

# Database Systems

## Concept

ผศ.กฤษฎา บุศรา

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



# คำนำ

หนังสือเล่มนี้ถูกเขียนขึ้นมาในท่ามกลางบริบท ของการเกิดกระแสการใช้งานโปรแกรมหรือระบบงานที่มีลักษณะการส่งข้อมูลแบบ IoT (Internet of Things) โดยนำข้อมูลทั่วทุกที่นำขึ้นไปเก็บบนระบบการประมวลผล คลาวด์ (Cloud Computing System) ที่มีการเก็บข้อมูลในฐานข้อมูลกระจายอยู่ในที่ต่างๆ โดยลักษณะข้อมูล เป็นแบบไม่เป็นโครงสร้าง (Unstructured Data) ประกอบด้วยข้อมูลแบบข้อความ (Text) รูปภาพ (Image) หรือ ภาพเคลื่อนไหว(Multimedia) ตามพัฒนาระบบของผู้ใช้งานในยุค Digital Economy ก่อให้เกิดข้อมูลมีปริมาณ มากมายที่เรียกว่า Big Data และทำให้เกิดแนวคิดหรือความเข้าใจคลาดเคลื่อนไปจากข้อเท็จจริง

- ปัจจุบันไม่มีใครใช้ฐานข้อมูลแล้ว เพราะมี Big Data
- ไม่ต้องเขียนภาษา SQL แล้วเพราะมี NoSQL
- ไม่ต้องออกแบบฐานข้อมูลแล้ว เพราะไม่สามารถกำหนดรูปแบบข้อมูลได้
- ไปทำงาน ไม่เห็นต้องเขียนคำสั่งภาษา SQL เลย แค่เรียกผ่าน Method โดยไม่ต้องรู้โครงสร้างตาราง
- โปรแกรมใช้ผ่าน GUI (Graphic User Interface) ไม่ต้องเขียนคำสั่งแค่ลากๆ เส้นก็พอ
- พัฒนาระบบงาน โดยเห็นข้อมูลที่มี Format เป็น XML หรือ JSON เท่านั้น

API

No Code / Low Code

=> จะขอให้เหตุผลว่า ทั้งหมดนี้ใช้ภาษา SQL ทั้งนั้นเพียงแต่ว่า ผู้พัฒนายังไม่มีประสบการณ์มากพอ และถ้ามองในประเด็นการพัฒนาระบบงานอย่างมีประสิทธิภาพโดยเน้นความเร็ว (Speed Performance) ก็ต้องเน้นเขียน Stored Procedure (PL/SQL) ที่ฝัง Database Tier เพื่อลดการติดต่อ (Session) และลดการดึงข้อมูลจำนวนมากผ่านระบบเครือข่าย ....และ Stored Procedure จะประกอบด้วยภาษา SQL + คำสั่งเชิงโปรแกรม (IF-THEN-ELSE , Loop) + การประกาศตัวแปรแล้วระบุชนิดข้อมูล + การดักจับ Error ที่เกิดขึ้นเพื่อไปดำเนินการต่อ เป็นต้น

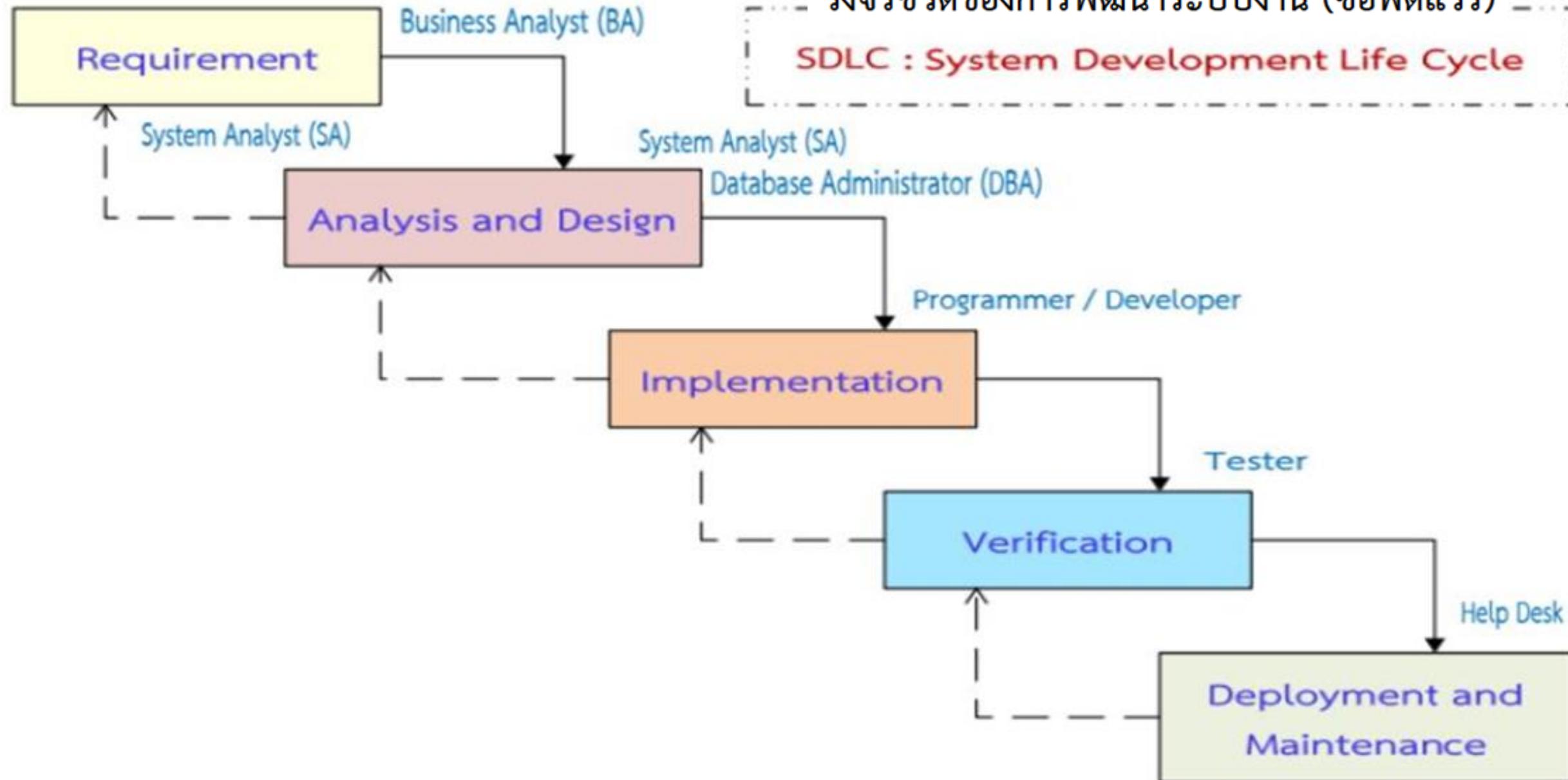
=> ขอแสดงความคิดเห็นว่า ในโลกความเป็นจริงมีการดำเนินธุรกิจในระบบห่วงโซ่อุปทาน (Supply Chain Management) และระบบ Logistic หรือระบบงานในธุรกิจภาคทั่วไป ดังนั้นจะทราบ Business Domain และ Work Flow ที่ชัดเจนทำให้ข้อมูลเป็น Structured Data ที่ถูกจัดเก็บในฐานข้อมูลหรือคลังข้อมูล

ยกเว้นลักษณะงานที่เป็นข้อมูลที่จัดเก็บจากพฤติกรรมลูกค้า ผู้บริโภค หรือกระแสใน Social Network – > Post-Like-Comment-Share หรือการเจรจา พูดคุย ตอบแบบสอบถามปลายเปิด(เหมือนข้อสอบอัตนัย) ....แต่ เมื่อต้องการใช้งานข้อมูลดังกล่าว ก็ต้องมีกระบวนการแปลง Big Data กลับมาอยู่ในรูปแบบของ Structured Data อีกรึหนึ่ง ด้วยเหตุผลว่า เมื่อต้องการวิเคราะห์หรือพยากรณ์หรือคาดความหมายเชื่อมโยงอะไร ก็ต้องใช้ สมการคณิตศาสตร์ หรือสมการสถิติมาประมวลผลอยู่ดีที่เรียกว่า **วิทยาการข้อมูล (Data Science)**

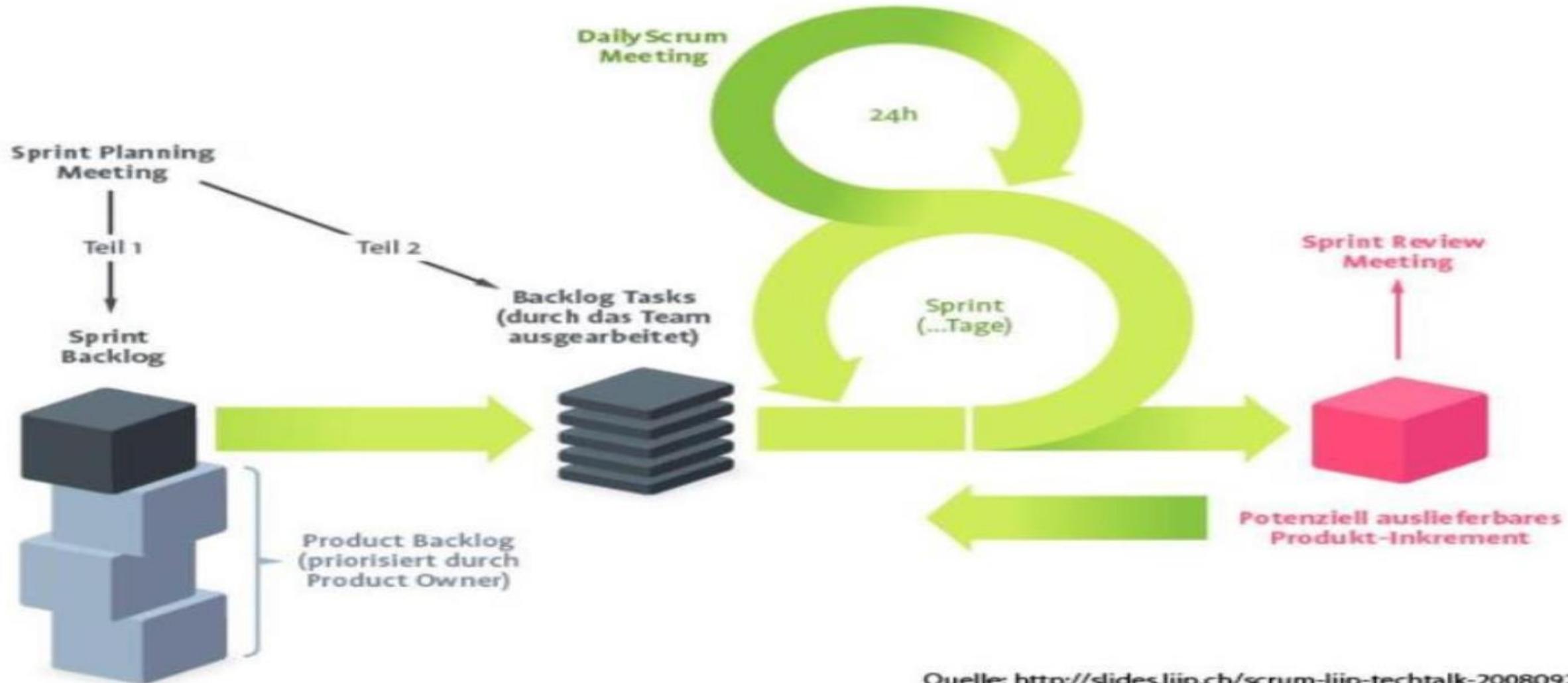
สุดท้ายนี้ ผู้เขียนหวังว่า หนังสือเล่มนี้คงอยู่ข้างกายนักพัฒนาระบบงานด้าน ICT นะครับ

# วงจรชีวิตของการพัฒนาระบบงาน (ซอฟต์แวร์)

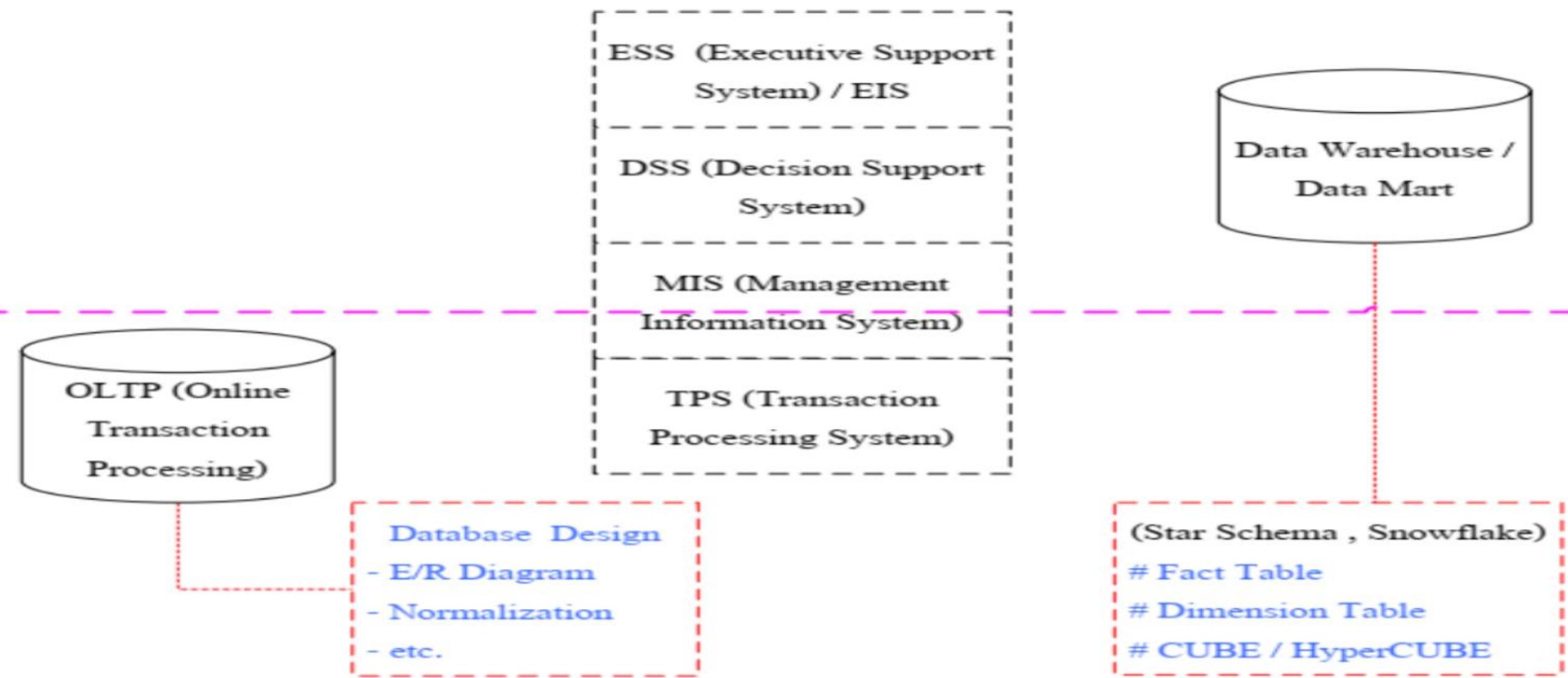
SDLC : System Development Life Cycle



# ตัวอย่าง Agile Process Model ที่เรียกว่า SCRUM



## สรุปความสัมพันธ์ระหว่างประเภทระบบงานสารสนเทศกับประเภทฐานข้อมูล



# บทที่ 1

## ทบทวนความรู้พื้นฐานเกี่ยวกับฐานข้อมูล

### 1.1 ฐานข้อมูล (Database)

ฐานข้อมูล คือ แหล่งเก็บข้อมูลและความสัมพันธ์ระหว่างข้อมูล โดยได้รับการออกแบบตามหลักการที่เป็นวิทยาศาสตร์(ก) เพื่อให้มีความซ้ำซ้อนของข้อมูลน้อยที่สุด และมีการควบคุมกฎข้อบังคับ(ข) เพื่อให้เกิดความถูกต้องของข้อมูลสูงสุด

โดยความสัมพันธ์ดังกล่าวสามารถมีหลายรูปแบบ ได้แก่ แบบแผนภูมิต้นไม้ (Hierarchical Model) แบบโครงข่าย (Network Model) แบบเชิงสัมพันธ์ (Relational Model) แบบเชิงวัตถุ (Object-Oriented Model) หรือแบบเชิงวัตถุสัมพันธ์ (Object-Relational Model) เป็นต้น โดยในที่นี้จะเน้นเฉพาะความสัมพันธ์แบบเชิงสัมพันธ์เท่านั้น

### 1.2 รูปแบบเชิงสัมพันธ์ (Relational Model)

โครงสร้างข้อมูลจะต้องเป็นรีเลชัน (Relation) เท่านั้น โดยที่ข้อมูลดังกล่าวจะถูกจัดเก็บในรูปแบบของตาราง (Table) ที่มีคุณสมบัติตามทฤษฎีของเซต (Set Theory) โดยไม่ต้องมีตัวชี้ (Pointer) (ค) เป็นตัวแสดงการเชื่อมโยงหรือความสัมพันธ์ระหว่างตาราง ดังแสดงได้ดังรูปที่ 1.1

## อริบายเพิ่มเติม

(ก) การออกแบบฐานข้อมูลในปัจจุบันมีหลายวิธี อย่างเช่น แผนภาพอีอาร์ (Entity Relationship Diagram) วิธีการทำให้เป็นบรรทัดฐาน(Normalization) แผนภาพクラス (Class Diagram) ในภาษาอูมอล (UML : Unified Modeling Language) เป็นต้น

และการออกแบบฐานข้อมูลเชิงสัมพันธ์จะนิยมใช้วิธี E/R Diagram ก่อนในเบื้องต้นแล้วใช้วิธีนอร์มัลไลเซชัน (Normalization) ทำการออกแบบซ้ำอีกรอบหนึ่ง เนื่องจาก วิธี E/R Diagram จะรับประกันตารางต่างๆที่เกิดจากการออกแบบให้มีคุณสมบัติในระดับ Second Normal Form(รูปแบบที่เป็นบรรทัดฐานระดับที่ 2) จาก 5 ระดับเท่านั้น

(ข) กฎข้อบังคับ (Integrity Constraint/Rule) เผาฐานข้อมูลเชิงสัมพันธ์ คือ สิ่งที่กำหนดให้ต้องมีเพื่อทำให้ข้อมูลมีความถูกต้องมากที่สุด ได้แก่

-การควบคุมความถูกต้องของคีย์หลัก (Primary Key) คือ การบังคับมีค่าข้อมูลแบบห้ามซ้ำ

-การควบคุมความถูกต้องของความสัมพันธ์ระหว่างคีย์หลัก (Primary Key) กับคีย์นอก(Foreign Key) ในรูปแบบเชิงสัมพันธ์ (Relational Model) คือ คีย์นอกจะมีค่าหรือไม่มีค่าข้อมูลก็ได้ แต่ถ้ามีค่า ค่าข้อมูลดังกล่าวจะต้องมีค่าสอดคล้องกับค่าข้อมูลในคีย์หลักที่มีความสัมพันธ์ด้วย

(ข) กฎข้อบังคับ (Integrity Constraint/Rule) เอกพาระฐานข้อมูลเชิงสัมพันธ์ คือ สิ่งที่กำหนดให้ต้องมีเพื่อทำให้ข้อมูลมีความถูกต้องมากที่สุด ได้แก่

-การควบคุมความถูกต้องของคีย์หลัก (Primary Key) คือ การบังคับมีค่าข้อมูลแบบห้ามซ้ำ

-การควบคุมความถูกต้องของความสัมพันธ์ระหว่างคีย์หลัก (Primary Key) กับคีย์นอก(Foreign Key) ในรูปแบบเชิงสัมพันธ์ (Relational Model) คือ คีย์นอกจะมีค่าหรือไม่มีค่าข้อมูลก็ได้ แต่ถ้ามีค่า ค่าข้อมูลดังกล่าวจะต้องมีค่าสอดคล้องกับค่าข้อมูลในคีย์หลักที่มีความสัมพันธ์ด้วย

-การควบคุมในลักษณะอื่นๆ ได้แก่ การห้ามซ้ำ (Unique) การบังคับมีค่า (Not Null) หรือการกำหนดขอบเขตของข้อมูลในคอลัมน์(Column) อย่างเช่น ข้อมูลเพศ มีค่าได้เพียง 2 ค่า คือ 'M' กับ 'F' เท่านั้น หรือเงินเดือนจะมีค่าตั้งแต่ 15,000 ถึง 500,000 บาทเท่านั้น เป็นต้น

(ค) ตัวชี้ (Pointer) คือ การเชื่อมโยงโดยเก็บค่า Address ของ Node ก่อนหน้า หรือถัดไป แบบ link List หรือ โครงสร้างต้นไม้ (Tree Structure) ในวิชาโครงสร้างข้อมูล (Data Structure)

ตาราง SetCountry



SctCode , SctTName , SctTAbbrev

ตาราง SetPart



F.K.

SprCode , SprTName , SprTAbbrev , SprSctCode

ตาราง SetProvince



F.K.

SpvCode , SpvTName , SpvTAbbrev , SpvSprCode

รูปที่ 1.1

ตัวอย่างตารางที่มีความสัมพันธ์แบบเชิงสัมพันธ์

โดยที่ตาราง SetCountry หมายถึง ตารางเก็บข้อมูลประเทศ  
ตาราง SetPart หมายถึง ตารางเก็บข้อมูลภาค โดยที่จะมีการอ้างอิงไปยังตารางประเทศ  
ตาราง SetProvince หมายถึง ตารางเก็บข้อมูลจังหวัดโดยที่จะมีการอ้างอิงไปยังตารางภาค  
และคุณสมบัติของรีเลชันที่อ้างอิงทฤษฎีของเซต ประกอบไปด้วย 4 คุณสมบัติดังต่อไปนี้

- 1) จะไม่มีทัพเพล (Tuple) หรือ แอทริบิวต์ (Attribute) ที่ซ้ำกันอยู่ในรีเลชันเดียวกัน
- 2) ลำดับของทัพเพลจากบนลงล่าง โดยอยู่ในรีเลชันเดียวกันนั้นจะไม่มีความสำคัญ
- 3) ลำดับของแอทริบิวต์จากซ้ายไปขวา โดยอยู่ในรีเลชันเดียวกันนั้นจะไม่มีความสำคัญ
- 4) ค่าของแอทริบิวต์ทุกค่าจะเป็นค่าเดียว หรือไม่มีกลุ่มซ้ำในแอทริบิวต์เดียวกัน

### 1.3 ตัวระบุชี้แผลข้อมูล (Key)

สำหรับตารางประวัตินักศึกษา (Student Table) ที่สร้างขึ้นมาเพื่อเป้าหมายในการเก็บข้อมูลของนักศึกษาเป็นหลักที่ประกอบไปด้วยทัพเพิล (Tuple) หรือแผล (Row) จำนวน 5 แผลและในแต่ละแผลประกอบด้วย 6 คอลัมม์ (Column) หรือแอทริบิวต์ (Attribute) ดังแสดงในรูปที่ 1.2

StdId	FName	LName	Address	DeptCode	CardId
59050026	สายลม	ขยันเรียน	11/11 ช. 3 กทม	1010111	1111111111111
59050300	สายลม	มารยาทดี	101/12 ช. 3 กทม	1010111	5555555555555
59050499	แสงเดด	ขยันเรียน	123 ถ.ดินคำ กทม	1010800	4444444444444
59050500	ภูเขา	มั่งคง	22 ถ.ดินแดง กทม	1010800	3333333333333
59051020	เมฆา	ทำมาค้าขึ้น	101/12 ช. 3 กทม	1010900	2222222222222

รูปที่ 1.2

ตารางประวัตินักศึกษา (Student Table)

ข้อมูลนักศึกษาแต่ละคนจะใช้ 1 แถวในการอธิบายคุณสมบัติ และเมื่อพิจารณาข้อมูลทั้งหมด แล้วจะเห็นว่า ในแต่ละคอลัมก็จะอธิบายสิ่งที่เป็นข้อมูลที่มีความหมายเป็นเรื่องเดียวกัน และเมื่อพิจารณาทั้งหมดจะได้ว่า จะไม่มีข้อมูลใดที่มีค่าข้อมูลเหมือนกัน แสดงว่านักศึกษาแต่ละคนจะมีบางคอลัมก์ที่มีลักษณะเฉพาะสำหรับอธิบายข้อมูลนักศึกษา

-คีย์ซุปเปอร์ (Super Key) คือ คอลัมก์หรือกลุ่มของคอลัมก์มีคุณสมบัติของความเป็นหนึ่ง (Uniqueness)

-คีย์คู่แข่งขัน (Candidate Key) คือ คอลัมก์หรือกลุ่มของคอลัมก์ที่มีคุณสมบัติของความเป็นหนึ่ง (Uniqueness) และมีคุณสมบัติของการมีค่าน้อยที่สุด (Minimality)

-คีย์หลัก (Primary Key) คือ คีย์คู่แข่งขัน(Candidate Key) ที่ถูกเลือกขึ้นมาใช้งาน โดยพิจารณาตามคุณสมบัติต่อไปนี้

1) เหมาะสมกับระบบงาน เช่น

-ในระบบงานประวัติทะเบียนราชภูร์ ควรเลือก เลขที่บัตรประจำตัวประชาชน

-ในระบบประวัตินักศึกษา ควรเลือก รหัสนักศึกษา เป็นต้น

2) ง่ายต่อการใช้งาน เช่น

ควรเลือก รหัสนักศึกษา เนื่องจากใส่ข้อมูลเพียง 8 ตัว เพื่อให้ข้อมูล 1 แถวขึ้นมา

-คีย์หลัก (Primary Key) คือ คีย์คู่แข่งขัน(Candidate Key) ที่ถูกเลือกขึ้นมาใช้งาน

-คีย์รอง (Alternate Key) คือ คีย์คู่แข่งขัน(Candidate Key) ที่ไม่ถูกเลือกขึ้นมาใช้งาน

โดยที่ Combine Key หรือ Composite Key หรือ Compound Key คือ คีย์หลัก (Primary Key) ที่ประกอบด้วย colum มากกว่า 1 colum รวมกัน

และเมื่อพิจารณา ตารางประวัตินักศึกษา (Student Table) จะวิเคราะห์ได้ว่า  
คีย์คู่แข่งขัน(Candidate Key) คือ StdId และ (StdFName รวมกับ StdLName)  
และ StdCardId

คีย์หลัก (Primary Key) คือ StdId

คีย์รอง (Alternate Key) คือ (StdFName รวมกับ StdLName) และ StdCardId

ตาราง Employee

A.K.

EmpNo , FName , LName , Sex , Salary , StartDate , ResignDate , WorkStatus,

ตาราง Department

F.K.

F.K.

F.K.

DeptCode , SuperEmpNo , SpvCode

F.K.

DeptCode , DeptName , TotalEmp , MgrEmpNo , MgrDate

การเชื่อมโยงตาราง Department กับตาราง Employee

- 1) ข้อมูลการทำงานในแผนก จะใช้ Employee.DeptCode = Department.DeptCode
- 2) ข้อมูลผู้บริหารแผนก จะใช้ Employee.EmpNo = Department.MgrEmpNo

## 1.4 กฎควบคุมความถูกต้องของข้อมูล (Data Integrity Constraint/Rule)

กฎควบคุมความถูกต้องของข้อมูล เป็นการจัดการเกี่ยวกับขอบเขตของข้อมูลและความคุ้มความถูกต้องของข้อมูล โดยการทำผ่านคีย์หลักและคีย์นอกร จำแนกเป็น 2 ข้อคือ

### 1) กฎควบคุมความถูกต้องของเอนติตี้ (The Entity Integrity Rule)

จะไม่มีแอ็ททริบิวต์ที่เป็นส่วนประกอบของคีย์หลักตัวใดในรีเลชันได้รับอนุญาตให้มีค่าเป็นคำว่าง (Null value) ค่านี้จะเป็นค่าที่ไม่สามารถทราบค่าได้ (Unknown value) ซึ่งจะไม่ใช่ค่าศูนย์ (Zero) หรือช่องว่าง (Blank) จะเห็นว่ากฎข้อนี้จะใช้ควบคุมความถูกต้องของคีย์หลัก

### 2) กฎควบคุมความถูกต้องของการอ้างอิง (The Referential Integrity Rule)

ในฐานข้อมูลจะต้องไม่มีการเก็บค่าของคีย์นอกรที่ไม่มีความสัมพันธ์กับค่าของคีย์หลัก ดังนั้นจะเป็น การควบคุมความถูกต้องของคีย์นอกรที่มีความสัมพันธ์กับคีย์หลัก โดยที่จะไม่ครอบคลุมความสัมพันธ์กับคีย์รอง จะเห็นว่า กฎข้อนี้จะใช้ควบคุมความถูกต้องของความสัมพันธ์ระหว่างข้อมูล

## 1.5 ระบบการจัดการฐานข้อมูล (DBMS : Database Management System)

ระบบการจัดการฐานข้อมูลเป็นซอฟต์แวร์ (Software) สำหรับจัดการฐานข้อมูลรวมทั้งมีภาษาสำหรับกำหนดโครงสร้างของข้อมูลในระบบงานสารสนเทศต่างๆ ตามสถาปัตยกรรมมาตรฐานของระบบฐานข้อมูล มีภาษาสำหรับการค้นหาและแก้ไขข้อมูลรวมทั้งการจัดการความปลอดภัยของข้อมูล การเข้าถึงข้อมูลรวมถึงการติดต่อกับผู้ใช้และโปรแกรมของผู้ใช้ที่เขียนด้วยโปรแกรมอื่น ดังมีรายละเอียดของประโยชน์การใช้งานดังนี้

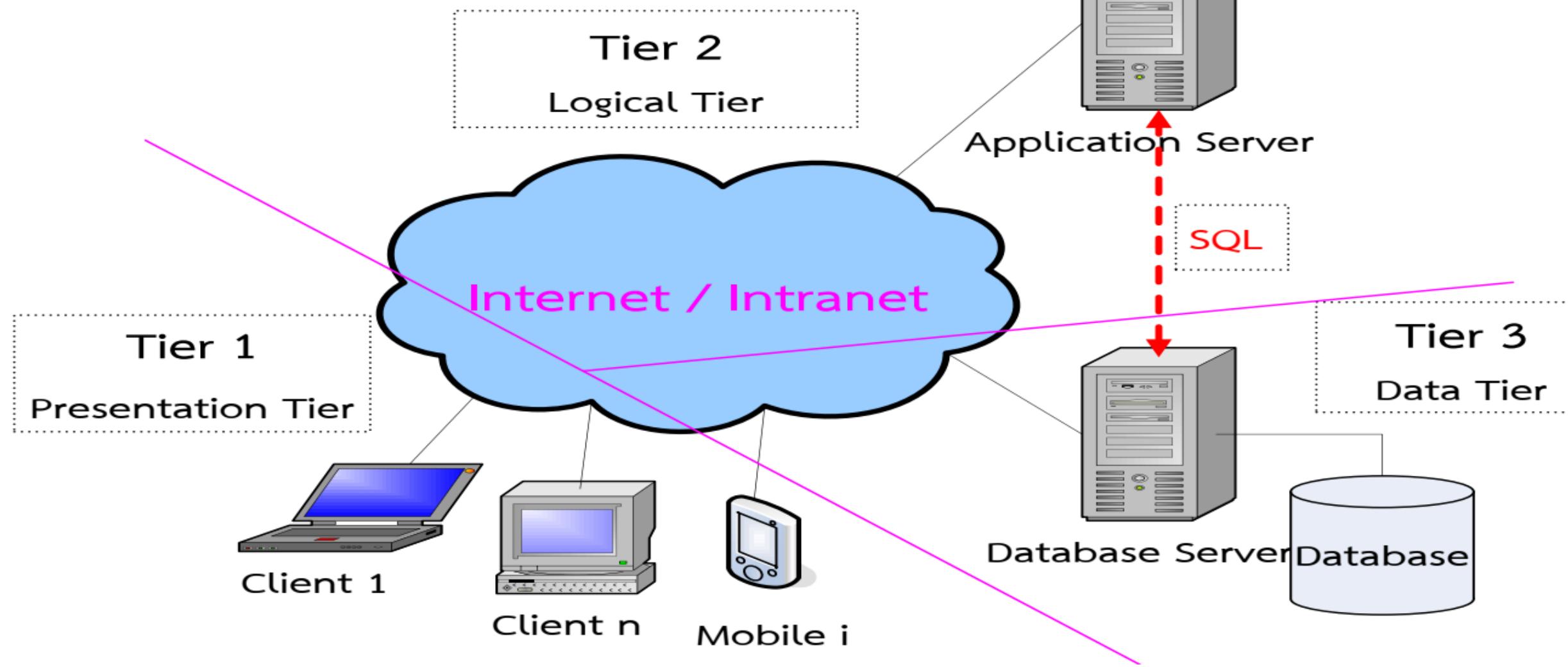
### 1.5.1 ภาษาเอสคิวแอล (SQL : Structured Query Language)

ช่วยจัดการการทำงานเกี่ยวกับโครงสร้างข้อมูล (Schema) ข้อมูลต่างๆที่เกิดขึ้น (Instance) โดยทุกผลลัพธ์ของการทำงานจะถูกจัดเก็บในตารางพจนานุกรมของข้อมูล (Data Dictionary Table) สามารถแบ่งตามลักษณะคำสั่งของการทำงานได้ 4 ส่วนคือ

#### 1) ส่วนจัดการเกี่ยวกับการกำหนดโครงสร้างข้อมูล (Data Definition Language : DDL)

เป็นคำสั่งกำหนดโครงสร้างข้อมูล (Schema) ตามการออกแบบฐานข้อมูลระบบงาน ไม่ว่าจะเป็นที่เก็บข้อมูลจริง หรือสิ่งที่ผู้ใช้แต่ละคนมองเห็น และเป็นการกำหนดโครงสร้างข้อมูลที่จัดเก็บจริงในอุปกรณ์เก็บข้อมูล (Hard Disk) ตัวอย่างเช่น CREATE , ALTER , DROP , TRUNCATE เป็นต้น

## 3-tier Client / Server Architecture



ตาราง Employee

A.K.

EmpNo , FName , LName , Sex , Salary , StartDate , ResignDate , WorkStatus,

ตาราง Department

F.K.

F.K.

F.K.

DeptCode , SuperEmpNo , SpvCode

F.K.

DeptCode , DeptName , TotalEmp , MgrEmpNo , MgrDate

การเชื่อมโยงตาราง Department กับตาราง Employee

- ข้อมูลการทำงานในแผนก จะใช้ Employee.DeptCode = Department.DeptCode
- ข้อมูลผู้บริหารแผนก จะใช้ Employee.EmpNo = Department.MgrEmpNo

## ตาราง Department (ข้อมูลแผนกของบริษัท)

DeptCode	หมายถึง รหัสแผนก	เป็น คีย์หลัก	เก็บค่าเป็นตัวอักษร 2 หลัก
DeptName	หมายถึง ชื่อแผนก	ต้องใส่ค่าข้อมูล	เก็บค่าเป็นตัวอักษร
TotalEmp	หมายถึง จำนวนพนักงานในปัจจุบันของแผนก(คำนวณ)	เก็บค่าเป็นตัวเลข	
MgrEmpNo	หมายถึง รหัสพนักงานที่เป็นผู้จัดการแผนกในปัจจุบัน		
		เป็น คีย์นอก (อ้างอิง Employee)	
MgrDate	หมายถึง วันที่เริ่มบริหารของผู้จัดการแผนกในปัจจุบัน	เก็บค่าเป็นวันที่	

```
CREATE TABLE Department
```

(1)	(DeptCode	CHAR(2)	PRIMARY KEY,
(2)	DeptName	VARCHAR2(20)	NOT NULL ,
(3)	TotalEmp	NUMBER(3),	
(4)	MgrEmpNo	CHAR(3),	
(5)	MgrDate	DATE);	

คำอธิบาย

- (1) สร้างกฎข้อบังคับคีย์หลัก (Primary Key) ให้กับ คีย์หลัก (PK.) ในระดับคอลัมภ์
  - (2) สร้างกฎข้อบังคับแบบต้องมีค่า(Not Null) ในระดับคอลัมภ์
- และการไม่ระบุชื่อของกฎข้อบังคับที่สร้างขึ้น ทาง Oracle DBMS จะสร้างชื่อให้โดยอัตโนมัติ

## ตาราง Employee (ข้อมูลพนักงานในบริษัท)

EmpNo	หมายเลขพนักงาน	เป็น <b>คีย์หลัก</b>	เก็บค่าเป็นตัวอักษร 3 หลัก
FName	นามสกุล	ต้องใส่ค่าข้อมูล	เก็บค่าเป็นตัวอักษร
LName	นามสกุล	ต้องใส่ค่าข้อมูล	เก็บค่าเป็นตัวอักษร
Sex	เพศ	ต้องใส่ค่าข้อมูล	เก็บค่าเป็น 'M' หรือ 'F'
Salary	เงินเดือน		เก็บค่าเป็นตัวเลขจำนวนจริง
StartDate	วันที่เริ่มทำงาน		เก็บค่าเป็นวันที่
ResignDate	วันที่ลาออกจากงาน		เก็บค่าเป็นวันที่
SpvCode	รหัสจังหวัดที่อยู่	เป็น <b>คีย์นอก</b> (อ้างอิง SetCountry)	
DeptCode	รหัสแผนกที่ทำงาน	เป็น <b>คีย์นอก</b> (อ้างอิง Department)	
SuperEmpNo	รหัสพนักงานที่เป็นหัวหน้างานสายตรง	เป็น <b>คีย์นอก</b> (อ้างอิง Employee)	
WorkStatus	สถานภาพการทำงาน จำแนกเป็นค่า 'W' แทนการเป็นพนักงานในปัจจุบัน (ResignDate มีค่าเป็น NULL) ค่า 'Q' แทนการเป็นอดีตพนักงาน (ResignDate มีค่าเป็น วันที่ลาออก)		

### 3) การทำลายตาราง ด้วยคำสั่ง DROP TABLE

```
DROP TABLE Employee;
```

```
DROP TABLE Department;
```

```
DROP TABLE SetProvince;
```

```
DROP TABLE SetPart;
```

```
DROP TABLE SetCountry;
```

เป็นการทำลาย ตารางทั้งหมดของโจทย์ตัวอย่าง (รูปที่ 3.2) ออกจากฐานข้อมูล โดยมีหลักการคือ ต้องเริ่มทำลาย  
จากตารางย่อย (ไม่มีการถูกอ้างอิง-Child) ไปยังตารางหลัก (ที่ถูกอ้างอิง-Parent)

### 3.5.3 การใช้คำสั่งสร้างดัชนี (CREATE INDEX)

การสร้างดัชนีจะใช้ประโยชน์เพื่อให้ สามารถเข้าถึงข้อมูลในตารางได้รวดเร็วขึ้น (Speed Performance) และ ควรสร้างบนคอลัมน์ของตารางขนาดใหญ่ที่มีการใช้งานบ่อยๆ ในส่วนของ WHERE และ ORDER BY โดยที่จะเป็น การสร้าง Index File ที่มีโครงสร้างเป็น B-Tree( $B^+$ -Tree)

```
CREATE INDEX Employee_Name_Idx ON Employee(FName,LName);
```

```
CREATE INDEX Employee_Salary_Idx ON Employee(Salary * 1.5 );
```

```
CREATE INDEX Department_Name_Idx ON Department (DeptName DESC);
```

```
SQL> CREATE INDEX Employee_Name_Idx ON Employee(FName,LName);
```

สร้างดัชนีแล้ว

```
SQL> CREATE INDEX Employee_Salary_Idx ON Employee(Salary * 1.5 );
```

สร้างดัชนีแล้ว

```
SQL> CREATE INDEX Department_Name_Idx ON Department (DeptName DESC);
```

สร้างดัชนีแล้ว

## 2) ส่วนจัดการเกี่ยวกับข้อมูล (Data Manipulation Language : DML)

เป็นคำสั่งจัดการเกี่ยวกับข้อมูล (Instance) ได้แก่ การสอบถามหรือเรียกค้นข้อมูล การเพิ่มเติมข้อมูลใหม่ การลบข้อมูลที่ไม่อยู่ การเปลี่ยนแปลงแก้ไขข้อมูลที่ต้องการที่อยู่ในฐานข้อมูล ตัวอย่างคำสั่ง ตัวอย่างเช่น SELECT , INSERT , DELETE และ UPDATE เป็นต้น

```
INSERT INTO DeptTest1 VALUES (99,'Computer','Bangkok');
```

```
UPDATE DeptTest1 SET DName = 'Test Dept '
WHERE DeptNo = 99;
```

```
DELETE FROM DeptTest1 WHERE DeptNo = 99;
```

```
SELECT * FROM View_Com_Manager ;
```

DEPTNAME	EMP_FNAME	LNAME	S	SALARY	STARTDATE
แผนกคอมพิวเตอร์	E07 พรญา	กล่องการค้า	F	40000	01 เม.ย. 2004
แผนกคอมพิวเตอร์	E15 กฤชฎา	วงศ์ใจดี	M	30000	16 ต.ค. 2004
แผนกคอมพิวเตอร์	E14 บุญมา	วงศ์ใจเรือน	F	42000	01 น.ค. 2004

ตัวอย่างที่ 9 ให้แสดงข้อมูล ชื่อพนักงาน เพศ เงินเดือน วันที่เริ่มทำงาน ของพนักงานที่ทำงานในปัจจุบัน ทั้งหมด โดยสนใจเฉพาะพนักงานเพศหญิง และ มีอัตราเงินเดือนมากกว่า 20,000 บาทแล้วให้เรียงลำดับข้อมูลตาม เพศ และเงินเดือน จากมากไปน้อย

```
SELECT FName , Sex , Salary , StartDate FROM Employee  
WHERE (Sex = 'F' AND Salary > 20000 )  
AND WorkStatus = 'W'  
ORDER BY Sex , Salary DESC;
```

FNAME	S	SALARY	STARTDATE
วนิดา	F	50,000.00	01 ม.ค. 2010
บุญมา	F	42,000.00	01 ม.ค. 2010
พัชรา	F	40,000.00	01 เม.ย. 2010
ญาดา	F	25,000.00	01 ม.ค. 2010

ตัวอย่างที่ 30 ให้แสดงข้อมูลรหัสแผนก จำนวนพนักงานที่ทำงานในปัจจุบัน โดยแบ่งกลุ่มข้อมูลตามรหัสแผนกและต้องการเฉพาะแผนกที่มีจำนวนพนักงานตั้งแต่ 3 คนขึ้นไป และให้เรียงลำดับตามรหัสแผนก

```
SELECT Employee.DeptCode , Min(DeptName) , COUNT(*)  
FROM Employee , Department  
WHERE WorkStatus = 'W' AND Employee.DeptCode = Department.DeptCode  
GROUP BY Employee.DeptCode  
HAVING COUNT(*) >= 3  
ORDER BY Employee.DeptCode;
```

DE	MIN(DEPTNAME)	COUNT(*)
3	แผนกการผลิต	4
4	แผนกวางแผนการผลิต	3
5	แผนกคณิตศาสตร์	3

```
SQL> SELECT * FROM DeptTest1;
```

<b>DEPTNO</b>	<b>DNAME</b>	<b>LOC</b>
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> INSERT INTO DeptTest1 VALUES (99,'Computer','Bangkok');
```

```
SQL> SELECT * FROM DeptTest1;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
99	Computer	Bangkok

```
SQL> UPDATE DeptTest1 SET DName = 'Test Dept'  
2 WHERE DeptNo = 99;
```

```
SQL> SELECT * FROM DeptTest1;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
99	Test Dept	Bangkok

```
DELETE FROM DeptTest1 WHERE DeptNo = 99;
```

```
SQL> SELECT * FROM DeptTest1;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

### 3) ส่วนการจัดการเกี่ยวกับการประมวลผลกลุ่มงาน (Transaction Processing)

เป็นคำสั่งที่ควบคุมการประมวลผลของกระบวนการทำงานจริง (Business Process) เพื่อให้มีความถูกต้องตามที่ได้ถูกออกแบบไว้ ได้แก่ คำสั่ง COMMIT , ROLLBACK , SAVEPOINT เป็นต้น

**COMMIT;**

**SQL> DELETE FROM DeptTest1;**

**SQL> COMMIT;**

**SQL> SELECT \* FROM DeptTest1;**

**ไม่มีข้อมูลแล้ว**

```
SQL> DELETE FROM DeptTest1;
```

ลบແກ້ວມາລົງຈະ 4 ແກ້ວ

```
SQL> SELECT * FROM DeptTest1;
```

ບໍ່ໄດ້ຮັບຜົນໄດ້

```
INSERT INTO DeptTest1 VALUES (99,'Computer','Bangkok');
```

```
SQL> SELECT * FROM DeptTest1;
```

DEPTNO	DNAME	LOC
99	Computer	Bangkok

```
SQL> ROLLBACK;
```

```
SQL> SELECT * FROM DeptTest1;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

#### 4) ส่วนการจัดการเกี่ยวกับสิทธิการใช้งานข้อมูล (Authority) ในฐานข้อมูล

เป็นคำสั่งที่เกี่ยวกับการบริหารสิทธิการใช้งานข้อมูล (Authority) ของแต่ละ Database User ในฐานข้อมูล ได้แก่ คำสั่ง GRANT , REVOKE เป็นต้น

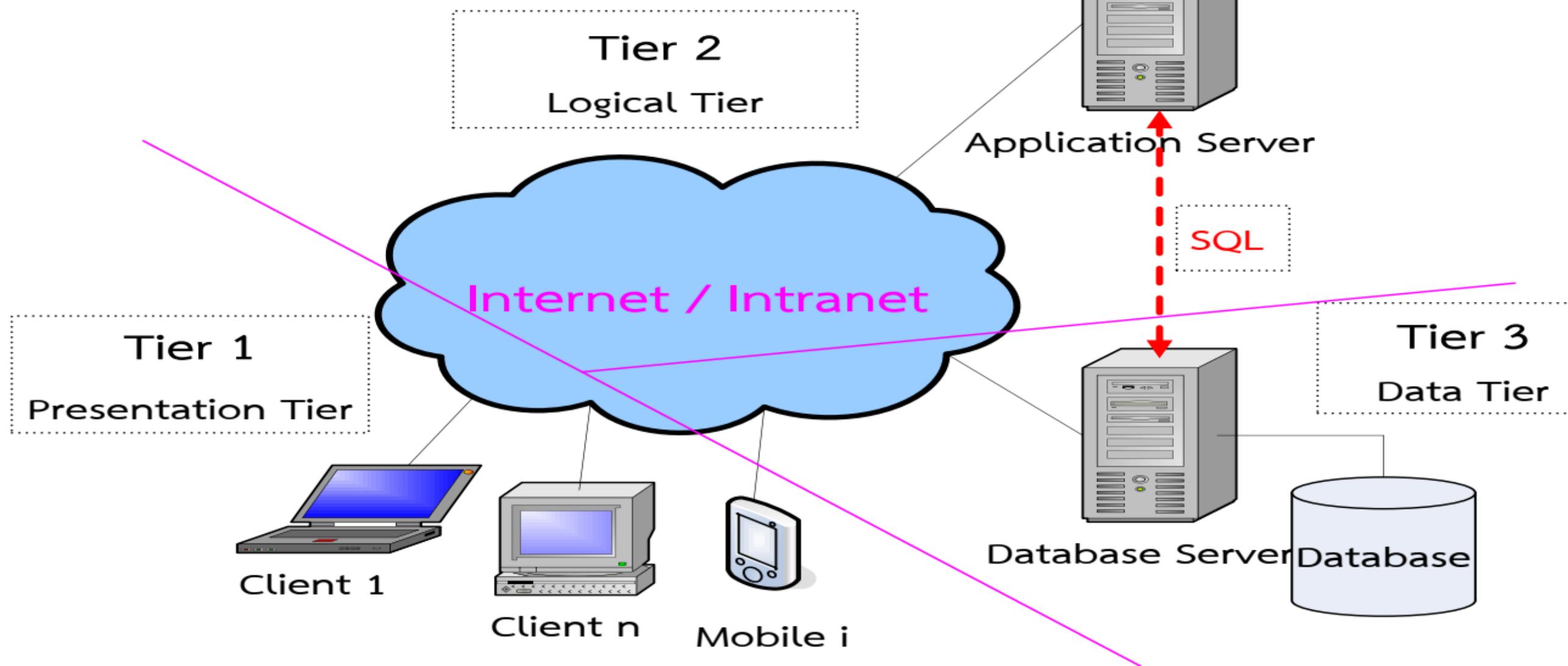
##### 1) สิทธิในระบบ (System Privilege)

```
SQL> GRANT CREATE SESSION TO com02;
```

##### 2) สิทธิในสิ่งที่สร้างขึ้น (Object Privilege)

```
GRANT SELECT , INSERT ON Test_A TO com01;
```

## 3-tier Client / Server Architecture



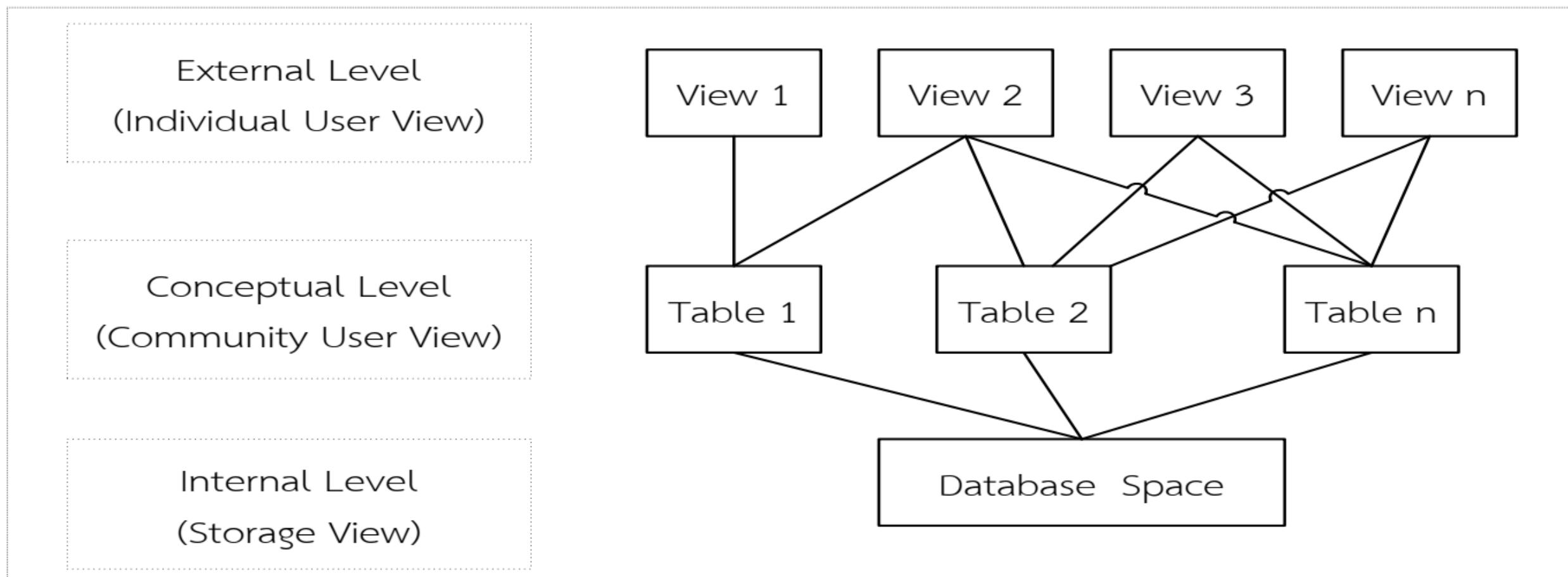
การเชื่อมต่อไปเพื่อใช้งานที่ Database Server นั้นต้องใช้งานผ่าน Database User ที่มีสิทธิในระบบ (System Privilege) ซึ่ง **CREATE SESSION** โดยที่ผู้ดูแลระบบการจัดการฐานข้อมูล ที่เรียกว่า Database Administrator (DBA) จะต้องทำการสร้างผู้ใช้งานฐานข้อมูล (Database User) ขึ้นมาด้วยคำสั่ง CREATE USER และต้องมอบสิทธิในการเข้าใช้งานด้วยคำสั่ง GRANT CREATE SESSION ให้กับ ผู้ใช้งานฐานข้อมูล ด้วย

```
SQL> conn com02/com02
ERROR:
ORA-01045: user COM02 lacks CREATE SESSION privilege; logon denied
```

```
SQL> GRANT CREATE SESSION TO com02;
```

```
SQL> conn com02/com02
เชื่อมต่อแล้ว
```

### 1.5.2 การจัดการด้านความเป็นอิสระของข้อมูล แบ่งเป็น 2 ลักษณะโดยอ้างอิงรูปที่ 1.3 คือ



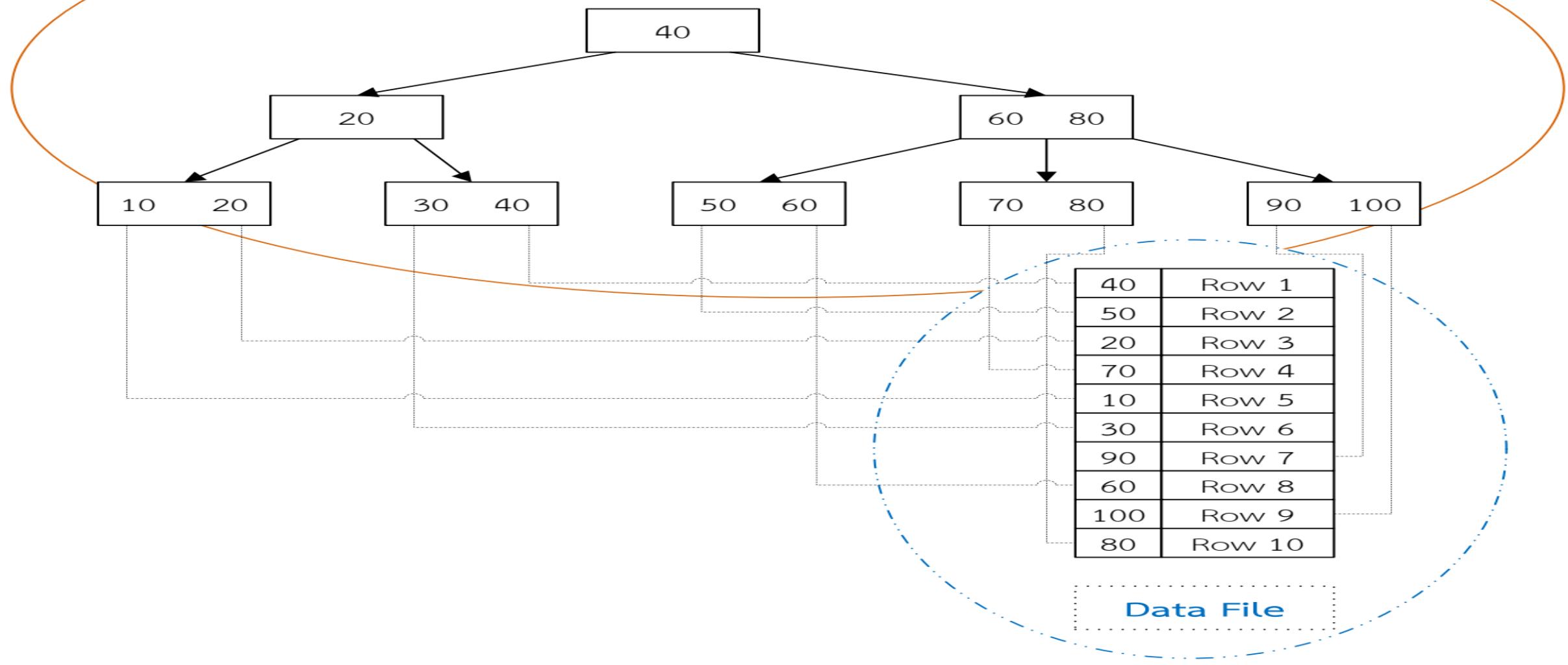
รูปที่ 1.3

สถาปัตยกรรม 3 นิยามข้อมูล (The 3-schema Architecture)

หมายเหตุ

เมื่อสั่งสร้างตารางจะมีการสร้างที่เก็บข้อมูลจริงใน Internal Level ตามสถาปัตยกรรม 3 นิยามข้อมูล (อ้างอิงรูปที่ 1.3) โดยลำดับการใส่ข้อมูลในตารางจะเรียงลำดับตามการใส่ข้อมูลจริงใน Data File (ซึ่งเป็นการเก็บข้อมูลจริง ในส่วนของ Data Segment ของ Permanent Tablespace ใน Oracle DBMS) แล้วจะมีการสร้าง Index File (ซึ่งเป็นการเก็บข้อมูลจริงที่มีโครงสร้างเป็น B-Tree( $B^+$ -Tree) ในส่วนของ Index Segment ของ Permanent Tablespace ใน Oracle DBMS) เพื่อใช้ในการเข้าถึงข้อมูลแบบ Index Sequential ดังแสดงได้ดังรูปที่ 3.3

### Index File ที่มีโครงสร้างเป็น B-Tree Order 3



รูปที่ 3.3

โครงสร้างตาราง (Table) และดัชนี (Index) ที่เกิดขึ้นเมื่อสร้างตารางพร้อมคีย์หลัก

1) ความเป็นอิสระของข้อมูลในระดับตรรกะ (Logical Data Independence)

เมื่อมีการแก้ไขโครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด (Table ต่างๆ) จะไม่มีผลกระทบถึงโครงสร้างข้อมูลในนิยามข้อมูลระดับภายนอก (View ต่างๆ) ที่ผู้ใช้งานงานกันอยู่ เช่น การเปลี่ยนแปลงขนาดคอลัมน์ หรือเพิ่มคอลัมน์เข้าไปในตาราง และวิว (View) ยังทำงานได้ตามปกติ

2) ความเป็นอิสระของข้อมูลในระดับกายภาพ (Physical Data Independence)

เมื่อมีการแก้ไขโครงสร้างข้อมูลในนิยามข้อมูลระดับภายนอก (File ต่างๆ) จะไม่มีผลกระทบถึงโครงสร้างข้อมูลในนิยามข้อมูลระดับแนวคิด (Table ต่างๆ) เช่น มีการเปลี่ยนแปลงวิธีการเข้าถึงข้อมูลในนิยามข้อมูลระดับภายนอกโดยการเปลี่ยนแปลงลักษณะการจัดเก็บจาก ไอแซม (ISAM) เป็นบีทรี (B-tree) หรือแม้กระทั่งเปลี่ยนแปลงอุปกรณ์ที่ใช้จัดเก็บข้อมูลอย่างเช่น ฮาร์ดดิสก์ (Hard Disk)

### 1.5.3 ความปลอดภัยของข้อมูล (Data Security)

มีระบบการสอบถามบัญชีรายชื่อพร้อมรหัสลับของผู้เข้ามาใช้งาน โดยจะอนุญาตให้เข้าทำงานในส่วนที่ผู้ใช้เกี่ยวข้องและมีสิทธิการเข้าถึงเท่านั้น นอกจากนี้ข้อมูลที่เก็บอยู่ในฐานข้อมูลจะถูกทำการเข้ารหัส (Encryption) เพื่อป้องกันการเข้าถึงข้อมูลด้วยภาษาอื่นๆ โดยไม่ผ่านระบบการจัดการฐานข้อมูล (DBMS)

### 1.5.4 การจัดการอำนาจสิทธิการใช้ข้อมูล (Data Authority)

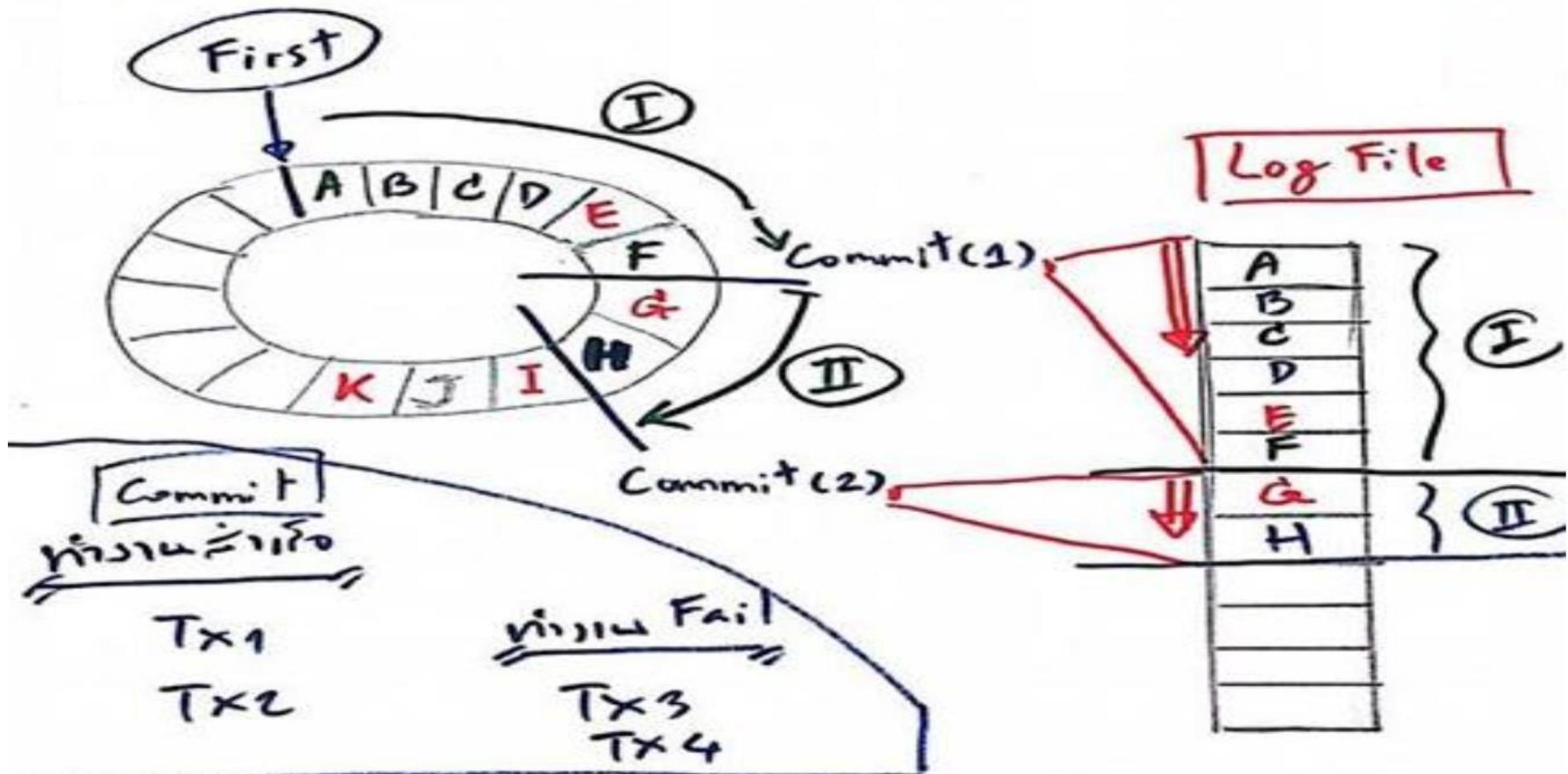
สามารถกำหนดสิทธิการใช้งานข้อมูลให้แก่ผู้ใช้ต่างๆ (Database User) ได้ โดยการใช้คำสั่ง GRANT หรือเพิกถอนสิทธิโดยใช้คำสั่ง REVOKE นอกจากนี้ยังใช้ VIEW เพื่อช่วยกำหนดสิทธิการใช้ข้อมูลของผู้ใช้แต่ละคนใน尼ยามข้อมูลระดับภายนอกได้อีกด้วย

### 1.5.5 การกู้ข้อมูลโดยอัตโนมัติเมื่อระบบเกิดความเสียหาย (Recovery Control)

หลักการการประมวลผลกลุ่มงาน (Transaction Processing) คือ เมื่อทำการประมวลผลกลุ่มงาน (Logical Unit of Work) จะมีผลลัพธ์ใน 2 แนวทางคือ

- 1) ถ้าทำงานเสร็จตามต้องการจะต้องเสร็จด้วยกันทั้งหมด
- 2) ถ้าเกิดปัญหาขณะประมวลผลกลุ่มงานจะต้องยกเลิกทั้งหมด

โดยที่ Commit คือ คำสั่งยืนยันการทำงานสำเร็จ และถ้าสั่งคำสั่ง Commit แล้วระบบเกิดความเสียหายขึ้น ระบบการจัดการฐานข้อมูลจะรับรองว่าข้อมูลที่ Commit แล้วจะไม่สูญหายไป ส่วน Rollback คือ คำสั่งยกเลิกการทำงานของกลุ่มงานนั้นๆ และระบบการจัดการฐานข้อมูลจะรับรองว่าข้อมูลเดิมก่อนการทำงานของกลุ่มงานจะถูกนำมาใช้งานโดยข้อมูลไม่เปลี่ยนแปลง



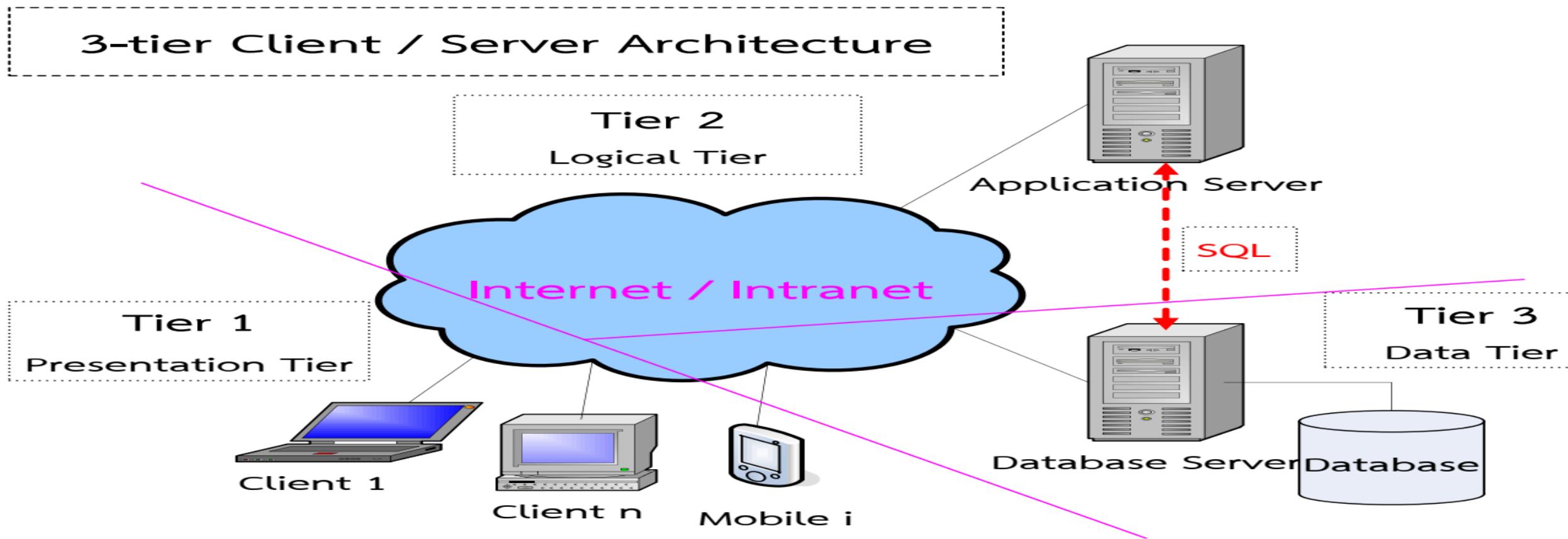
## 1.5.6 การดูแลผู้ใช้หลายคนให้สามารถทำงานได้พร้อมๆ กัน (Concurrency Control)

สามารถรองรับการทำงานพร้อมกันของผู้ใช้หลายคน โดยเกิดปัญหาได้ 3 ประเภท คือ

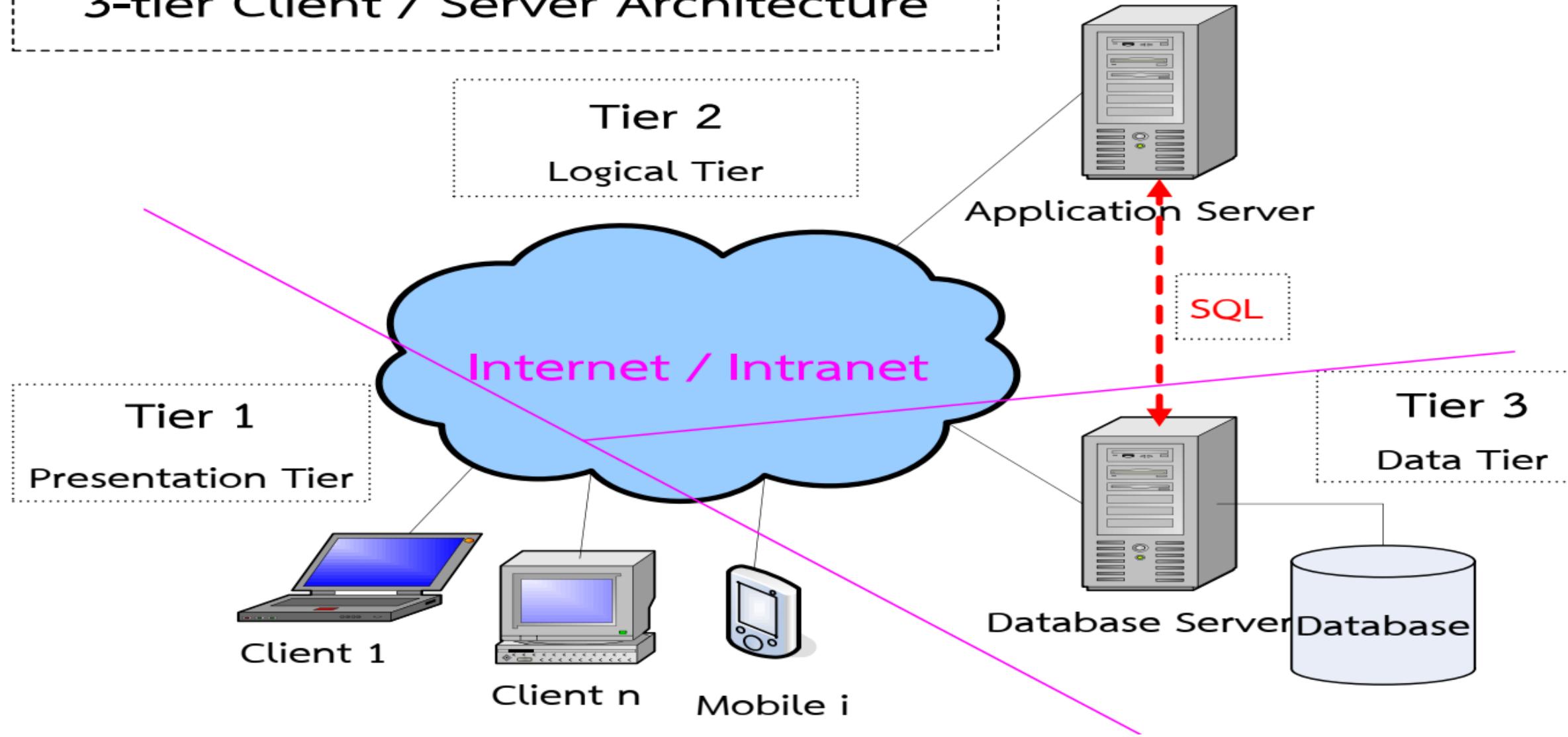
- 1) ปัญหาการสูญเสียการเปลี่ยนแปลงข้อมูล (The Lost Update Problem) เกิดจากการอ่านข้อมูลตัวเดียว กันไปใช้งาน โดยที่มีการเปลี่ยนแปลงข้อมูลตัวนั้นในเวลาที่ไม่พร้อมกัน
- 2) ปัญหาที่เกิดจากการยกเลิกการเปลี่ยนแปลงข้อมูล (The Uncommitted Dependency Problem) เกิดจากการนำเอาข้อมูลใหม่ที่ถูกยกเลิกการประมวลผลกลุ่มงานไปใช้งาน
- 3) ปัญหาที่เกิดจากการขัดแย้งของการทำงาน (The Inconsistent Analysis Problem) เกิดจากการนำเอาข้อมูลที่ยังไม่เสร็จสิ้นการเปลี่ยนแปลง (การโอนเงินจากบัญชี A ไปยังบัญชี B) และส่งผลต่ออีกการทำงานหนึ่งไปใช้งาน (การคำนวณยอดเงินคงเหลือของทุกบัญชีในสาขาของธนาคาร)

## 1.6 สถาปัตยกรรมการพัฒนาระบบงาน

ลักษณะการทำงานของการพัฒนาระบบงานสารสนเทศที่มีการใช้งานฐานข้อมูล จะต้องทำงานกับเครื่องแม่ข่ายฐานข้อมูล (Database Server) โดยต้องติดตั้งระบบการจัดการฐานข้อมูล (DBMS) และมีลักษณะการทำงานอยู่ในรูปแบบของสถาปัตยกรรมการทำงานแบบ 3 ส่วนการทำงาน (3-tier Client/Server Architecture) ดังแสดงได้ดังรูปที่ 1.4 และต้องใช้ภาษาเอสคิวแอล (SQL) ในการทำงานกับฐานข้อมูลเท่านั้น

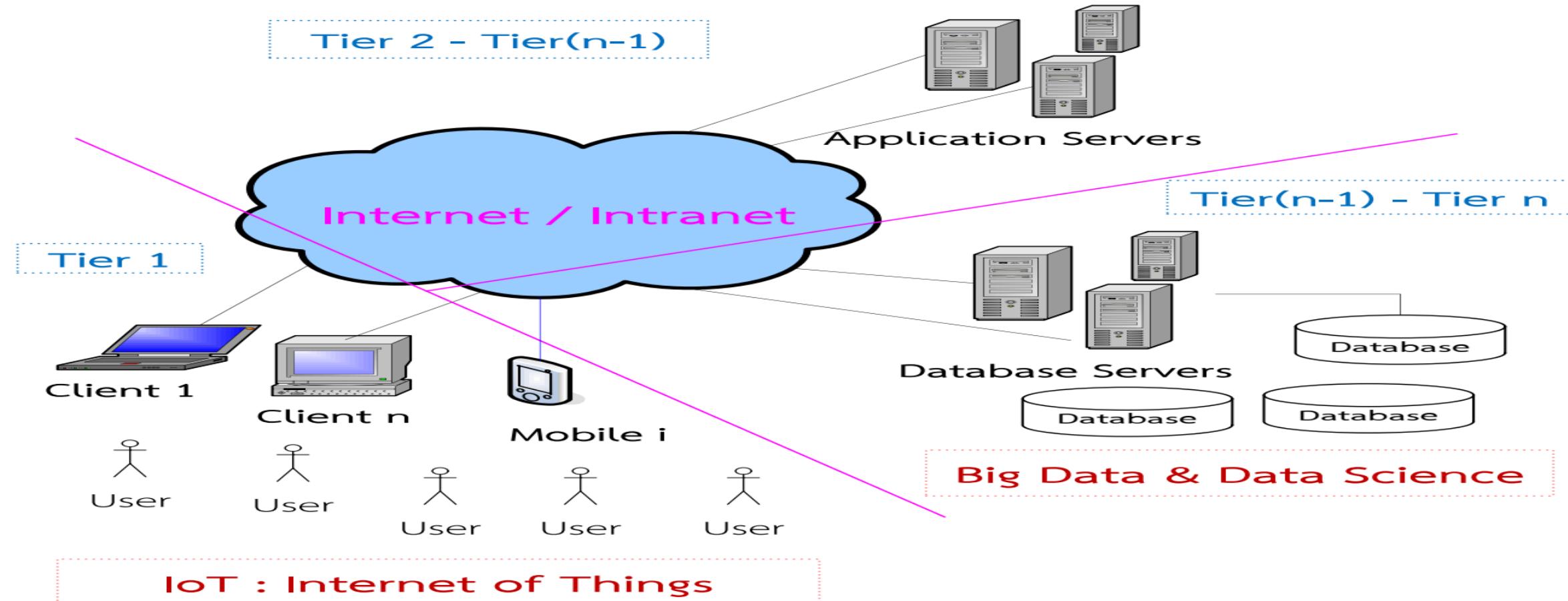


## 3-tier Client / Server Architecture

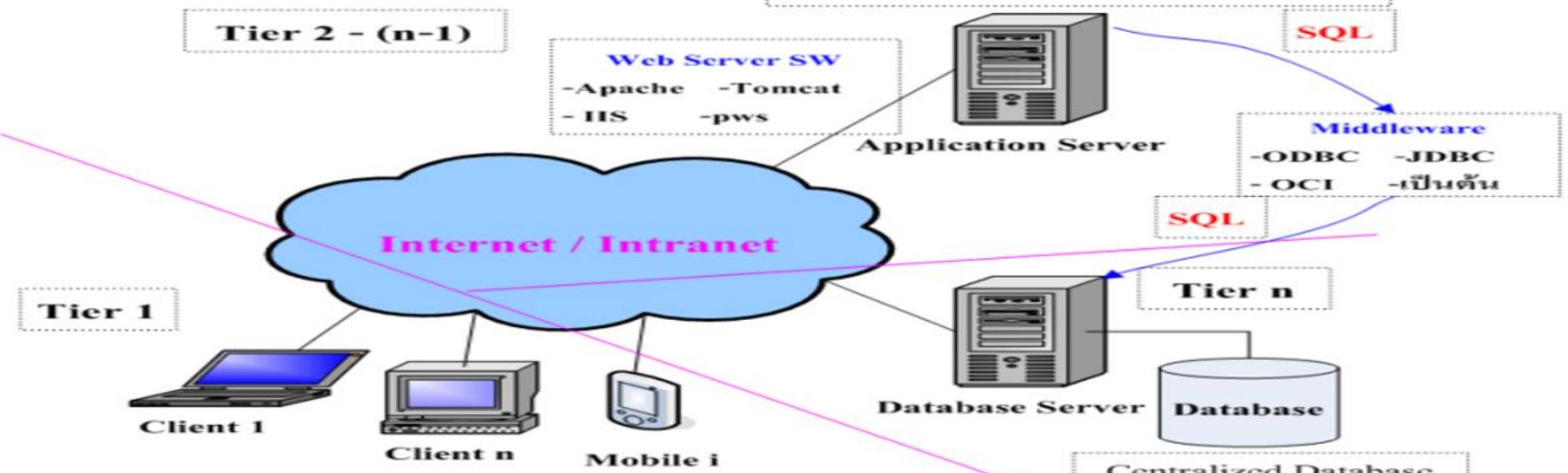


และการใช้งานระบบงานหรือโปรแกรมต่างๆ ในปัจจุบันเป็นรูปแบบของการประมวลผลแบบคลาวด์เมฆ (Cloud Computing System) และมีรูปแบบการทำงานเป็นแบบสถาปัตยกรรมการทำงานแบบ n ส่วนการทำงาน (n-tier Client/Server Architecture) แสดงดังรูปที่ 1.5

### n-tier Client / Server Architecture



## **n-tier Client / Server Architecture**



## ตัวอย่าง Web Browser SW.

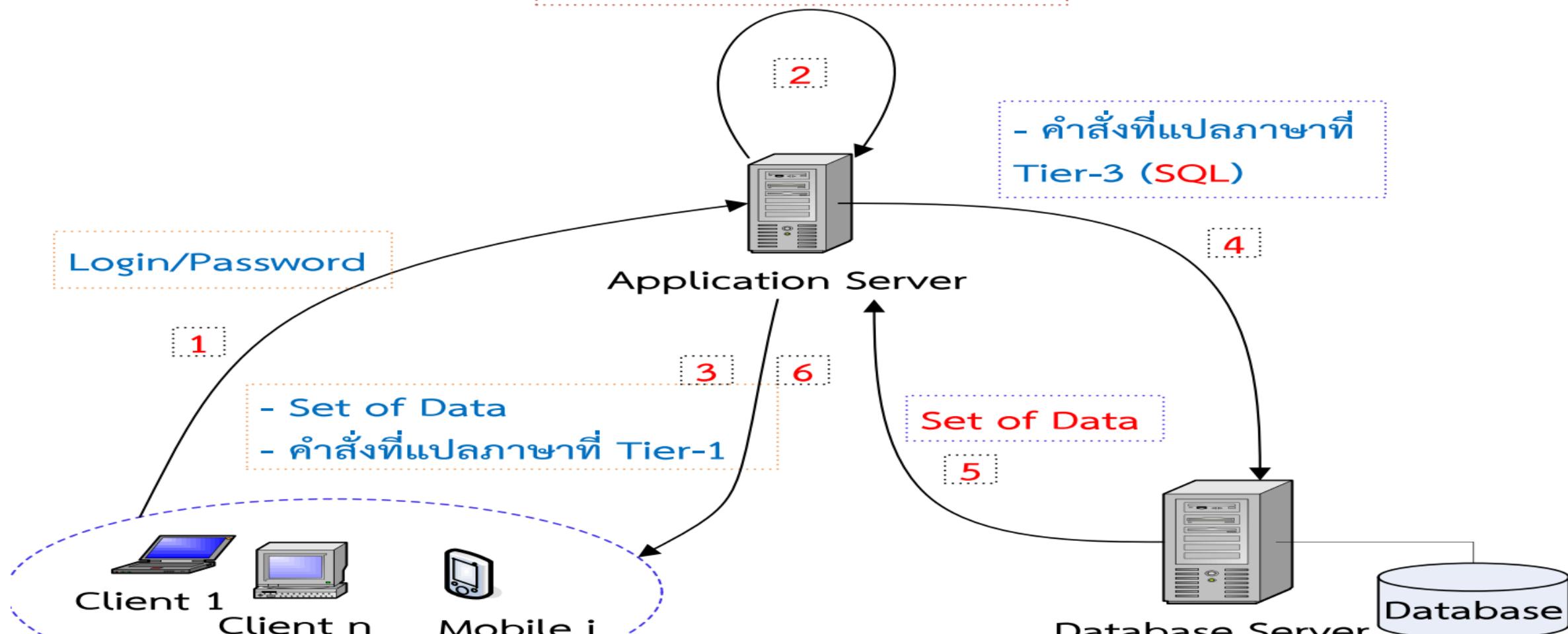
- Internet Explorer (IE)
  - Firefox - Google Chrome - Opera - Netscape - เป็นต้น
  - โดยที่สามารถแปลงภาษาต่อไปนี้ได้ + JSON
  - HTML - Java Script - Ajax (Java Script + XML) - Flash เป็นต้น

ตัวอย่าง DBMS

(Database Management System)

- |                       |      |           |         |
|-----------------------|------|-----------|---------|
| - Oracle              | -DB2 | -Informix | -Ingres |
| -Microsoft SQL*Server |      | -MySQL    |         |
| -Microsoft Access     |      | -เป็นต้น  |         |
| -ProgreSQL            |      | -NoSQL    |         |

ประมวลผลตามตัวแปลภาษา  
ที่ติดตั้งบน Tier-2



เมื่อมีการร้องขอใช้การทำงาน (Request) จากโปรแกรมประเภท Web Browser หรือโปรแกรมที่ติดตั้งบน Smart Phone/Tablet (การทำงานบน Tier-1) ไปยังเครื่องคอมพิวเตอร์ที่ให้บริการด้านโปรแกรมระบบงาน (Web Site / Application Server) ที่มีติดตั้งโปรแกรมประเภท Web Sever และมีตัวแปลงภาษา (Translator) และเมื่อได้รับ การตอบสนอง (Response) จะเรียกว่า มีการสร้างการเชื่อมต่อ(Session) อย่างสมบูรณ์ [หมายเลข 1] และจะทำการ ประมวลผลตาม Business Domain และ Process Flow (การทำงานบน Tier-2) [หมายเลข 2] โดยจะมีส่งผลลัพธ์ การประมวลผลกลับไปยัง Tier-1 [หมายเลข 3] และถ้ามีคำสั่งภาษา SQL จะดำเนินการติดต่อขอใช้งานผ่านโปรแกรม การเชื่อมต่อ (Database Middleware) ไปยังเครื่องคอมพิวเตอร์ที่ให้บริการด้านฐานข้อมูล (การทำงานบน Tier-3) และ เมื่อสามารถเชื่อมต่อ(Session) อย่างสมบูรณ์ก็จะส่งคำสั่งภาษา SQL ไปให้ DBMS ประมวลผล [หมายเลข 4] และ ดำเนินการส่งกลุ่มข้อมูล (กรณีเป็นคำสั่ง SELECT) กลับไปยัง Tier-2 [หมายเลข 5] และส่งผ่านข้อมูลไปให้แสดงผลที่ Tier-1 [หมายเลข 6]

## การแสดงความสัมพันธ์ระหว่าง ประเภทระบบงานสารสนเทศ กับ ประเภทของฐานข้อมูล



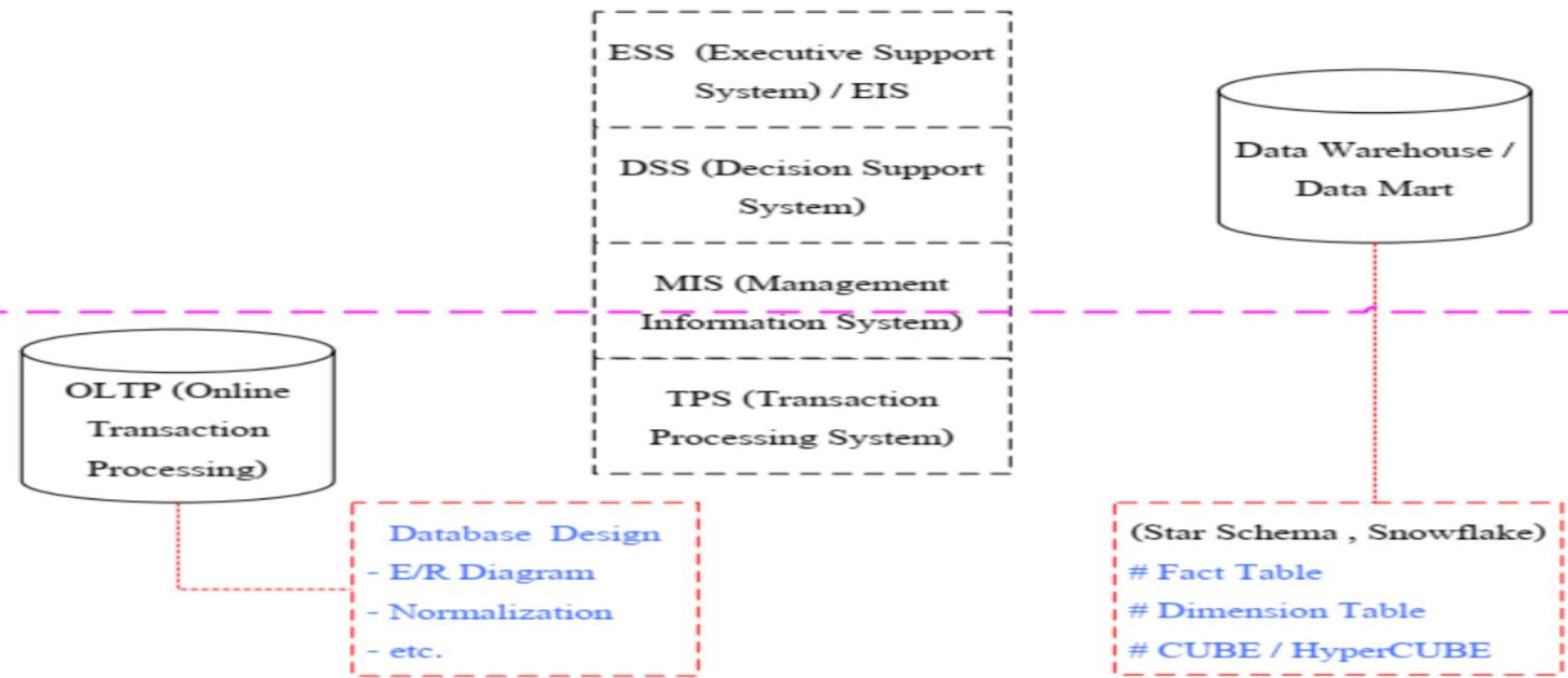
ระบบสารสนเทศเพื่อการตัดสินใจระดับสูง  
(ESS : Executive Support System) /  
(EIS : Executive Information System)

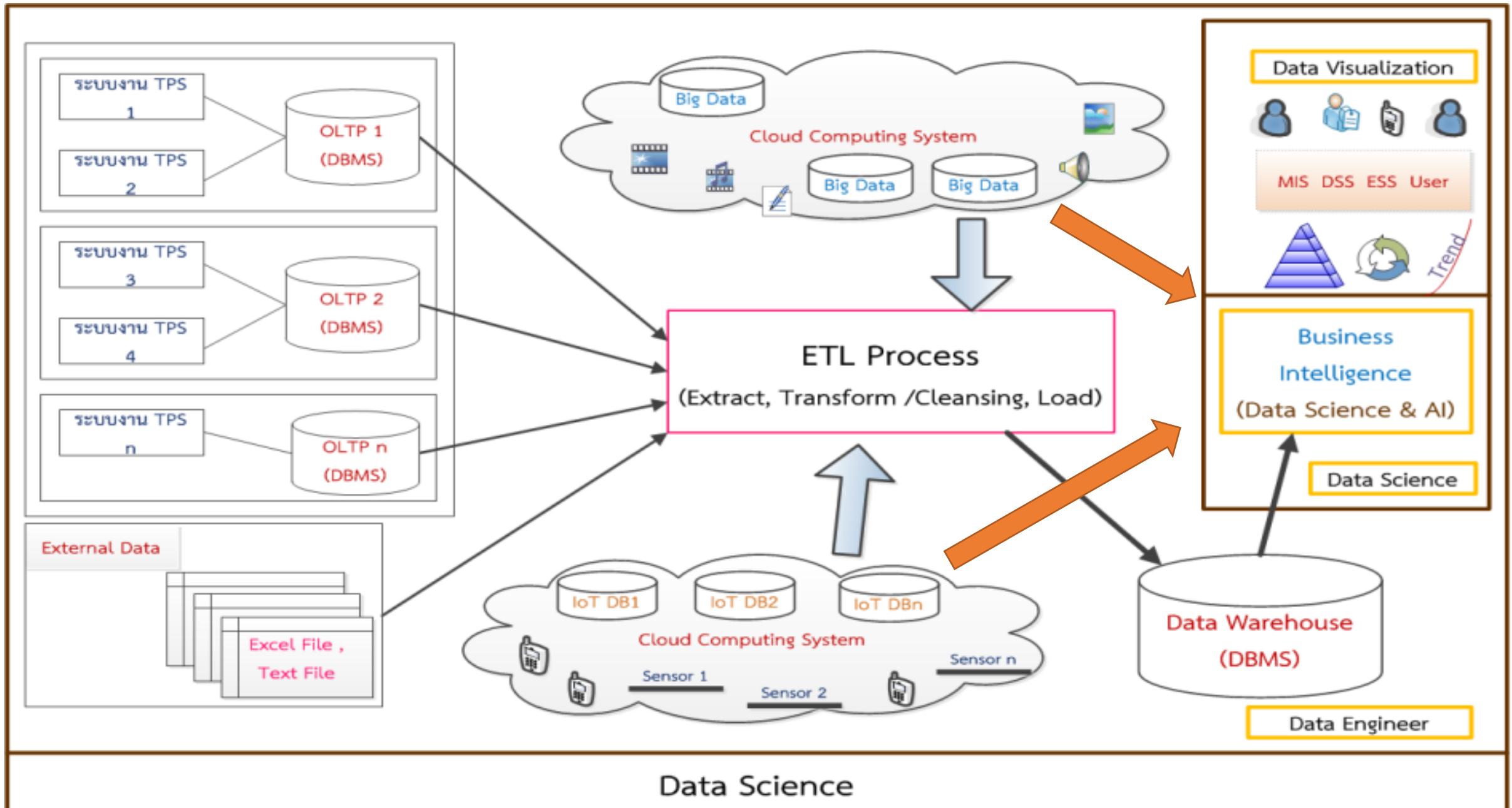
ระบบสารสนเทศเพื่อการตัดสินใจ  
(DSS : Decision Support System)

ระบบสารสนเทศเพื่อการบริหาร  
(MIS : Management Information System)

ระบบการประมวลผลความการทำงานปกติ  
(TPS : Transaction Processing System)

## สรุปความสัมพันธ์ระหว่างประเภทระบบงานสารสนเทศกับประเภทฐานข้อมูล





# บรรณานุกรม

1. C.J. Date , " An Introduction to Database Systems 8th Edition " , Addison-Wesley, 2004.
2. E.F. Codd , " The Relational Model for Database Management : Version 2" , Addison-Wesley, 1990.
3. Ramez Elmasri, Shamkant B. Navathe , " Fundamentals of Database Systems 6th Edition " , Addison-Wesley, 2011.
4. Abraham Siberschatz , Henry F. Korth , S. Sudarshan , " Database System Concept 6th Edition " , McGraw-Hill , 2011.
5. G. M. Nyssen , E. D. Falkenberg , " Introduction to IBM SQL 1984.
6. Michael V. Mannino , Database Design , Application Development , & Administration 3rd Edition " , McGraw-Hill, 2007.
7. Peter Rob , Carlos Coronel , " Database Systems Design , Implementation , Management 7th Edition " , 2007.
8. Connolly , Thomas and Begg , " Database System : A Practical Approach to Design,Implementation,Management 5th Edition " , Pearson Addison Wesley , 2010.
9. Oracle Co,Ltd. , O10g CD Pack Document Set.
10. ศุภมิตร จิตตะยศธรรม , " เอกสารประกอบคำสอนวิชาระบบฐานข้อมูล " , คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
11. สมนึก เจียมเจริญเดช , " Oracle 9i รีลีส 2 " , เอส พี บีค , 2546.

Q & A