

## วิว(View)

การควบคุมความปลอดภัยให้กับข้อมูลสามารถสร้าง โครงสร้างข้อมูลใหม่ ที่ทำให้ผู้ใช้ เห็นเพียงโครงสร้างบางส่วน ของฐานข้อมูลเท่านั้น ที่เป็นการป้องกันไม่ให้ผู้ใช้ได้เห็นข้อมูลทั้งหมดของฐานข้อมูล เราเรียกตารางข้อมูลประเภทนี้ว่า “ตารางเสมือน” หรือ “วิว” บางครั้งการออกแบบฐานข้อมูลในระดับกายภาพก็ต้องกำหนดตารางเป็นลักษณะวิว เพื่อเป็นการป้องกันรักษาความปลอดภัยของข้อมูลและการควบคุมการใช้งานในระบบฐานข้อมูล เพราะหากสร้างเป็นตารางข้อมูลจริงอาจเกิดความซ้ำซ้อนของข้อมูลมากซึ่งยากต่อการควบคุมการใช้งาน

### 1. โครงสร้างของวิว

วิว(VIEW)หรือตารางเสมือนเป็นตารางข้อมูลที่มีรายละเอียดหรือได้รายละเอียดมาจากตารางหลัก วิว(view) ถูกสร้างขึ้นจากฐานข้อมูล โดยตารางที่สร้างขึ้นนี้จะสอดคล้องกับการใช้งานของผู้ใช้และยังเป็นการป้องกันข้อมูลที่แท้จริงภายในฐานข้อมูล วิว(view) ถูกสร้างขึ้นจากฐานข้อมูล โดยวิวที่สร้างขึ้นนี้จะสอดคล้องกับการใช้งานของผู้ใช้และยังเป็นการป้องกันข้อมูลที่แท้จริงภายในฐานข้อมูล ตารางเหล่านี้จะทำงานเช่นเดียวกับตารางธรรมดา แต่ไม่มีข้อมูลเป็นของตนเอง วิวใช้เป็นเครื่องมือในการควบคุมไม่ให้ผู้ใช้เข้าไปดูข้อมูลทั้งหมดของตารางได้ วิวเป็นกลไกรักษาความปลอดภัยในการปกปิดส่วนต่าง ๆ ของตารางที่เป็นความลับหรือเกินความจำเป็นสำหรับผู้ใช้ เช่น ถ้าต้องการให้พนักงานขายดูตารางข้อมูลพนักงานขายได้ แต่ไม่ต้องการให้เห็นค่าคอมมิชชั่นของแต่ละคน ก็ควรสร้างวิวของตารางพนักงานขายที่ไม่มีคอลัมน์ค่าคอมมิชชั่นไว้ให้พนักงานขายได้เรียกดู

ดังนั้นในการป้องกันและรักษาความปลอดภัยของข้อมูลในการเรียกใช้ข้อมูลที่แตกต่างจากตารางข้อมูลที่ออกแบบไปแล้ว ก็อาจทำได้โดยการสร้างเป็นวิว

### 2. การทำงานของวิว

เมื่อมีการเรียกใช้วิว ระบบจัดการฐานข้อมูลจะทำหน้าที่โดยเริ่มค้นหาข้อกำหนดของวิวในคำสั่ง SQL ที่เก็บไว้ในฐานข้อมูล แล้วแปลคำสั่งของวิวเพื่อไปนำข้อมูลมาจากตารางข้อมูลจริง ทำให้วิวรักษาความถูกต้องของโครงสร้างข้อมูล (integrity) ไว้ได้

สำหรับวิวแบบง่าย ๆ ระบบจัดการฐานข้อมูลอาจสร้างแต่ละแถวของวิวขึ้นมาจากตารางข้อมูลจริงเลย ส่วนวิวที่ซับซ้อนนั้นระบบจัดการฐานข้อมูลจะเก็บแถวของวิวไว้ในตารางชั่วคราว แล้วจึงแสดงผลจากตารางชั่วคราว และเลิกใช้ตารางชั่วคราวนั้น เมื่อหมดความต้องการใช้งานอีกต่อไป คือเมื่อสิ้นสุดคำสั่งของ SQL อย่างไรก็ตามไม่ว่าระบบจัดการฐานข้อมูลจะจัดการกับวิวอย่างไร ผลที่ได้ที่ผู้ใช้ได้รับก็ไม่แตกต่างกัน นั่นคือ วิวสามารถอ้างอิงได้จากคำสั่ง SQL เสมือนหนึ่งว่าอ้างอิงไปยังตารางข้อมูลจริงเลย

### 3. การสร้างวิว

การสร้างวิวได้จากการ query ในคำสั่ง SQL ทำให้เห็นข้อมูลในแถวและคอลัมน์ตามต้องการ การสร้างวิวอาจสร้างมาจาก

ตารางข้อมูลเดียวหรือมากกว่าหนึ่งตารางได้ ภาษา SQL จะมีการสร้างวิวโดยใช้คำสั่ง **CREATE VIEW**ซึ่งมีรูปแบบทั่วไปดังนี้

**CREATE VIEW** <view name>

[column1 , column2 ....>)]

AS <SELECT statement>;

CREATE VIEW เป็นคำสั่งที่ต้องมีทุกครั้งที่ต้องการสร้างวิว

view name ชื่อวิวที่ต้องการสร้าง

column 1 ชื่อของคอลัมน์ที่ต้องการตั้งชื่อในวิวที่สร้างขึ้น

AS <SELECT statement > เป็นไปตามเงื่อนไขในการเรียกค้น

ถ้าต้องการสร้างวิวชื่อ SALESOWN จากตารางพนักงานขาย( SALESTAB) โดยให้มีคอลัมน์ SALENO SALENAME และ ADDRESS จะใช้คำสั่งสร้างวิวดังนี้

CREATE VIEW SALESOWN

AS SELECT SALENO,SALENAME,ADDRESS

FROM SALESTAB;

ตาราง SALESTAB

SALENO	SALENAME	ADDRESS	SALECOM
1001	Chaiwat	Bangkok	0.12
1002	Mitree	Puket	0.13
1004	Benjawan	Bangkok	0.11
1007	Kanjana	Chiangmai	0.15
1003	Ternjai	Nonthaburi	0.10

วิว SALESOWN

SALENO	SALENAME	ADDRESS
1001	Chaiwat	Bangkok
1002	Mitree	Puket
1004	Benjawan	Bangkok
1007	Kanjana	Chiangmai
1003	Ternjai	Nonthaburi

ตารางที่ 12. 1 แสดงวิว SALESOWN

ผลของคำสั่งจะได้ตาราง SALESOWN ที่ไม่มีคอลัมน์ SALECOM อยู่ดังตารางที่ 12.1

ถ้าต้องการสร้างวิวชื่อ BANGKOKSTAFF จากตาราง SALESTAB โดยให้มีคอลัมน์ ADDRESS ที่เป็น “Bangkok” จะใช้คำสั่งดังนี้

```
CREATE VIEW BANGKOKSTAFF
```

```
AS SELECT *
```

```
FROM SALESTAB
```

```
WHERE ADDRESS = 'Bangkok';
```

ตาราง SALESTAB

SALENO	SALENAME	ADDRESS	SALECOM
1001	Chaiwat	Bangkok	0.12
1002	Mitree	Puket	0.13
1004	Benjawan	Bangkok	0.11
1007	Kanjana	Chiangmai	0.15
1003	Ternjai	Nonthaburi	0.10

## วิว BANGKOKSTAFF

SALENO	SALENAME	ADDRESS	SALECOM
1001	Chaiwat	Bangkok	0.12
1004	Benjawan	Bangkok	0.11

ตารางที่ 12.2 แสดงวิว BANGKOKSTAFF

ผลจากคำสั่งนี้จะทำให้ได้วิวที่ชื่อ BANGKOKSTAFF ดังตารางที่ 12.2 ซึ่งเป็นวิวที่สามารถใช้ได้เหมือนตารางอื่น ๆ คือสามารถสอบถามข้อมูล แก้ไขปรับปรุง นำไปใส่ ลบออกจากและรวมกับตารางอื่น ๆ ได้

ถ้าต้องการสอบถามข้อมูลจากวิว BANGKOKSTAFF

```
SELECT * FROM BANGKOKSTAFF;
```

ผลของคำสั่งนี้จะแสดงทุกคอลัมน์ในวิว BANGKOKSTAFF

วิวที่สร้างขึ้นนี้จะถูกเปลี่ยนแปลงโดยอัตโนมัติไปตามตารางข้อมูลที่สร้างวิว เช่น วิว BANGKOKSTAFF ถูกสร้างมาจากตารางพนักงานขาย (SALESTAB) ถ้าตารางพนักงานขาย (SALESTAB) มีการปรับปรุง โดยการเพิ่มพนักงานขายที่อยู่ใน Bangkok เข้าไปอีกคนหนึ่ง ในตารางพนักงานขาย (SALESTAB) พนักงานขายผู้นี้ก็จะปรากฏอยู่ในวิว BANGKOKSTAFF เองโดยอัตโนมัติ

การสร้างวิวให้เกิดประสิทธิภาพในการรักษาความปลอดภัยของข้อมูลและให้เกิดความเร็วในการสอบถามข้อมูลควรกำหนดขอบเขตโครงสร้างข้อมูลที่ต้องการ เพื่อจัดทำเป็นโครงสร้างของวิว โดยกำหนดเฉพาะชื่อคอลัมน์ไม่ต้องกำหนดชนิดข้อมูล เพราะจะเป็นไปตามที่กำหนดไว้ในตารางข้อมูลที่เลือกแล้ว นอกจากการกำหนดขอบเขตแล้วในการสร้างวิวจะระบุเฉพาะชื่อคอลัมน์ของวิวที่ต้องการเท่านั้น

## 4. การลบโครงสร้างของวิว

เมื่อต้องการลบโครงสร้างของวิวที่ถูกนิยามขึ้น สามารถทำได้ด้วยคำสั่ง DROP TABLE แล้วตามด้วยตารางที่ต้องการลบรูปแบบทั่วไปมีดังนี้

```
DROP VIEW < view name>;
```

DROP VIEW เป็นคำสั่งที่ต้องมีทุกครั้งที่ต้องการลบวิว

view name ชื่อวิวที่ต้องการลบ

ถ้าต้องการลบโครงสร้างตารางเสมือนชื่อ BANGKOKSTAFF

```
DROP VIEW BANGKOKSTAFF;
```

ผลของคำสั่งนี้จะทำให้วิว BANGKOKSTAFF ถูกลบไป

## 5 คุณสมบัติของวิว

วิวมีคุณสมบัติดังนี้

-วิวเกิดจากส่วนย่อยของข้อมูลจากตารางข้อมูลจริงได้มากกว่าหนึ่งตาราง

-ค่าที่ปรากฏบนวิวเป็นค่าจริงในตารางข้อมูล

-สามารถปรับปรุงข้อมูลในวิวได้ ถ้าได้รับการอนุญาตจากระบบจัดการฐานข้อมูล ได้แก่ การเพิ่มแถวในวิว(INSERT) การปรับปรุงคอลัมน์ในวิว(UPDATE) การลบข้อมูลในวิว(DELETE)

-สามารถกำหนดสิทธิการใช้งานให้ผู้ใช้งานได้ว่าจะให้ใช้วิวหรือไม่ให้ใช้ เช่นเดียวกับตารางข้อมูล โดยไม่ต้องยุ่งเกี่ยวกับตารางข้อมูลจริง

-สามารถกำหนดเงื่อนไขการเรียกใช้ลงในวิวอย่างถาวร เพื่อให้ง่ายต่อผู้ใช้

-วิวช่วยให้ทำงานง่ายขึ้น เพราะทำให้สามารถสอบถามข้อมูลที่ต้องการได้ง่ายขึ้น

ตัวอย่าง เมื่อสร้างวิวขึ้นมา ดังนี้

```
CREATE VIEW view-emp
```

```
SELECT EMPNO,ENAME,JOB,MGR,SAL
```

```
FROM demp
```

```
WHERE DEPTNO=10;
```

จากวิว view-emp สามารถใช้คำสั่ง SQL เพื่อสอบถามข้อมูลอย่างง่าย ๆ ดังนี้

```
SELECT ENAME,JOB
```

```
FROM view-emp
```

```
WHERE HIREDATE= '21-AMR-89';
```

หากไม่สร้างวิว ก็จะต้องใช้คำสั่ง SQL ที่ยุ่งยากกว่า เพื่อเรียกข้อมูลดังนี้

```
SELECT ENAME, JOB
```

```
FROM emp
```

```
WHERE DEPTNO=10 AND HIREDATE= '21-AMR-89';
```

วิวไม่ได้ช่วยเรื่องความเร็ว บางครั้งการใช้วิวก็มีปัญหาในเรื่องความเร็ว

ตัวอย่างที่เมื่อสร้างวิวขึ้นมา ดังนี้

```
CREATE VIEW vsummary (DEPTNO, MINSAL, MAXSAL, AVGSAL)
```

```
SELECT DEPTNO, MIN(SAL), MAX(SAL), AVG(SAL)
```

```
FROM emp
```

```
GROUP BY DEPTNO;
```

และเรียกข้อมูลจากวิวด้วยคำสั่งต่อไปนี้

```
SELECT DEPTNO, MINSAL
```

```
FROM vsummary WHERE DEPTNO = 20 ;
```

ผลก็คือเครื่องจะทำงานได้ผลช้ากว่าคำสั่งที่เรียกข้อมูลจากตารางข้อมูลโดยตรง ต่อไปนี้

```
SELECT DEPTNO, MIN(SAL) FROM emp
```

```
GROUP BY DEPTNO HAVING DEPTNO = 20;
```

หลีกเลี่ยงการสร้างวิวจากวิวหรือ join วิว เพราะจะทำให้ยุ่งยากซับซ้อนต่อการจัดการข้อมูล

## 6. ข้อดีของวิว

วิวเป็นข้อมูลที่ถูกคัดลอกออกมาจากฐานข้อมูลเพื่อเป็นการป้องกันและรักษาความปลอดภัยในระบบการจัดการฐานข้อมูล วิวมีความสำคัญต่อการรักษาความปลอดภัยในการจัดการฐานข้อมูล ข้อดีของวิวในระบบจัดการฐานข้อมูลได้แก่

6.1 วิวช่วยสนับสนุนการรักษาความปลอดภัยของข้อมูลตามมุมมองของผู้ใช้ได้อย่างมีประสิทธิภาพ ได้แก่ ผู้ใช้สามารถมีมุมมองต่างกันในเรื่องข้อมูลที่แตกต่างกันได้โดยการจัดรูปแบบของโครงสร้างข้อมูลของวิวนี้จะทำให้ได้ข้อมูลทั้งแถวและคอลัมน์ตามที่ผู้ใช้งานต้องการเท่านั้น

6.2 วิวช่วยรักษาความปลอดภัยในการปรับปรุงข้อมูลทั้งการเพิ่ม ลบ แก้ไข ข้อมูล ด้วยคำสั่ง INSERT , DELETE และ UPDATE โดยปรับปรุงข้อมูลเฉพาะบางส่วนของฐานข้อมูลผ่านวิว

6.3 วิวช่วยรักษาความปลอดภัยของข้อมูล โดยจำกัดผู้ใช้ไม่ให้เข้าถึงข้อมูลจริงและยังช่วยไม่ให้กระทบต่อฐานข้อมูล

6.4 วิวช่วยให้ง่ายต่อการสอบถามข้อมูล เพราะวิวสามารถเรียกข้อมูลจากหลายๆ ตารางและแสดงผลเสมือนตารางเดียว โดยเปลี่ยนการสอบถามจากตารางข้อมูลจริงหลายตารางมายังตารางเสมือนตารางเดียว

6.5 วิวมีความง่ายทางโครงสร้าง เพราะวิวจะให้โครงสร้างข้อมูลตามที่ผู้ใช้งานต้องการ โดยไม่ต้องคำนึงถึงเรื่องความซับซ้อนของข้อมูล

6.6 วิวไม่มีผลกระทบต่อการเปลี่ยนแปลงโครงสร้างตาราง วิวสามารถควบคุมความคงเส้นคงวา (consistent) ในการใช้งาน แม้ว่าจะมีความจำเป็นต้องปรับเปลี่ยนตารางข้อมูลจริงอย่างไรก็ตาม ไม่ว่าจะป็นชื่อตาราง ชื่อคอลัมน์ หรือ แม้แต่จะแบ่งแยกตารางข้อมูลออกเป็นหลายตารางก็ตาม

6.7 วิวช่วยให้ความถูกต้องของข้อมูล เพราะข้อมูลถูกเรียกใช้ผ่านวิว ไม่ได้ผ่านข้อมูลจริง ระบบจัดการฐานข้อมูลสามารถตรวจสอบข้อมูลว่าตรงกับกฎควบคุมความถูกต้องของโครงสร้างข้อมูลที่กำหนดไว้หรือไม่

## 7. ข้อดีของวิว

ข้อดีบางประการในการใช้วิวคือ

7.1 ข้อจำกัดในด้านประสิทธิภาพ เพราะการเรียกใช้ข้อมูลผ่านวิวที่มีความซับซ้อนมาก จะทำให้มีการเรียกคำสั่งสอบถามข้อมูลหลายคำสั่ง ดังนั้นแม้ว่าจะเรียกข้อมูลผ่านวิวที่ง่ายๆ แต่ถ้าวิวสร้างมาจากตารางข้อมูลที่เชื่อมกันอย่างซับซ้อน ก็จะต้องกินเวลาในการทำงานนาน

7.2 ข้อจำกัดในการปรับปรุงข้อมูล เมื่อผู้ใช้ต้องการปรับปรุงข้อมูลบางแถวจากวิว ระบบจัดการฐานข้อมูลจะแปลความต้องการไปยังแถวที่ต้องการปรับปรุงในตารางข้อมูลจริง ซึ่งจะกระทำได้กับวิวแบบง่ายๆ แต่สำหรับวิวที่ซับซ้อนก็จะจำกัดให้อ่านข้อมูลได้อย่างเดียว

ที่มา <http://sot.swu.ac.th/Portals/156/sot/CP342/lesson12/ms3t1.htm>