SC 612 104 Essential Data Science — Coding Challenge

กลุ่มที่ 7 : โรงแรม (Hotel) — ระบบจองห้องพัก

ขั้นที่ 1 — เข้าใจองค์กรของกลุ่มก่อนเขียนโค้ด

คำถามที่ 1 : องค์กรนี้มี “สิ่งของหลัก” อะไรบ้างที่ต้องเก็บข้อมูล?

ธุรกิจโรงแรมมีสิ่งของหลัก 4 อย่างที่ต้องเก็บข้อมูล ดังนี้

1) ห้องพัก (Room) — ห้อง 1 ห้องในโรงแรม ต้องจำได้เองว่าวันไหนถูกจองไปแล้ว ไม่งั้นจะจองซ้อนกัน

2) ลูกค้า (Guest) — ลูกค้า 1 คน ลูกค้าคนเดิมกลับมาจองได้หลายครั้ง
จึงต้องแยกออกมาเพื่อสะสมสิทธิ์สมาชิก

3) ใบจอง (Booking) — การจอง 1 ใบ เป็นตัวเชื่อมว่าลูกค้าคนไหน จองห้องไหน วันไหน ราคาเท่าไร

4) โรงแรม (Hotel) — ตัวโรงแรมทั้งระบบ ถือห้องและลูกค้าทั้งหมด
และเป็นคนตัดสินใจว่าจะรับจองหรือปฏิเสธ

ในธุรกิจนี้ “ใบจอง” คือสิ่งหลัก เทียบได้กับ “ออเดอร์ซักผ้า” ของร้านซักรีด
ส่วนห้องพักและลูกค้าเป็นสิ่งของที่ใบจองไปอ้างอิงถึง จึงต้องแยกเก็บคนละตาราง

คำถามที่ 2 : แต่ละสิ่งของหลักนั้น มี “คุณสมบัติ” (ข้อมูล) อะไรบ้าง?

ห้องพัก (Room)

room_id, room_number (เลขห้อง เช่น 101), room_type (Standard / Deluxe / Suite / Family), base_price (ราคาต่อคืนก่อนบวกฤดูกาล), capacity (พักได้กี่คน), floor (ชั้น), booked_dates (เซ็ตของวันที่ถูกจองไปแล้ว)

ลูกค้า (Guest)

guest_id, name, phone, nationality (สัญชาติ 12 ชาติ), member_tier (Regular / Silver / Gold), stay_count (พักมาแล้วกี่ครั้ง)

ใบจอง (Booking)

booking_id, guest (ลูกค้าที่จอง), room (ห้องที่ได้), check_in, nights, check_out, channel (OTA / Website / Walk-in / Phone), booking_date, lead_time_days (จองล่วงหน้ากี่วัน), breakfast,

upgraded (ถูกอัปเกรดห้องฟรีหรือไม่), adults, children, status (CONFIRMED / CANCELLED / NO_SHOW / CHECKED_OUT), price_detail (รายละเอียดราคาทั้งใบ)

โรงแรม (Hotel)

name, rooms (ห้องทั้งหมด 15 ห้อง), guests (ลูกค้าทั้งหมดในระบบ), bookings (ใบจองที่สำเร็จ), rejected_log (รายการที่จองไม่ได้เพราะห้องเต็ม)

คำถามที่ 3 : แต่ละสิ่งของหลักนั้น "ทำอะไรได้" บ้าง?

ห้องพัก (Room) ทำได้

- is_available() — ตรวจว่าตัวเองว่างตลอดช่วงวันที่ขอมาหรือไม่ เจอวันชนแม้แต่วันเดียวก็จองไม่ได้
- reserve() — ยึดวันทั้งช่วงไว้ หลังยืนยันการจองแล้ว
- release() — คืนวันว่างกลับเข้าระบบ ใช้ตอนลูกค้ายกเลิก
- price_per_night() — คิดราคาต่อคืนของตัวเองตามฤดูกาล
- nights_sold() — บอกว่าตัวเองขายได้กี่คืนแล้ว

ลูกค้า (Guest) ทำได้

- record_stay() — บันทึกว่าพักเพิ่มอีก 1 ครั้ง
- upgrade_tier() — เลื่อนขั้นสมาชิกอัตโนมัติเมื่อพักครบเกณฑ์
- discount_rate() — บอกว่าตัวเองได้ส่วนลดกี่เปอร์เซ็นต์
- is_vip() — บอกว่าตัวเองเป็น VIP (ระดับ Gold) หรือยัง

ใบจอง (Booking) ทำได้

- calculate_price() — คำนวณราคาทั้งใบ แยกให้เห็นทุกองค์ประกอบ
- cancel() — ยกเลิก แล้วสั่งให้ห้องคืนวันว่าง
- mark_no_show() — ลูกค้าไม่มาเช็คอินโดยไม่แจ้งล่วงหน้า
- check_out_guest() — เช็คเอาท์เรียบร้อย
- is_revenue() — บอกว่าใบนี้นับเป็นรายได้จริงไหม
- to_dict() — แปลงตัวเองเป็นแถวข้อมูลสำหรับบันทึกลง CSV

โรงแรม (Hotel) ทำได้

- find_available_room() — หาห้องว่างประเภทที่ลูกค้าขอ
- create_booking() — ตัดสินใจว่าจะรับจองหรือปฏิเสธ
- occupancy_on() — ดูอัตราการเข้าพักของวันหนึ่ง ๆ เพื่อใช้ปรับราคา
- occupancy_rate() / adr() / revpar() — คำนวณตัวชี้วัดมาตรฐานของธุรกิจโรงแรม

คำถามที่ 4 : ธุรกิจนี้มีการ “ตัดสินใจ” หรือ “คำนวณ” อะไรที่ซับซ้อนกว่าการบวกเลขตรง ๆ บ้าง?

ธุรกิจโรงแรมมีจุดที่ซับซ้อนกว่าการคูณราคากับจำนวนคืน อยู่ 6 จุด

1. ตรวจไม่ให้จองซ้อนวัน

ห้องเดียวกันจองซ้อนช่วงวันเดียวกันไม่ได้ ต้องไล่เช็คทีละวันตลอดช่วงที่ขอมา
เจอวันชนแม้แต่วันเดียวก็จองไม่ได้

2. อัปเกรดห้องฟรีอัตโนมัติ

ถ้าห้องประเภทที่ลูกค้าขอเต็ม ระบบจะลองหาห้องที่สูงกว่าให้ฟรีก่อน (Standard เป็น Deluxe เป็น Suite)
ถ้ายังไม่ได้จึงปฏิเสธและบันทึกไว้ใน rejected_log

3. ราคาต่อคืนไม่คงที่ (Dynamic Pricing)

ราคาต่อคืน = ราคาฐาน x ตัวคูณฤดูกาล x ตัวคูณดีมานด์

ตัวคูณฤดูกาล : High season 1.25 / Low season 0.90 / ปกติ 1.00

ตัวคูณดีมานด์ : ห้องเต็มเกิน 85% เป็น 1.20 / เกิน 70% เป็น 1.10 / ต่ำกว่า 40% เป็น 0.92

ห้องเดียวกัน คืนเดียวกัน จึงมีราคาต่างกันได้ตามวันที่จองและความหนาแน่นของวันนั้น

4. ส่วนลดสมาชิกที่เลื่อนขั้นเองอัตโนมัติ

พักครบ 3 ครั้ง เลื่อนเป็น Silver ลด 5% และพักครบ 6 ครั้ง เลื่อนเป็น Gold ลด 10%
ลูกค้าคนเดิมจึงจ่ายไม่เท่ากันในแต่ละครั้งที่กลับมาพัก

5. โครงสร้างราคาเป็นชั้น ๆ ต้องคิดตามลำดับ

ค่าห้อง = ราคาต่อคืน x จำนวนคืน

+ ค่าอาหารเช้า = 350 x คืน x ผู้ใหญ่ (ถ้าซื้อ)

= ยอดรวมก่อนลด

- ส่วนลดสมาชิก

+ ค่าบริการ 10%

+ VAT 7% <- คิดจาก (ยอดหลังลด + ค่าบริการ)

= ยอดที่ลูกค้าจ่าย

- ค่าคอมมิชชันช่องทาง

= รายได้สุทธิที่โรงแรมได้จริง

จุดที่พลาดง่ายคือ VAT ต้องคิดจากยอดหลังบวกค่าบริการแล้ว ไม่ใช่คิดจากยอดตั้งต้น

6. ค่าคอมมิชชันต่างกันตามช่องทาง

OTA 15% / Website 2% / Walk-in และ Phone 0% ยอดขายเท่ากันแต่คนละช่องทาง

โรงแรมได้เงินจริงไม่เท่ากัน เป็นที่มาของข้อเสนอเชิงธุรกิจข้อสำคัญของกลุ่ม

สรุปสิ่งที่ได้จากขั้นที่ 1

จากคำตอบทั้ง 4 ข้อ กลุ่มนำไปออกแบบเป็นโครงสร้างโปรแกรมได้ดังนี้

- คำตอบข้อ 1 (สิ่งของหลัก) กลายเป็น class Room, class Guest, class Booking และ class Hotel

- คำตอบข้อ 2 (คุณสมบัติ) กลายเป็น attribute ใน __init__ ของแต่ละ class และคอลัมน์ในตาราง CSV กับ SQLite

- คำตอบข้อ 3 (ทำอะไรได้) กลายเป็น method ของแต่ละ class

- คำตอบข้อ 4 (การคำนวณซับซ้อน) กลายเป็นฟังก์ชันช่วยงาน เช่น seasonal_multiplier(), dynamic_rate_multiplier() และ calculate_price()

ผลลัพธ์ที่ได้จริงหลังเขียนโค้ดตามการออกแบบนี้ คือ ใบจอง 960 รายการ ลูกค้า 748 คน ห้องพัก 15 ห้อง อัตราการเข้าพัก 76.0% ADR 2,326 บาท และ RevPAR 1,768 บาท

ซึ่งอยู่ในช่วงเดียวกับตลาดโรงแรมกรุงเทพจริงปี 2025 (อัตราเข้าพัก 75.1% และ ADR 1,819–4,260 บาท)