

ปฏิบัติการที่ 2 สร้าง ER-Diagram

จุดประสงค์การปฏิบัติการ

- ☐ นักศึกษาสามารถสร้าง ER Diagram จากโปรแกรมได้
- ☐ นักศึกษาสามารถออกแบบฐานข้อมูลแบบง่ายได้
- ☐ สร้าง ER Diagram จากตารางที่มีอยู่
- ☐ สามารถเขียนคำสั่งกระทำกับตารางได้

แบบทดสอบ 2.1 (บรรจุ PI9.1)

- ☐ ให้นักศึกษาทำการออกแบบ ER-Diagram ตามเงื่อนไขดังต่อไปนี้ ทำการสร้าง ER Diagram ออกมาและสร้างเป็นตารางที่ถูกต้อง
 - ☐ จากฐานข้อมูลโรงภาพยนตร์ (Cinema) มีการเก็บข้อมูลของรายการหนังทั้งหมดที่ฉายอยู่ (OnAir_Movie) ซึ่งประกอบด้วยรหัสหนัง (MovieID) ชื่อหนัง (MovieName) ประเภทหนัง (MovieGenre) ความยาวหนัง (MovieDuration)
 - ☐ ข้อมูลโรงหนัง(Theater) ประกอบด้วย หมายเลขโรงหนัง(TheaterNumber) ความจุ(Capacity)
 - ☐ เก็บข้อมูลรอบหนัง (Schedule) มีรหัสรอบหนัง(ScheduleID) วันที่เวลาที่ฉาย (ScheduleDateTime) โรงที่ฉาย(TheaterNumber) ซึ่งมีความสัมพันธ์แบบหนึ่งต่อหลายกับหนังที่ฉาย และมีความสัมพันธ์แบบหนึ่งต่อหลายกับโรงหนัง
 - ☐ ข้อมูลของตั๋วหนัง(Ticket) ประกอบด้วย รหัสตั๋วหนัง(TicketID) เลขที่นั่ง(SeatNo) ราคา(Price) มีความสัมพันธ์กับข้อมูลรอบหนังแบบหลายต่อหนึ่ง
- ☐ แสดงการใช้งาน Forward และ Reverse Engineer

แบบทดสอบ 2.2 (บรรจุ PI9.1)

- ☐ จงเขียนคำสั่ง CREATE สร้างตาราง ชื่อ Doctor ซึ่งประกอบด้วย doctor_id, doctor_name, doctor_gender, ward (กำหนด Data type ให้เหมาะสม)
- ☐ จงเขียนคำสั่ง CREATE สร้างตาราง ชื่อ Patient ซึ่งประกอบด้วย patient_id, patient_name, doctor_id (กำหนด Data type ให้เหมาะสม) doctor_id เป็น Foreign Key จากตาราง Doctor

- ☐ doctor_id จากตาราง Doctor และ patient_id จากตาราง Patient เป็น Primary Key

แบบทดสอบ 2.3 (บรรลุ PI9.1)

- ☐ จงเขียนคำสั่งสร้างตาราง patientType ซึ่งประกอบด้วย id (ยังไม่ต้องกำหนด Primary key)
- ☐ จงสร้าง Column ชื่อ type_name เพิ่มในตาราง
- ☐ จงเปลี่ยนชื่อจาก id เป็น type_id
- ☐ ทำการแก้ไขตารางโดยเพิ่ม Primary key ที่ type_id
- ☐ เชื่อมตารางด้วย Foreign Key โดย
 - เพิ่ม Column ชื่อ tid ที่ตาราง patient
 - เชื่อม Foreign Key ระหว่างตาราง patient ที่ tid และ patientType ที่ type_id
- ☐ สร้างตาราง Disease ประกอบด้วย id, name, level
- ☐ ลบตาราง Disease ที่