Database Security

01418221 Fundamentals of Database Systems

Source: Fundamental of IBM DB2 Database System

Outlines

- Transactions
- Concurrency & Locking
- Lock Wait
- Deadlocks

Source: Fundamental of IBM DB2 Database System

ภาพรวมของความปลอดภัยของฐานข้อมูล

- โดยปกติ ปัญหาที่เกี่ยวข้องกับการรักษาความปลอดภัยของข้อมูลเป็นปัญหาที่ซับซ้อน และเกี่ยวข้องกับประเด็นของกฎหมาย และทางด้านสังคมหรือจริยธรรม
- ปัญหาที่เกี่ยวข้องกับนโยบายการใช้งาน หรือเกี่ยวข้องกับการควบคุมอุปกรณ์ทางกายภาพ ความปลอดภัยของฐานข้อมูลจะเกี่ยวข้องกับการป้องกันฐานข้อมูลจากภัยคุกคามที่เกิดขึ้นโดยความตั้งใจหรือไม่ตั้งใจ
- การรักษาความปลอดภัยของฐานข้อมูลเพียงอย่างเดียวนั้น อาจจะไม่ได้ทำให้ข้อมูลมีความปลอดภัย แต่จำเป็นต้องพิจารณาถึงองค์ประกอบอื่นๆทั้งระบบเครือข่าย ระบบปฏิบัติการ และสถานที่ที่เราติดตั้งระบบฐานข้อมูล เช่น อาคาร รวมถึงบุคคลที่สามารถเข้าถึงระบบฐานข้อมูล ก็เป็นสิ่งจำเป็นในการคำนึงถึงการรักษาความปลอดภัย

Source: Fundamental of IBM DB2 Database System

การสร้างความเสียหาย (Threats)

- การกระทำหรือเหตุการณ์ใด ๆ ที่เกิดขึ้นด้วยความตั้งใจหรือด้วยเหตุสุดวิสัยที่ทำให้ข้อมูลถูกเปิดเผย (Data Confidentiality) หรือมีการเปลี่ยนแปลงข้อมูล (Data Integrity) หรือทำให้ระบบหยุดบริการ (System Availability) ทำให้เกิดความเสียหายต่อข้อมูลในฐานข้อมูล หรือข้อมูลไม่ปลอดภัย ส่งผลต่อการทำงานของระบบ องค์การ ยกตัวอย่างเช่น
  - ข้อมูลสูญหายหรือเสียหายโดยเหตุสุดวิสัย
  - การโกงและการหลอกลวง
  - การสูญเสียความเป็นส่วนตัวหรือความลับ
  - การสูญเสียความคงสภาพของข้อมูล
  - การสูญเสียความพร้อมใช้

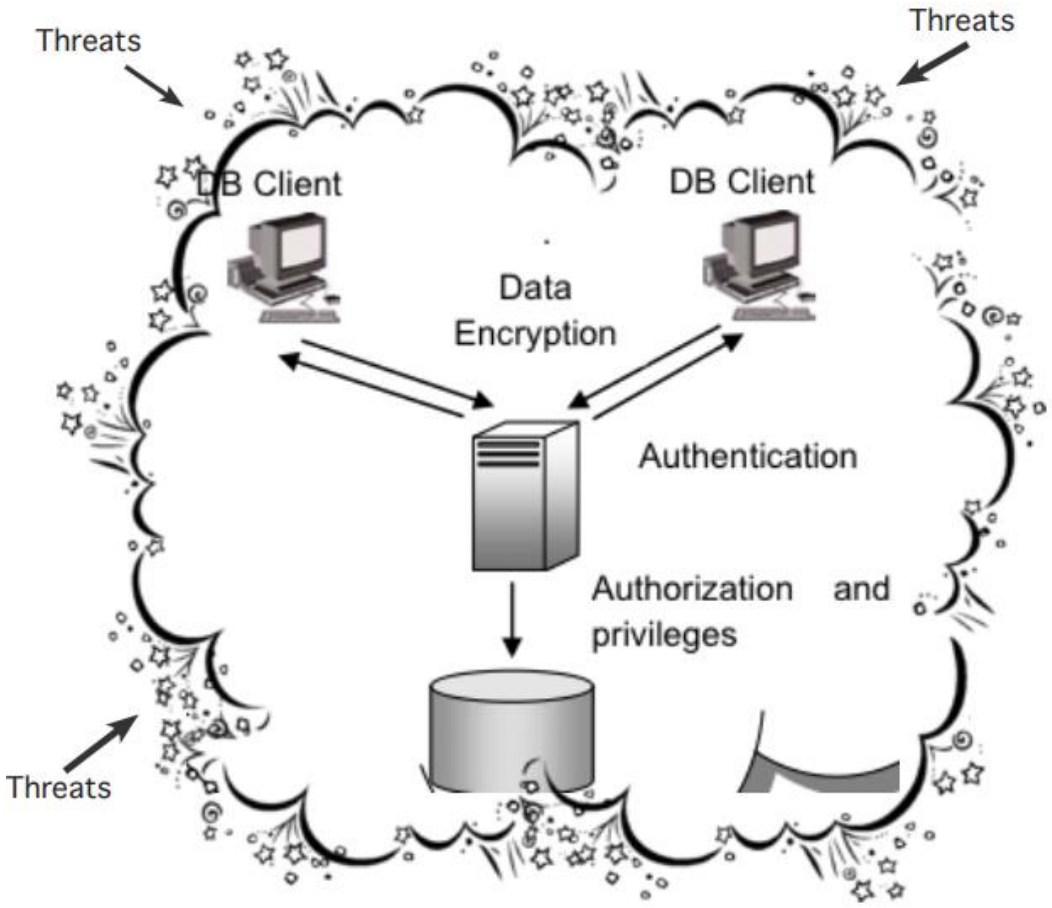
Source: Fundamental of IBM DB2 Database System

การควบคุมการเข้าถึงข้อมูล

- การออกแบบ และการติดตั้งความปลอดภัยให้กับฐานข้อมูลเกี่ยวข้องกับวัตถุประสงค์ต่อไปนี้:
  - ข้อมูลส่วนบุคคล (Privacy) หมายถึง ผู้ใช้ที่ไม่มีสิทธิ์ จะต้องไม่สามารถเห็นข้อมูล
  - ความสมบูรณ์ (Integrity) หมายถึงการที่อนุญาตให้ เฉพาะผู้ใช้ที่มีสิทธิ์ สามารถเปลี่ยนแปลงข้อมูล
  - การมีให้ (Availability) หมายถึงการที่ผู้ใช้ที่มีสิทธิ์จะต้องสามารถ ใช้สิทธิ์ที่เขามีได้

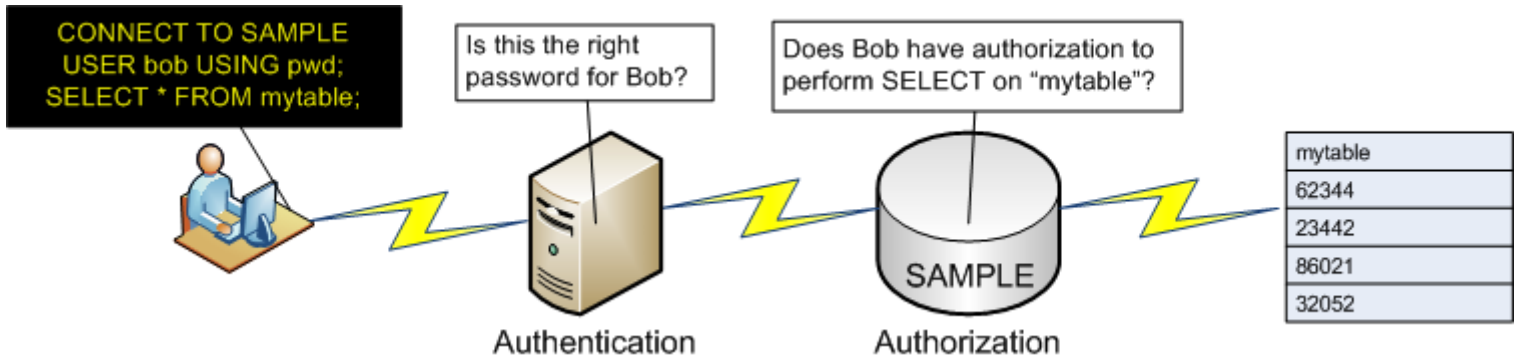
Source: Fundamental of IBM DB2 Database System

ความปลอดภัยของฐานข้อมูล



Source: Fundamental of IBM DB2 Database System

DB2 security overview



DB2 security has two steps:

- Authentication ตรวจสอบชื่อผู้ใช้งานและรหัสผ่าน
- Authorization ตรวจสอบสิทธิการใช้งานของผู้ใช้งานในการประมวลผลคำสั่งที่ต้องการ

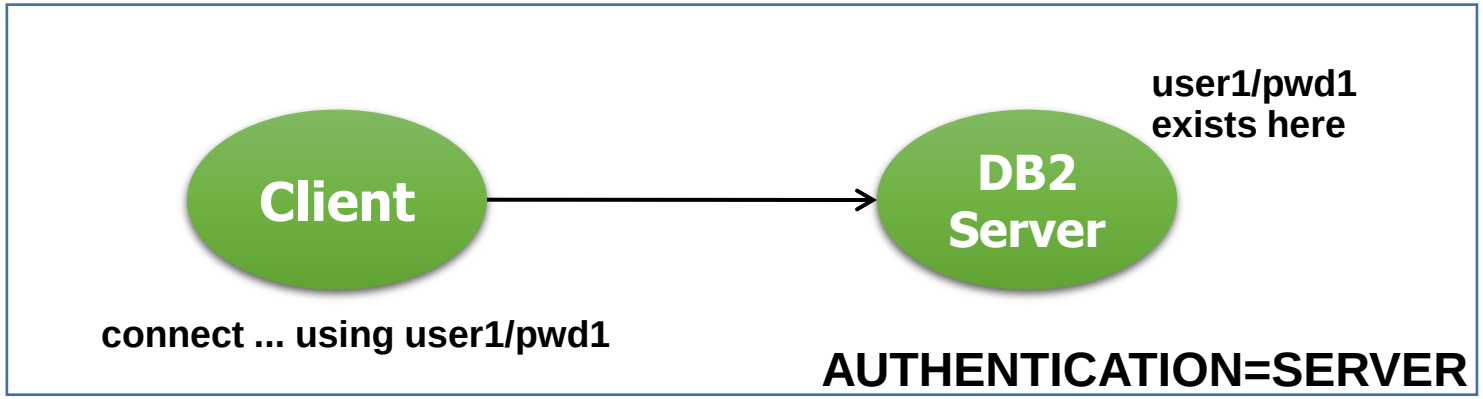
Source: Fundamental of IBM DB2 Database System

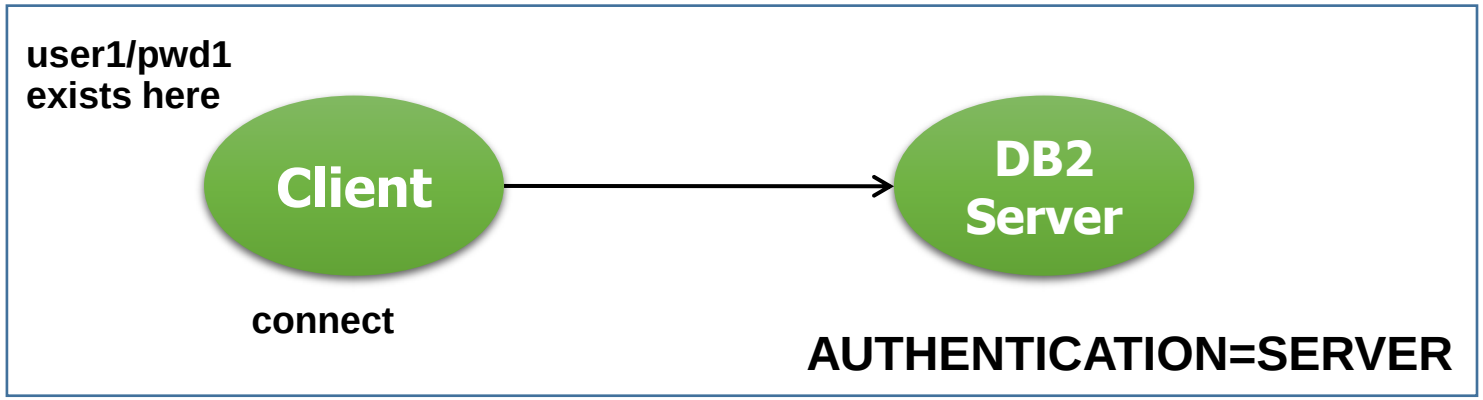
การรักษาความปลอดภัยของฐานข้อมูล

- **การยืนยันตัวตน (Authentication)** เป็นการกระทำที่เกิดขึ้นเป็นสิ่งแรกเมื่อมีการเชื่อมต่อกับฐานข้อมูล DB2 โดยเป็นกระบวนการที่ผู้ใช้จะต้องพิสูจน์ตัวตนก่อนการเข้าใช้งานฐานข้อมูล การตรวจสอบข้อมูลเฉพาะตัวระดับผู้ใช้งานแต่ละคนหรือระดับกลุ่มนั้น DB2 ทำงานร่วมกับระบบการรักษาความปลอดภัยอื่นๆที่อยู่ภายนอกระบบฐานข้อมูล เช่น
  - การทำงานร่วมกับระบบปฏิบัติการ หรือบางครั้งเป็นการใช้ฟังก์ชันการรักษาความปลอดภัยระบบอื่นๆเช่น Kerberos หรือ Lightweight Directory Access Protocol (LDAP) เพื่อรับรองความถูกต้อง ซึ่งผู้ใช้จำเป็นต้องมี รหัสผู้ใช้ และรหัสผ่านในการยืนยันตัวตน
- **การอนุญาต (Authorization)** หลังจากที่ผู้ใช้ผ่านขั้นตอนการยืนยันตัวตน แล้ว สิ่งที่ต้องทำต่อไปคือการตรวจสอบว่าผู้ใช้นั้นได้รับอนุญาตให้เข้าถึงข้อมูลหรือทรัพยากรใดได้บ้าง
  - การอนุญาตเป็นการกระบวนการการให้สิทธิแก่ผู้ใช้ (grant privilege) ในการเข้าใช้งานระบบหรือเข้าถึงออบเจ็กต์ตามนโยบายที่กำหนดในระบบ คำจำกัดความของการอนุญาตจะประกอบด้วย Subject และ Object
    - Subject จะเกี่ยวข้องกับผู้ใช้ และ โปรแกรม
    - Object จะเกี่ยวข้องกับ ตาราง วิว แอพพลิเคชัน procedure หรือออบเจ็กต์อื่นๆที่มีอยู่ในระบบ

Source: Fundamental of IBM DB2 Database System

Authentication





Source: Fundamental of IBM DB2 Database System

สิทธิ (Privileges)

สิทธิ (Privileges) เป็นการกำหนด ความสามารถให้แก่ผู้ใช้กลุ่ม หรือRole ซึ่งเป็นการยอมให้ผู้ใช้งานฐานข้อมูลสามารถกระทำกิจกรรมต่างๆที่เกี่ยวข้องกับออบเจ็กต์ของระบบฐานข้อมูล ได้แก่ การค้นหา (Search) การเพิ่มข้อมูล (Insert) การลบข้อมูล (Delete) การเปลี่ยนแปลงข้อมูล (Change)

ระบบอาจจะพิจารณาจากเมตริกซ์การเข้าถึง (Access Matrix) ซึ่งเป็นตาราง 2 มิติ ที่กำหนดให้แถว หมายถึงผู้ใช้งาน (User) และคอลัมน์ หมายถึงฐานข้อมูล

บางระบบจะใช้ระบบ Access Control List (ACL) แทน ACLเป็นลิสต์ที่บรรจุผู้ใช้งานระบบและสิทธิการเข้าใช้งาน โดยระบุประเภทของผู้ใช้ (User Type) ระดับการเข้าถึงฐานข้อมูล (Access Level) สิทธิ (Privileges) บทบาท (Role)

	ฐานข้อมูล 1	**ฐานข้อมูล 2**	**ฐานข้อมูล 3**
ผู้ใช้งาน 1	ค้นหา	ค้นหา เพิ่ม แก้ไข ลบ	
ผู้ใช้งาน 2		ค้นหา	ค้นหา เพิ่ม แก้ไข ลบ
ผู้ใช้งาน 3	ค้นหา เพิ่ม แก้ไข ลบ		ค้นหา

Source: Fundamental of IBM DB2 Database System

Privileges - Examples

- ■ **Schema privileges**
  –CREATEIN: can create objects within the schema
  –ALTERIN: can alter objects within the schema
  –DROPIN: can drop objects from within the schema
- ■ **Table and View privileges**
  –CONTROL: Full control on a table or view including drop, grant/revoke
    - DELETE: can delete rows
    - INSERT: can insert rows and run the IMPORT utility.
    - SELECT: can retrieve rows, create a view, run the EXPORT utility.
    - UPDATE: can change an entry in a column, table or view
    - ALTER: can modify a table
    - INDEX: can create an index on a table
    - REFERENCES can create and drop a foreign key

Source: Fundamental of IBM DB2 Database System

Privileges – Granting / Revoking

- **Explicit**
  - ใช้คำสั่ง GRANT and REVOKE เพื่อมอบหรือถอนสิทธิการใช้งาน user or group

```
grant select on table db2inst1.employee to user mary
revoke select on table db2inst1.employee from user mary
```

- **Implicit**
  - DB2 ให้สิทธิการใช้งานแก่ user ทันทีโดยอัตโนมัติเมื่อมีการเรียกคำสั่ง

```
create table mytable
```

User automatically gains full access to the table

- **Indirect**
  - เมื่อการรันคำสั่ง SQL statements ใด ๆ ผู้ใช้งานต้องมีสิทธิ EXECUTE ในการเรียกใช้งานหรือรันคำสัง
  - Example: package1 contains the following static SQL statements

```
select * from test
insert into test values (1,2,3)
```

  - ในกรณีนี้ user ที่มีสิทธิ EXECUTE ใน package1 ได้รับสิทธิทางอ้อม (indirectly granted) SELECT and INSERT ในตาราง TEST

Source: Fundamental of IBM DB2 Database System

Privileges – Granting / Revoking (Examples)

```
GRANT    SELECT     ON     TABLE  CUSTOMERS  TO  USER       user1

GRANT     ALL         ON     TABLE  CUSTOMERS  TO  GROUP     group1

REVOKE   ALL         ON     TABLE CUSTOMERS FROM  GROUP    group1

GRANT    EXECUTE   ON      PROCEDURE PROC1 TO USER    user1

REVOKE  EXECUTE   ON      PROCEDURE PROC1 FROM USER user1

REVOKE  CONNECT   ON     DATABASE  FROM  USER     user2
```

Source: Fundamental of IBM DB2 Database System

บทบาท (Role)

- Role คือออบเจกต์ในฐานข้อมูลที่อาจจัดกลุ่มสิทธิ (Privileges) หนึ่งรายการหรือมากกว่าเข้าด้วยกันและสามารถกำหนดให้กับผู้ใช้กลุ่ม Public หรือไปยังบทบาทอื่น ๆ ผ่านคำสั่ง GRANT บทบาทช่วยให้การบริหารและการจัดการสิทธิ์ง่ายขึ้น
- Role สามารถสร้างแบบจำลองได้หลังจากโครงสร้างขององค์กร พวกเขาสามารถสร้างขึ้นเพื่อทำแผนที่โดยตรงกับฟังก์ชั่นงานเฉพาะภายในองค์กร แทนที่จะให้สิทธิ์ชุดเดียวกันแก่ผู้ใช้แต่ละคนในฟังก์ชั่นงานเฉพาะ
  - ชุดของสิทธิ์นี้สามารถมอบให้กับ Role
  - จากนั้นผู้ใช้จะได้รับการเป็นสมาชิกใน role เดียวกัน
- Role ที่สะท้อนถึงความรับผิดชอบในงานของพวกเขา เมื่อความรับผิดชอบในงานเปลี่ยนไปสมาชิกภาพในบทบาทสามารถได้รับอนุญาตและเพิกถอนได้ง่าย

Source: Fundamental of IBM DB2 Database System

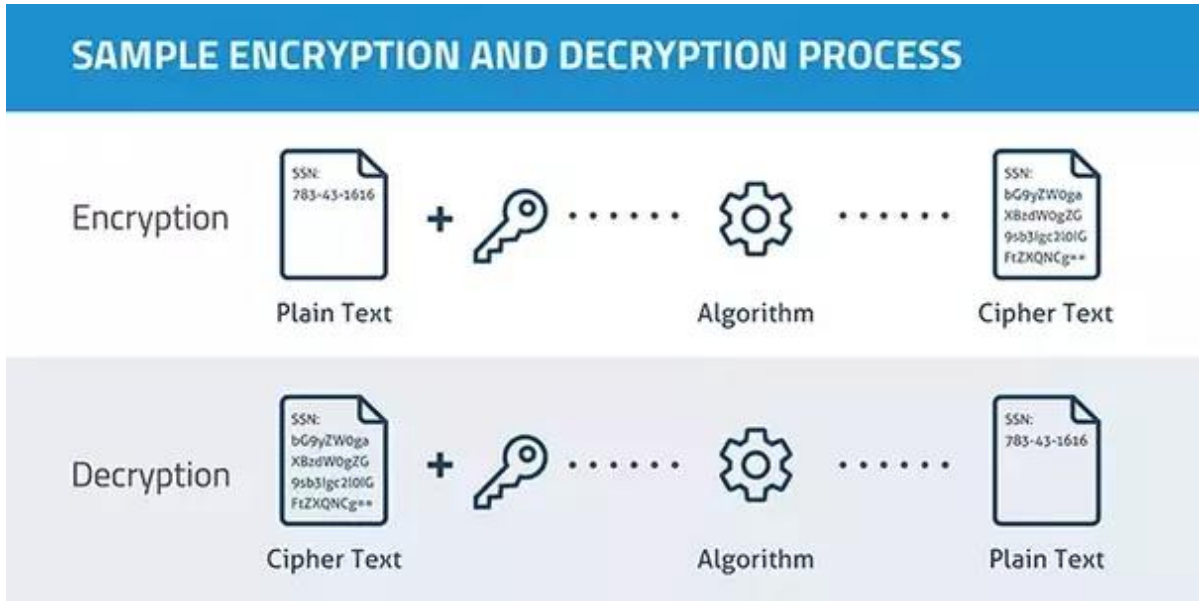
บทบาท (Role)

```
CONNECT TO SAMPLE
CREATE ROLE DEVELOPER

GRANT CONNECT ON DATABASE TO ROLE DEVELOPER
GRANT SELECT, INSERT, UPDATE, DELETE ON TABLE employee TO ROLE DEVELOPER

GRANT ROLE DEVELOPER TO USER USERDEV
REVOKE ROLE DEVELOPER FROM USER USERDEV
```

Source: Fundamental of IBM DB2 Database System

การเข้ารหัสข้อมูล (Data Encryption)

- การเข้ารหัส (Encryption) หมายถึง วิธีการที่ช่วยให้ข้อมูลมีความปลอดภัยมากยิ่งขึ้นด้วยการเปลี่ยนแปลงข้อมูล (Plain Text) ให้กลายเป็นข้อมูลถูกรหัส (Cipher Text) ที่ไม่สามารถอ่านได้รู้เรื่อง นอกจากจะมีคีย์ถอดรหัส (Decryption Key) ข้อมูลออกมาก่อน
  - ระบบจะใช้คีย์ที่มีความซับซ้อนสูง โดยพิจารณาว่าความยาวของคีย์หรือจำนวนบิตที่ใช้ในคีย์ เพื่อให้บุคคลที่เป็นผู้ประสงค์ร้ายทำลักลอบเข้าใช้ข้อมูลยากขึ้นหรือใช้เวลานานมากในการคาดเดาคีย์ที่ถูกต้อง

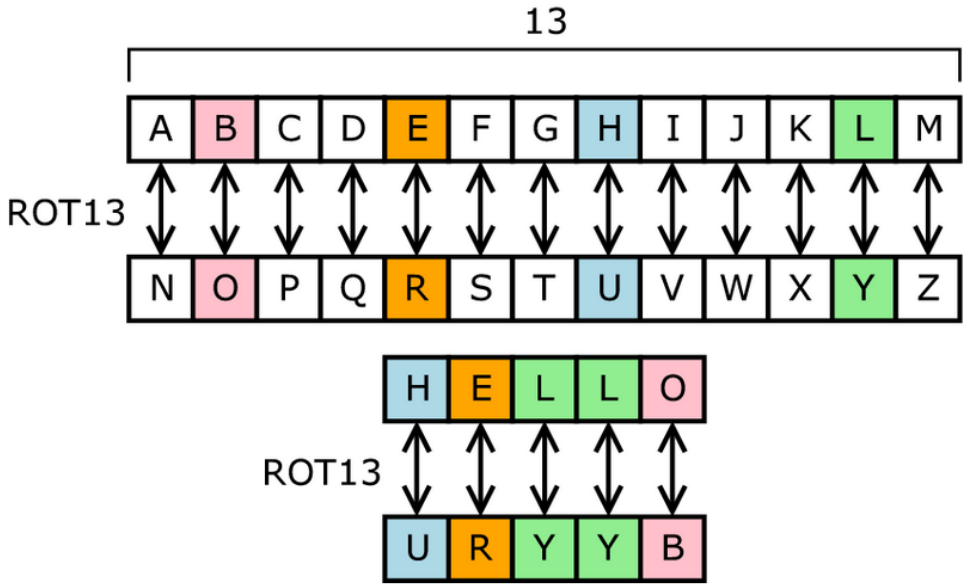


https://medium.com/@daser/a-lazy-mans-introduction-to-multi-party-encryption-and-decryption-59f62b8616d8

Source: Fundamental of IBM DB2 Database System

เทคนิคที่ใช้ในการเข้ารหัสข้อมูล

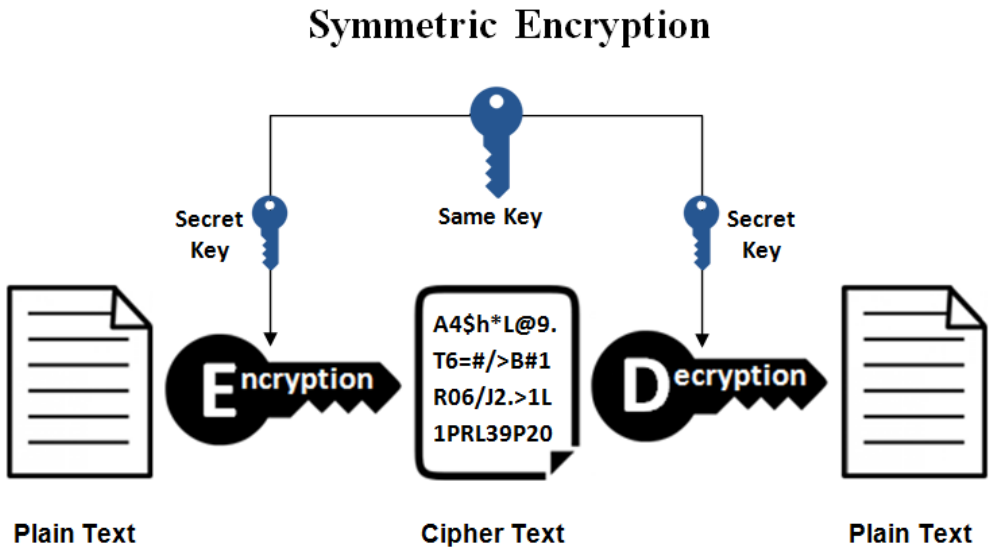
- **รหัสลับแบบสับเปลี่ยน (Substitution Cypher)** เป็นวิธีการเข้ารหัสข้อมูลดั้งเดิมที่ถูกค้นพบมาตั้งแต่โบราณ โดยใช้วิธีการเขียนข้อความลับของข้อความตั้งต้นด้วยการสุ่มจับคู่ตัวอักษรระหว่างข้อความตั้งต้นกับตัวอักษรที่ได้จากตารางคู่ตัวอักษร จากนั้นสลับเปลี่ยนตัวอักษรแต่ละตัวด้วยอักษรที่เป็นคู่กัน แล้วเขียนเป็นข้อความลับ



Source: Fundamental of IBM DB2 Database System

เทคนิคที่ใช้ในการเข้ารหัสข้อมูล

- **กุญแจแบบสมมาตร (Symmetrical Encryption)** เป็นวิธีการเข้ารหัสข้อมูลที่ใช้กุญแจดอกเดียวกันเพื่อใช้เป็นกุญแจเข้ารหัส (Encryption Key) ในการเข้ารหัสข้อความต้นฉลับเป็นรหัสลับ และใช้เป็นกุญแจถอดรหัส (Decryption Key) เพื่อทำการถอดรหัสลับเป็นข้อความตั้งต้น

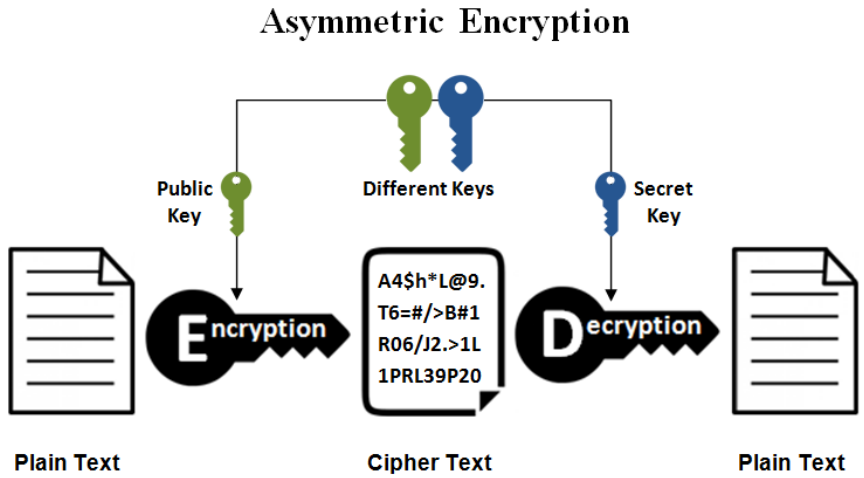


https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Source: Fundamental of IBM DB2 Database System

เทคนิคที่ใช้ในการเข้ารหัสข้อมูล

- **กุญแจอสมมาตร (Asymmetric Key)** เป็นการใช้กุญแจสองตัวที่แตกต่างกัน กุญแจตัวแรกในการเข้ารหัสลับ และใช้กุญแจอีกตัวหนึ่งในการถอดรหัสออกเป็นข้อความปกติ โดยที่
  - กุญแจตัวหนึ่งจะเรียกว่า กุญแจสาธารณะ (Public Key) ที่สามารถปรากฏและส่งให้ผู้อื่นหรือเซิร์ฟเวอร์ใช้งานได้
  - กุญแจอีกตัวหนึ่งคือ กุญแจส่วนตัว (Private key) ที่เก็บไว้เพื่อใช้งานเป็นการส่วนตัวเฉพาะผู้ใช้งานเท่านั้น



https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Source: Fundamental of IBM DB2 Database System

สคีมา (Schemas)

- Schemas คือการกำหนดชื่อให้กับกลุ่มของออบเจกต์ในฐานข้อมูล เพื่อจัดให้มีการจำแนกประเภทของวัตถุในฐานข้อมูล อาทิเช่น ตาราง วิว Procedure เป็นต้น
- วัตถุส่วนใหญ่ในฐานข้อมูลนั้นตั้งชื่อโดยใช้หลักการตั้งชื่อที่ประกอบด้วยสองส่วน คือ
  1. ส่วนแรก (ซ้ายสุด) ของชื่อเรียกว่าชื่อ Schemas หรือตัวระบุ
  2. ส่วนที่สอง (ขวาสุด) เรียกว่าชื่อวัตถุ โดยที่ทั้งสองส่วนนี้ถูกต่อกันและคั่นด้วย .

schema_name.object_name

- Schemas ยังเป็นวัตถุในฐานข้อมูล Schemas สามารถสร้างได้ 2 วิธีหลัก ๆ :
  1. สามารถสร้างขึ้นโดยปริยายเมื่อวัตถุอื่นถูกสร้างขึ้นโดยมีเงื่อนไขว่าผู้ใช้มีสิทธิ์ฐานข้อมูล IMPLICIT_SCHEMA
  2. ถูกสร้างขึ้นอย่างชัดเจนโดยใช้คำสั่ง CREATE SCHEMA กับผู้ใช้ปัจจุบัน

Source: Fundamental of IBM DB2 Database System

สคีมา (Schemas)

LOGIN USER1

CREATE TABLE TABLE1 .....

FULL NAME OF TABLE : USER1.TABLE1

- เพื่อสร้างสกีมาโดยใช้คำสั่ง CREATE SCHEMA:

```
CREATE SCHEMA <name> AUTHORIZATION <name>

CREATE SCHEMA myschema AUTHORIZATION user01

CREATE TABLE myschema.store
(storeid INTEGER,
address CHAR(50))
```

- ชื่อตารางที่ระบุจะต้องไม่ซ้ำกันภายในสคีมาเดียวกัน

Source: Fundamental of IBM DB2 Database System

วิว (Views)

- วิวเป็นองค์ประกอบที่สำคัญของกลไกการรักษาความปลอดภัยโดยระบบฐานข้อมูล โดยที่วิวเป็นการสร้างมุมมองใหม่ของฐานข้อมูลที่อนุญาตให้ผู้ใช้เห็นข้อมูลในมุมมองต่างๆ ที่เราต้องการให้เห็น และเรายังสามารถซ่อนข้อมูลใดๆ ที่ไม่ต้องการให้ผู้ใช้ เข้าถึงผ่านวิวได้อีกด้วย
- วิวเป็นการสร้างมุมมอง เพื่อให้ได้ผลลัพธ์เป็นแบบไดนามิก ที่ถูกสร้างขึ้นมาจากโอเปอเรชั่นต่างๆ เช่น union โดยเกี่ยวข้องกับตารางปกติ (base table) หนึ่งตารางหรือมากกว่า เพื่อให้เกิดเป็น ตารางใหม่ วิว จะแสดงผลลัพธ์ที่เป็นข้อมูลปัจจุบันที่ได้จากตารางปกติ ที่มันอ้างอิงถึง
- ประโยชน์ของวิวในมุมมองด้านความปลอดภัยคือวิวถูกสร้างขึ้นมาเพื่อใช้สำาหรับการนำเสนอข้อมูลในแก่ผู้ใช้ในมุมมองต่างๆ ตามที่เราต้องการ
- วิวจะไม่แสดงข้อมูลที่เป็นความลับที่เราต้องการปกปิดไว้ เราสามารถกำาหนดสิทธิให้แก่ผู้ใช้ในการทำางานกับวิว โดยที่ไม่จำเป็นต้องให้สิทธิ์ในการเข้าถึงตารางปกติ ที่วิวอ้างอิงถึง

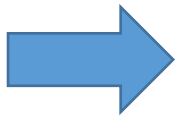
Source: Fundamental of IBM DB2 Database System

การเข้าถึงผ่านมุมมองวิว

- อนุญาตให้ผู้ใช้งานหลายคนสามารถเห็นข้อมูลเดียวกันในหลายมุมมอง
- เป็นวิธีการที่ง่ายและดีในการควบคุมความปลอดภัย แต่จะมีความยุ่งยากในการจัดการที่มีขนาดใหญ่
- **ขั้นตอนการทำงาน:**
  - สร้างวิวใหม่ (ชุดข้อมูลย่อยในตารางข้อมูลหลัก)
  - สามารถให้สิทธิการเข้าถึงข้อมูลแก่ผู้ใช้งานที่ต้องการ
  - ยกเลิกสิทธิการเข้าถึงข้อมูลตารางหลักจากผู้ใช้งาน

EMPLOYEE

ID	Name	AGE	PHONE	SALARY
12	Peter	30	99999	10000
3	Mary	23	88888	15000



EMP_VIEW

ID	Name	AGE	PHONE
12	Peter	30	99999
3	Mary	23	88888

```
CREATE VIEW EMP_VIEW AS (
    SELECT ID, NAME, AGE,PHONE
    FROM EMPLOYEE);
```

Source: Fundamental of IBM DB2 Database System