06026212 DATA WAREHOUSING (การสร้างคลังข้อมูล)

3 -Requirement Gathering for DW

หัวข้อ: Subject Area, KPI, Fact vs Dimension, Bus Matrix

ผู้สอน: อ.เฉลิมพล ศิริกายน

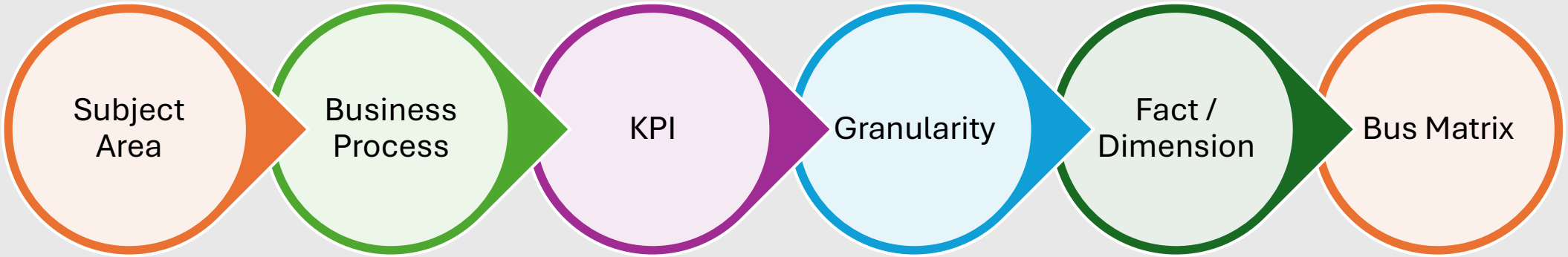
ภาคเรียนที่ 1 ปีการศึกษา 2569

วัตถุประสงค์การเรียนรู้

- อธิบายความหมายของ Subject Area ได้อย่างถูกต้อง
- ระบุและวิเคราะห์ Business Process ที่เกี่ยวข้องในแต่ละ Subject Area ได้
- อธิบายแนวคิดของ KPI (Key Performance Indicator) และความสำคัญต่อการออกแบบ Data Warehouse
- อธิบาย Granularity และเลือกระดับความละเอียดของข้อมูลที่เหมาะสมกับ KPI และ Business Process
- แยกประเภทของข้อมูลเป็น Fact และ Dimension ได้อย่างเหมาะสม

Requirements Gathering

- การวิเคราะห์ความต้องการข้อมูล
- จุดเริ่มต้นของการออกแบบ Data Warehouse



Subject Area

- กลุ่มหัวข้อข้อมูลที่ธุรกิจต้องการเก็บ วิเคราะห์ หรือรายงาน
- คิดแบบ “หมวดหมู่ข้อมูลใหญ่”
- เป็นสิ่งที่องค์กรใช้ในการตัดสินใจเชิงกลยุทธ์

ตัวอย่าง Subject Area (ร้านชานมไข่มุก)

Subject Area (ตามกระบวนการธุรกิจ)	คำอธิบาย
1. การขายและการชำระเงิน (Sales & Payment)	รายการขาย, ยอดขายต่อวัน
2. การจัดการสินค้าและวัตถุดิบ (Inventory & Menu)	เมนู, วัตถุดิบ, ราคา
3. ระบบสมาชิกและการตลาด (Loyalty & Marketing)	ประวัติ, การสะสมแต้ม
4. การบริหารงานบุคคลสาขา (Store HR/Shift)	เวลากะ, ตารางงาน
5. การดำเนินงานสาขา (Store Operations)	ข้อมูลสาขา, ยอดขายแยกสาขา

วิธีหา Subject Area

- สัมภาษณ์ผู้ใช้ (User Interviews)
- ดูรายงานเดิม / Dashboard ที่มีอยู่แล้ว
- พิจารณาโมดูลในระบบ OLTP เดิม
- ดูเอกสาร / คู่มือกระบวนการทางธุรกิจ

กิจกรรมในห้องเรียน

ถ้าคุณเป็นเจ้าของกิจการร้านถ่ายเอกสาร

คุณอยากรู้อะไรจากข้อมูลที่มีอยู่

Business Process

- กิจกรรมที่ก่อให้เกิดข้อมูลในแต่ละ Subject Area
- มักอยู่ในรูปของคำกริยา เช่น สั่งซื้อ, ชำระเงิน, สมัครสมาชิก
- ช่วยให้เราเข้าใจว่าข้อมูลเกิดขึ้นเมื่อไร และในลักษณะใด

ตัวอย่างเชื่อมโยง SA ⟶ BP (ร้านชานมไข่มุก)

Subject Area	Business Process (เหตุการณ์ที่เกิดขึ้น)
1. การขายและการชำระเงิน	เปิดออร์เดอร์, ชำระเงิน, ยกเลิกออร์เดอร์, คืนเงิน, ปิดยอดขายประจำวัน
2. การจัดการสินค้าและวัตถุดิบ	เพิ่ม/แก้ไขเมนูใหม่, ปรับเปลี่ยนราคาสินค้า, ตรวจนับวัตถุดิบ, รับเข้าวัตถุดิบ
3. ระบบสมาชิกและการตลาด	สมัครสมาชิก, สะสมแต้ม, Redeem แต้ม, การใช้สิทธิ์โปรโมชั่น/คูปอง
4. การบริหารงานบุคคลสาขา	Clock-in / Clock-out, การเข้ากะพนักงาน, การจัดตารางงาน
5. การดำเนินงานสาขา	เปิดร้าน-ปิดร้าน, การส่งยอดเงินสดประจำกะ

ความสำคัญของ Business Process

- เชื่อมโยงความเข้าใจระหว่าง “สิ่งที่อยากรู้” กับ “เหตุการณ์ที่ก่อให้เกิดข้อมูล”
- เป็นรากฐานในการกำหนด Granularity และวิเคราะห์ KPI

กิจกรรมในห้องเรียน

- "เลือก 1 Subject Area แล้วระบุ Business Process อย่างน้อย 3 รายการ"
- เขียนคำตอบของแต่ละกลุ่ม ⟶ อาจารย์วิจารณ์คำตอบ ⟶ เน้นว่าต้องเป็น “กิจกรรม” ที่ก่อให้เกิดข้อมูลจริง

KPI

- KPI (Key Performance Indicator) คือ ตัวเลขที่ใช้วัดความสำเร็จของธุรกิจ
- ใช้ประเมินว่า Business Process ที่เกิดขึ้นใน Subject Area นั้น ๆ มีประสิทธิภาพหรือไม่

ตัวอย่าง KPI (ร้านชานมไข่มุก)

Subject Area	Business Process	KPI
1. การขายและการชำระเงิน	ชำระเงิน	ยอดขายรวม, ยอดซื้อต่อบิล
2. การจัดการสินค้าและวัตถุดิบ	ตรวจนับวัตถุดิบ	ความคลาดเคลื่อนยอดสต็อกในระบบ
3. ระบบสมาชิกและการตลาด	สะสมแต้ม	ยอดขายจากสมาชิก
4. การบริหารงานบุคคลสาขา	Clock-in / Clock-out	สาย-ขาด-ลา
5. การดำเนินงานสาขา	การส่งยอดเงินสดประจำกะ	เงินขาด/เงินเกิน

หลักการเลือก KPI ที่ดี

- ต้อง **วัดได้** เป็นตัวเลขจริง
- ต้อง **ชัดเจน** ไม่คลุมเครือ
- ต้อง **สอดคล้องกับเป้าหมายของธุรกิจ**
- ต้องสามารถ **วัดซ้ำได้** (รายวัน/รายเดือน)

KPI ⟶ Fact ใน Data Warehouse

- “KPI จะกลายเป็นตัวกำหนดว่าเราควรเก็บข้อมูลตัวไหนเป็น Fact”
- KPI: ยอดขายรายวัน ⟶ Fact: amount, date
- KPI: ลูกค้าใหม่ ⟶ Fact: customer_id, signup_date

ปัญหาเมื่อไม่มีมาตรฐาน KPI

- “สูตรไม่ตรงกัน = รายงานไม่ตรงกัน = ความสับสน”
- ทีม A: เขียน SUM(amount) ใน Excel
- ทีม B: เขียน SUM(order_total) ใน Power BI
- ทีม C: เขียน SQL ด้วยชื่อฟิลด์อื่น ⟶ ค่าต่างกันหมด

แนวคิด Metric Layer

- "จัดการสูตร KPI ทั้งหมดไว้ในที่เดียว"
- เป็น ‘ชั้นกลาง’ ที่นิยามสูตร KPI อย่างเป็นทางการ
- ระบบ Dashboard หรือ BI ทุกตัวจะดึงสูตรจากที่เดียวกัน

ตัวอย่าง Metric Layer (dbt / LookML)

```yaml
metric:
  name: daily_revenue
  label: Daily Revenue
  model: fact_sales
  calculation_method: sum
  expression: amount
  time_grain: day
```

- ใช้ daily_revenue ได้ในทุกระบบ ไม่ต้องเขียนสูตรใหม่

ประโยชน์ของ Metric Layer

- ทุกทีมใช้สูตรเดียวกัน ⟶ ลดความสับสน
- เปลี่ยนสูตร KPI ที่เดียว ทุกระบบอัปเดตอัตโนมัติ
- รองรับแนวคิด Data Governance

Granularity

- “ระดับความละเอียดของข้อมูลในแต่ละ Business Process”
- คือระดับ “หนึ่งแถวต่อ...” เช่น:
- 1 แถวต่อออร์เดอร์
- 1 แถวต่อรายการในออร์เดอร์
- 1 แถวต่อวัน
- ส่งผลโดยตรงต่อการออกแบบ Fact Table

ตัวอย่าง Granularity (ร้านชานมไข่มุก)

Business Process	Granularity
ชำระเงิน	รายการสินค้าแต่ละแถวบนใบเสร็จ (Line Item) Ex: ใบเสร็จเลขที่ A001 แถวที่ 1 ขาย "ชาไทยไข่มุก" จำนวน 1 แก้ว เวลา 12.05 น.
ตรวจนับวัตถุดิบ	รายการวัตถุดิบแต่ละชนิด ต่อการนับ 1 รอบ ของแต่ละสาขา Ex: สาขาลาดพร้าว นับ "นมสด" ได้ 15 กล่อง ประจำวันที่ 7 ก.ค. 2026 (เทียบกับค่าในระบบ)
สะสมแต้ม	รายการใช้แต้มหรือสะสมแต้มต่อครั้งของสมาชิก (Per Loyalty Transaction) Ex: สมาชิก M08 ใช้แต้ม 50 แต้ม แลกส่วนลด 50 บาท ตอนซื้อกาแฟเย็น วันนี้
Clock-in / Clock-out	การสับเปลี่ยนกะหรือการกดบันทึกเวลาเข้า-ออก 1 ครั้ง ของพนักงาน 1 คน Ex: พนักงาน A สแกนนิ้วเข้างาน (Clock-in) กะเช้า เวลา 07.45 น. ของวันที่ 7 ก.ค. 2026
การส่งยอดเงินสดประจำกะ	การสรุปปิดลิ้นชักเก็บเงิน 1 รอบตอนเปลี่ยนกะ (Per Shift Summary) Ex: เครื่อง POS 1 ปิดกะบ่าย เงินในเครื่องขาดไป 20 บาท ผู้รับผิดชอบ พนักงาน B

ทำไม Granularity ถึงสำคัญ?

- กำหนดระดับการวิเคราะห์ (ระดับรายละเอียด)
- ส่งผลต่อขนาดของตาราง (แถวมากหรือน้อย)
- ส่งผลต่อ Index และ Performance
- มีผลต่อ KPI ที่คำนวณได้จากตารางนั้น

จาก Granularity ⟶ Fact

- “Granularity เป็นตัวกำหนดว่า 1 แถวใน Fact Table จะหมายถึงอะไร”
- เช่น: Granularity = 1 แถวต่อออร์เดอร์ ⟶ Fact Table: fact_order
- เช่น: Granularity = 1 แถวต่อรายการสินค้าในออร์เดอร์ ⟶ fact_order_item

การแยก Fact กับ Dimension

- Fact Table = ตารางที่เก็บค่าที่วัดได้ เช่น ยอดขาย จำนวนลูกค้า เวลาเฉลี่ย
- Dimension Table = ตารางที่อธิบายมิติต่าง ๆ เช่น ลูกค้า เมนู สาขา เวลา

ตัวอย่าง Fact & Dimension (ร้านชานมไข่มุก)

Table	ประเภท	ตัวอย่างฟิลด์
fact_order	Fact	order_id, order_date, amount
dim_customer	Dimension	customer_id, name, gender, birth_date
dim_product	Dimension	product_id, name, category
dim_time	Dimension	date, day_of_week, is_holiday

หลักการแยก Fact / Dimension

- สิ่งที่ **เปลี่ยนบ่อยและคำนวณได้** ⟶ Fact
- สิ่งที่ **ใช้บรรยาย / แยกตามมิติ** ⟶ Dimension
- ใช้ Granularity เป็นจุดตัดในการตัดสินใจแยกตาราง

Workshop (ในห้องเรียน)

- 📌 “เลือก 1 Business Process และกำหนด Granularity + Fact + Dimension ที่เหมาะสม”
- เช่น:
- Process: เปิดออร์เดอร์
- Granularity: 1 แถวต่อออร์เดอร์
- Fact: order_id, amount, datetime
- Dimension: customer, product, time, branch

สรุป Granularity และ Fact/Dimension

- Granularity = ระดับความละเอียดของข้อมูล
- เป็นฐานของการออกแบบ Fact Table
- ช่วยให้แยก Fact กับ Dimension อย่างมีเหตุผล
- เตรียมพร้อมสำหรับการสร้าง Bus Matrix ในขั้นตอนถัดไป

Bus Matrix

- “แผนภาพตารางที่เชื่อมโยง Subject Area กับ Business Process และ KPI ที่เกี่ยวข้อง”
- มักใช้เพื่อเตรียมการออกแบบ Data Mart และ Star Schema
- ช่วยให้มองเห็นภาพรวมของโครงสร้างข้อมูลเชิงธุรกิจ

ตัวอย่างของ Bus Matrix

Subject Area	Business Process	KPI / Fact	Dimension
การขายและการชำระเงิน	เปิดออร์เดอร์	amount order_count	customer time product branch
การบริหารงานบุคคลสาขา	ปิดยอดกะ	total_sales shift_hours	employee time branch

ประโยชน์ของ Bus Matrix

- เชื่อมโยง Business Process ⟶ KPI ⟶ Fact Table ได้ชัดเจน
- ช่วยวางแผนการออกแบบ Star Schema
- ใช้ตรวจสอบความครบถ้วนของความต้องการข้อมูล

Workshop (ในห้องเรียน)

- ✏️ "จากสิ่งที่กลุ่มคุณทำไว้: Subject Area, Process, KPI ⟶ สร้าง Bus Matrix กลุ่มของคุณ"
- 📌 เขียนลงกระดาษ หรือทำใน diagrams.net ⟶ อาจารย์ให้คำแนะนำ ⟶ ใช้ในการออกแบบ Star Schema สัปดาห์ถัดไป

สรุปภาพรวม Requirements Gathering

- ระบุ Subject Area
- กำหนด Business Process
- นิยาม KPI และ Granularity
- แยก Fact / Dimension
- สร้าง Bus Matrix

< พร้อมออกแบบ Schema สัปดาห์หน้า >