	<b>ALTER TABLE</b> table_name <b>DROP COLUMN</b> column_name;	<b>ALTER TABLE</b> Customers <b>DROP COLUMN</b> Email;
ALTER/MODIFY COLUMN	<b>ALTER TABLE</b> table_name <b>ALTER COLUMN</b> column_name datatype	<b>ALTER TABLE</b> Persons <b>ALTER COLUMN</b> DateOfBirth year;

# SQL Constraints

- คำสั่ง Constraints เป็นคำสั่งในการสร้างกฎเพื่อควบคุมค่าในคอลัมน์ให้ถูกต้องอยู่เสมอ ระบบฐานข้อมูลจะไม่อนุญาตให้บันทึกข้อมูลลงในคอลัมน์ กรณีที่ค่าดังกล่าวไม่สอดคล้องกับกฎเกณฑ์ที่สร้างไว้

CONSTRAINTS	DESCRIPTION
NOT NULL	ควบคุมค่าในคอลัมน์ไม่เป็นค่าว่าง หรือมีค่าเสมอ
UNIQUE	ควบคุมค่าไม่ซ้ำกัน
PRIMARY KEY	กำหนดให้คอลัมน์เป็นคีย์หลัก
FOREIGN KEY	กำหนดให้คอลัมน์เป็นคีย์นอก ที่เชื่อมโยงความสัมพันธ์ไปยังอีกตารางหนึ่ง
CHECK	กำหนดให้มีการบันทึกค่าในคอลัมน์ได้ก็ต่อเมื่อค่าดังกล่าวก็ต่อเงื่อนไขตรวจสอบนั้น
DEFAULT	กำหนดให้ค่าปริยายของคอลัมน์ในกรณีที่ไม่มีระบุค่า
CREATE INDEX	ใช้สำหรับการสร้างดัชนีในการสืบค้นและสร้างข้อมูลได้อย่างรวดเร็ว

# SQL ALTER TABLE Statement

Edit Type	CREATE	ALTER
NOT NULL	<pre>CREATE TABLE Persons (   ID int NOT NULL,   LastName varchar(255) NOT NULL,   FirstName varchar(255) NOT NULL,   Age int );</pre>	<pre>ALTER TABLE Persons MODIFY Age int NOT NULL;</pre>
UNIQUE	<pre>CREATE TABLE Persons (   ID int NOT NULL UNIQUE,   LastName varchar(255) NOT NULL,   FirstName varchar(255),   Age int );</pre>	<pre>ALTER TABLE Persons ADD CONSTRAINT UC_Person UNIQUE (ID,LastName);</pre>
		<pre>ALTER TABLE Persons DROP CONSTRAINT UC_Person;</pre>
PRIMARY KEY	<pre>CREATE TABLE Persons (   ID int NOT NULL PRIMARY KEY,   LastName varchar(255) NOT NULL,   FirstName varchar(255),   Age int );</pre>	<pre>ALTER TABLE Persons ADD PRIMARY KEY (ID);</pre> <p>หรือ</p> <pre>ALTER TABLE Persons ADD CONSTRAINT PK_Person PRIMARY KEY (ID,LastName);</pre>
		<pre>ALTER TABLE Persons DROP PRIMARY KEY;</pre> <p>หรือ</p> <pre>ALTER TABLE Persons DROP CONSTRAINT PK_Person;</pre>

# SQL ALTER TABLE Statement

Edit Type	CREATE	ALTER
FOREIGN KEY	<b>CREATE TABLE</b> Orders ( OrderID int <b>NOT NULL</b> PRIMARY KEY, OrderNumber int <b>NOT NULL</b> , PersonID int <b>FOREIGN KEY</b> REFERENCES Persons(PersonID) );	<b>ALTER TABLE</b> Orders <b>ADD FOREIGN KEY</b> (PersonID) REFERENCES Persons(PersonID); หรือ <b>ALTER TABLE</b> Orders <b>ADD CONSTRAINT</b> FK_PersonOrder <b>FOREIGN KEY</b> (PersonID) REFERENCES Persons (PersonID);
		<b>ALTER TABLE</b> Orders <b>DROP CONSTRAINT</b> FK_PersonOrder;
CHECK	<b>CREATE TABLE</b> Persons ( ID int <b>NOT NULL</b> , LastName varchar(255) <b>NOT NULL</b> , FirstName varchar(255), Age int <b>CHECK</b> (Age>=18) );	<b>ALTER TABLE</b> Persons <b>ADD CHECK</b> (Age>=18); หรือ <b>ALTER TABLE</b> Persons <b>ADD CONSTRAINT</b> CHK_PersonAge <b>CHECK</b> (Age>=18 AND City='Sandnes');
		<b>ALTER TABLE</b> Persons <b>DROP CONSTRAINT</b> CHK_PersonAge;

# SQL ALTER TABLE Statement

Edit Type	CREATE	ALTER
DEFAULT	<b>CREATE TABLE</b> Persons ( ID int <b>NOT NULL</b> , LastName varchar(255) <b>NOT NULL</b> , FirstName varchar(255), Age int, City varchar(255) <b>DEFAULT</b> 'Sandnes' );	<b>ALTER TABLE</b> Persons <b>ADD CONSTRAINT</b> df_City <b>DEFAULT</b> 'Sandnes' <b>FOR</b> City;
		<b>ALTER TABLE</b> Persons <b>ALTER COLUMN</b> City <b>DROP DEFAULT</b> ;
CREATE INDEX	<b>CREATE INDEX</b> idx_lastname <b>ON</b> Persons (FirstName, LastName);	<b>DROP INDEX</b> Persons.idx_lastname;

# SQL CREATE VIEW Statement

- วิวเป็นตารางเสมือน (virtual table) ที่สร้างจากตารางผลลัพธ์ของคำสั่ง SQL ซึ่งมีแถวและคอลัมน์เหมือนตารางข้อมูล โดยที่คอลัมน์อาจจะมาจากตารางเดียวหรือหลายตารางก็ได้

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

เช่น

```
CREATE VIEW [Brazil Customers] AS  
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'Brazil';
```

```
CREATE VIEW [Products Above Average Price] AS  
SELECT ProductName, Price  
FROM Products  
WHERE Price > (SELECT AVG(Price) FROM Products);
```

# Grant/Revoke Privileges

คำสั่ง GRANT ให้สิทธิการใช้งาน (Privileges) ให้กับผู้ใช้งาน

**GRANT privileges ON object TO user\_name;**

คำสั่ง REVOKE ถอนสิทธิการใช้งาน (Privileges) ออกจากผู้ใช้งาน

**REVOKE privileges ON object FROM user\_name;**

Privilege	Description
SELECT	Ability to perform SELECT statements on the table.
INSERT	Ability to perform INSERT statements on the table.
UPDATE	Ability to perform UPDATE statements on the table.
DELETE	Ability to perform DELETE statements on the table.
REFERENCES	Ability to create a constraint that refers to the table.
ALTER	Ability to perform ALTER TABLE statements to change the table definition.
ALL	ALL does not revoke all permissions for the table. Rather, it revokes the ANSI-92 permissions which are SELECT, INSERT, UPDATE, DELETE, and REFERENCES.

เช่น

**GRANT SELECT ON Customers TO User1**

**REVOKE INSERT ON Customers FROM User1**