

## Constraints

Constraints คือ ข้อบังคับหรือเงื่อนไขในการอนุญาตให้เก็บเฉพาะข้อมูลที่เหมาะสมลงในฐานข้อมูลเพื่อให้การเลือกข้อมูลจากฐานข้อมูลเป็นไปอย่างถูกต้อง ประโยชน์อย่างอื่นของการมีเงื่อนไข ได้แก่

เพื่อตรวจสอบข้อผิดพลาดในการกรอกข้อมูลลงในฐานข้อมูล

เพื่อความถูกต้องในการปรับปรุงข้อมูล

เพื่อรักษาความถูกต้องของข้อมูลโดยรวม

เพื่อบอกให้ฐานข้อมูลทราบว่าผู้ใช้ต้องการจะเก็บข้อมูลหรือค้นหาข้อมูล

รูปแบบของการแสดงฐานข้อมูล เช่น E/R model เงื่อนไขที่สามารถแสดงได้มีเพียงการประกาศคีย์ ชนิดความสัมพันธ์ ลักษณะการเป็นสมาชิก และความซ้ำซ้อนเท่านั้น ต่างจาก Relational model หรือ SQL ที่สามารถรองรับเงื่อนไขที่มากกว่านี้ได้ คือ

Key constraint

Not Null constraint

Referential Integrity constraint

Check

Assertion constraint

ในการประกาศและบังคับใช้เงื่อนไขเพื่อให้มีผลกับฐานข้อมูล สามารถทำได้โดยการประกาศไว้ในรูปแบบของความสัมพันธ์ หรือประกาศภายหลัง ซึ่งจะมีผลใช้กับฐานข้อมูลที่ใช้อยู่ปัจจุบัน เมื่อประกาศเงื่อนไขบังคับไปแล้ว และมีเหตุการณ์ใดที่ทำให้ผิดเงื่อนไข เช่น คำสั่ง SQL ระบบจัดการฐานข้อมูลก็จะต้องยกเลิกเหตุการณ์หรือคำสั่งนั้นและแจ้งข้อผิดพลาดให้ผู้ใช้ทราบทันที ตัวอย่างที่จะกล่าวต่อไปจะอ้างอิงโดยใช้คำสั่ง SQL ซึ่งเป็นคำสั่งที่นิยมใช้มากในการค้นหาข้อมูลจากฐานข้อมูล

Key constraint

ในฐานข้อมูลเชิงสัมพันธ์มีเงื่อนไขเกี่ยวกับคีย์อยู่ 2 แบบ คือ

คีย์หลัก เมื่อมีการให้ค่าแก่ คีย์หลัก ในความสัมพันธ์ใดๆ ก็แล้วแต่ ค่าของคีย์ชนิดนี้จะไม่เป็น Null เสมอ (Not Null) และจะถูกใช้เป็นดัชนีในการเรียงลำดับเสมอ เมื่อผู้ใช้ค้นหาข้อมูลจากความสัมพันธ์โดยใส่ค่า คีย์หลักเป็น Null ก็จะไม่พบข้อมูลที่ต้องการ ดังนั้นผู้ใช้จึงจำเป็นต้องทราบถึงเงื่อนไขนี้ เพื่อให้ได้ข้อมูลที่ถูกต้องจากการค้นหา

การกำหนดคีย์ใน SQL สามารถทำได้สองแบบ คือ การกำหนดให้แอททริบิวต์แต่ละตัว หรือกำหนดไว้ตั้งแต่ต้นเมื่อสร้างความสัมพันธ์ ดังตัวอย่าง

```
CREATE TABLE Emp (EmpID integer คีย์หลัก,
```

```
Name char(40),
```

```
Address char(80)
```

```
Dept char(3)
```

```
Office integer,
```

```
Phone char(10)
```

```
UNIQUE (name,address))
```

Unique ความสัมพันธ์ใดๆ สามารถมีแอททริบิวต์ที่เป็นเอกลักษณ์ได้มากกว่าหนึ่งนอกจาก คีย์หลัก โดยการกำหนดให้แอททริบิวต์นั้นเป็น Unique และทุกแอททริบิวต์ที่เป็น Unique นั้นจะถูกใช้เป็นดัชนีในการเรียงรายการในความสัมพันธ์นั้นโดยอัตโนมัติ ต่อจาก คีย์หลัก ตัวอย่างดังการสร้างตารางในภาษา SQL ข้างบน

Referential Integrity

การอ้างอิงถึงข้อมูลจากความสัมพันธ์อื่น มีเงื่อนไข คือ

แอททริบิวต์ที่อ้างอิงมาจากความสัมพันธ์อื่นจะต้องเป็น คีย์หลัก เสมอ เพื่อป้องกันไม่ให้เกิดการซ้ำซ้อนของข้อมูลที่อ้างอิงได้

แอททริบิวต์ที่ถูกอ้างอิงและนำมาอยู่ในความสัมพันธ์จะถูกเรียกว่า Foreign key

ในการบังคับใช้ Referential Integrity จาก SQL สามารถประกาศไว้กับตัวแอททริบิวต์โดยตรงหรือประกาศแยกไว้ในความสัมพันธ์ที่ต้องการอ้างอิงถึงแอททริบิวต์นั้น

```
CREATE TABLE Apply(ID integer REFERENCES Emp(EmpID),
```

```
Location char(20),
```

```
Dept char(3),
```

```
FOREIGN KEY (location) REFERENCES Building(location))
```

## Check

การตรวจสอบค่าของข้อมูลในความสัมพันธ์เป็นเงื่อนไขอย่างหนึ่งที่จะต้องทำตาม เพื่อกรองเฉพาะข้อมูลที่เหมาะสมลงไปเก็บในฐานข้อมูล คำสั่ง CHECK ในภาษา SQL สามารถใช้ร่วมกับคำสั่งสร้างตารางความสัมพันธ์ CREATE TABLE ใน SQL เพื่อตรวจสอบการรับข้อมูลว่าสมควรจะเก็บข้อมูลแถวนั้นในตารางหรือไม่ ยกตัวอย่างเช่น ความสัมพันธ์ Emp มีการเพิ่มอายุของพนักงาน จึงต้องมีการตรวจสอบว่าพนักงานมีอายุเกิน 20 ปีหรือไม่

```
CREATE TABLE Emp (EmpID integer คีย์หลัก,
```

```
Name char(40),
```

```
Age integer,
```

```
Address char(80)
```

```
Dept char(3)
```

```
Office integer,
```

```
Phone char(10)
```

```
CHECK (age > 20))
```

```
Assertion
```

Assertion คือ เงื่อนไขที่ใช้ทั่วไปในการตรวจสอบความถูกต้องโดยรวมของทุกความสัมพันธ์หรือทุกฐานข้อมูล ซึ่งสามารถใส่เป็นเงื่อนไข เช่น การเปรียบเทียบได้โดยตรง การสร้าง Assertion โดยใช้ SQL สามารถทำได้โดยใช้คำสั่งในรูปแบบต่อไปนี้

```
CREATE ASSERTION <name> CHECK (<condition>)
```

อาศัยตัวอย่างความสัมพันธ์ Emp ที่ปรับปรุงแล้วข้างต้น

```
CREATE ASSERTION ageover CHECK (NOT EXISTS (SELECT age FROM Emp) > 20)
```

Assertion ที่สร้างไว้จะเป็นเงื่อนไขที่ทำให้ระบบจัดการฐานข้อมูลคอยเช็คว่าการผิดเงื่อนไขตามที่กำหนดไว้ใน Assertion หรือไม่ ถ้าเกิดการผิดพลาดระบบจัดการฐานข้อมูลก็จะรายงานต่อผู้ใช้ด้วยชื่อของ Assertion ที่สร้างไว้ทันที

ที่มา <http://www.srisangworn.go.th/home/databaselearnx/ms3t1-1.htm>