



สถาบัน **THE BEST CENTER**

2145/7 ซ.รามคำแหง 43/1 ถ.รามคำแหง แขวงหัวหมาก เขตบางกะปิ กรุงเทพฯ 10240

โทร. 0-2318-6868, 0-2314-1492 โทรสาร 0-2718-6274

www.thebestcenter.com facebook.com/bestcentergroup

คุณภาพทางวิชาการต้องมาที่ 1

คู่มือเตรียมสอบ

นักวิชาการคอมพิวเตอร์

การประสานครหลวง เจาะข้อสอบมากกว่า 900 ข้อ



ประกอบด้วย

1. ความรู้พื้นฐานด้านคอมพิวเตอร์
 - 1.1 ความเข้าใจการออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)
 - 1.2 ความเข้าใจวงจรพัฒนาระบบงาน (System Development Life Cycle: SDLC) (พื้นฐาน)
 - 1.3 ความเข้าใจระบบปฏิบัติการ (Operating System) หลักการทำงานของระบบปฏิบัติการ การจัดการหน่วยความจำ การจัดการกระบวนการ การจัดการไฟล์
 - 1.4 มาตรฐาน IT และหลักประกันคุณภาพและความปลอดภัยในการทำงาน การบริหารความเสี่ยง
2. ด้านพัฒนาระบบงาน (Software)
 - 2.1 ความสามารถ ทักษะ ในการเขียนโปรแกรม เช่น JAVA, PHP, Javascript, Mobile Application, Web Application, Python, C#, SQL, Microservice, DevOps, DevSecOps, เขียนฟังก์ชันรับข้อมูลจาก API และจัดเก็บในฐานข้อมูล หรือเขียนเป็น Pseudo Code เพื่อดู logic ในการเขียนโปรแกรม
 - 2.2 ความเข้าใจการพัฒนาระบบสารสนเทศประกอบด้วย การออกแบบระบบงาน วงจรพัฒนาระบบงาน (System Development life Cycle: SDLC) (เชิงลึก), การวิเคราะห์ออกแบบระบบงาน เช่น ต้องการออกแบบระบบ e-Commerce รองรับการรับลูกค้าหลายล้านคน
3. ด้านโครงสร้างสร้างพื้นฐาน (Hardware)
 - 3.1 ความเข้าใจเครือข่ายคอมพิวเตอร์ (Computer Network) โมเดล OSI, TCP/IP Protocol ต่าง ๆ เช่น HTTP, FTP, SMTP การออกแบบและบริหารจัดการเครือข่าย
 - 3.2 ความรู้ด้าน Cybersecurity, Cloud Computing operating system
4. ด้านเทคโนโลยีสมัยใหม่
 - 4.1 การจัดการข้อมูลขนาดใหญ่ (Big Data), แนวคิดการคิดค้นข้อมูล (Data Analytics), IoT, AI
 - 4.2 ความรู้เบื้องต้นเกี่ยวกับเทคโนโลยีคอมพิวเตอร์

สนใจสั่งซื้อ หรือสอบถามเพิ่มเติม โทร. 081-496-9907



LINE: @thebestcenter

299.-

คู่มือเตรียมสอบ
นักวิชาการคอมพิวเตอร์
การประสานครหลวง

ราคา 299.-

1. ความรู้พื้นฐานด้านคอมพิวเตอร์

1.1 ความเข้าใจการออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

1. การทำแบบจำลอง Entity-Relationship (ER) มีจุดประสงค์หลักเพื่ออะไร

- ก. เพื่อกำหนดชนิดข้อมูลและความยาวของแต่ละฟิลด์ในตารางอย่างละเอียดทั้งสำหรับระบบฐานข้อมูลต้นทางและปลายทาง
- ข. เพื่อใช้ในการสร้าง Query ประมวลผลข้อมูลที่ซับซ้อนที่สุดภายในระบบฐานข้อมูลเชิงสัมพันธ์
- ค. เพื่อกำหนดโครงสร้างเชิงตรรกะ (Logical Structure) ของข้อมูล เช่น Entity, Attribute และ Relationship ก่อนออกแบบฐานข้อมูลเชิงสัมพันธ์จริง
- ง. เพื่อออกแบบ Front-end User Interface ให้สามารถรองรับการกรอกข้อมูลในรูปแบบต่างๆ ได้ครบถ้วนตามความต้องการของผู้ใช้งาน

เฉลย: ค

คำอธิบาย: การทำแบบจำลอง ER มุ่งเน้นการกำหนดโครงสร้างเชิงตรรกะของข้อมูล ได้แก่ การระบุหน่วยข้อมูลหลัก (Entity) คุณสมบัตินี้ (Attribute) และความสัมพันธ์ (Relationship) ระหว่าง Entity ต่างๆ เพื่อให้เห็นภาพรวมก่อนนำไปสู่การสร้างตารางเชิงสัมพันธ์ (Relational Tables) ในขั้นตอนถัดไป จึงไม่ใช่การกำหนดรายละเอียดทางกายภาพ (Physical) เช่น ขนาดฟิลด์ หรือการสร้าง UI โดยตรง

2. ในกระบวนการแปลง ER Diagram ไปเป็น Relational Schema สิ่งใดต้องทำหลังจากแปลง Entity เป็น Table เสร็จสิ้น

- ก. กำหนด Constraint มาตรฐานข้อมูล เช่น CHECK, UNIQUE, และ DEFAULT ให้กับคอลัมน์ทั้งหมด
- ข. แปลงความสัมพันธ์แบบ Many-to-Many ระหว่าง Entity สองชุดให้เป็นตารางเชื่อม (Junction Table) พร้อมกำหนด Foreign Key
- ค. สร้าง Index แบบ Composite บนทุกตารางเพื่อให้ Query ทำงานเร็วขึ้นโดยอัตโนมัติ
- ง. รันคำสั่ง SQL สำหรับสร้างตารางจริงบน DBMS ทุกตารางทันที แล้วทดสอบการ Insert, Update, และ Delete

เฉลย: ข

คำอธิบาย: หลังจากสร้างตารางหลัก (Entity) แล้ว เราจำเป็นต้องแปลงความสัมพันธ์แบบ Many-to-Many ซึ่ง ER Diagram ไม่สามารถรองรับในเชิงฟิสิกส์ได้ จึงต้องสร้างตารางเชื่อม (Junction Table) ที่มีคอลลัมน์ Foreign Key ขึ้นไปยัง Primary Key ของทั้งสองตาราง พร้อม Primary Key ของ Junction Table เอง ซึ่งขั้นตอนการกำหนด Constraint อื่น ๆ หรือการสร้าง Index แบบ Composite สามารถทำได้ในภายหลังตาม Performance Tuning แต่หัวใจสำคัญคือการสร้างตารางเชื่อมก่อน

3. Functional Dependency (FD) ที่สัมพันธ์กับการพิจารณาสภาพปกติ (Normal Form) ขั้นที่ 2 (2NF)

ก. $A \rightarrow B$ และ $B \rightarrow C$

ข. $AB \rightarrow C$ และ C ไม่สามารถพึ่งพิงแยกกัน (partial dependency) บน A หรือ B เดียวๆ

ค. $A \rightarrow B$, $B \rightarrow C$ และ $C \rightarrow D$ โดยทุกค่า A, B, C, D มีความขึ้นต่อกัน

ง. $A \rightarrow B$ และ $C \rightarrow D$ โดย A, B, C, D มาจาก Entity คนละตารางกัน

เฉลย: ข

คำอธิบาย: เงื่อนไขของ 2NF คือ ตารางต้องอยู่ใน 1NF (Atomic data) และไม่มี Partial Dependency โดยหมายความว่า เมื่อ Primary Key ประกอบด้วยคอลัมน์มากกว่า 1 คอลัมน์ (Composite Key) จะต้องไม่มี Functional Dependency ที่คอลัมน์ย่อยใดย่อยหนึ่งของ Composite Key สามารถกำหนดค่าแยกกันไปยังคอลัมน์อื่น ดังนั้น $AB \rightarrow C$ ที่ C ไม่พึ่งพาแค่ A หรือแค่ B แต่พึ่งพา AB จึงสอดคล้องกับ 2NF

4. แต่ละตัวเลือกต่อไปนี้ เป็นข้อดีของการใช้ Database Normalization ยกเว้นข้อใด

ก. ลดความซ้ำซ้อนของข้อมูล (Data Redundancy)

ข. ป้องกันการเกิด Anomaly ในการ Insert, Update, Delete

ค. เพิ่มประสิทธิภาพการ Query ทุกประเภทให้เร็วขึ้นเสมอ

ง. ช่วยให้การจัดโครงสร้างข้อมูลมีความยืดหยุ่นต่อการขยายระบบในอนาคต

เฉลย: ค

คำอธิบาย: แม้การ Normalization จะช่วยลดความซ้ำซ้อนและป้องกัน Anomaly ได้ แต่ไม่ได้รับประกันว่าจะทำให้ทุก Query ทำงานเร็วขึ้น เพราะบางกรณีอาจต้อง join ตารางจำนวนมาก จึงอาจส่งผลให้การ Query บางประเภททำงานช้าลง

5. เมื่อออกแบบ Primary Key สำหรับตารางที่มีข้อมูลลูกค้าทั้งหมด 1 ล้านระเบียน ควรเลือกชนิดข้อมูลใด และเพราะเหตุใด

ก. ใช้คอลัมน์ Email (VARCHAR(255)) เพราะมีความเป็น Unique อยู่แล้ว ตรงกับ Primary Key ได้ทันที

ข. สร้างคอลัมน์ CustomerID (CHAR(36)) เก็บ UUID เพื่อความเป็น Unique และป้องกันการเดา ID

ค. สร้างคอลัมน์ CustomerID (INT AUTO_INCREMENT) เพราะขนาดเล็กและประมวลผล Index ได้เร็ว

ง. ใช้คอลัมน์ Composite Key ระหว่าง Email และวันที่ลงทะเบียน เพราะข้อมูลทั้งสองรวมกันมั่นใจว่าไม่ซ้ำ

เฉลย: ค

คำอธิบาย: INT AUTO_INCREMENT มีขนาดเล็ก (4 ไบต์) เวลาเก็บและดัชนี (Index) ทำงานได้เร็วกว่า VARCHAR หรือ CHAR ที่เก็บ UUID (16 หรือ 36 ไบต์) ขณะที่ Composite Key อาจเพิ่มความซับซ้อนและขนาด Index ใหญ่ขึ้น ซึ่งไม่เหมาะกับปริมาณข้อมูลมากๆ

6. ตัวอย่างใดต่อไปนี้เป็นข้อเสียของการใช้ Natural Key เป็น Primary Key

- ก. ลดขนาดของ Index เพราะใช้ข้อมูลที่มีอยู่แล้ว
- ข. มีโอกาสที่ค่าของข้อมูลจะเปลี่ยนแปลงในอนาคต ทำให้ต้อง Update ทุกที่ที่อ้างอิงถึง
- ค. ช่วยให้ Query สำหรับการรายงานตามธรรมชาติของข้อมูลทำได้ง่ายขึ้น
- ง. ไม่จำเป็นต้องสร้างคอลัมน์เพิ่มเติม ทำให้ลดขั้นตอนการออกแบบ

เฉลย: ข

คำอธิบาย: Natural Key คือคอลัมน์ที่มีความหมายทางธุรกิจ เช่น เลขบัตรประชาชน แม้จะลดขั้นตอนสร้างคอลัมน์ใหม่ แต่หากค่าธรรมชาติเปลี่ยนแปลง (เช่น การแก้ไขเลขประจำตัวหรือกรณีพิเศษ) จะเกิดปัญหา Referential Integrity ไหนๆ ก็ต้องปรับปรุง Foreign Key ทั่วทั้งระบบ จึงเป็นข้อเสียหลักของ Natural Key

7. Schema design แบบใดที่จะลดโอกาสการเกิด Update Anomaly ได้มากที่สุด

- ก. 1NF แล้วไม่ Normalize ต่อ
- ข. 2NF แต่ไม่ถึง 3NF
- ค. 3NF แต่ไม่ถึง BCNF
- ง. BCNF (Boyce-Codd Normal Form)

เฉลย: ง

คำอธิบาย: BCNF เป็นรูปแบบ Normal Form ที่เข้มงวดกว่ารูปแบบอื่น โดยทุก Functional Dependency $A \rightarrow B$ ต้องเป็นไปตามเงื่อนไขของ Primary Key ทำให้ลดโอกาสการเกิด Anomaly ในทุกกรณีมากที่สุด

8. Foreign Key Constraint มีความสำคัญอย่างไรในการออกแบบฐานข้อมูลเชิงสัมพันธ์

- ก. เป็นตัวกำหนดชนิดข้อมูลและความยาวของฟิลด์ภายในตาราง
- ข. ช่วยกำหนดความสัมพันธ์ระหว่างตารางและรักษาความสมบูรณ์ของข้อมูล (Referential Integrity)
- ค. ใช้ในการกำหนด Default Value ของคอลัมน์ที่สัมพันธ์กับตารางอื่น
- ง. ทำให้สามารถสร้าง View ที่ซับซ้อนโดยไม่ต้องเขียน SQL ใหม่

เฉลย: ข

คำอธิบาย: Foreign Key Constraint คือการกำหนดให้ค่าของฟิลด์ในตารางลูก (Child Table) ต้องตรงกับค่าของ Primary Key ในตารางแม่ (Parent Table) ช่วยป้องกันการกระทำอันไม่สอดคล้องกัน เช่น การลบหรืออัปเดตคีย์หลักที่ยังถูกร่องอ้างอิงอยู่ในตารางอื่น และรักษาความสมบูรณ์เชิงสัมพันธ์ของข้อมูล

9. ตัวอย่างสถานการณ์ใดควรใช้การ Denormalization

- ก. ระบบ OLTP ที่มี Transaksi ข้อมูลจำนวนมากครั้งต่อวัน ต้องการ Insert/Update รวดเร็วที่สุด
- ข. ระบบ OLAP สำหรับการวิเคราะห์รายงานธุรกิจ ที่เน้น Query อ่านข้อมูลจำนวนมากและซับซ้อน
- ค. ระบบเก็บ Log การเข้าชมเว็บไซต์ที่ไม่ต้องการวิเคราะห์แบบ Real-time
- ง. ระบบจัดเก็บเอกสารแบบ NoSQL เพื่อรองรับข้อมูลกึ่งโครงสร้าง

เฉลย: ข

คำอธิบาย: ระบบ OLAP เน้นการอ่านข้อมูลจำนวนมากและต้องการประสิทธิภาพ Query สูง การ Denormalization ช่วยลดการ Join ตารางหลายตาราง ทำให้ Query อ่านทำงานเร็วขึ้น แม้ว่าจะเพิ่มความซ้ำซ้อนของข้อมูลและต้องแลกกับ Storage และการ Update อาจซับซ้อนขึ้น

.....

10. Surrogate Key แตกต่างจาก Natural Key อย่างไร

- ก. Surrogate Key มีความหมายทางธุรกิจโดยตรง ขณะที่ Natural Key เป็นเพียงตัวเลขอัตโนมัติ
- ข. Surrogate Key มักสร้างขึ้นเป็น INT AUTO_INCREMENT ขณะที่ Natural Key ใช้ข้อมูลจริงจากธุรกิจ
- ค. Surrogate Key ไม่สามารถใช้เป็น Foreign Key ในตารางลูกได้ ขณะที่ Natural Key สามารถใช้ได้เสมอ
- ง. Surrogate Key ไม่สามารถเปลี่ยนแปลงได้ ขณะที่ Natural Key เปลี่ยนแปลงบ่อย

เฉลย: ข

คำอธิบาย: Surrogate Key คือคีย์เทียมที่สร้างขึ้นใหม่ เช่น INT AUTO_INCREMENT หรือ GUID ไม่มีความหมายทางธุรกิจ ขณะที่ Natural Key เป็นคีย์ที่มีความหมายในบริบทของข้อมูล เช่น รหัสสินค้า เลขบัตรประชาชน ซึ่งอาจเปลี่ยนแปลงหรือมีข้อผิดพลาดได้

.....

11. การเลือก Data Type ที่เหมาะสมมีผลต่อเรื่องใดมากที่สุด

- ก. การรักษาความสมบูรณ์ของข้อมูล (Data Integrity)
- ข. ขนาดพื้นที่จัดเก็บ (Storage Space) และประสิทธิภาพ Index
- ค. การจัดการการเชื่อมต่อ (Concurrency Control)
- ง. รูปแบบการแสดงผลบนหน้าเว็บ (Presentation Layer)

เฉลย: ข

คำอธิบาย: การเลือก Data Type ที่เหมาะสมส่งผลต่อขนาดพื้นที่จัดเก็บข้อมูลและประสิทธิภาพการสร้างดัชนี (Index) ช่วยให้ Query ทำงานได้รวดเร็วและลดพื้นที่จัดเก็บโดยรวม ในขณะที่การรักษาความสมบูรณ์ข้อมูลควบคุมโดย Constraint และการออกแบบ Schema อื่น ๆ

.....

12. ER Diagram ต่อไปนี้ ถ้า Cardinality ระหว่าง Customer และ Order เป็น 1:N ควรออกแบบใน Relational Schema อย่างไร

- ก. สร้างตาราง Customer_Order ที่เก็บทั้ง Customer และ Order ไว้ร่วมกันเพื่อความเร็ว
- ข. เพิ่มคอลัมน์ CustomerID เป็น Foreign Key ในตาราง Order
- ค. เพิ่มคอลัมน์ OrderID เป็น Foreign Key ในตาราง Customer
- ง. สร้าง Junction Table ระหว่าง Customer และ Order พร้อม Primary Key ประกอบ (CustomerID, OrderID)

เฉลย: ข

คำอธิบาย: เมื่อความสัมพันธ์เป็น 1:N ระหว่าง Customer (1) และ Order (N) ให้เก็บ Foreign Key ของด้าน 1 (CustomerID) ไปไว้ในตารางด้าน N (Order) เพื่อเชื่อมโยงคำสั่งซื้อแต่ละรายการกับลูกค้าหนึ่งราย

13. ข้อใดอธิบายถึง Multi-valued Attribute ใน ER Model ได้ดีที่สุด

- ก. คอลัมน์ในตารางที่เก็บค่าตัวเลขลอยตัว และสามารถเก็บค่าสูงสุดได้หลายหลัก
- ข. คุณสมบัติของ Entity ที่อาจมีค่ามากกว่า 1 ค่า เช่น เบอร์โทรศัพท์หลายหมายเลขของลูกค้า
- ค. คุณสมบัติที่ต้องเก็บค่าสองค่าควบคู่กันเสมอ เช่น กรณีนับทีกเวลาเริ่มต้นและเวลาสิ้นสุดของเหตุการณ์
- ง. คุณสมบัติที่เปลี่ยนแปลงได้ตามเวลาปัจจุบัน เช่น สถานะ Availability ของสินค้า

เฉลย: ข

คำอธิบาย: Multi-valued Attribute คือคุณสมบัติที่สามารถมีหลายค่าได้ในหนึ่ง Entity เช่น ลูกค้าอาจมีเบอร์โทรศัพท์ 2-3 หมายเลข ER Model จึงไม่อนุญาตให้เก็บในคอลัมน์เดียว แต่ต้องแยกออกเป็น Entity ย่อยหรือสร้างตารางแยกเพื่อรองรับ

14. การออกแบบ Index สำหรับคอลัมน์ที่ใช้ในการ Join ข้ามตาราง มีประโยชน์อย่างไร

- ก. ช่วยลดการใช้ Disk Space เพราะ Index จะเก็บข้อมูลที่จำเป็นเท่านั้น
- ข. ทำให้การ Join ระหว่างตารางทำงานเร็วขึ้นเพราะสามารถค้นหา Foreign Key ได้โดยตรง
- ค. ป้องกันการเกิด Deadlock ในการ Execute Transaction ระหว่างตาราง
- ง. ช่วยจัดเรียงผลลัพธ์ของ Query ให้อยู่ในลำดับที่ต้องการทันทีโดยไม่ต้องใช้ ORDER BY

เฉลย: ข

คำอธิบาย: Index บนคอลัมน์ที่ใช้เป็น Foreign Key หรือ Join Key จะช่วยให้การค้นหาแถวที่ตรงตามเงื่อนไข JOIN ทำได้รวดเร็วขึ้น เนื่องจากไม่ต้องสแกนทั้งตาราง (Full Table Scan)

15. When designing a large relational database, partitioning tables by range, list, or hash can help solve which problem most directly?

- ก. ปัญหาการทำงานของ Query ช้าเนื่องจากการ Join ตารางเล็กหลายตาราง
- ข. การจัดการข้อมูลแบบทวิคูณในแอปพลิเคชัน NoSQL
- ค. การจัดเก็บและประมวลผลข้อมูลปริมาณมาก (Big Data) ที่อาจเกินขนาดตารางหนึ่ง
- ง. ปัญหาเรื่อง Referential Integrity ระหว่างชาร์ดข้อมูลในหลายโหนด

เฉลย: ค

คำอธิบาย: Partitioning ช่วยแบ่งตารางขนาดใหญ่ให้ออกเป็นส่วนย่อยตาม Range, List, หรือ Hash ทำให้การสืบค้นและบำรุงรักษาข้อมูลปริมาณมากง่ายขึ้น ลดเวลา Full Scan และลด I/O บนโหนดเดียว แต่ไม่ได้แก้ปัญหาด้าน Referential Integrity โดยตรง

16. Migration Strategy แบบใดเหมาะกับการเปลี่ยนระบบฐานข้อมูลโดยไม่หยุดงาน (Zero Downtime)

- ก. Big Bang Migration คือย้ายตารางทั้งหมดพร้อมกันในช่วงเวลาสั้นๆ
- ข. Phased Migration คือย้ายเป็นส่วน ๆ ไปใช้งานทีละโมดูล พร้อม Sync ข้อมูลสองระบบชั่วคราว
- ค. Manual Data Export/Import ทุกตารางผ่านเครื่องมือ ETL
- ง. การใช้ DB Link เพื่ออ่านเขียนข้อมูลพร้อมกันทั้งสองระบบโดยไม่มีการ Sync

เฉลย: ข

คำอธิบาย: Phased Migration ย้ายทีละส่วน (เช่น ตาราง หรือโมดูล) พร้อมการ Sync ข้อมูลระหว่างระบบเก่าและใหม่ ทำให้ลด Downtime และลดความเสี่ยงกว่า Big Bang Migration ที่ย้ายทั้งหมดพร้อมกันในครั้งเดียว

.....

17. ข้อใดกล่าวถึงข้อดีของการใช้ Entity-Attribute-Value (EAV) Model ได้ถูกต้องที่สุด

- ก. ทำให้รองรับ Attribute ใหม่ ๆ ได้ทันทีโดยไม่ต้องปรับโครงสร้างตาราง
- ข. ช่วยให้ Query มีประสิทธิภาพสูงโดยไม่ต้อง Join ตารางจำนวนมาก
- ค. ลดความซ้ำซ้อนของข้อมูล (Data Redundancy) ในทุกกรณี
- ง. ป้องกัน Anomaly ด้าน Update และ Delete ในทุกสถานการณ์

เฉลย: ก

คำอธิบาย: EAV Model เก็บ Attribute แบบไดนามิกในตารางเดียว ทำให้เพิ่มหรือเปลี่ยน Attribute ใหม่ ๆ ได้ทันทีโดยไม่ต้อง ALTER TABLE แต่มีข้อเสียเช่น Query ซับซ้อน ประสิทธิภาพต่ำ และการรักษาความสมบูรณ์ข้อมูลยาก

.....

18. In a fact table of a data warehouse, surrogate keys are often used instead of natural keys because:

- ก. Surrogate keys ทำให้การ Join กับ dimension table ช้าลง และลดการใช้ Index
- ข. Surrogate keys สร้างขึ้นในฐานข้อมูลปลายทาง ทำให้ไม่ขึ้นกับระบบต้นทางและช่วยจัดการ Slowly Changing Dimensions ได้ง่ายขึ้น
- ค. Surrogate keys ย่อมาจากคีย์ที่มีความหมายในทางธุรกิจ ช่วยในการวิเคราะห์ได้ตรงประเด็น
- ง. Surrogate keys ป้องกันการเกิดเชิงสัมพันธ์แบบ Circular Reference between tables

เฉลย: ข

คำอธิบาย: Surrogate Key ใน Data Warehouse มักเป็น Primary Key ที่สร้างขึ้นภายในเพื่อไม่พึ่งพาระบบต้นทาง ทำให้ควบคุม Slowly Changing Dimensions และประวัติการเปลี่ยนแปลงข้อมูลได้ง่าย และลดปัญหาการปรับโครงสร้างเมื่อระบบต้นทางเปลี่ยนแปลง

.....

19. การทดสอบ Schema Design สามารถทำได้ด้วยวิธีใดต่อไปนี้

- ก. การรัน SQL EXPLAIN PLAN บนทุก Query ที่คาดว่าจะใช้งานจริง
- ข. การ Insert ข้อมูลทดสอบ 100 ระเบียบ ดูว่าทำงานครบทุก Use Case
- ค. การเขียน Stored Procedure เพื่อตรวจสอบ Referential Integrity อัตโนมัติ
- ง. การใช้ Data Profiling Tool เพื่อวิเคราะห์ความสอดคล้องของข้อมูลกับ Schema ที่ออกแบบ

เฉลย: ง

คำอธิบาย: Data Profiling Tool ช่วยตรวจสอบโครงสร้างข้อมูล วิเคราะห์คุณภาพ (Data Quality) และความสอดคล้องระหว่างข้อมูลที่มีจริงกับ Schema ที่ออกแบบ ช่วยให้พบข้อผิดพลาดก่อนการใช้งานจริง ขณะที่ EXPLAIN PLAN ช่วยเรื่องประสิทธิภาพ แต่ไม่ครอบคลุมการทดสอบภาพรวม

20. Entity ใดควรถูกทำเป็น Weak Entity (Dependent Entity) ในการออกแบบ ER Model

- ก. InvoiceHeader ซึ่งมีข้อมูลหัวเอกสารใบแจ้งหนี้
- ข. LineItem ซึ่งต้องอ้างอิง InvoiceHeader เพราะไม่มีความหมายตัวเองหากไม่อ้างอิงถึง Invoice
- ค. Customer ซึ่งมีตัวตนอิสระและสามารถเก็บข้อมูลได้โดยไม่ต้องอ้างอิงอื่น
- ง. ProductCategory ซึ่งจัดหมวดหมู่สินค้าและมีตัวตนเป็นอิสระ

เฉลย: ข

คำอธิบาย: Weak Entity คือ Entity ที่ไม่มี Primary Key ของตัวเองและจำเป็นต้องอิงกับ Owner Entity เพื่อความหมาย เช่น LineItem ต้องอ้างอิง InvoiceHeader จึงถูกออกแบบให้เป็น Weak Entity และมี Primary Key ประกอบด้วย Foreign Key ของ InvoiceHeader

21. การประยุกต์ใช้ Constraint CHECK จะเหมาะกับกรณีใด

- ก. ตรวจสอบความซ้ำซ้อนของข้อมูลในคอลัมน์เดียวกัน เช่น รหัสลูกค้าต้องไม่ซ้ำในตารางเดียวกัน
- ข. ตรวจสอบค่าของคอลัมน์ที่ต้องอยู่ในช่วงที่กำหนด เช่น อายุของพนักงานต้อง ≥ 18 และ ≤ 65
- ค. กำหนดให้ค่าฟิลด์ที่สัมพันธ์กับ Foreign Key ต้องมีค่าในตารางแม่เท่านั้น
- ง. กำหนด Default Value ของคอลัมน์วันที่สร้างเอกสารให้เป็น CURRENT_TIMESTAMP

เฉลย: ข

คำอธิบาย: CHECK Constraint ใช้ตรวจสอบเงื่อนไขบนค่าฟิลด์ เช่น ช่วงตัวเลข ค่าที่อนุญาต หรือรูปแบบข้อความ ไม่เหมือน UNIQUE (ป้องกันซ้ำ) หรือ Foreign Key (ตรวจสอบความสัมพันธ์)

22. ในขั้นตอนการออกแบบ Physical Schema ควรพิจารณาประเด็นใดบ้าง

- ก. ประเภทของ Storage Engine, การจัด Partition, และการกำหนด Tablespace
- ข. รูปแบบการแสดงผลบน UI และการเชื่อมต่อผ่าน API
- ค. การกำหนดชื่อฟิลด์ให้สั้นลงเพื่อให้สวยงาม
- ง. การตั้งชื่อ User และ Password สำหรับแต่ละตาราง

เฉลย: ก

คำอธิบาย: การออกแบบ Physical Schema เกี่ยวกับการเลือก Storage Engine (เช่น InnoDB, MyISAM), การจัด Partition ตารางเพื่อลด I/O, การกำหนด Tablespace และ File Group เพื่อจัดการพื้นที่จัดเก็บและประสิทธิภาพของ DBMS

.....

23. Granularity ของ Data Warehouse Fact Table หมายถึงอะไร

- ก. จำนวนของฟิลด์ในแต่ละบันทึกของ Fact Table
- ข. ระดับความละเอียดของเมตริกซ์ที่เก็บใน Fact Table เช่น รายวัน รายเดือน หรือรายการเดียว
- ค. ขนาดของไฟล์ Data Warehouse ในแต่ละ Partition
- ง. จำนวน Dimension ที่เชื่อมโยงกับ Fact Table

เฉลย: ข

คำอธิบาย: Granularity คือระดับความละเอียดของบันทึกใน Fact Table ที่จะเก็บข้อมูลในระดับใด เช่น รายวัน รายเดือน หรือระดับ Transaction เดียวๆ โดยมีผลต่อขนาดและรายละเอียดของชุดข้อมูลที่วิเคราะห์

.....

24. Snowflake Schema แตกต่างจาก Star Schema ตรงไหน

- ก. Snowflake Schema จัดเก็บ Dimension เป็นตารางเดียวแบบ denormalized ขณะที่ Star Schema แยกตาราง Dimension ย่อยๆ
- ข. Star Schema ต้องใช้ Surrogate Key เท่านั้น ขณะที่ Snowflake ใช้ Natural Key ได้เสมอ
- ค. Snowflake Schema นำ Dimension มา Normalize หลายชั้น ขณะที่ Star Schema นำ Dimension มา Denormalize เป็นตารางเดียว
- ง. Star Schema ไม่สามารถรองรับ Slowly Changing Dimension ได้ ในขณะที่ Snowflake รองรับ

เฉลย: ค

คำอธิบาย: Snowflake Schema จะ Normalize Dimension Tables ให้แยกเป็นตารางย่อยหลายระดับ ส่วน Star Schema จะ Denormalize Dimension เป็นตารางเดียวทำให้ Query Join ยากขึ้นแต่เพิ่ม Data Redundancy

.....

25. Entity-Relationship Diagram ต่อไปนี้ เมื่อแปลงเป็น Relational Schema แล้ว การกำหนด PRIMARY Key ของตาราง CustomerOrderDetail ควรเป็นอย่างไร

- ก. PRIMARY KEY(OrderDetailID) ซึ่งเป็น Surrogate Key แบบ AUTO_INCREMENT
- ข. PRIMARY KEY(CustomerID, OrderID, ProductID) ซึ่งเป็น Composite Key
- ค. PRIMARY KEY(OrderID, ProductID) เนื่องจากแต่ละคำสั่งซื้อจะมีสินค้าลำดับเดียว
- ง. PRIMARY KEY(CustomerID, ProductID) เนื่องจากลูกค้าหนึ่งคนไม่สามารถสั่งซื้อสินค้าเดิมซ้ำได้

เฉลย: ข

คำอธิบาย: ตาราง CustomerOrderDetail จัดเก็บรายละเอียดสินค้าในแต่ละ Order จึงต้องระบุ Composite Key ได้แก่ CustomerID (FK), OrderID (FK) และ ProductID (FK) เพื่อให้แน่ใจว่าแต่ละบรรทัดรายละเอียดไม่ซ้ำกัน

.....

26. In a highly concurrent OLTP system, what lock granularity is generally preferred to maximize throughput?

- ก. Table-level locks to ensure consistency across entire tables
- ข. Page-level locks to balance between concurrency and overhead
- ค. Row-level locks to allow maximum concurrent transactions on different rows
- ง. Database-level locks to simplify transaction management

เฉลย: ค

คำอธิบาย: Row-level locks จะล็อกเฉพาะแถวที่กำลังถูกแก้ไข ช่วยให้ transaction อื่นที่ทำงานบนแถวต่างๆ สามารถทำงานพร้อมกันได้มากที่สุด เพิ่ม Throughput ในระบบที่มี Concurrent สูง

.....

27. การใช้ View ช่วยในด้านใดบ้าง ยกเว้นข้อใด

- ก. แสดงผลข้อมูลเฉพาะบางฟิลด์หรือกรองเฉพาะบาง Record ให้ผู้ใช้งาน
- ข. ปิดบังความซับซ้อนของการ Join หลายตารางจากผู้ใช้
- ค. ช่วยเพิ่มประสิทธิภาพการ Modify ข้อมูล (INSERT/UPDATE/DELETE) ให้เร็วขึ้นทุกกรณี
- ง. ใช้เป็น Layer Security สำหรับกำหนดสิทธิ์การเข้าถึงข้อมูล

เฉลย: ค

คำอธิบาย: View ช่วยในเรื่องการกรองข้อมูล ปิดบังความซับซ้อน และใช้ควบคุมสิทธิ์ข้อมูล แต่ไม่รับประกันว่าจะทำให้การ Modify ข้อมูลเร็วขึ้น เพราะบางระบบอาจต้องสร้าง View แบบ Materialized View เพื่อให้ Query Write มีประสิทธิภาพ

.....

28. ข้อใดอธิบาย Third Normal Form (3NF) ได้ถูกต้องที่สุด

- ก. เป็นแบบฟอร์มที่กำหนดให้ทุกคอลัมน์ต้องพึ่งพิงโดยตรงกับ Primary Key เท่านั้นและไม่มี Transitive Dependency
- ข. ตารางต้องไม่มี Multi-valued Attribute และทุก Attribute ต้องเป็น Atomic
- ค. ตารางต้องไม่มี Partial Dependency แต่ยังมี Transitive Dependency ได้
- ง. เป็นแบบฟอร์มที่กำหนดให้ต้องใช้ Surrogate Key เท่านั้น

เฉลย: ก

คำอธิบาย: 3NF คืออยู่ใน 2NF และไม่มี Transitive Dependency ($A \rightarrow B$ และ $B \rightarrow C$) ใดๆ ทำให้ทุก Attribute (C) พึ่งพิงกับ Primary Key ตรงๆ ไม่ผ่าน Attribute อื่น

.....

29. การใช้ Materialized View ในระบบ Data Warehouse ช่วยอะไรได้บ้าง

- ก. เก็บเฉพาะ Definition ของ Query โดยไม่เก็บข้อมูลจริง
- ข. เก็บผลลัพธ์ของ View ช่วยให้ Query อ่านข้อมูลชุดเดิมเร็วขึ้น

- ค. บังคับให้การ Insert/Update ในตารางต้นทางเกิดขึ้นบน View แทนโดยตรง
- ง. ทำให้สามารถเขียน Transaction ที่ซับซ้อนโดยไม่ต้องระบุ SQL ใหม่

เฉลย: ข

คำอธิบาย: Materialized View เก็บผลลัพธ์ของ Query จริงลงในตาราง ทำให้สามารถอ่านข้อมูลเดิมได้เร็วขึ้น ลดการประมวลผลซ้ำ แต่ต้องมีการ Refresh เพื่อนำข้อมูลจากตารางต้นทางมาทดแทนเมื่อข้อมูลเปลี่ยนแปลง

30. What is the primary benefit of using normalization up to BCNF in OLTP systems?

- ก. เพิ่มความซับซ้อนของข้อมูลเพื่อลดจำนวน Join ใน Query
- ข. ลดข้อมูลซ้ำซ้อนและป้องกันทุกประเภทของ Anomaly ทำให้การ Insert/Update/Delete มีความถูกต้องสูง
- ค. ทำให้ทุก Query ในระบบทำงานเร็วขึ้น โดยไม่ต้องออกแบบ Index ใดๆ เพิ่มเติม
- ง. ช่วยให้ Data Warehouse สามารถนำข้อมูลไปใช้งานได้ง่ายขึ้น

เฉลย: ข

คำอธิบาย: การ Normalize ถึง BCNF จะช่วยลด Data Redundancy และป้องกันทุกประเภทของ Anomaly ในระบบ OLTP ทำให้การจัดการข้อมูล Insert/Update/Delete มีความถูกต้องและมั่นใจได้ว่าข้อมูลจะไม่เกิดปัญหา Inconsistency

31. ข้อใดต่อไปนี้กล่าวถึงความหมายของแบบจำลองเชิงสัมพันธ์ (Relational Model) ได้ถูกต้องและครอบคลุมมากที่สุด

- ก. แบบจำลองเชิงสัมพันธ์คือรูปแบบการจัดเก็บข้อมูลที่แต่ละหน่วยข้อมูลเรียกว่าเรคคอร์ด และมีโครงสร้างเป็นแฟ้มข้อมูลที่ไม่จำเป็นต้องเชื่อมโยงกัน
- ข. แบบจำลองเชิงสัมพันธ์คือการจัดข้อมูลให้อยู่ในรูปแบบตารางที่สามารถจัดการด้วยคำสั่ง SQL ได้และมีคีย์หลักที่ไม่จำเป็นต้องไม่ซ้ำกันเสมอ
- ค. แบบจำลองเชิงสัมพันธ์คือระบบที่จัดเก็บข้อมูลในรูปแบบของตาราง (Relations) โดยแต่ละตารางมีแถว (Tuples) และคอลัมน์ (Attributes) ซึ่งสามารถเชื่อมโยงกันด้วยคีย์ต่าง ๆ
- ง. แบบจำลองเชิงสัมพันธ์คือการจำลองโครงสร้างข้อมูลตามโครงสร้างแฟ้มแบบลำดับชั้น และใช้ความสัมพันธ์แบบแม่-ลูกในการจัดการข้อมูล

เฉลย: ค

อธิบาย: แบบจำลองเชิงสัมพันธ์ (Relational Model) เป็นการจัดโครงสร้างฐานข้อมูลให้อยู่ในรูปของตาราง (Relation) ซึ่งแต่ละตารางประกอบด้วยแถว (Tuple) และคอลัมน์ (Attribute) โดยใช้คีย์หลัก (Primary Key) และคีย์ต่างประเทศ (Foreign Key) เพื่อสร้างความสัมพันธ์ระหว่างตาราง ช่วยให้ข้อมูลมีความสอดคล้อง (Consistency) และง่ายต่อการบริหารจัดการ

32. หากนักออกแบบฐานข้อมูลต้องการกำหนดให้ข้อมูลในตารางไม่มีความซ้ำซ้อน (Redundancy) และสามารถลดความผิดพลาดจากการแทรก ลบ หรือปรับปรุงข้อมูล (Anomaly) ได้ ควรใช้แนวคิดใด
- ก. การทำ Indexing แบบ Full-Text เพื่อให้ค้นหาข้อมูลจากหลายฟิลด์ได้ง่ายขึ้นและลดเวลาประมวลผลลง
 - ข. การออกแบบฐานข้อมูลโดยใช้การทำ Clustering และ Replication เพื่อเพิ่มความเร็วในการเข้าถึงข้อมูลในหลาย Site
 - ค. การออกแบบฐานข้อมูลโดยใช้กฎของนอร์มัลไลเซชัน (Normalization) เพื่อแบ่งข้อมูลออกเป็นตารางย่อยตามระดับของความสัมพันธ์
 - ง. การสร้าง Entity-Relationship Diagram (ERD) ให้มีรายละเอียดมากที่สุด โดยใช้คีย์หลักซ้อนกันหลายระดับของ Subtype Entity

เฉลย: ค

อธิบาย: การใช้หลักการ นอร์มัลไลเซชัน (Normalization) เป็นกระบวนการหลักในการออกแบบฐานข้อมูลเชิงสัมพันธ์เพื่อลดความซ้ำซ้อนของข้อมูล ป้องกันความผิดพลาดจากการแทรก ลบ หรือปรับปรุงข้อมูล โดยการแยกข้อมูลออกเป็นหลายตารางที่มีความสัมพันธ์กันอย่างเหมาะสมและกำหนดคีย์ต่าง ๆ ให้ถูกต้อง

-
33. ข้อใดแสดงถึงลักษณะสำคัญของคีย์หลัก (Primary Key) ที่ดีในฐานข้อมูลเชิงสัมพันธ์
- ก. ควรเป็นข้อมูลที่สามารถเปลี่ยนแปลงได้บ่อยเพื่อความยืดหยุ่นและความสามารถในการสืบค้น
 - ข. ควรเป็นข้อมูลที่มีขนาดใหญ่และซับซ้อนเพื่อป้องกันไม่ให้มีค่าซ้ำในระบบ
 - ค. ควรเป็นแอตทริบิวต์ที่มีค่าซ้ำได้บางกรณี เพื่อรองรับการออกแบบที่ซับซ้อนของข้อมูล
 - ง. ควรเป็นแอตทริบิวต์ที่มีค่าที่ไม่ซ้ำกันในแต่ละเรคคอร์ด และไม่ควรเปลี่ยนแปลงได้ง่าย เพื่อให้สามารถระบุแต่ละแถวได้อย่างชัดเจน

เฉลย: ง

อธิบาย: คีย์หลัก (Primary Key) คือแอตทริบิวต์หรือชุดของแอตทริบิวต์ที่ใช้ระบุเอกลักษณ์หรือแถวข้อมูลในตารางได้อย่างไม่ซ้ำกัน และต้องไม่เป็นค่าว่าง (NULL) รวมถึงไม่ควรเปลี่ยนแปลง เพราะจะส่งผลกระทบต่ออ้างอิงในตารางอื่นที่ใช้คีย์นี้เป็น Foreign Key

-
34. การใช้คีย์ต่างประเทศ (Foreign Key) มีวัตถุประสงค์หลักเพื่ออะไรในระบบฐานข้อมูลเชิงสัมพันธ์
- ก. เพื่อใช้ในการสร้างตารางชั่วคราวเพื่อจัดการข้อมูลในระดับแคช
 - ข. เพื่อเชื่อมโยงตารางต่าง ๆ เข้าด้วยกันผ่านความสัมพันธ์ และควบคุมความสมบูรณ์ของข้อมูลระหว่างตาราง
 - ค. เพื่อจัดการกับข้อมูลในระดับฟิลด์แบบไดนามิกโดยไม่ต้องกำหนดโครงสร้างแน่นอน
 - ง. เพื่อแบ่งข้อมูลออกเป็นกลุ่ม ๆ โดยใช้กลไกเชิงฟังก์ชัน (Functional Partitioning)

เฉลย: ข

อธิบาย: Foreign Key คือคีย์ที่ใช้เพื่อสร้างความสัมพันธ์ระหว่างตาราง โดยอ้างอิงถึง Primary Key ของอีกตารางหนึ่ง เพื่อควบคุมความถูกต้องของข้อมูล (Referential Integrity) เช่น หากมีการลบข้อมูลในตารางแม่ ควรกำหนดว่าจะให้เกิดอะไรขึ้นกับข้อมูลในตารางลูก (เช่น Cascade Delete หรือ Restrict)

.....

35. ในกระบวนการทำ Normalization จาก First Normal Form (1NF) ไปยัง Third Normal Form (3NF) ต้องผ่านเงื่อนไขใดบ้าง

- ก. ต้องแยกข้อมูลให้อยู่ในรูปแบบทรีดี (3D Table) โดยตัดข้อมูลที่ซ้ำซ้อนทิ้งไปทั้งหมด
- ข. ต้องตรวจสอบให้แน่ใจว่าไม่มีการขึ้นต่อกันแบบฟังก์ชันระหว่าง Non-Key Attribute กับ Candidate Key และไม่มีการขึ้นต่อกันแบบทรานซิทีฟ
- ค. ต้องตรวจสอบให้มีการกำหนด Foreign Key ที่สัมพันธ์กับตารางแม่ทุกตารางที่เกี่ยวข้องกับความสัมพันธ์
- ง. ต้องสร้างตารางใหม่สำหรับทุกกลุ่มข้อมูลที่มีความซ้ำซ้อนโดยไม่ต้องคำนึงถึงความสัมพันธ์ระหว่างแอตทริบิวต์

เฉลย: ข

อธิบาย: การทำ Normalization จนถึง 3NF ต้องตรวจสอบว่า (1) อยู่ใน 1NF คือ ไม่มีข้อมูลซ้ำซ้อนภายในฟิลด์, (2) 2NF คือ ไม่มี Partial Dependency กับ Primary Key และ (3) 3NF คือ ไม่มี Transitive Dependency หมายความว่า Non-Key Attribute ต้องขึ้นกับ Candidate Key โดยตรงเท่านั้น

.....

36. ข้อใดคือข้อดีที่สำคัญของการออกแบบฐานข้อมูลผ่านกระบวนการนอร์มัลไลเซชัน (Normalization) อย่างครบถ้วนจนถึงระดับ 3NF หรือสูงกว่า

- ก. ทำให้สามารถรวมข้อมูลจากหลายระบบได้ง่ายโดยไม่ต้องมีการแปลงโครงสร้างมากนัก
- ข. ช่วยให้สามารถคำนวณหาค่าเฉลี่ยและค่ารวมของข้อมูลได้แม่นยำยิ่งขึ้นในระดับแถว
- ค. ลดการซ้ำซ้อนของข้อมูล, ป้องกัน Anomaly ต่าง ๆ และเพิ่มความสอดคล้องของข้อมูลโดยรวมในระบบ
- ง. ทำให้สามารถใช้งานร่วมกับ NoSQL Database ได้ทันทีโดยไม่ต้องปรับโครงสร้างหรือออกแบบซ้ำ

เฉลย: ค

อธิบาย: ข้อดีของการนอร์มัลไลซ์ข้อมูลคือสามารถ ลดความซ้ำซ้อน (Redundancy) ได้อย่างมีประสิทธิภาพ, ป้องกันปัญหา Insertion, Update, และ Deletion Anomalies และเพิ่มความ Consistency ของข้อมูลโดยรวมในระบบที่มีการอ้างอิงข้ามตารางอย่างถูกต้อง

.....

37. สมมติว่าตาราง OrderDetails มีคอลัมน์ OrderID, ProductID, ProductName, UnitPrice, Quantity, และ Total. หากต้องการลด Redundancy ให้เหลือน้อยที่สุด ควรดำเนินการอย่างไร

- ก. รวมข้อมูลทุกตารางไว้ในตารางเดียวเพื่อความสะดวกในการ Query และลดการใช้ JOIN
- ข. แยกตาราง Product ออกจาก OrderDetails โดยใช้ ProductID เป็น Foreign Key และย้าย ProductName, UnitPrice ไปยังตาราง Product
- ค. คงตารางไว้เช่นเดิมเพราะโครงสร้างนี้สามารถ Query ได้ง่ายและไม่ต้อง JOIN หลายตาราง
- ง. ย้ายคอลัมน์ Total ไปยังตาราง Product เพื่อให้สามารถดึงข้อมูลได้โดยตรงโดยไม่ต้องคำนวณซ้ำ

เฉลย: ข

อธิบาย: การที่ตาราง OrderDetails มี ProductName และ UnitPrice แสดงว่ามีข้อมูลซ้ำซ้อน เพราะข้อมูลของ

สินค้าไม่ควรอยู่ในตารางรายละเอียดคำสั่งซื้อ การ แยกข้อมูลสินค้าส่วนนี้ไปไว้ในตาราง Product แล้วอ้างอิงผ่าน ProductID เป็น Foreign Key จะทำให้ฐานข้อมูล อยู่ในรูปแบบ Normalize มากขึ้น ลด Redundancy และ Update Anomaly

.....

38. ในการออกแบบฐานข้อมูลเชิงสัมพันธ์ที่ดี สิ่งใดต่อไปนี้ไม่ใช่เป้าหมายหลัก

- ก. การเก็บข้อมูลที่มีความสัมพันธ์กันในรูปแบบที่ลดความซ้ำซ้อนของข้อมูลลงให้มากที่สุด
- ข. การทำให้การจัดการข้อมูลผ่านระบบสามารถทำได้มีประสิทธิภาพและประสิทธิผล
- ค. การรวมข้อมูลทุกประเภทไว้ในตารางเดียวเพื่อความสะดวกในการจัดเก็บและลดจำนวนตาราง
- ง. การออกแบบให้สามารถปรับขยาย (Scalable) ได้ง่ายเมื่อข้อมูลหรือผู้ใช้งานเพิ่มขึ้น

เฉลย: ค

อธิบาย: การรวมข้อมูลไว้ในตารางเดียวเป็นสิ่งที่ ขัดกับหลักการของ Relational Design เพราะจะทำให้เกิด ข้อมูลซ้ำซ้อน, ยากต่อการดูแลรักษา, และ เสี่ยงต่อการเกิด Anomaly ดังนั้นจึงไม่ใช่เป้าหมายของการออกแบบฐานข้อมูลที่ดี

.....

39. ถ้านักออกแบบฐานข้อมูลพบว่าในตารางมีแอตทริบิวต์ B ที่ขึ้นกับ A และ C ขึ้นกับ B แต่ไม่ขึ้นกับ A จะแสดงถึงการละเมิดเงื่อนไขของ Normal Form ไດ

- ก. ละเมิด First Normal Form เพราะข้อมูลมีโครงสร้างซ้ำซ้อนกัน
- ข. ละเมิด Second Normal Form เพราะมีการขึ้นต่อแบบบางส่วน (Partial Dependency)
- ค. ละเมิด Third Normal Form เพราะมีการขึ้นต่อแบบทรานซิทีฟ (Transitive Dependency)
- ง. ละเมิด Boyce-Codd Normal Form เพราะมี Functional Dependency ซ้ำซ้อนระหว่าง Candidate Keys

เฉลย: ค

อธิบาย: ถ้า C ขึ้นกับ B แต่ไม่ขึ้นกับ A (ซึ่ง A เป็น Primary Key) แสดงว่ามี Transitive Dependency อยู่ในตาราง ซึ่งขัดกับเงื่อนไขของ Third Normal Form (3NF) เพราะใน 3NF ต้องไม่มีการขึ้นต่อของ Non-Key Attribute ที่ขึ้นกับ Non-Key Attribute ตัวอื่น

.....

40. คำสั่ง SQL ใดด้านล่างนี้สามารถใช้สร้างความสัมพันธ์ระหว่างตาราง Orders กับ Customers ได้อย่างเหมาะสม หากใช้ CustomerID เป็น Foreign Key

- ก. CREATE FOREIGN KEY ON Orders REFERENCES Customers(CustomerID);
- ข. ALTER TABLE Orders ADD CONSTRAINT fk_customer FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);
- ค. UPDATE TABLE Orders SET FOREIGN KEY = Customers.CustomerID;
- ง. LINK TABLE Orders TO Customers USING CustomerID AS RELATION;

เฉลย: ข

อธิบาย: การเพิ่ม Foreign Key ใน SQL จะใช้คำสั่ง ALTER TABLE ... ADD CONSTRAINT ... FOREIGN KEY (...) REFERENCES ... เช่น ALTER TABLE Orders ADD CONSTRAINT fk_customer FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID); ซึ่งจะเชื่อมตาราง Orders กับ Customers ผ่าน CustomerID อย่างมีประสิทธิภาพ และรักษา Referential Integrity

41. ในการออกแบบฐานข้อมูลเชิงสัมพันธ์ การกำหนด Domain ของแอตทริบิวต์ในตารางมีความสำคัญอย่างไร
- ก. ใช้เพื่อควบคุมสิทธิ์ในการเข้าถึงข้อมูลในระดับ User ผ่าน Role-Based Authorization
 - ข. ใช้เพื่อกำหนดขนาดของข้อมูลในเชิงกายภาพเท่านั้น โดยไม่เกี่ยวข้องกับประเภทข้อมูล
 - ค. ใช้เพื่อกำหนดชนิดข้อมูลที่ยอมรับได้ในแอตทริบิวต์นั้น ๆ เพื่อควบคุมความถูกต้องของข้อมูลตั้งแต่ขั้นตอนการป้อนข้อมูล
 - ง. ใช้เพื่อจัดกลุ่มข้อมูลตามเขตการปกครองของประเทศเพื่อให้รองรับระบบภูมิภาคแบบกระจายศูนย์

เฉลย: ค

อธิบาย: การกำหนด Domain ของแอตทริบิวต์ หมายถึงการระบุ ชนิดข้อมูลและขอบเขตของค่าที่อนุญาตให้เก็บ เช่น VARCHAR(100), DATE, INT, หรืออาจเป็นค่าที่จำกัด เช่น ('ชาย', 'หญิง') การกำหนด Domain ที่เหมาะสมจะช่วยให้สามารถ ควบคุมความถูกต้องของข้อมูล ได้ตั้งแต่ต้น ลดโอกาสที่ข้อมูลจะผิดพลาดหรือขัดกับตรรกะของระบบ

42. ข้อใดคือลักษณะของความสัมพันธ์แบบ Many-to-Many ในการออกแบบฐานข้อมูลเชิงสัมพันธ์ และแนวทางการจัดการที่เหมาะสมคืออะไร
- ก. เป็นความสัมพันธ์ที่สามารถจัดการได้โดยตรงในตารางเดียว และไม่จำเป็นต้องมีตารางกลางหากใช้ Index
 - ข. เป็นความสัมพันธ์ที่ต้องแปลงให้เป็น One-to-Many ด้วยการรวมข้อมูลของทั้งสองเอนทิตีไว้ในตารางเดียว
 - ค. เป็นความสัมพันธ์ที่ต้องใช้ตารางกลางเพื่อเก็บคีย์จากทั้งสองตาราง และสามารถเพิ่มแอตทริบิวต์เฉพาะของความสัมพันธ์ได้
 - ง. เป็นความสัมพันธ์ที่ใช้ได้เฉพาะกับฐานข้อมูลแบบ NoSQL เท่านั้น เนื่องจาก RDBMS ไม่รองรับ

เฉลย: ค

อธิบาย: ความสัมพันธ์แบบ Many-to-Many (M:N) เช่น นักเรียนลงทะเบียนเรียนหลายวิชา และวิชาหนึ่งมีนักเรียนหลายคน จึงไม่สามารถจัดเก็บโดยตรงใน 2 ตาราง ต้องใช้ ตารางกลาง (Junction Table) เช่น Enrollment(StudentID, CourseID, Grade) ซึ่งเก็บ Foreign Key จากทั้งสองตาราง และสามารถเพิ่มแอตทริบิวต์เฉพาะของความสัมพันธ์ เช่น เกรด วันที่ลงทะเบียน ฯลฯ ได้

43. การสร้าง Index ในฐานข้อมูลเชิงสัมพันธ์มีผลต่อประสิทธิภาพในลักษณะใด
- ก. ทำให้ข้อมูลไม่สามารถอัปเดตหรือลบได้อย่างอัตโนมัติ หากมี Index อยู่ในตารางนั้น
 - ข. ช่วยเพิ่มความเร็วในการค้นหา (SELECT) ข้อมูล แต่มีผลกระทบต่อประสิทธิภาพของการแทรก (INSERT) และปรับปรุง (UPDATE)
 - ค. ทำให้ข้อมูลสามารถลิงก์ถึงกันได้อย่างรวดเร็วโดยไม่ต้องใช้ Foreign Key ในการอ้างอิง
 - ง. ช่วยให้สามารถจำลองความสัมพันธ์แบบซ้อนกันได้หลายชั้นในตารางเดียวโดยอัตโนมัติ

เฉลย: ข

อธิบาย: การสร้าง Index โดยเฉพาะ B-Tree Index หรือ Hash Index จะช่วย เพิ่มความเร็วในการค้นหา อย่างมากในกรณีที่มีเงื่อนไข WHERE หรือ JOIN แต่ในทางกลับกัน การทำ INSERT, UPDATE และ DELETE อาจ ช้าลง เพราะระบบต้องอัปเดต Index ทุกครั้งที่ข้อมูลเปลี่ยนแปลง จึงควรเลือกสร้าง Index เฉพาะฟิลด์ที่ใช้ค้นหาบ่อย

.....

44. คำว่า "Entity Integrity" ในฐานข้อมูลเชิงสัมพันธ์หมายถึงหลักการใด
- ก. หลักการที่กำหนดให้ Foreign Key ในตารางลูกต้องอ้างอิงกับค่าใน Primary Key ของตารางแม่
 - ข. หลักการที่ระบุว่า Primary Key ในทุกตารางต้องมีค่าที่ไม่ซ้ำกันและต้องไม่เป็น NULL
 - ค. หลักการที่บังคับให้ทุกแถวต้องมี Foreign Key เพื่อให้เชื่อมโยงกับแหล่งข้อมูลภายนอกเสมอ
 - ง. หลักการที่ใช้ควบคุมสิทธิ์ในการแก้ไขข้อมูลในตารางสำคัญเพื่อป้องกันการละเมิดข้อมูลส่วนบุคคล

เฉลย: ข

อธิบาย: Entity Integrity คือหลักการที่ระบุว่า Primary Key ต้องไม่เป็น NULL และต้องไม่ซ้ำกัน ในแต่ละแถวของตาราง ซึ่งมีความสำคัญเพราะใช้ในการระบุเรคคอร์ดเฉพาะหนึ่งเดียวในตาราง และทำให้สามารถอ้างอิงข้ามตารางผ่าน Foreign Key ได้อย่างถูกต้อง

.....

45. ถ้าต้องการเขียน SQL เพื่อดึงข้อมูลชื่อพนักงาน (EmployeeName) และชื่อแผนก (DepartmentName) ที่พนักงานแต่ละคนสังกัดอยู่ คำสั่งใดต่อไปนี้ถูกต้อง
- ก.

```
SELECT e.EmployeeName, d.DepartmentName
FROM Employee e LEFT OUTER JOIN Department d
ON e.DepartmentID = d.ID;
```
 - ข.

```
SELECT EmployeeName, DepartmentName
FROM Employee JOIN Department
WHERE Employee.DepartmentID = Department.ID;
```
 - ค.

```
SELECT EmployeeName, DepartmentName
FROM Employee INNER JOIN Department
ON Employee.DepartmentID = Department.ID;
```
 - ง.

```
SELECT Employee.EmployeeName AND Department.DepartmentName
FROM Employee AND Department
```

WHERE Department.ID = Employee.DepartmentID;

เฉลย: ค

อธิบาย: การเขียน SQL ที่ถูกต้องสำหรับการดึงข้อมูลจาก 2 ตารางที่มีความสัมพันธ์กัน ใช้ INNER JOIN พร้อมเงื่อนไข ON ระบุว่า Employee.DepartmentID = Department.ID ซึ่งจะแสดงเฉพาะเรคคอร์ดที่มีการจับคู่ข้อมูลทั้งสองฝั่ง **รูปแบบข้อ ค** ถูกต้องตามหลักการ JOIN และ Syntax

.....

46. ถ้าในระบบฐานข้อมูลมีตาราง Books, Authors และ BookAuthor โดยที่ BookAuthor ใช้เก็บความสัมพันธ์ระหว่างหนังสือและผู้แต่ง (Many-to-Many) การกำหนดคีย์หลักของ BookAuthor ควรเป็นแบบใดเพื่อความถูกต้องของข้อมูล
- ก. ใช้เพียงคอลัมน์ BookID เป็น Primary Key เพื่อป้องกันการซ้ำของหนังสือ
 - ข. ใช้เพียงคอลัมน์ AuthorID เป็น Primary Key เพราะผู้แต่งแต่ละคนจะไม่ซ้ำกัน
 - ค. ใช้คอลัมน์ BookID และ AuthorID ร่วมกันเป็น Primary Key แบบคอมโพสิต เพื่อป้องกันการซ้ำซ้อนของการจับคู่
 - ง. ไม่จำเป็นต้องใช้ Primary Key หากมี Foreign Key ครบทั้งสองฟิลด์ในตาราง BookAuthor

เฉลย: ค

อธิบาย: ตาราง BookAuthor ทำหน้าที่เป็น ตารางกลาง (junction table) ในความสัมพันธ์แบบหลายต่อหลาย (M:N) การใช้ BookID และ AuthorID ร่วมกันเป็น คีย์หลักแบบผสม (composite primary key) จะช่วยป้องกันไม่ให้เกิดการจับคู่ระหว่างหนังสือและผู้แต่งซ้ำกัน เช่น คนเขียนคนเดิมถูกบันทึกซ้ำสำหรับเล่มเดียวกัน

.....

47. คำว่า “Referential Integrity” ในระบบฐานข้อมูลเชิงสัมพันธ์ หมายถึงแนวปฏิบัติอย่างไร
- ก. การกำหนดให้ Foreign Key ในตารางถูกต้องมีค่าที่ตรงกับ Primary Key ที่มีอยู่จริงในตารางแม่เสมอ
 - ข. การป้องกันไม่ให้ข้อมูลที่ซ้ำกันมากเกินไปถูกจัดเก็บอยู่ในตารางเดียวกัน
 - ค. การออกแบบระบบฐานข้อมูลให้มีความสัมพันธ์ระหว่างตารางอย่างซับซ้อนสูงสุด
 - ง. การควบคุมไม่ให้แถวในตารางใดถูกลบหรือแก้ไขเมื่อมีข้อมูลอื่นในตารางเดียวกันเชื่อมโยงอยู่

เฉลย: ก

อธิบาย: Referential Integrity (RI) คือกฎที่ใช้ควบคุมความสัมพันธ์ของ Foreign Key โดยระบุว่า ค่าของ Foreign Key ต้องตรงกับค่าของ Primary Key ที่มีอยู่จริงในอีกตารางหนึ่ง เพื่อไม่ให้เกิดข้อมูลที่ไม่มีจุดเชื่อมโยง (orphan records) เช่น ถ้า Orders.CustomerID ไม่มีใน Customers.CustomerID จะละเมิด RI

.....

48. แนวคิดของการใช้ ER Model (Entity-Relationship Model) ก่อนออกแบบฐานข้อมูลจริงในเชิงเทคนิคมีประโยชน์อย่างไร
- ก. เพื่อจำลองโครงสร้างการจัดเก็บข้อมูลแบบเสมือนจริงโดยไม่ต้องใช้ฐานข้อมูล
 - ข. เพื่อสร้างระบบฐานข้อมูลที่สามารถทำงานกับ NoSQL ได้อย่างอัตโนมัติ

- ค. เพื่อระบุความสัมพันธ์, เอนทิตี, และแอตทริบิวต์อย่างชัดเจนก่อนแปลงเป็นโครงสร้างฐานข้อมูลจริง
ง. เพื่อแปลงข้อมูลเชิงภาพให้เป็นโค้ด SQL โดยตรงผ่านระบบ AI-based Schema Mapping

เฉลย: ค

อธิบาย: ER Model ใช้ในการออกแบบฐานข้อมูลเชิงแนวคิด โดยระบุว่า มี Entity อะไร, Attribute อะไร, และมีความสัมพันธ์อย่างไร (Relationship) จากนั้นจะถูกแปลงเป็นโครงสร้างเชิงตรรกะ (Logical Design) และสุดท้ายจึงเป็น Physical Database Model เช่น SQL Schema การวางแผนผ่าน ER Model จึงช่วยให้การออกแบบชัดเจนและมีประสิทธิภาพ

49. หากนักออกแบบฐานข้อมูลพบว่าคอลัมน์ CustomerAddress ในตาราง Customers ประกอบด้วยที่อยู่แบบเต็ม (บ้านเลขที่, ถนน, เขต, จังหวัด, รหัสไปรษณีย์) ซึ่งอาจต้องใช้แยกในอนาคต ควรดำเนินการอย่างไร
- ก. แยกข้อมูลที่อยู่เป็นตารางใหม่แล้วเชื่อมด้วย Foreign Key
 - ข. คงข้อมูลไว้เหมือนเดิมและจัดทำ Index เพิ่มเพื่อให้ค้นหาเร็วขึ้น
 - ค. แยกฟิลด์ย่อยภายใน Address เป็นคอลัมน์ใหม่ในตารางเดิม เช่น Street, District, Province, ZipCode
 - ง. ลบคอลัมน์ CustomerAddress และให้ระบบบันทึกที่อยู่เป็นภาพหรือ PDF แทน

เฉลย: ค

อธิบาย: การเก็บข้อมูล Address ไว้รวมกันจะทำให้การค้นหา, กรอง, หรือแสดงผลในอนาคตทำได้ยาก การแยกออกเป็นฟิลด์ย่อยช่วยให้สามารถ Query เช่น WHERE Province = 'กรุงเทพฯ' ได้อย่างง่ายดาย ซึ่งสอดคล้องกับหลักการออกแบบข้อมูลที่ Atomic (ข้อมูลควรย่อยได้) ตาม 1NF (First Normal Form)

50. การกำหนด Foreign Key Constraint พร้อม ON DELETE CASCADE มีผลต่อระบบฐานข้อมูลอย่างไร
- ก. เมื่อลบข้อมูลในตารางแม่ ข้อมูลที่อ้างอิงอยู่ในตารางลูกจะถูกลบตามไปโดยอัตโนมัติ
 - ข. ป้องกันไม่ให้มีการลบข้อมูลในตารางแม่ หากยังมีตารางลูกอ้างอิงอยู่
 - ค. บังคับให้ต้องใช้ Transaction ทุกครั้งเมื่อลบข้อมูล
 - ง. ลบเฉพาะข้อมูลในตารางลูกที่ไม่มีการใช้งานจริง แต่ไม่ได้ลบข้อมูลที่อ้างอิง

เฉลย: ก

อธิบาย: การใช้ ON DELETE CASCADE ในการกำหนด Foreign Key หมายความว่า หากลบแถวหนึ่งในตารางแม่ (เช่น Customers), แถวทั้งหมดในตารางลูก (Orders) ที่อ้างอิง Foreign Key เดียวกันจะถูก ลบตามโดยอัตโนมัติ เหมาะกับกรณีที่ความสัมพันธ์เป็นของกันและกันโดยสมบูรณ์

51. ข้อใดคือข้อดีหลักของการทำ Normalization จนถึงระดับ Third Normal Form (3NF)
- ก. ช่วยลดขนาดของฐานข้อมูลโดยบีบอัดข้อมูลซ้ำ ๆ ให้เหลือเพียงครั้งเดียว
 - ข. ทำให้สามารถแยก Entity ออกเป็นหลายตารางได้โดยอัตโนมัติ ไม่ต้องออกแบบเอง
 - ค. ลดความซ้ำซ้อนของข้อมูล และลดโอกาสที่ข้อมูลจะไม่สอดคล้องกัน (Anomaly)
 - ง. ทำให้ระบบสามารถจัดเรียงข้อมูลอัตโนมัติตามลำดับตัวอักษรทุกคอลัมน์

เฉลย: ค

อธิบาย: การทำ Normalization ถึงระดับ 3NF จะช่วยลดปัญหา ข้อมูลซ้ำซ้อน (Redundancy) และป้องกัน ข้อมูลผิดพลาด (Update Anomaly, Insertion Anomaly, Deletion Anomaly) ทำให้การอัปเดตหรือลบ ข้อมูลมีความแม่นยำ และประสิทธิภาพของฐานข้อมูลดีขึ้นในระยะยาว โดยการแยกข้อมูลให้อยู่ในตารางที่มีความสัมพันธ์อย่างมีระบบ

.....

52. ข้อใดแสดงถึงปัญหาของการไม่ทำ Normalization ที่อาจเกิดขึ้นกับระบบฐานข้อมูล

- ก. ฐานข้อมูลไม่สามารถ Query ได้ด้วย SQL เนื่องจากไม่มีโครงสร้างที่เป็นระบบ
- ข. ระบบจะต้องใช้ NoSQL แทน RDBMS เพื่อจัดการกับข้อมูลที่ไม่มีแบบแผน
- ค. เกิดความซ้ำซ้อนของข้อมูล และการลบหรืออัปเดตข้อมูลบางรายการอาจเกิดความไม่สอดคล้องกัน
- ง. การเชื่อมโยงระหว่างตารางจะล้มเหลวทันทีเมื่อมีการเพิ่มฟิลด์ใหม่ในตารางแม่

เฉลย: ค

อธิบาย: หากไม่มีการทำ Normalization อย่างเหมาะสม จะทำให้ ข้อมูลซ้ำซ้อนมาก เช่น ชื่อลูกค้าถูกบันทึกซ้ำหลายครั้งใน Orders และหากมีการแก้ไขชื่อเพียงบางรายการ จะเกิด Inconsistency ข้อมูลไม่ตรงกันทุกจุด หรือหากลบลูกค้าคนนั้นออก ก็อาจลบรายการสั่งซื้อที่ยังต้องใช้อยู่โดยไม่ตั้งใจ

.....

53. การออกแบบฐานข้อมูลให้รองรับการ Query ที่ซับซ้อนและรวดเร็วควรใช้หลักการใดเพิ่มเติมนอกจากการทำ Normalization

- ก. ใช้การทำ Denormalization อย่างมีกลยุทธ์ในตารางที่ถูก Query บ่อย ๆ
- ข. ยกเลิก Primary Key เพื่อให้สามารถใช้ Index ได้มากขึ้น
- ค. ใช้การเขียน Stored Procedure แทน SQL Query เพื่อให้ไม่ต้องใช้ JOIN
- ง. ปรับฐานข้อมูลให้เป็นแบบ Graph Database เพื่อการดึงข้อมูลได้เร็วขึ้นโดยไม่เปลี่ยนโครงสร้าง

เฉลย: ก

อธิบาย: ในกรณีที่ระบบมี Query บ่อย ๆ และมีความซับซ้อนสูง อาจเลือก Denormalize บางตาราง เช่น การรวม Customer และ Orders ไว้ใน View หรือตารางเดียวเพื่อความเร็วในการค้นหา โดยยอมให้ข้อมูลบางอย่างซ้ำได้ เพื่อแลกกับความเร็วในการดึงข้อมูล ซึ่งต่างจาก Normalization ที่เน้นความถูกต้องสูงสุด

.....

54. ในฐานข้อมูลขนาดใหญ่ที่มีตารางหลักอย่างน้อย 50 ตาราง การใช้ ER Diagram จะมีประโยชน์ในแง่ใดมากที่สุด

- ก. ช่วยจัดการความสัมพันธ์ทางคณิตศาสตร์ระหว่างแอตทริบิวต์ได้อย่างเป็นระบบ
- ข. ช่วยแสดงภาพรวมความสัมพันธ์ระหว่าง Entity และเข้าใจโครงสร้างข้อมูลทั้งระบบง่ายขึ้น
- ค. ช่วยให้สามารถจำลองข้อมูลจริงแบบ 3 มิติได้ผ่านการใช้เครื่องมือ BI
- ง. ช่วยให้ระบบสามารถแปลงข้อมูลไปใช้กับ Big Data โดยไม่ต้องมีการ Mapping เพิ่มเติม

เฉลย: ข

อธิบาย: ER Diagram มีหน้าที่ในการ แสดงภาพรวมโครงสร้างข้อมูลทั้งหมด เช่น ตาราง, คีย์, ความสัมพันธ์, จำนวน (Cardinality) ซึ่งเป็นเครื่องมือช่วยนักพัฒนา, DBA และ Business Analyst ทำความเข้าใจระบบ โดยไม่ต้องอ่านโค้ด SQL ทั้งหมด ทำให้เข้าใจระบบเร็วและลดข้อผิดพลาดในการพัฒนา

.....

55. หากมีการออกแบบฐานข้อมูลที่มี Foreign Key หลายชั้น (เช่น ลูกค้า -> ใบสั่งซื้อ -> รายการสินค้า) การลบลูกค้าคนหนึ่งควรระมัดระวังในประเด็นใดมากที่สุด

- ก. คำสั่ง DELETE ที่ใช้กับตารางลูกค้าอาจลบสินค้าทั้งระบบทั้งหมดตามไปด้วย
- ข. หากไม่มีการกำหนด ON DELETE CASCADE ข้อมูลในตารางลูกจะกลายเป็น NULL โดยอัตโนมัติ
- ค. ถ้าไม่มีการจัดการ Referential Integrity จะเกิด Error ทันทีเมื่อพยายามลบข้อมูลในตารางแม่
- ง. ระบบจะเปลี่ยนสถานะลูกค้าคนนั้นเป็น "ลบแล้ว" โดยอัตโนมัติเพื่อไม่ให้กระทบกับตารางลูก

เฉลย: ค

อธิบาย: ถ้ามี Foreign Key Constraint ระหว่าง Customers → Orders → OrderDetails แล้วผู้ใช้งานลบ Customer โดยไม่จัดการการอ้างอิงจาก Orders ก่อน และไม่มีการตั้ง ON DELETE CASCADE หรือ SET NULL ระบบจะแจ้ง Error ทันทีเพราะ ละเมิด Referential Integrity ต้องจัดการลำดับการลบให้เหมาะสม หรือใช้ Transaction ช่วย

.....

56. ในการออกแบบฐานข้อมูลสำหรับระบบร้านค้าออนไลน์ ตาราง Product ควรมี StockQuantity หรือไม่ หากต้องการรองรับการอัปเดตแบบเรียลไทม์เมื่อมีการขายสินค้า

- ก. ไม่ควรมีฟิลด์ StockQuantity ควรคำนวณทุกครั้งจากยอดขายย้อนหลัง
- ข. ควรมีฟิลด์ StockQuantity ที่อัปเดตผ่าน Trigger หรือ Stored Procedure ทันทีหลังมีการขาย
- ค. ควรเก็บข้อมูลสต็อกไว้ในระบบไฟล์ภายนอกเท่านั้นเพื่อให้ระบบเร็วขึ้น
- ง. ควรใช้ฐานข้อมูล NoSQL จัดเก็บจำนวนสินค้าเพราะเปลี่ยนแปลงบ่อย

เฉลย: ข

อธิบาย: ในระบบที่ต้อง อัปเดตสต็อกแบบทันที (Real-Time) เช่น เมื่อมีคำสั่งซื้อใหม่ ต้องลดจำนวน StockQuantity ทันที ควรมีฟิลด์ StockQuantity อยู่ใน Product และเขียน Trigger หรือ Stored Procedure ที่ทำงานทันทีเมื่อมีการเพิ่มใน OrderDetails เพื่อรักษาความถูกต้องของสต็อกและป้องกันการขายเกิน

.....

57. คำว่า “Cardinality” ในการออกแบบฐานข้อมูลเชิงสัมพันธ์หมายถึงอะไร

- ก. จำนวนแอตทริบิวต์ที่มีในตารางที่ใช้ในการจัดกลุ่มข้อมูล
- ข. จำนวนความสัมพันธ์ที่เอนทิตีหนึ่งสามารถมีต่ออีกเอนทิตีหนึ่ง เช่น 1:1, 1:N, N:M
- ค. จำนวนของ Foreign Key ที่ถูกใช้ร่วมในแต่ละตาราง
- ง. จำนวนครั้งที่ระบบสามารถดึงข้อมูลจากตารางนั้นได้ในช่วงเวลาหนึ่ง

เฉลย: ข

อธิบาย: **Cardinality** คือการอธิบายความสัมพันธ์ระหว่าง Entity ว่า 1 ต่อ 1 (1:1), 1 ต่อ หลาย (1:N), หรือ หลายต่อหลาย (M:N) ซึ่งมีผลต่อการออกแบบตาราง เช่น ถ้าเป็น 1:N จะใส่ Foreign Key ในฝั่ง N ส่วน M:N จะต้องมีการวางกลาง เป็นแนวคิดพื้นฐานที่สำคัญมากในการออกแบบความสัมพันธ์

.....

58. ข้อใดเป็นข้อเสียที่อาจเกิดขึ้นจากการทำ Denormalization มากเกินไป

- ก. ไม่สามารถใช้ Foreign Key ได้ ทำให้เกิดข้อมูลหลงเหลือในระบบ
- ข. ขนาดของฐานข้อมูลจะลดลงอย่างมากจนไม่สามารถเก็บประวัติได้
- ค. ข้อมูลซ้ำซ้อนสูงและเสี่ยงต่อการไม่สอดคล้องกันเมื่อมีการแก้ไขข้อมูลเพียงบางส่วน
- ง. ไม่สามารถเขียน Query แบบ JOIN ได้อีกเนื่องจากข้อมูลรวมอยู่ในตารางเดียวหมดแล้ว

เฉลย: ค

อธิบาย: การ Denormalize ทำให้ข้อมูลบางอย่าง **ถูกรวมอยู่ในตารางเดียว** เพื่อความเร็ว แต่จะทำให้เกิด **ข้อมูลซ้ำ** เช่น ชื่อลูกค้าอาจซ้ำหลายแถว หากต้องเปลี่ยนชื่อ อาจเปลี่ยนไม่ครบทุกจุด จึงทำให้ **ข้อมูลไม่สอดคล้องกัน** (Data Inconsistency) ได้ง่าย

.....

59. ในระบบฐานข้อมูลขนาดใหญ่ ควรใช้เทคนิคใดในการป้องกันการสูญหายของข้อมูลจากการทำธุรกรรมที่ไม่สมบูรณ์

- ก. ใช้ Foreign Key พร้อม ON DELETE CASCADE ทุกตาราง
- ข. ใช้ระบบ NoSQL แทน RDBMS เพราะ NoSQL ไม่เกิด Transaction Error
- ค. ใช้แนวคิด Transaction และ Commit/Rollback เพื่อควบคุมความถูกต้องของการเปลี่ยนแปลง
- ง. ใช้ Index ร่วมกับการ Query แบบ Select Only เพื่อไม่ให้มีการเปลี่ยนข้อมูล

เฉลย: ค

อธิบาย: การใช้ Transaction (BEGIN... COMMIT / ROLLBACK) เป็นเทคนิคมาตรฐานใน RDBMS เพื่อให้การเปลี่ยนแปลงหลาย ๆ รายการในคำสั่งเดียว **สำเร็จทั้งหมดหรือไม่เลย** เช่น ถ้าการโอนเงินหักบัญชีผู้โอนสำเร็จ แต่เพิ่มบัญชีผู้รับไม่สำเร็จ คำสั่งจะ **Rollback** กลับ ทั้งหมด ป้องกันข้อมูลไม่ถูกต้อง

.....

60. การใช้ View ใน SQL มีประโยชน์ต่อการออกแบบฐานข้อมูลในแง่ใด

- ก. ใช้เพื่อสร้างตารางจริงที่เก็บข้อมูลซ้ำจากหลายตารางเพื่อเพิ่มประสิทธิภาพ
- ข. ใช้เพื่อซ่อนความซับซ้อนของโครงสร้างฐานข้อมูลและให้ผู้ใช้งาน Query ได้ง่ายขึ้น
- ค. ใช้เพื่อสร้าง Index ชั่วคราวที่มีประสิทธิภาพมากกว่าคีย์ปกติ
- ง. ใช้เพื่อให้สามารถสร้าง Trigger ได้บน View เพื่อแทนที่ตารางจริงในการอัปเดตข้อมูล

เฉลย: ข

อธิบาย: **View** คือ ตารางเสมือน ที่รวบรวมข้อมูลจากหลายตารางจริง ๆ มาแสดงผลในรูปแบบที่ต้องการ เช่น รวม

Customers, Orders, Products เข้าด้วยกัน ใช้เพื่อ **ซ่อนความซับซ้อนของ JOIN หรือ Subquery** และให้ผู้ใช้งานสามารถดึงข้อมูลได้สะดวกโดยไม่ต้องรู้โครงสร้างเบื้องหลัง

-
61. ในกระบวนการออกแบบฐานข้อมูล หากนักพัฒนาไม่กำหนด Primary Key ให้กับตาราง จะส่งผลเสียในข้อใดมากที่สุด
- ก. ไม่สามารถสร้างตารางใหม่ได้เลยหากไม่มี Primary Key อย่างน้อย 1 คอลัมน์
 - ข. ไม่สามารถกำหนด Foreign Key จากตารางอื่นมายังตารางนี้ได้ ทำให้ความสัมพันธ์ไม่สมบูรณ์
 - ค. ระบบฐานข้อมูลจะลบข้อมูลในตารางนั้นอัตโนมัติเมื่อไม่มี Primary Key
 - ง. ไม่สามารถใช้งาน SQL JOIN กับตารางนั้นได้ในทุกกรณี

เฉลย: ข

อธิบาย: การไม่กำหนด Primary Key จะทำให้ไม่สามารถสร้างความสัมพันธ์ผ่าน Foreign Key ได้ เพราะระบบไม่สามารถอ้างอิงแบบแน่นอนได้ว่า **แต่ละแถวในตารางนั้นมีเอกลักษณ์เฉพาะหรือไม่** ซึ่งเป็นพื้นฐานสำคัญของฐานข้อมูลเชิงสัมพันธ์และการ JOIN ตารางต่าง ๆ

-
62. ถ้ามีการออกแบบฐานข้อมูลระบบห้องสมุด โดยมีตาราง Book, Member, Borrow, แล้วความสัมพันธ์ระหว่าง Borrow กับตารางอื่นควรเป็นแบบใด
- ก. Borrow มีความสัมพันธ์แบบ N:1 กับ Book และแบบ 1:N กับ Member
 - ข. Borrow มีความสัมพันธ์แบบ 1:N กับ Book และแบบ N:1 กับ Member
 - ค. Borrow มีความสัมพันธ์แบบ N:1 กับทั้ง Book และ Member
 - ง. Borrow มีความสัมพันธ์แบบ 1:N กับทั้ง Book และ Member

เฉลย: ค

อธิบาย: ตาราง Borrow คือ ตารางกลางที่บันทึกประวัติการยืม ซึ่งแต่ละรายการมี หนังสือ 1 เล่ม และสมาชิก 1 คน ดังนั้น Borrow จึงเป็น N:1 กับ Book (ยืมได้หลายครั้งต่อหนังสือ) และ N:1 กับ Member (สมาชิกแต่ละคนสามารถยืมหลายเล่มได้)

-
63. การใช้ Index บนฐานข้อมูลมีจุดมุ่งหมายหลักในด้านใด
- ก. ลดขนาดของฐานข้อมูลให้เล็กลงเพื่อใช้พื้นที่ได้น้อยที่สุด
 - ข. เพิ่มความเร็วในการอ่านข้อมูลโดยลดเวลาค้นหาในตาราง
 - ค. ป้องกันข้อมูลซ้ำซ้อนจากการบันทึกข้อมูลซ้ำ
 - ง. ใช้แทน Foreign Key เพื่อเชื่อมโยงตารางโดยอัตโนมัติ

เฉลย: ข

อธิบาย: Index คือโครงสร้างพิเศษที่สร้างขึ้นเพื่อ **ช่วยเพิ่มความเร็วในการค้นหา (Query)** เช่น การค้นหาข้อมูลจาก Primary Key หรือชื่อพนักงาน หากไม่มี Index ระบบจะต้องไล่ดูทุกแถว (Full Scan) ซึ่งช้า Index จึงเปรียบเหมือนสารบัญที่ช่วยให้ฐานข้อมูลเข้าถึงข้อมูลได้เร็วขึ้น

.....

64. ในสถานการณ์ใดที่ควรใช้ Composite Primary Key

- ก. เมื่อมีข้อมูลจำนวนมากเกิน 1 ล้านเรคคอร์ดต่อปี
- ข. เมื่อข้อมูลหนึ่งรายการไม่สามารถระบุได้ด้วยคอลัมน์เดียว เช่น ตาราง Enrollment
- ค. เมื่อมีตารางที่ไม่สามารถเชื่อมโยงกับตารางอื่นได้เลย
- ง. เมื่อมีการใช้ Data Warehouse แทนฐานข้อมูลแบบ Online

เฉลย: ข

อธิบาย: **Composite Primary Key** คือ Primary Key ที่ประกอบด้วย **หลายคอลัมน์ร่วมกัน** ใช้ในกรณีที่ไม่มีคอลัมน์ใดคอลัมน์เดียวที่ระบุแถวได้ เช่น ในตาราง Enrollment (นักเรียนลงทะเบียนวิชาเรียน) ต้องใช้ StudentID + CourseID จึงจะระบุเรคคอร์ดได้ชัดเจน

.....

65. ตารางใดต่อไปนี้จะทำหน้าที่เป็น Associative Entity ในระบบฐานข้อมูล

- ก. ตารางที่เก็บข้อมูลลูกค้าพร้อมที่อยู่ทั้งหมดในบรรทัดเดียว
- ข. ตารางกลางที่เชื่อมความสัมพันธ์แบบหลายต่อหลาย (M:N) ระหว่างสอง Entity
- ค. ตารางหลักที่ใช้เก็บข้อมูลสินค้าในคลัง
- ง. ตารางที่ใช้คำนวณยอดขายรวมโดยใช้ Aggregate Function

เฉลย: ข

อธิบาย: **Associative Entity** หรือ **ตารางกลาง** คือ ตารางที่ใช้เชื่อมความสัมพันธ์แบบ **หลายต่อหลาย (M:N)** เช่น Student_Course เชื่อม Student กับ Course โดยตารางนี้จะมี Foreign Key ไปยังตารางทั้งสอง และมักมี Composite Primary Key

.....

66. การใช้ NULL ในฐานข้อมูลมีข้อควรระวังใดมากที่สุด

- ก. การใช้ NULL จะเพิ่มความเร็วในการ Query โดยไม่ต้องตรวจสอบเงื่อนไข
- ข. การใช้ NULL จะทำให้ระบบฐานข้อมูลลบข้อมูลที่เข้าเงื่อนไขอัตโนมัติ
- ค. NULL ไม่สามารถใช้เปรียบเทียบกับค่าอื่น ๆ โดยตรงได้ เช่น = NULL
- ง. NULL ทำให้ไม่สามารถสร้าง Foreign Key ได้ในทุกกรณี

เฉลย: ค

อธิบาย: ค่า NULL หมายถึงไม่มีค่า (Unknown/Not Available) ซึ่ง **ไม่สามารถเปรียบเทียบกับ = ได้** เช่น WHERE price = NULL จะไม่มีผล ต้องใช้ IS NULL แทน และต้องระวังเมื่อใช้ในการคำนวณหรือเงื่อนไข เพราะอาจทำให้ผลลัพธ์ผิดพลาด

.....

67. คำสั่ง ALTER TABLE มักถูกใช้เพื่อวัตถุประสงค์ใด

- ก. ลบตารางออกจากฐานข้อมูลอย่างถาวร
- ข. ปรับปรุงข้อมูลภายในตารางโดยอัตโนมัติ
- ค. แก้ไขโครงสร้างตาราง เช่น เพิ่ม/ลบคอลัมน์ หรือเปลี่ยนชนิดข้อมูล
- ง. นำเข้าข้อมูลจำนวนมากจากแหล่งข้อมูลภายนอกเข้าสู่ฐานข้อมูล

เฉลย: ค

อธิบาย: คำสั่ง ALTER TABLE ใช้สำหรับ **เปลี่ยนแปลงโครงสร้างของตาราง** เช่น เพิ่มฟิลด์ใหม่ (ADD COLUMN), ลบฟิลด์ (DROP COLUMN), เปลี่ยนชนิดข้อมูล (MODIFY) เป็นต้น ซึ่งต่างจาก UPDATE ที่ใช้แก้ไขข้อมูลในตาราง

.....

68. ตาราง Employee ที่มีข้อมูลตำแหน่งงานและแผนกอยู่ในฟิลด์เดียวกัน จะละเมิดกฎ Normalization ข้อใด

- ก. ละเมิด 1NF เพราะมีค่าหลายค่าในหนึ่งคอลัมน์
- ข. ละเมิด 2NF เพราะฟิลด์ไม่มีความสัมพันธ์กับ Primary Key
- ค. ละเมิด 3NF เพราะฟิลด์ขึ้นกับฟิลด์ที่ไม่ใช่ Primary Key
- ง. ไม่ละเมิด Normalization ใดเพราะเป็นฟิลด์รวมข้อมูล

เฉลย: ก

อธิบาย: การใส่หลายค่าลงในฟิลด์เดียว เช่น “Manager – IT” คือการ **ละเมิด First Normal Form (1NF)** เพราะ 1NF ระบุว่าฟิลด์หนึ่งต้องมี **ค่าหนึ่งเดียวเท่านั้น** หากต้องการแยกความสัมพันธ์จริง ต้องแยกฟิลด์เป็น Position และ Department

.....

69. เหตุผลที่ควรแยก Entity Customer และ Order ออกเป็นคนละตาราง แทนที่จะเก็บรวมในตารางเดียวคืออะไร

- ก. เพื่อให้สามารถลบลูกค้าได้แม้จะมีคำสั่งซื้ออยู่
- ข. เพื่อให้สามารถเก็บข้อมูลลูกค้าได้แบบ JSON โดยไม่ต้องใช้ JOIN
- ค. เพื่อความสะดวกในการ Query ยอดขายรวมโดยไม่ต้องมีข้อมูลลูกค้า
- ง. เพื่อหลีกเลี่ยงการซ้ำซ้อนของข้อมูลลูกค้าในหลายรายการสั่งซื้อ

เฉลย: ง

อธิบาย: ถ้าเก็บ Customer และ Order ไว้ในตารางเดียวกัน จะเกิด **ข้อมูลลูกค้าซ้ำ** ทุกครั้งที่มีคำสั่งซื้อใหม่ เช่น ชื่อลูกค้าถูกเก็บซ้ำหลายครั้ง ซึ่งไม่ดีต่อการจัดการข้อมูล การแยกเป็นสองตารางแล้วเชื่อมกันด้วย Foreign Key จะทำให้ประหยัดพื้นที่และลด Anomaly

.....

70. การสร้าง Foreign Key ระหว่างตารางสองตารางจะมีผลอย่างไรในระบบฐานข้อมูล
- ก. ระบบจะอนุญาตให้ลบข้อมูลในตารางแม่โดยไม่ต้องตรวจสอบความสัมพันธ์
 - ข. ข้อมูลในตารางลูกจะกลายเป็น NULL โดยอัตโนมัติเมื่อมีการอัปเดตในตารางแม่
 - ค. ระบบจะตรวจสอบความถูกต้องของข้อมูลระหว่างสองตารางทุกครั้งที่มีการ Insert หรือ Delete
 - ง. การใช้ Foreign Key จะลดประสิทธิภาพของระบบจนไม่สามารถ Query ได้เร็ว

เฉลย: ค

อธิบาย: Foreign Key ทำให้ระบบ ตรวจสอบความถูกต้องของความสัมพันธ์ (Referential Integrity) ทุกครั้ง เช่น หากใส่รหัสลูกค้าที่ไม่มีอยู่จริงในตาราง Customers ลงในตาราง Orders ระบบจะไม่อนุญาต เพราะละเมิดความสัมพันธ์ ถือเป็นการป้องกันข้อผิดพลาดในการบันทึกข้อมูล

71. เมื่อมีการใช้ Foreign Key ที่เชื่อมโยงตาราง Order กับ Customer โดยใช้กฎ ON DELETE CASCADE ผลกระทบที่จะเกิดขึ้นคืออะไร
- ก. หากลบลูกค้าออกจากระบบ รายการสั่งซื้อของลูกค้าจะยังคงอยู่ในตาราง Order เพื่อรักษาประวัติ
 - ข. หากลบลูกค้าออกจากระบบ รายการสั่งซื้อทั้งหมดที่อ้างอิงถึงลูกค้ารายนั้นจะถูกลบออกไปโดยอัตโนมัติ
 - ค. หากมีคำสั่งซื้ออยู่ จะไม่สามารถลบลูกค้ารายนั้นได้จนกว่าจะลบคำสั่งซื้อทิ้งเอง
 - ง. รายการสั่งซื้อของลูกค้าจะถูกเปลี่ยนให้ CustomerID เป็น NULL โดยอัตโนมัติเมื่อมีการลบลูกค้า

เฉลย: ข

อธิบาย: เมื่อกำหนด Foreign Key พร้อม ON DELETE CASCADE หมายถึง หากมีการ ลบข้อมูลในตารางแม่ (Customer) แล้ว ข้อมูลในตารางลูก (Order) ที่อ้างอิงถึงลูกค้ารายนั้น จะถูกลบอัตโนมัติ เพื่อลดการเกิดข้อมูลขาดความสัมพันธ์ (Orphaned Record)

72. ในแง่ของการทำ Normalization การแยกฟิลด์ “ที่อยู่” ออกเป็น เลขที่, ถนน, ตำบล, อำเภอ, จังหวัด และ รหัสไปรษณีย์ เป็นการส่งเสริม Normal Form ในระดับใด
- ก. First Normal Form (1NF) เพราะลดค่าซ้ำ
 - ข. Second Normal Form (2NF) เพราะแยกค่าออกจาก Composite Key
 - ค. Third Normal Form (3NF) เพราะกำจัดการพึ่งพาแบบ Transitively
 - ง. Boyce-Codd Normal Form (BCNF) เพราะสร้างความสัมพันธ์แบบฟังก์ชันสมบูรณ์

เฉลย: ก

อธิบาย: การแยกฟิลด์ที่อยู่ซึ่งรวมหลายข้อมูลไว้ในคอลัมน์เดียวออกเป็นหลายคอลัมน์ ถือเป็นการทำให้เป็น First Normal Form (1NF) โดยตรง เพราะช่วยให้ แต่ละฟิลด์มีค่าเดียว (Atomic Value) ตามข้อกำหนดหลักของ 1NF

73. ถ้าต้องการบันทึกการเปลี่ยนแปลงข้อมูลสำคัญ เช่น การเปลี่ยนแปลงเงินเดือนพนักงานในระบบ HR ควรออกแบบอย่างไรให้เหมาะสม
- ก. ใช้คำสั่ง UPDATE ไปยังฟิลด์เงินเดือนในตารางพนักงานเพื่อแทนที่ค่าเดิม
 - ข. สร้างตารางแยกชื่อ SalaryHistory เพื่อเก็บบันทึกการเปลี่ยนแปลงเงินเดือนพร้อมวันที่
 - ค. เพิ่มฟิลด์ใหม่ในตารางพนักงานชื่อ PreviousSalary เพื่อเก็บค่าก่อนหน้า
 - ง. ไม่ควรเก็บการเปลี่ยนแปลงเงินเดือนเพราะผิดหลักความเป็นส่วนตัว

เฉลย: ข

อธิบาย: วิธีที่ถูกต้องและเป็นมาตรฐานในการเก็บการเปลี่ยนแปลงข้อมูล คือการเก็บแบบประวัติ (Historical Data) ด้วยการสร้างตารางใหม่ เช่น SalaryHistory ซึ่งเก็บ EmployeeID, OldSalary, NewSalary, EffectiveDate เพื่อให้สามารถตรวจสอบย้อนหลังได้แบบ Audit Trail

.....

74. คำว่า “ความสมบูรณ์ของข้อมูล” (Data Integrity) ในระบบฐานข้อมูล หมายถึงข้อใดต่อไปนี้มากที่สุด
- ก. การที่ข้อมูลไม่ถูกเข้ารหัสเพื่อความปลอดภัย
 - ข. การที่ข้อมูลไม่มีการซ้ำซ้อนหรือผิดพลาดในเชิงโครงสร้าง
 - ค. การที่ข้อมูลถูกนำเสนอในรูปแบบที่สวยงามและอ่านง่าย
 - ง. การที่ข้อมูลในระบบสามารถใช้ร่วมกันระหว่างผู้ใช้หลายคนได้พร้อมกัน

เฉลย: ข

อธิบาย: Data Integrity คือความถูกต้อง สม่ำเสมอ และเชื่อถือได้ของข้อมูลในระบบฐานข้อมูล เช่น ข้อมูลลูกค้ามีหมายเลขประจำตัวที่ไม่ซ้ำ ข้อมูลที่สัมพันธ์กันมี Foreign Key ที่เชื่อมถูกต้อง ไม่มีข้อมูลผิดพลาดประเภทหรือละเมิดข้อจำกัด

.....

75. ในแนวทางการออกแบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Design) แนวคิดใดที่สำคัญที่สุดก่อนเริ่มออกแบบโครงสร้างตารางจริง
- ก. การสร้างแบบฟอร์มการกรอกข้อมูลให้ผู้ใช้
 - ข. การกำหนดชนิดข้อมูล (Data Type) ให้กับฟิลด์ทุกคอลัมน์
 - ค. การวิเคราะห์ Entity และความสัมพันธ์ระหว่างกัน (ER Diagram)
 - ง. การออกแบบหน้าจอแสดงผลและรายงานตามที่ต้องการ

เฉลย: ค

อธิบาย: ก่อนจะออกแบบตาราง ต้องเข้าใจก่อนว่า ระบบประกอบด้วย Entity อะไรบ้าง และมีความสัมพันธ์แบบใด (ER Diagram) เช่น 1:1, 1:N, N:M เพื่อให้ออกแบบโครงสร้างได้ถูกต้อง เป็นรากฐานของการกำหนด Primary Key, Foreign Key และ Normalization

.....

76. ความแตกต่างระหว่าง Domain Constraint กับ Referential Integrity Constraint คืออะไร
- ก. Domain Constraint ควบคุมค่าข้อมูลแต่ละฟิลด์ ส่วน Referential Integrity ควบคุมการอ้างอิงระหว่างตาราง
 - ข. Domain Constraint ใช้ควบคุม Foreign Key ส่วน Referential Integrity ใช้ควบคุม Primary Key
 - ค. Domain Constraint เป็นเรื่องของความปลอดภัย ส่วน Referential Integrity เป็นเรื่องของความเร็ว
 - ง. ทั้งสองแบบทำหน้าที่เหมือนกัน ต่างกันแค่ชื่อเรียกเท่านั้น

เฉลย: ก

อธิบาย:

- **Domain Constraint** = ข้อจำกัดในแต่ละคอลัมน์ เช่น อายุต้องเป็นเลขบวก, เพศต้องเป็น “ชาย” หรือ “หญิง”
- **Referential Integrity Constraint** = ตรวจสอบความสัมพันธ์ของข้อมูลระหว่างตาราง เช่น การอ้างอิง Foreign Key ต้องมีข้อมูลที่สอดคล้องในตารางหลัก

-
77. ในระบบฐานข้อมูลที่รองรับการทำงานพร้อมกัน (Concurrency) ปัญหาใดเกิดขึ้นเมื่อมีการอ่านค่าข้อมูลที่เพิ่งถูกเขียนแต่ยังไม่ Commit
- ก. Lost Update
 - ข. Dirty Read
 - ค. Phantom Read
 - ง. Uncommitted Insert

เฉลย: ข

อธิบาย: Dirty Read เกิดขึ้นเมื่อ Transaction หนึ่งอ่านข้อมูลที่ Transaction อื่นยังไม่ Commit ถ้าอีก Transaction ทำ Rollback ข้อมูลที่อ่านไปอาจไม่ถูกต้อง เป็นปัญหาคลาสสิกของระบบที่ไม่มีการควบคุม Isolation อย่างเพียงพอ

-
78. คำว่า ACID ซึ่งเป็นหลักสำคัญในการออกแบบระบบจัดการฐานข้อมูล หมายถึงอะไร
- ก. Atomicity, Consistency, Isolation, Durability
 - ข. Addition, Correction, Integration, Distribution
 - ค. Alignment, Compatibility, Interface, Design
 - ง. Architecture, Configuration, Iteration, Dataflow

เฉลย: ก

อธิบาย:

- **A – Atomicity:** ทำงานทั้งหมดหรือไม่ทำเลย
- **C – Consistency:** ไม่ละเมิดกฎฐานข้อมูล
- **I – Isolation:** Transaction ไม่รบกวนกัน

- **D – Durability:** เมื่อ Commit แล้วจะไม่สูญหายแม้ระบบล่ม

.....

79. ปัญหา “Redundancy” ในการออกแบบฐานข้อมูลหมายถึงอะไร

- ก. การที่มี Foreign Key ซ้ำกันในหลายตาราง
- ข. การที่ข้อมูลเดียวกันถูกจัดเก็บไว้หลายแห่งโดยไม่จำเป็น
- ค. การที่ข้อมูลถูกออกแบบให้อ่านได้เพียงครั้งเดียว
- ง. การที่ข้อมูลไม่สามารถจัดเรียงตามลำดับเวลาได้

เฉลย: ข

อธิบาย: Redundancy คือ การเก็บข้อมูลซ้ำซ้อนกันหลายจุด เช่น ชื่อลูกค้าปรากฏทุกครั้งในรายการสั่งซื้อ ทำให้เสียพื้นที่และยากต่อการดูแลรักษาข้อมูล เช่น หากชื่อผิด ต้องแก้ทุกเรคคอร์ด แนวทางที่ถูกต้องคือแยกข้อมูลไว้ต่างหากแล้วใช้ความสัมพันธ์เชื่อมโยงกัน

.....

80. การที่ฐานข้อมูลรองรับการทำธุรกรรมแบบหลายคนพร้อมกันโดยไม่มีข้อมูลขัดแย้ง เรียกว่าการรองรับลักษณะใด

- ก. Data Distribution
- ข. Data Concurrency Control
- ค. Data Fragmentation
- ง. Data Aggregation

เฉลย: ข

อธิบาย: การที่หลายคนสามารถทำงานกับฐานข้อมูลได้ในเวลาเดียวกันอย่างปลอดภัย เรียกว่า **Concurrency Control** เช่นการ Lock ตาราง, Row-Level Locking, ใช้ Isolation Level เพื่อควบคุมไม่ให้ Transaction แต่ละชุดรบกวนกันจนเกิดปัญหาเช่น Lost Update หรือ Dirty Read

.....

81. หากมีการออกแบบฐานข้อมูลให้รองรับระบบสต็อกสินค้า และต้องการให้ระบบสามารถตรวจสอบได้ว่าการนำสินค้าชนิดหนึ่งออกจากคลังมากกว่าที่มีอยู่หรือไม่ ควรใช้แนวทางใดต่อไปนี้เป็นวิธีป้องกันปัญหาได้ดีที่สุด

- ก. ใช้ SQL View เพื่อกรองเฉพาะสินค้าที่เหลืออยู่
- ข. ใช้ Trigger ในการตรวจสอบจำนวนคงเหลือก่อนอนุญาตให้หักสต็อก
- ค. ใช้ Foreign Key เชื่อมโยงระหว่างตารางสินค้าและตารางรายการเบิก
- ง. ใช้การทำ Index แบบ UNIQUE กับจำนวนสินค้าที่คงเหลือ

เฉลย: ข

อธิบาย: การใช้ **Trigger** คือแนวทางที่ดีที่สุดกรณีนี้ โดยสามารถกำหนดให้ Trigger ทำงานแบบ BEFORE INSERT หรือ BEFORE UPDATE ที่ตารางการเบิกสินค้า ซึ่งจะตรวจสอบว่าจำนวนสินค้าที่ต้องการเบิกมากกว่าจำนวนคงเหลือหรือไม่ หากมากกว่าจะยกเลิกการเบิกทันที ป้องกันปัญหาสินค้าติดลบ

.....

82. ในการวิเคราะห์ความสัมพันธ์แบบ N:M (Many-to-Many) เช่น นักเรียนกับรายวิชา ควรออกแบบฐานข้อมูลโดยใช้เทคนิคใด
- ก. ใช้การรวมตารางทั้งสองเข้าด้วยกันให้เป็นตารางเดียว
 - ข. ใช้ Trigger สร้างความสัมพันธ์โดยอัตโนมัติ
 - ค. สร้างตารางกลางที่มี Foreign Key ชี้กลับไปยังทั้งสองตาราง
 - ง. ใช้ฟิลด์ TEXT เก็บชื่อหลาย ๆ วิชาในตารางนักเรียน

เฉลย: ค

อธิบาย: ความสัมพันธ์แบบ Many-to-Many (N:M) ไม่สามารถกำหนดได้โดยตรงใน RDBMS จำเป็นต้อง แยกออกมาเป็นตารางกลาง (Join Table) ซึ่งมี Foreign Key ที่เชื่อมโยงกับทั้งตารางนักเรียนและตารางวิชา เช่น ตาราง Enroll ที่มี StudentID, CourseID เพื่อให้ระบบสามารถจัดการความสัมพันธ์หลายต่อหลายได้อย่างถูกต้อง

83. ถ้าในระบบฐานข้อมูลมีการอ้างอิง Foreign Key ที่ไม่ได้เปิดใช้งานการตรวจสอบความสมบูรณ์ของข้อมูล จะก่อให้เกิดผลเสียเช่นใด
- ก. ข้อมูลจะถูกจัดเก็บได้เร็วขึ้นเพราะไม่ต้องตรวจสอบความสัมพันธ์
 - ข. สามารถเก็บข้อมูลได้ยืดหยุ่นมากขึ้นแม้ไม่ถูกต้องตามโครงสร้าง
 - ค. อาจเกิดข้อมูลลูกที่ไม่มีพ่อแม่หรือข้อมูลขาดความสัมพันธ์ (Orphan Record)
 - ง. ไม่มีผลใด ๆ เพราะ Foreign Key เป็นเพียงคำสั่งทางตรรกะที่ไม่บังคับ

เฉลย: ค

อธิบาย: หากไม่มีการเปิดใช้ Referential Integrity การเก็บข้อมูลแบบ Foreign Key จะไม่ถูกตรวจสอบจริง ๆ ทำให้เกิด ข้อมูลกำพร้า (Orphaned Record) เช่น รายการสั่งซื้อที่อ้างอิงไปยังลูกค้าที่ไม่มีอยู่จริงในระบบ เป็นความเสี่ยงร้ายแรงในการวิเคราะห์หรือประมวลผลข้อมูล

84. แบบจำลองข้อมูล (Data Model) ชนิดใดที่เป็นพื้นฐานสำคัญของฐานข้อมูลเชิงสัมพันธ์
- ก. Entity-Relationship Model ซึ่งแสดงความสัมพันธ์ของ Entity ด้วย Cardinality
 - ข. Hierarchical Model ที่เชื่อมโยงแบบโครงสร้างต้นไม้
 - ค. Network Model ที่สามารถมีหลายเส้นทางในการอ้างอิงข้อมูล
 - ง. Object-Oriented Model ที่รวมคุณลักษณะการเขียนโปรแกรมเชิงวัตถุเข้าไปด้วย

เฉลย: ก

อธิบาย: แบบจำลองที่ใช้วางรากฐานในการออกแบบฐานข้อมูลเชิงสัมพันธ์คือ Entity-Relationship Model (ER Model) ซึ่งช่วยให้ผู้ออกแบบสามารถกำหนด Entity, Attribute, และความสัมพันธ์ (Relationship) ได้ชัดเจนก่อนจะแปลงเป็นตารางในระบบฐานข้อมูลเชิงสัมพันธ์ (RDBMS) เช่น MySQL, PostgreSQL

85. การเลือกประเภทของข้อมูล (Data Type) ให้เหมาะสมกับชนิดของข้อมูลมีผลต่อสิ่งใดมากที่สุด
- ก. ทำให้คำสั่ง SQL อ่านง่ายขึ้นและสวยงาม
 - ข. ป้องกันความผิดพลาดในการป้อนข้อมูลและประหยัดพื้นที่จัดเก็บ
 - ค. ทำให้ฐานข้อมูลมีขนาดใหญ่ขึ้นและเสถียร
 - ง. ช่วยให้ระบบสามารถรองรับการค้นหาแบบ Full-Text Search ได้ทุกคอลัมน์

เฉลย: ข

อธิบาย: การเลือกประเภทข้อมูลที่เหมาะสม เช่น ใช้ INT แทน VARCHAR สำหรับข้อมูลตัวเลข หรือใช้ DATE แทน TEXT สำหรับวันที่ จะช่วยป้องกันการป้อนค่าผิดประเภทและลดขนาดของฐานข้อมูลลง นอกจากนี้ยังช่วยให้การ Query เร็วขึ้นและระบบจัดการดัชนีได้อย่างมีประสิทธิภาพ

.....

86. หากมีความจำเป็นต้องเก็บไฟล์ภาพหรือไฟล์ PDF ลงในระบบฐานข้อมูล ควรออกแบบอย่างไรให้มีประสิทธิภาพสูงสุด
- ก. เก็บไฟล์ไว้ในฟิลด์ชนิด BLOB โดยตรงในฐานข้อมูล
 - ข. แปลงไฟล์ให้เป็น Base64 แล้วเก็บในคอลัมน์ TEXT
 - ค. เก็บเพียงพาธไฟล์หรือ URL ไว้ในฐานข้อมูล ส่วนไฟล์จริงเก็บไว้ในไฟล์ระบบ
 - ง. ใช้ฐานข้อมูลแบบ NoSQL แทน RDBMS

เฉลย: ค

อธิบาย: แนวทางที่มีประสิทธิภาพสูงสุดในการจัดเก็บไฟล์ขนาดใหญ่ เช่น PDF, รูปภาพ คือ เก็บไว้ที่ File System หรือ Cloud Storage แล้วเก็บเพียง Path หรือ URL ไว้ในฐานข้อมูลเท่านั้น วิธีนี้ช่วยลดภาระของ RDBMS และทำให้ระบบทำงานเร็วขึ้นโดยไม่ต้องโหลดไฟล์ใหญ่เข้าสู่หน่วยความจำทุกครั้ง

.....

87. คำสั่ง SQL ใดที่เหมาะสมในการสร้างตารางที่มี Primary Key และ Foreign Key
- ก. CREATE TABLE พร้อมระบุคีย์ภายนอกและภายในในคำสั่งเดียว
 - ข. INSERT INTO แล้วค่อยกำหนดคีย์ทีหลัง
 - ค. ALTER TABLE เพิ่ม Primary Key ภายหลังจากการสร้าง
 - ง. DROP TABLE แล้วสร้างใหม่พร้อมคีย์

เฉลย: ก

อธิบาย: การใช้คำสั่ง CREATE TABLE พร้อมระบุ PRIMARY KEY และ FOREIGN KEY ภายในคำสั่งเดียว เป็นวิธีที่เหมาะสมที่สุด ช่วยให้ความสัมพันธ์ทางตรรกะถูกกำหนดตั้งแต่แรก ลดโอกาสผิดพลาด และฐานข้อมูลจะตรวจสอบความถูกต้องตั้งแต่ตอนเริ่มต้น เช่น:

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    CustomerID INT,  
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) );
```

.....

88. คำว่า “Third Normal Form (3NF)” มีหลักเกณฑ์การออกแบบอย่างไร

- ก. ทุกคอลัมน์ต้องไม่ซ้ำกันในทุกแถว
- ข. ข้อมูลต้องไม่มีการขึ้นอยู่แบบ Transitive Dependency
- ค. คอลัมน์ที่ไม่ใช่คีย์ ต้องขึ้นกับคีย์บางส่วน
- ง. ตารางต้องมี Composite Key อย่างน้อยหนึ่งชุด

เฉลย: ข

อธิบาย: 3NF (Third Normal Form) มีหลักการคือ ทุกฟิลด์ที่ไม่ใช่คีย์ ต้องขึ้นตรงกับ Primary Key เท่านั้น ไม่ขึ้นกับฟิลด์อื่นที่ไม่ใช่คีย์ กล่าวคือ ต้องกำจัดความขึ้นต่อกันแบบทางอ้อม (Transitive Dependency) เพื่อให้การเปลี่ยนแปลงข้อมูลในคอลัมน์ใด ๆ ไม่มีผลต่อคอลัมน์อื่นโดยไม่จำเป็น

.....

89. การใช้ Index บนคอลัมน์ในฐานข้อมูลส่งผลต่อระบบอย่างไร

- ก. ช่วยลดขนาดของฐานข้อมูล
- ข. เพิ่มความเร็วในการ Insert ข้อมูล
- ค. เพิ่มความเร็วในการ Query ข้อมูลโดยเฉพาะคำสั่ง SELECT
- ง. ช่วยให้ข้อมูลสามารถเรียงลำดับอัตโนมัติได้ตลอดเวลา

เฉลย: ค

อธิบาย: การสร้าง Index ทำให้การค้นหาข้อมูลด้วย SELECT ทำได้รวดเร็วขึ้นมาก โดยเฉพาะเมื่อตารางมีข้อมูลจำนวนมาก เช่น ค้นหาด้วย WHERE, JOIN, ORDER BY อย่างไรก็ตาม Index จะทำให้การ INSERT, UPDATE ช้าลงเล็กน้อย เพราะต้องอัปเดต Index ไปพร้อมกัน

.....

90. การออกแบบฐานข้อมูลโดยไม่ใช้ Normalization อาจก่อให้เกิดผลเสียในลักษณะใด

- ก. ทำให้ระบบใช้งานง่ายขึ้นแต่ช้าลง
- ข. ทำให้เกิดการซ้ำซ้อนของข้อมูลและยากต่อการดูแล
- ค. ช่วยให้เขียน SQL ได้ยืดหยุ่นมากขึ้น
- ง. ไม่มีผลกระทบเพราะระบบฐานข้อมูลจะจัดการให้อัตโนมัติ

เฉลย: ข

อธิบาย: การไม่ทำ Normalization หรือออกแบบตารางโดยไม่แยกข้อมูลให้ถูกต้องตามหลัก 1NF, 2NF, 3NF จะก่อให้เกิด การซ้ำซ้อนของข้อมูล (Redundancy) และ ความไม่สอดคล้อง (Inconsistency) เช่น ชื่อลูกค้าถูกกรอกผิดในบางรายการ หรือข้อมูลล้นจนยากต่อการจัดการระยะยาว

.....

91. ในการกำหนดรหัสพนักงานให้ไม่ซ้ำกัน และสามารถตรวจสอบได้ว่าเป็นค่าเฉพาะของแต่ละบุคคลในระบบฐานข้อมูล ควรใช้แนวทางใดเพื่อควบคุมได้ดีที่สุด

© THE BEST CENTER INTER GROUP CO., LTD.

All rights reserved

ห้ามผู้ใดทำการคัดลอก ตีพิมพ์ แจกจ่าย ปรับเปลี่ยน
ดัดแปลง หรือแก้ไขส่วนใดส่วนหนึ่งของหนังสือเล่มนี้
เพื่อการเผยแพร่หรือนำไปใช้ในเชิงพาณิชย์โดยเด็ดขาด

หากตรวจพบจะดำเนินการตามกฎหมายถึงที่สุด

หากผู้ใดพบเห็น สามารถแจ้งเบาะแสที่
081-496-9907 มีรางวัลตอบแทน