Discuss pro and con of each database structure as well as discuss situation in which each structure is appropriate.

1. Relational Database

ข้อดี (Pro):

- จัดเก็บข้อมูลเป็นตารางที่เข้าใจง่าย มีโครงสร้างที่ชัดเจน
- มีมาตรฐานชัดเจน ใช้ภาษา SQL ที่เป็นมาตรฐานสากล
- รองรับธุรกรรมที่ซับซ้อนและแม่นยำ (ACID: ข้อมูลถูกต้องสมบูรณ์)

ข้อเสีย (Con):

- ขยายระบบแนวราบ (เพิ่มเซิร์ฟเวอร์หลายตัวพร้อมกัน) ได้ยาก
- ไม่ยืดหยุ่นกับข้อมูลที่เปลี่ยนแปลงโครงสร้างบ่อยๆ

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- ระบบการเงิน, ธนาคาร, ระบบจองตั๋วเครื่องบิน, ระบบบัญชี, ระบบร้านค้าออนไลน์ เช่น Lazada, Shopee, ระบบลงทะเบียนนักศึกษา

2. NoSQL Database

ข้อดี (Pro):

- เก็บข้อมูลที่โครงสร้างไม่แน่นอนหรือเปลี่ยนแปลงบ่อยได้ดีมาก
- ขยายระบบ (Scaling) แนวราบง่ายมาก สามารถเพิ่มเครื่องเซิร์ฟเวอร์ได้หลายตัว
- เขียน-อ่านข้อมูลได้เร็วมาก

ข้อเสีย (Con):

- บางครั้งข้อมูลซ้ำซ้อน ไม่สมบูรณ์แบบ 100% (Eventual consistency)
- ไม่เหมาะกับงานที่ต้องการความแม่นยำในธุรกรรมสูงมาก เช่นระบบการเงินที่ซับซ้อน

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- Social media (Facebook, Twitter), แอปพลิเคชันมือถือ, ระบบ IoT, ระบบแชทออนไลน์, แอปพลิเคชันสตรีมมิ่ง (Netflix, YouTube)

3. Hierarchical Database

ข้อดี (Pro):

- โครงสร้างข้อมูลเป็นลำดับชั้นชัดเจน เรียกใช้งานง่าย
- อ่านข้อมูลที่เป็นโครงสร้างลำดับชั้นได้เร็วมาก

ข้อเสีย (Con):

- ไม่ยืดหยุ่นกับข้อมูลที่เปลี่ยนแปลงโครงสร้างบ่อย
- การเชื่อมโยงข้อมูลจำกัดแค่แบบ Parent-Child เท่านั้น (เชื่อมโยงหลายทางไม่ได้)

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- ระบบจัดการไฟล์, ระบบข้อมูลเก่าในองค์กรใหญ่ (Legacy system), ระบบข้อมูลพนักงานที่มีหัวหน้า-ลูกน้องชัดเจน

4. Network Database

ข้อดี (Pro):

- สามารถเชื่อมโยงข้อมูลได้หลายทิศทาง (Many-to-many)
- มีความยืดหยุ่นในการจัดการความสัมพันธ์ข้อมูลดีกว่าแบบ Hierarchical

ข้อเสีย (Con):

- ซับซ้อนกว่า Hierarchical มาก ทำให้จัดการและดูแลระบบยาก
- การออกแบบโครงสร้างฐานข้อมูลค่อนข้างยุ่งยาก

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- ระบบเฉพาะทางเก่าแก่ เช่น ระบบโทรคมนาคมยุคก่อน, ระบบจองสายการบินยุคเก่า

5. Object-Oriented Database

ข้อดี (Pro):

- เก็บข้อมูลในรูปแบบ Object เช่นเดียวกับภาษาเขียนโปรแกรมแบบ OOP
- สะดวกในการนำข้อมูลไปใช้งานต่อในโปรแกรมที่เขียนแบบ OOP

ข้อเสีย (Con):

- ประสิทธิภาพในการ Query ข้อมูลซับซ้อนต่ำกว่า Relational
- มีเครื่องมือสนับสนุนน้อยกว่า ไม่ได้รับความนิยมมากนัก

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- ระบบออกแบบผลิตภัณฑ์ (CAD/CAM), ระบบเกม, แอปพลิเคชันเฉพาะที่พัฒนาด้วยภาษา OOP (เช่น Java, C#, Python)

6. Graph Database

ข้อดี (Pro):

- เก็บข้อมูลที่เน้นความสัมพันธ์ระหว่างข้อมูลจำนวนมากได้ดีมาก
- สามารถ Query ข้อมูลที่มีความสัมพันธ์ซับซ้อนได้เร็วกว่าแบบอื่นมากๆ

ข้อเสีย (Con):

- ไม่เหมาะกับข้อมูลที่ไม่ค่อยมีความสัมพันธ์ หรือข้อมูลจำนวนมหาศาลที่ไม่มีการเชื่อมโยง
- ต้องมีความรู้เฉพาะทางในการออกแบบโครงสร้างที่เหมาะสม

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- Social networks (Facebook, LinkedIn), ระบบแนะนำสินค้า (Recommendation system), ระบบค้นหาเส้นทาง, ระบบตรวจสอบการโกงทางการเงิน

7. Time-series Database

ข้อดี (Pro):

- ประสิทธิภาพสูงมากในการเก็บข้อมูลที่เปลี่ยนแปลงตามเวลา
- วิเคราะห์ข้อมูลตามช่วงเวลาได้เร็วและแม่นยำมาก

ข้อเสีย (Con):

- ไม่เหมาะกับข้อมูลที่ไม่มีความเกี่ยวข้องกับเวลา
- การจัดการข้อมูลทั่วไปที่ไม่ใช่ข้อมูลเชิงเวลามีประสิทธิภาพต่ำ

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- ระบบเก็บข้อมูลเซ็นเซอร์ IoT, ข้อมูลหุ้น, การเก็บ Log ของ Server, ข้อมูลสภาพอากาศ, ข้อมูลราคาคริปโต

8. Columnar Database

ข้อดี (Pro):

- สามารถ Query และวิเคราะห์ข้อมูลจำนวนมากได้เร็วมาก โดยเฉพาะข้อมูลขนาดใหญ่
- เหมาะสำหรับการทำ Aggregation (Sum, Average, Count) ได้เร็ว

ข้อเสีย (Con):

- การเขียนข้อมูล (Insert, Update) ช้ากว่าแบบ Row-based
- ไม่เหมาะกับระบบที่มีการเขียนข้อมูลบ่อยครั้งและรวดเร็ว

สถานการณ์ที่เหมาะสม (Appropriate Situations):

- ระบบวิเคราะห์ข้อมูล Big Data, Data Warehouse, ระบบ Business Intelligence (BI), ระบบสรุปรายงานขององค์กรขนาดใหญ่