Group Assignment #1 – Mini Data Warehouse & Analytics Dashboard

ElderCare Insight

คลังข้อมูลและแดชบอร์ดสำหรับธุรกิจสถานดูแลผู้สูงอายุ จากข้อมูลเปิดของ CMS สหรัฐอเมริกา ปี 2562–2569

เอกสารนี้คือ **แผนการออกแบบระบบ** ก่อนลงมือพัฒนา ครอบคลุมตั้งแต่ปัญหาทางธุรกิจ คำถามที่ต้องตอบ แหล่งข้อมูลทั้งห้าแหล่ง ปัญหาคุณภาพข้อมูลที่ต้องรับมือ ไปจนถึงการประกาศ Grain และผัง Star Schema

รูปทุกรูปในเอกสารนี้เป็นภาพอธิบายที่วาดขึ้นเอง **ยังไม่มีรูปผลลัพธ์จริง** เพราะยังไม่ได้รัน ETL ส่วนตัวเลขขนาดข้อมูลทุกตัวมาจากการเปิดไฟล์จริงแล้ว และตรวจซ้ำได้ด้วยคำสั่งในภาคผนวก ค

เอกสารนี้เป็น *เอกสารทำงาน* จึงยาวกว่ารายงานฉบับส่งที่โจทย์จำกัดไว้ 10–15 หน้า รายงานฉบับส่งจะย่อจากภาค 1–4 ของเอกสารนี้

กลุ่มที่ 05 สมาชิก: Kitsadee Thongbankao (6710503721) , Kawinphop Junjuae (6710503739) , Pirapat Rojbadin (6710504182)

ข้อมูล: CMS Provider Data Catalog – Nursing Homes (data.cms.gov) · สถานะ: ออกแบบเสร็จ ยังไม่เริ่ม ETL

ปรับปรุงล่าสุด: 12 สิงหาคม 2569

สารบัญ

เอกสารนี้อ่านอย่างไร 4

1 ธุรกิจและคำถาม — เราจะตอบอะไร 6

1 ธุรกิจสถานดูแลผู้สูงอายุทำงานอย่างไร 6

1.1 ปัญหาทางธุรกิจที่ต้องการแก้ 7

2 ใครใช้ข้อมูลนี้ (Stakeholders) 8

3 คำถามทางธุรกิจแปดข้อ 8

4 Measures และกับดักของการรวมค่า 9

2 แหล่งข้อมูล — มาจากไหน มีอะไร ใช้ทำอะไร 11

5 ภาพรวมทั้งห้าแหล่ง 11

6 แหล่งที่ 1 — CMS Nursing Homes Archived Snapshots 11

6.1 มาจากไหน 11

6.2 มีอะไรอยู่ข้างใน 12

6.3 ใช้ทำอะไร 13

7 แหล่งที่ 2 — CMS Provider Data Catalog REST API 14

7.1 มาจากไหน 14

7.2 มีอะไร 14

7.3 ใช้ทำอะไร 15

8 แหล่งที่ 3 — SNF VBP Facility Performance 16

8.1 มาจากไหน มีอะไร 16

8.2 ใช้ทำอะไร 16

9 แหล่งเสริมอีกสองแหล่ง 16

9.1 S4 — ประชากรอายุ 65 ปีขึ้นไป รายรัฐ 16

9.2 S5 — ตาราง lookup ของ CMS 17

10 ตารางสืบย้อน — แหล่งไหนตอบคำถามไหน 17

3 คุณภาพข้อมูล — อะไรพังบ้าง 19

11 ปัญหาแปดประเภทและวิธีรับมือ 20

4 คลังข้อมูล — Grain และ Star Schema 22

12 สี่ขั้นตอนของ Kimball 22

12.1 ขั้นที่ 1 — เลือกกระบวนการทางธุรกิจ 22

13 ขั้นที่ 2 — ประกาศ Grain 22

13.1 ทำไมสองตารางนี้ถึงรวมกันไม่ได้ 23

14 ขั้นที่ 3 — เลือก Dimensions 24

14.1 SCD ชนิดที่ 2 ทำงานอย่างไร 25

15 ขั้นที่ 4 — ผัง Star Schema 26

16 ข้อควรระวังของผังนี้ 27

5 ETL และ Dashboard — แผนงานถัดไป 28

17 กระบวนการ ETL ห้าขั้น 28

18 แผนแดชบอร์ด 28

19 คะแนนพิเศษที่ตั้งเป้า 29

20 สิ่งที่ต้องทำก่อนเริ่มพัฒนา 29

ภาคผนวก 31

ภาคผนวก 31

A อภิธานศัพท์ ไทย–อังกฤษ 31

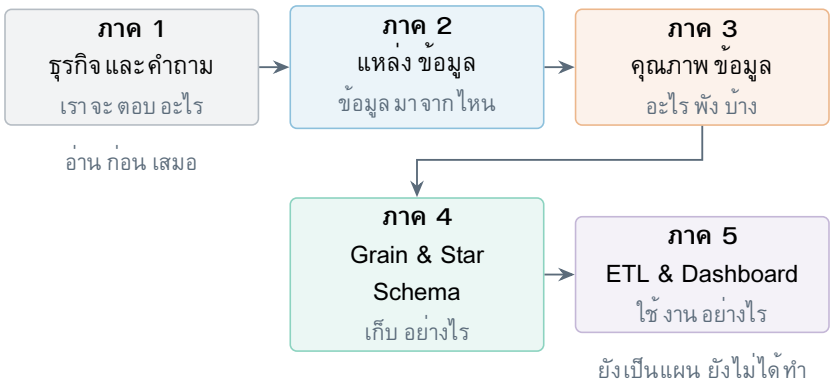
B คอลัมน์สำคัญของ NH_ProviderInfo ที่จะใช้จริง 32

C คำสั่งที่ใช้ตรวจสอบตัวเลขในเอกสารนี้ 32

D แหล่งอ้างอิง 33

เอกสารนี้อ่านอย่างไร

เอกสารแบ่งเป็นห้าภาค **เรียงตามทางเดินของข้อมูลจริง ไม่ใช่เรียงตามหัวข้อทฤษฎี** ผู้อ่านที่หลงทางสามารถถามตัวเองว่า “ตอนนี้ข้อมูลอยู่ตรงไหนของท่อ” แล้วจะรู้ทันทีว่าอยู่ภาคใด



รูปที่ 1: แผนที่ของเอกสาร ห้าภาคเรียงตามทางเดินของข้อมูล จากคำถามทางธุรกิจไปจนถึงหน้าจอที่ผู้บริหารเห็น

ระบบสีที่ใช้ตลอดเอกสาร

สีสี่สีนี้แทน *ระยะของไปป์ไลน์* และจะถูกใช้ความหมายเดิมทุกที่ ทั้งในเนื้อความ ในผัง และในกราฟของแดชบอร์ดภายหลัง ชุดสีเป็น Okabe–Ito ซึ่งปลอดภัยต่อผู้มีภาวะตาบอดสี

- แหล่งข้อมูล — ข้อมูลดิบจากภายนอกที่เรายังแก้อะไรไม่ได้
- ETL — ขั้นตอนแปลงข้อมูล จุดที่ความผิดพลาดเกิดขึ้นมากที่สุด
- คลังข้อมูล — ตาราง Fact และ Dimension ที่ออกแบบเอง
- Dashboard — สิ่งที่ผู้ใช้เห็นและตัดสินใจจากมัน

กล่องสี่แบบ และความหมายของมัน

กล่องด้านล่างนี้ *คือคำนิยามของตัวมันเอง* — เนื้อหาในกล่องบอกว่ากล่องสีนั้นใช้เมื่อไร

กล่องน้ำเงิน: ข้อสรุปที่ต้องจำ

ใช้กับใจความที่ร้อยหลายบทเข้าด้วยกัน ถ้าอ่านเอกสารนี้แบบเร็ว ให้อ่านเฉพาะกล่องน้ำเงิน

กล่องส้ม: กับดัก หรือข้อแม้ที่ถ้าลืมแล้วข้อสรุปจะผิด

ใช้กับสมมติฐานที่เปราะ และกับจุดที่คนทำงาน DW พลาดกันบ่อย

กล่องเขียว: สิ่งที่ตรวจสอบกับข้อมูลจริงแล้ว

ใช้เฉพาะกับตัวเลขหรือข้อเท็จจริงที่เปิดไฟล์ดูจริงหรือเรียก API จริงแล้ว ไม่ใช้กับสิ่งที่คาดว่าน่าจะเป็น

กล่องแดง: วิธีที่ผิดจริง ๆ

ใช้น้อยที่สุด — เฉพาะแนวทางที่ถ้าทำตามแล้วผลลัพธ์จะผิด ไม่ใช่แค่ไม่สวย

สถานะของข้อมูลในเอกสารนี้

ข้อความที่กำกับ **[ตรวจแล้ว]** หมายถึงเปิดไฟล์จริงหรือเรียก API จริงแล้ว ส่วน **[ยังไม่ตรวจ]** คือสมมติฐานที่ยังต้องพิสูจน์ในขั้น ETL — อย่าอ้างอิงข้อความกลุ่มหลัง ในรายงานฉบับส่งจนกว่าจะตรวจแล้ว

ภาคที่ 1

ธุรกิจและคำถาม — เราจะตอบอะไร

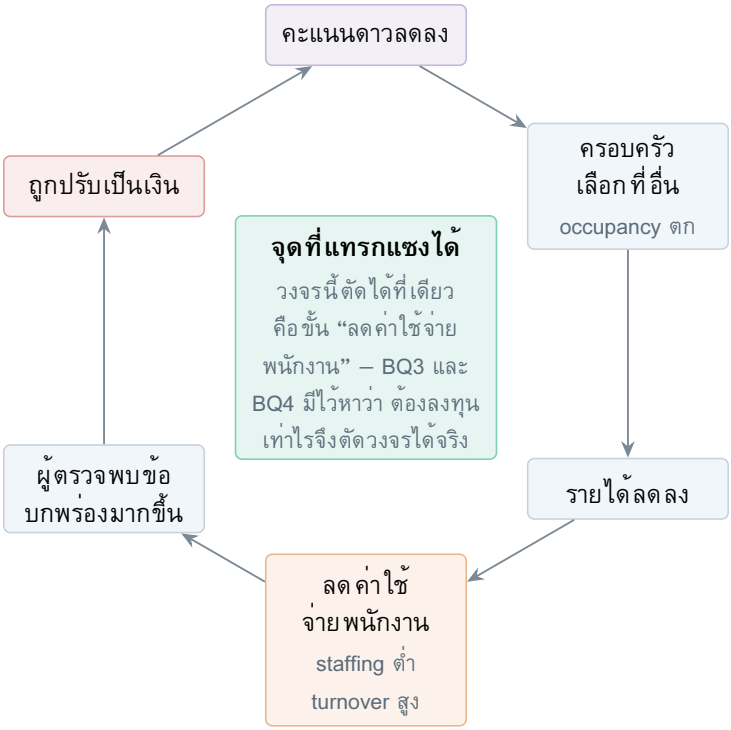
ภาคนี้ตอบคำถามว่า: ก่อนแตะข้อมูลแม้แต่แถวเดียว เราต้องการรู้อะไรกันแน่

1 ธุรกิจสถานดูแลผู้สูงอายุทำงานอย่างไร

สถานดูแลผู้สูงอายุแบบ Skilled Nursing Facility (SNF) ในสหรัฐอเมริกา เป็นธุรกิจที่มีลักษณะเฉพาะสามอย่างพร้อมกัน

1. **รายได้ผูกกับอัตราการเข้าพัก** (occupancy) เพราะต้นทุนต่อเตียงเป็นต้นทุนคงที่ เตียงว่างหนึ่งเตียงคือรายได้ที่หายไปโดยที่ต้นทุนไม่ลด
2. **ต้นทุนหลักคือค่าแรงพยาบาลและผู้ช่วยพยาบาล** ซึ่งกินสัดส่วนต้นทุนดำเนินงานสูงที่สุด
3. **ถูกกำกับเข้มโดยรัฐ** ผ่านระบบให้คะแนน Five-Star Quality Rating การตรวจสถานประกอบการ และบทลงโทษทางการเงิน

สิ่งที่ทำให้ธุรกิจนี้น่าสนใจในเชิงวิเคราะห์คือ ทั้งสามอย่างนี้ *ไม่ได้แยกกัน* แต่ป้อนกลับเข้าหากันเป็นวงจร



รูปที่ 2: วงจรป้อนกลับของธุรกิจสถานดูแลผู้สูงอายุ — คะแนนคุณภาพไม่ใช่แค่ตัวชี้วัด แต่เป็นตัวขับรายได้ ระบบคลังข้อมูลนี้จึงต้องเชื่อมคุณภาพ ต้นทุนแรงงาน และค่าปรับ ไว้ในที่เดียวกัน

ทำไมต้องสร้างคลังข้อมูลเอง ทั้งที่ CMS เปิดข้อมูลให้ดูอยู่แล้ว

เว็บของ CMS ให้ดูได้เฉพาะ **สถานะปัจจุบัน** ของแต่ละแห่ง ข้อมูลเดือนเก่าถูกย้ายไป คลังเอกสารเก่าเป็นไฟล์บีบอัดแยกกันเดือนต่อเดือน ไม่มีใครต่อให้เป็นอนุกรมเวลา คำถามอย่าง “สถานที่ไหนคะแนนกำลังตกต่อเนื่อง” หรือ “อุตสาหกรรมฟื้นจาก COVID แล้วหรือยัง” จึงตอบไม่ได้เลยถ้าไม่ประกอบคลังข้อมูลขึ้นมาเอง — นี่คือเหตุผลที่โครงงานนี้มีอยู่

1.1 ปัญหาทางธุรกิจที่ต้องการแก้

เอกสารนี้สมมติบทบาทเป็นทีมวางแผนกลยุทธ์ของ **ผู้ประกอบการเครือสถานดูแลผู้สูงอายุ** หรือกองทุนที่กำลังพิจารณาลงทุนในธุรกิจนี้ ปัญหาที่เผชิญมีสี่ข้อ

1. **ตัดสินใจขยายตลาดโดยไม่มีข้อมูลรองรับ** — ไม่รู้ว่ารัฐใดมีอุปสงค์เหลืออยู่ (ผู้สูงอายุมากแต่เตียงน้อย) และคู่แข่งในพื้นที่นั้นอ่อนแอพอให้เข้าไปแข่งได้
2. **ไม่รู้ว่าควรลงทุนด้านพนักงานเท่าไรจึงคุ้ม** — การเพิ่มชั่วโมงพยาบาลคือต้นทุน ที่เห็นทันทีในงบเดือนนี้ แต่ผลตอบแทนมาช้าและวัดยาก
3. **ความเสี่ยงด้านกฎระเบียบมองไม่เห็นล่วงหน้า** — รู้ตัวอีกทีเมื่อถูกปรับไปแล้ว ทั้งที่สัญญาณเตือนปรากฏในข้อมูลมาก่อนหลายเดือน
4. **ข้อมูลกระจัดกระจายตามเวลา** — ตามที่อธิบายในกล่องน้ำเงินด้านบน

2 ใครใช้ข้อมูลนี้ (Stakeholders)

ผู้ใช้	ต้องตัดสินใจอะไร	ต้องการอะไรจากแดชบอร์ด
CEO / หัวหน้าฝ่ายขยายกิจการ	เลือกตลาดที่จะเข้าไปลงทุนหรือซื้อกิจการ	ภาพเปรียบเทียบรายรัฐ: อุปสงค์ เทียบอุปทาน เทียบ คุณภาพคู่แข่ง
COO / ผู้จัดการภูมิภาค	จัดลำดับว่าจะเข้าไปแก้ที่ไหนก่อน	รายชื่อสถานที่ที่คะแนนกำลังตก พร้อมระดับความเร่งด่วน
ผู้อำนวยการฝ่ายบุคคล	วางแผนอัตรากำลังและลดการลาออก	ความสัมพันธ์ ชั่วโมงพยาบาล – อัตราลาออก – คะแนนคุณภาพ
CFO / นักลงทุนสัมพันธ์	ประเมินผลตอบแทนและความเสี่ยง	มูลค่าค่าปรับ อัตราการเข้าพัก และตัวคูณเงินจูงใจ VBP
ฝ่ายกำกับการปฏิบัติตามกฎ	บริหารความเสี่ยงด้านกฎระเบียบ	แนวโน้มค่าปรับ ประเภทข้อบกพร่องที่พบบ่อย สถานะ Special Focus
(ภายนอก) ครอบครัวผู้สูงอายุ	เลือกสถานที่ให้ญาติผู้ใหญ่	เปรียบเทียบคุณภาพของสถานที่ในพื้นที่เดียวกัน

ตารางที่ 1: ผู้ใช้งานข้อมูลหกกลุ่ม คอลัมน์กลางคือสิ่งที่ทำให้แดชบอร์ดมีประโยชน์จริง — ถ้ากราฟใดไม่ช่วยตัดสินใจข้อใดข้อหนึ่งในคอลัมน์นี้ ก็ไม่ควรอยู่บนหน้าจอ

3 คำถามทางธุรกิจแปดข้อ

โจทย์กำหนดอย่างน้อย 5 ข้อ ตั้งไว้ 8 ข้อเพื่อให้มีระยะเผื่อ คอลัมน์ขวาสุดคือเครื่องมือตรวจสอบตัวเอง: ถ้าคำถามข้อใดไม่มีตารางรองรับ แปลว่าผังที่ออกแบบไว้ยังไม่พอ

#	คำถาม	ผู้ใช้	ตอบด้วยตารางใด
BQ1	รัฐใดน่าเข้าไปลงทุนที่สุด เมื่อดูจำนวนเตียงต่อประชากรสูงอายุพันคน อัตราการเข้าพักเฉลี่ย และคะแนนคุณภาพของคู่แข่ง	CEO	Fact_Facility_Monthly × Dim_Geography + ข้อมูลประชากร
BQ2	รูปแบบการถือครองแบบใดให้ผลต่างกันอย่างไร ทั้งด้านคุณภาพ ชั่วโมงพยาบาล และค่าปรับต่อเตียง	CEO, CFO	ทั้งสอง Fact × Dim_Ownership
BQ3	ชั่วโมงพยาบาลต่อผู้พักอาศัยต่อวันสัมพันธ์กับคะแนนดาวและค่าปรับอย่างไร และจุดคุ้มอยู่ที่ระดับใด	ฝ่ายบุคคล, CFO	Fact_Facility_Monthly
BQ4	อัตราการลาออกของพยาบาลสูงแค่ไหนจึงเริ่มฉุดคุณภาพและอัตราการเข้าพัก	ฝ่ายบุคคล	Fact_Facility_Monthly
BQ5	แนวโน้มปี 2562–2569 ของอัตราการเข้าพัก ชั่วโมงพยาบาล และคะแนนดาว เป็นอย่างไร ฟื้นจาก COVID แล้วหรือยัง	CEO, CFO	Fact_Facility_Monthly × Dim_Date
BQ6	สถานที่ใดกำลังเสื่อมคุณภาพและมีแนวโน้มถูกลงโทษ	COO, กำกับกฎ	ทั้งสอง Fact × Dim_Facility (SCD2)
BQ7	ค่าปรับกระจุกตัวที่รัฐใดและช่วงเวลาใด รัฐใดบังคับใช้กฎเข้มที่สุดต่อเตียง	กำกับกฎ	Fact_Penalty_Event × Dim_Geography, Dim_Date
BQ8	เครือขนาดใหญ่ได้เปรียบจริงหรือไม่ — คุณภาพดีกว่า หรือแค่กดต้นทุนแรงงาน	CEO	Fact_Facility_Monthly × Dim_Chain

ตารางที่ 2: คำถามทางธุรกิจแปดข้อ ทุกข้อมีตารางรองรับครบ ซึ่งเป็นหลักฐานว่าผัง Star Schema ในภาค 4 ออกแบบมาพอดีกับสิ่งที่ต้องตอบ ไม่ได้ออกแบบลอย ๆ

BQ1 เป็นข้อเดียวที่พึ่งข้อมูลนอก CMS

ถ้าดึงข้อมูลประชากรอายุ 65 ปีขึ้นไปไม่สำเร็จ จะลดรูปเหลือ “เตียงต่อประชากรทั้งรัฐ” หรือตัด BQ1 ทิ้ง แล้วยังเหลือ 7 ข้อ ซึ่งยังเกินเกณฑ์ วางแผนทางถอยไว้ตั้งแต่ตอนนี้ เพราะถ้าไปเจอตอนเขียนแดชบอร์ดจะแก้ไม่ทัน

4 Measures และกับดักของการรวมค่า

โจทย์กำหนดอย่างน้อย 3 ตัว และให้อธิบาย *ความหมาย วิธีคำนวณ และประโยชน์* เอกสารนี้เพิ่มคอลัมน์ **การรวมค่า (additivity)** เข้าไปด้วย เพราะเป็นสิ่งที่ตัดสินว่า แดชบอร์ดจะสรุปตัวเลขถูกหรือผิด

Measure	วิธีคำนวณ	การรวมค่า	ความหมายและประโยชน์
M1 อัตราการเข้าพัก	SUM(avg_residents_per_day) / SUM(certified_beds)	รวมตรง ๆ ไม่ได้ (อัตราส่วน)	ตัวขับรายได้หลัก ต่ำกว่า 80% มักหมายถึงขาดทุน
M2 วัน-ผู้พักอาศัย	avg_residents_per_day × จำนวนวันในงวด	**รวมได้**	ปริมาณบริการที่ขายได้จริง ใช้ถ่วงน้ำหนัก measure อื่น
M3 ชั่วโมงพยาบาลต่อคนต่อวัน	reported_total_nurse_hprd (ถ่วงน้ำหนักด้วย M2)	รวมตรง ๆ ไม่ได้	ตัวแทนของต้นทุนแรงงาน และเป็นตัวพยากรณ์คุณภาพที่แรงที่สุด
M4 อัตราการลาออก	total_nursing_turnover_pct (ถ่วงน้ำหนักด้วย M2)	รวมตรง ๆ ไม่ได้	ต้นทุนแฝงจากการสรรหาและฝึกอบรม และเป็นสัญญาณเตือนคุณภาพ
M5 มูลค่าค่าปรับ	SUM(fine_amount_usd) จาก Fact_Penalty_Event	**รวมได้**	ต้นทุนความเสี่ยงด้านกฎระเบียบที่จับต้องได้เป็นเงิน
M6 จำนวนครั้งที่ถูกลงโทษ	COUNT(*) จาก Fact_Penalty_Event	**รวมได้**	ใช้คู่กับ M5 เพื่อแยก “ถูกปรับบ่อยทีละน้อย” ออกจาก “ถูกปรับหนักครั้งเดียว”
M7 คะแนนดาวเฉลี่ย	AVG(overall_rating) ช่วง 1–5	รวมตรง ๆ ไม่ได้	ตัวแปรที่ครอบครัวผู้สูงอายุใช้ตัดสินใจมากที่สุด
M8 จำนวนข้อบกพร่อง	SUM(cycle1_total_-deficiencies)	**รวมได้**	ตัวชี้วัดปฏิบัติการที่ “มาก่อน” การถูกปรับ
M9 ค่าปรับต่อวัน-ผู้พักอาศัย	M5 / M2	คำนวณตอน query	ทำให้เทียบความเสี่ยงระหว่างสถานที่ต่างขนาดได้อย่างเป็นธรรม
M10 เตียงต่อผู้สูงอายุพันคน	SUM(certified_beds) / (ประชากร 65+ / 1000)	คำนวณตอน query	ความอิ่มตัวของตลาด ค่าต่ำแปลว่ายังมีอุปสงค์เหลือ (ใช้ตอบ BQ1)

ตารางที่ 3: Measures สิบตัว M1–M8 เก็บในตาราง Fact ส่วน M9–M10 ไม่เก็บแต่คำนวณตอนสอบถาม เพราะเป็นอัตราส่วนที่หาได้จากตัวอื่น การเก็บซ้ำมีแต่จะทำให้ข้อมูลขัดกันเอง

ความผิดพลาดที่พบบ่อยที่สุดในงานคลังข้อมูล: เฉลี่ยของอัตราส่วน

สมมติมีสองแห่ง แห่งแรก 10 เตียง มีผู้พัก 5 คน (50%) แห่งที่สอง 200 เตียง มีผู้พัก 180 คน (90%)

`AVG(occupancy)` ให้ $(50 + 90)/2 = 70\%$
วิธีที่ถูกคือ `SUM(residents)/SUM(beds)` ให้ $185/210 = 88.1\%$

ต่างกัน 18 จุด และค่าที่ผิดจะ *ต่ำกว่าความจริงเสมอ* เมื่อสถานที่ใหญ่ทำได้ดีกว่า แดชบอร์ดที่ลากอัตราส่วนไปวางแล้วให้เครื่องมือเฉลี่ยให้เอง จะผิดแบบนี้ทุกครั้ง จึงต้องเก็บ **ตัวตั้งและตัวหารแยกกัน** ในตาราง Fact แล้วหารตอนแสดงผล

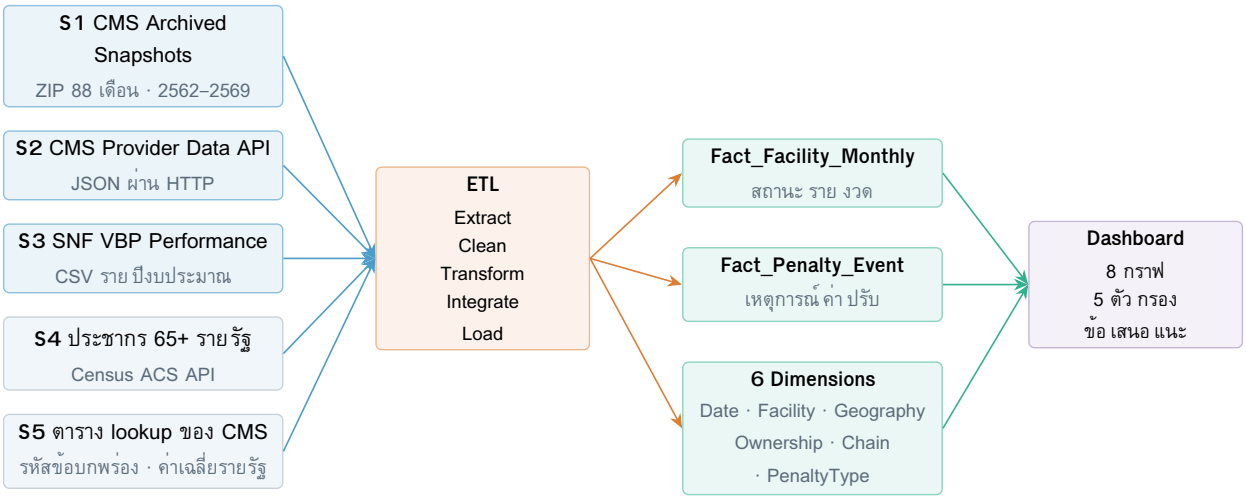
ภาคที่ 2

แหล่งข้อมูล — มาจากไหน มีอะไร ใช้ทำอะไร

ภาคนี้ตอบคำถามว่า: ข้อมูลทุกตัวเลขที่จะขึ้นบนแดชบอร์ด สืบกลับไปถึงไฟล์ไหนบนเซิร์ฟเวอร์ใคร

โจทย์กำหนดให้ใช้ **อย่างน้อย 3 แหล่ง** และอย่างน้อยหนึ่งแหล่งต้องซับซ้อนกว่า CSV ตารางทั่วไป โครงงานนี้ใช้ **3 แหล่งหลัก** และ **2 แหล่งเสริม**

5 ภาพรวมทั้งห้าแหล่ง



รูปที่ 3: ทางเดินของข้อมูลทั้งระบบ ห้าแหล่งไหลผ่าน ETL ชุดเดียวลงสู่สองตาราง Fact และหกตาราง Dimension แล้วจึงขึ้นแดชบอร์ด สีในผังนี้คือสีเดียวกับที่ใช้เรียกระยะต่าง ๆ ตลอดเอกสาร

6 แหล่งที่ 1 — CMS Nursing Homes Archived Snapshots

6.1 มาจากไหน

- **ผู้เผยแพร่:** Centers for Medicare & Medicaid Services (CMS) กระทรวงสาธารณสุขสหรัฐอเมริกา ผ่านระบบ Provider Data Catalog
- **หน้าเว็บ:** `https://data.cms.gov/provider-data/archived-data/nursing-homes`
- **ลิงก์ไฟล์จริง:** หน้าเว็บนั้นเป็นแอปแบบ JavaScript จึงไม่มีลิงก์ในโค้ด HTML ต้องดึงรายการผ่าน API (แหล่ง S2) ซึ่งคืนที่อยู่ไฟล์ในรูปแบบ
  `/provider-data/sites/default/files/dataset-archives/theme/nursing-homes/nursing-homes_2026-06-24.zip`
- **สิทธิ์การใช้งาน:** ข้อมูลสาธารณะของรัฐบาลสหรัฐฯ ใช้ได้เสรี ต้องอ้างอิงแหล่งที่มา

สิ่งที่ตรวจแล้วเกี่ยวกับแหล่งนี้

มี **88 snapshot รายเดือน** ตั้งแต่ **17 ม.ค. 2562** ถึง **6 ส.ค. 2569** และมีชุดรวมรายปีอีก 8 ชุด ไฟล์รายเดือนมีขนาดราว 32–40 MB เมื่อบีบอัด ยกเว้นบางเดือนที่ผิดปกติ เช่น 29 ก.ค. 2569 มีขนาด 622 MB และ 6 ส.ค. 2569 มีเพียง 3.5 MB (น่าจะเป็นการอัปเดตบางส่วน) — ขั้น ETL จึงต้องไม่สมมติว่าทุก snapshot มีไฟล์ครบเท่ากัน

6.2 มีอะไรอยู่ข้างใน

เปิดไฟล์ `nursing-homes_2026-06-24.zip` (38 MB บีบอัด) พบ **20 ไฟล์** รวม **614 MB** เมื่อคลาย รายการด้านล่างคือของจริงทั้งหมด

ไฟล์	แถว	คอลัมน์	เนื้อหา
NH_ProviderInfo	14,695	99	โปรไฟล์ คะแนนดาว ชั่วโมงพยาบาล อัตราลาออก เตียง ที่ตั้ง เจ้าของ
NH_Penalties	16,180	14	ค่าปรับและการระงับการจ่ายเงินรายรายการ พร้อมวันที่และจำนวนเงิน
NH_SurveySummary	43,952	41	สรุปผลการตรวจแยกตามหมวดข้อบกพร่อง 30 หมวด
FY_2026_SNF_VBP_Facility_-Performance	13,900	49	คะแนนและตัวคูณเงินจูงใจ (ดูแหล่ง S3)
NH_CitationDescriptions	643	5	คำอธิบายรหัสข้อบกพร่องแต่ละรหัส
NH_StateUSAverages	54	51	ค่าเฉลี่ยรายรัฐและระดับประเทศ
NH_HealthCitations	165 MB		ข้อบกพร่องรายรายการ ไฟล์ใหญ่ที่สุด
NH_Ownership	56 MB		นิติบุคคลผู้ถือครองแต่ละราย หนึ่งแห่งมีได้หลายราย
NH_QualityMsr_MDS	75 MB		ตัวชี้วัดคุณภาพจากแบบประเมินผู้พักอาศัย
NH_QualityMsr_Claims	17 MB		ตัวชี้วัดคุณภาพจากข้อมูลการเบิกจ่าย
NH_FireSafetyCitations	69 MB		ข้อบกพร่องด้านความปลอดภัยจากอัคคีภัย
NH_SurveyDates	7.8 MB		วันที่เข้าตรวจแต่ละรอบ

อีก 8 ไฟล์ที่เหลือเป็นข้อมูลประกอบขนาดเล็กและคู่มือ NH_Data_Dictionary.pdf

ตารางที่ 4: ไฟล์ทั้งหมดใน snapshot เดือนมิถุนายน 2569 ตัวเลขแถวและคอลัมน์มาจากการเปิดไฟล์จริง โครงงานนี้จะใช้เพียง 6 ไฟล์แรก ส่วนที่เหลือระบุไว้เพื่อให้เห็นว่าตัดอะไรออกและตัดเพราะอะไร

6.3 ใช้ทำอะไร

ไฟล์	ป้อนตารางใด	ตอบคำถามข้อใด
NH_ProviderInfo	Fact_Facility_Monthly และ Dimension ทั้ง 5 ตัวแรก	BQ1–BQ6, BQ8
NH_Penalties	Fact_Penalty_Event, Dim_Penalty_Type	BQ2, BQ6, BQ7
NH_SurveySummary	สำรองไว้สำหรับ Fact ที่ 3 (ถ้ามีเวลา)	ขยาย BQ7
NH_CitationDescriptions	ตาราง lookup แปลรหัสเป็นคำอธิบาย	BQ7
NH_StateUSAverages	ค่าอ้างอิงเปรียบเทียบบนกราฟ	BQ1, BQ5

ตารางที่ 5: การจับคู่ไฟล์กับตารางปลายทาง ไฟล์ที่ไม่ปรากฏในตารางนี้จะไม่ถูกคลายออกจาก ZIP เลย เพื่อไม่ให้เสียเวลาและพื้นที่ไปกับข้อมูล 165 MB ที่ไม่ได้ใช้

ทำไมแหล่งนี้จึงนับว่า "ซับซ้อนกว่า CSV ตารางทั่วไป"

โจทย์ยกตัวอย่างความซับซ้อนไว้ข้อหนึ่งว่า *"ข้อมูลจากหลายไฟล์หรือหลายช่วงเวลา"* ซึ่งแหล่งนี้ตรงเป๊ะ: ต้องวนดาวน์โหลดหลายสิบไฟล์ คลายบีบอัด จับคู่ชื่อไฟล์ที่เปลี่ยนไปเรื่อย แล้วต่อกันเป็นอนุกรมเวลาเอง ไม่มีไฟล์ใดไฟล์หนึ่งที่เปิดแล้วได้ข้อมูลย้อนหลังทันที

ชื่อไฟล์มีอย่างน้อย 3 แบบ ไม่ใช่แค่เปลี่ยนเดือน [ตรวจแล้ว]

เปิด snapshot ปี 2562 เทียบกับปี 2569 จริงแล้วพบว่าไม่ได้ต่างกันแค่เดือน แต่ต่างกันทั้ง *ระบบตั้งชื่อ*

ยุค	รูปแบบชื่อไฟล์
2562	`ProviderInfo_Download.csv`, `Penalties_Download.csv` — ไม่มี `NH_` ไม่มีเดือน
2569 (ปกติ)	`NH_ProviderInfo_Jun2026.csv` — มี prefix และเดือน
2569 (บางส่วน)	`4pq5-n9py_2026-07-29_NH_ProviderInfo_Jul2026.csv` — มีรหัสชุดข้อมูลนำหน้าอีกชั้น

การ จับ คู่ ด้วย `glob` ธรรมดา จึง พัง แน่นอน ต้อง ใช้ การ จับ คู่ แบบ **regex ต่อ ยุค** หรือ อ่าน จาก `manifest.json` ถ้ามี

snapshot "บางส่วน" มีอยู่จริง และ ETL ต้องรับมือ [ตรวจแล้ว]

งวด 2026-08-06 ที่มีขนาดผิดปกติ 3.5 MB เปิดดูแล้ว **มีเพียง 4 ไฟล์**: `NH_ProviderInfo`, `NH_StateUSAverages`, `manifest.json` และคู่มือ — *ไม่มี* `NH_Penalties` เลย

ถ้า ETL สมมติว่าทุก snapshot มีครบ 6 ไฟล์ที่ต้องใช้ งวดนี้จะทำให้ท่อพัง หรือแย่กว่านั้นคือโหลด Fact 2 ได้ศูนย์แถวแบบเงียบ ๆ

กฎที่ต้องเขียน: ตรวจรายชื่อไฟล์ก่อนประมวลผลทุกงวด งวดใดขาดไฟล์ให้ข้ามเฉพาะ Fact ที่เกี่ยวข้อง ไม่ใช่ข้ามทั้งงวด และบันทึกลงตารางบันทึกการทำงาน

ขอบเขตที่ตัดสินใจไว้แล้ว: โหลดรายไตรมาส ไม่ใช่ทุกเดือน

88 snapshot × 38 MB คือการดาวน์โหลดราว 3.3 GB และคะแนนดาวเปลี่ยนช้าเกินกว่าจะ ให้ข้อมูลเพิ่มรายเดือน จึงเลือกไตรมาสละหนึ่ง snapshot ได้ประมาณ 28–31 จุดเวลา ครอบคลุมช่วงเดียวกัน แต่ดาวน์โหลดลดลงเหลือราวหนึ่งในสาม ถ้าภายหลังพบว่าความละเอียดไม่พอ เพิ่ม snapshot ได้โดยไม่ต้องรื้อผัง เพราะ Grain กำหนดไว้ที่ “ต่อ snapshot” ไม่ใช่ “ต่อเดือน”

7 แหล่งที่ 2 — CMS Provider Data Catalog REST API

7.1 มาจากไหน

เป็น API สาธารณะของ CMS ที่อยู่ใต้โดเมนเดียวกัน ใช้สามปลายทาง

#	ปลายทาง (ต่อท้าย `https://data.cms.gov/provider-data/api/1/`)	คืนอะไร
A	`archive/aggregate/theme/nursing-homes/relative`	รายการ snapshot ทั้งหมด 96 รายการ พร้อม URL ขนาด และวันที่
B	`metastore/schemas/dataset/items`	แคตาล็อกชุดข้อมูลทั้งหมด 234 ชุด ในจำนวนนี้เป็นของ Nursing Homes 11 ชุด
C	`datastore/query/4pq5-n9py/0?limit=1`	ข้อมูลแถวจริงของชุด Provider Information เป็น JSON ชื่อคีย์แบบ snake_case

ตารางที่ 6: สามปลายทางของ API ที่โครงงานนี้ใช้ รหัสอย่าง 4pq5-n9py คือรหัสชุดข้อมูลที่ได้มาจากปลายทาง B จึงไม่ต้องจดไว้เอง

7.2 มีอะไร

ปลายทาง A คืนข้อมูลหน้าตาแบบนี้ (ตัดมาหนึ่งรายการ)

```
{"name": "Theme: nursing-homes (2026-06-24)", "id": "8681",
"type": "theme", "theme": "nursing-homes",
"url": "/provider-data/sites/default/files/dataset-archives/theme/
nursing-homes/nursing-homes_2026-06-24.zip",
"size": "38990000", "date": "2026-06-24", "access_level": "public"}
```

ส่วนปลายทาง C คืนแถวข้อมูลจริง ตัวอย่างแถวแรกคือ

```
{"cms_certification_number_ccn": "015009",
 "provider_name": "BURNS NURSING HOME, INC.",
 "citytown": "RUSSELLVILLE", "state": "AL", "zip_code": "35653",
 "ownership_type": "For profit - Corporation",
 "number_of_certified_beds": "57",
 "average_number_of_residents_per_day": "51.6", ...}
```

แถวตัวอย่างนี้เรียกมาจริงแล้ว และใช้เป็นกรณีทดสอบได้

สถานที่ CCN `015009` มี 57 เตียง มีผู้พักอาศัยเฉลี่ย 51.6 คนต่อวัน ⇒ อัตราการเข้าพัก $51.6/57 =$ **90.5%**

กับดัก: แถวทดสอบนี้ผูกกับ "งวด" ห้ามใช้ลอย ๆ

[ตรวจแล้ว] ตรวจซ้ำแล้วพบว่า **ค่าเดียวกันนี้ไม่ตรงกันระหว่างสองแหล่ง** เพราะเป็นคนละงวด

แหล่ง	Processing Date	ผู้พักอาศัย	อัตราเข้าพัก
API ปลายทาง C (ข้อมูลปัจจุบัน)	2026-07-01	51.6	90.53%
CSV `NH_ProviderInfo_Jun2026`	2026-06-01	52.4	91.93%

ETL โหลดจาก *CSV* ไม่ใช่จาก API ดังนั้นการทดสอบต้องคาดหวัง **91.93% สำหรับงวด มิ.ย. 2569** ถ้าเขียนเทสต์ให้เช็ค 90.5% ลอย ๆ เทสต์จะแดงทั้งที่ท่อถูกต้อง

บทเรียนที่ใหญ่กว่านั้น: **ค่าทดสอบทุกตัวต้องระบุงวดกำกับเสมอ** และนี่คือหลักฐานตรง ๆ ว่าปลายทาง C ใช้ "ตรวจความถูกต้องหลัง Load" ได้เฉพาะกับงวดล่าสุดเท่านั้น

7.3 ใช้ทำอะไร

1. **ค้นหา snapshot อัตโนมัติ** (ปลายทาง A) — ทำให้ ETL รู้เองว่ามีข้อมูลเดือนใหม่ โดยไม่ต้องมีคนไปคัดลอกลิงก์มาแปะ นี่คือสิ่งที่ทำให้ *รันซ้ำได้* ตามที่โจทย์บังคับ
2. **ทำ Incremental Load** — เทียบรายการจาก A กับไฟล์ที่มีอยู่แล้วในเครื่อง แล้วโหลดเฉพาะส่วนต่าง
3. **ตรวจความถูกต้องหลัง Load** — ดึงข้อมูลปัจจุบันจากปลายทาง C มาเทียบกับแถวล่าสุดในคลังข้อมูล ถ้าไม่ตรงแปลว่า ETL แปลงค่าผิดที่ไหนสักแห่ง

API คือสิ่งที่แยก "ไปป์ไลน์" ออกจาก "การดาวน์โหลดด้วยมือ"

โจทย์ระบุชัดว่า *"กระบวนการควรสามารถรันซ้ำได้ โดยไม่ต้องแก้ไขข้อมูลด้วยมือทุกขั้นตอน"* และให้คะแนนพิเศษกับ *การใช้ API* กับ *การทำ Incremental Load* ปลายทาง A ตัวเดียวตอบทั้งสามข้อนี้พร้อมกัน ด้วยโค้ดไม่กี่บรรทัด

8 แหล่งที่ 3 — SNF VBP Facility Performance

8.1 มาจากไหน มีอะไร

ไฟล์ `FY_2026_SNF_VBP_Facility_Performance.csv` อยู่ใน ZIP เดียวกับแหล่ง S1 แต่นับเป็นคนละแหล่ง เพราะเป็นคนละโครงการของ CMS คนละรอบเวลา และ **คนละ Grain** — เป็นข้อมูลราย *ปีงบประมาณ* ขณะที่ทุกอย่างใน S1 เป็นรายเดือน

ขนาด **13,900 แถว** × **49 คอลัมน์** [ตรวจแล้ว] เนื้อหาคือคะแนนของโครงการ Skilled Nursing Facility Value-Based Purchasing ซึ่งวัดสี่ด้าน

ตัวชี้วัด	ความหมายทางธุรกิจ
อัตราการกลับเข้าโรงพยาบาล	ผู้ป่วยที่ต้องกลับเข้าโรงพยาบาลหลังออกจากสถานดูแล
อัตราการติดเชื้อในสถานพยาบาล	ตัวชี้วัดคุณภาพการควบคุมการติดเชื้อ
อัตราการลาออกของพยาบาล	เชื่อมกับ M4 โดยตรง ยืนยันว่า CMS เองก็ถือว่านี่คือคุณภาพ
ชั่วโมงพยาบาลปรับตามความหนักของผู้ป่วย	เชื่อมกับ M3
ตัวคูณเงินจูงใจ	**ผลลัพธ์สุดท้าย: ตัวคูณที่ใช้กับเงินที่ CMS จ่ายให้จริง**

ตารางที่ 7: สี่ตัวชี้วัดของโครงการ VBP และผลลัพธ์ที่เป็นตัวเงิน คอลัมน์สุดท้ายคือสิ่งที่ทำให้แหล่งนี้มีค่าต่อ CFO เพราะเป็นจุดเดียวในข้อมูลทั้งหมดที่แปลงคุณภาพเป็นรายได้ได้โดยตรง

8.2 ใช้ทำอะไร

เชื่อมเข้ากับ `Fact_Facility_Monthly` ด้วย CCN แล้วเพิ่มมิติ "ผลตอบแทน" ให้กับคำถาม BQ2 และ BQ3 — เปลี่ยนคำถามจาก "ลงทุนพนักงานแล้วคะแนนดีขึ้นไหม" เป็น "ลงทุนพนักงานแล้ว *ได้เงินคืน* เท่าไร" ซึ่งเป็นคำถามที่ CFO สนใจจริง

> **การเชื่อมข้าม Grain ต้องระวังการนับซ้ำ**
>
> VBP มีหนึ่งแถวต่อสถานที่ต่อ *ปีงบประมาณ* แต่ Fact 1 มีหลายแถวต่อสถานที่ต่อปี ถ้า `JOIN` ตรง ๆ ค่าตัวคูณเงินจูงใจจะถูกทำซ้ำในทุกงวด และการ `SUM` จะได้ค่าเกินจริงหลายเท่า
>
> วิธีจัดการ: เก็บตัวคูณไว้เป็น **แอตทริบิวต์ระดับปี** แล้วใช้เฉพาะกับการรวมค่า ที่ระดับปีเท่านั้น ห้ามให้มันไหลลงไปที่ระดับงวด

9 แหล่งเสริมอีกสองแหล่ง

9.1 S4 — ประชากรอายุ 65 ปีขึ้นไป รายรัฐ

มาจาก U.S. Census Bureau American Community Survey ผ่าน API ที่คืน nested JSON ใช้เป็น **ฝั่งอุปสงค์** ของตลาด ซึ่งข้อมูล CMS ทั้งหมดไม่มีให้ เพราะ CMS มองแต่ฝั่งผู้ให้บริการ

การคำนวณไม่ตรงไปตรงมา: ตาราง `B01001` แยกประชากรตามเพศและช่วงอายุย่อย จึงต้องรวมตัวแปรหลายตัว (ชาย 65–66, 67–69, 70–74, ...และหญิงชุดเดียวกัน) เข้าด้วยกันเองจึงจะได้ "ประชากร 65+"

หาสาเหตุที่เรียกไม่สำเร็จได้แล้ว — Census บังคับใช้ API key

[ตรวจแล้ว] อาการ "คืนค่าว่าง" ไม่ใช่พารามิเตอร์ผิดหรือปีไม่มีข้อมูล แต่เป็นเพราะ Census ตอบกลับด้วย HTTP 302 ชี้ไปที่ `https://api.census.gov/data/missing_key.html` ซึ่งเป็นหน้า HTML ถ้า `curl` ไม่ได้ใส่ `-L` จะได้ body ว่างเปล่าโดยไม่มี error — ตรงกับอาการที่บันทึกไว้ทุกประการ

วิธีแก้: ขอคีย์ฟรีที่ `api.census.gov/data/key_signup.html` แล้วต่อ `&key=...` ท้าย URL เก็บคีย์ไว้ในตัวแปรสภาพแวดล้อม ห้าม hard-code ลงในโค้ดที่ commit

ตัวแปร 12 ตัวที่ต้องบวกกันเพื่อให้ได้ "ประชากร 65+"

ยืนยันชื่อตัวแปรจาก `groups/B01001.json` แล้ว [ตรวจแล้ว] (metadata ไม่ต้องใช้คีย์)

ชาย: `B01001_020E` (65–66), `021E` (67–69), `022E` (70–74), `023E` (75–79), `024E` (80–84), `025E` (85+)

หญิง: `B01001_044E` ถึง `049E` ตามช่วงอายุเดียวกัน

ทางถอยที่ทดสอบแล้วว่าใช้ได้ และไม่ต้องใช้คีย์เลย

[ตรวจแล้ว] ถ้าขอคีย์ไม่ทัน ใช้แฟ้ม Population Estimates แบบ CSV นิ่งแทนได้ ไม่มีระบบคีย์มาเกี่ยวข้อง

`www2.census.gov/programs-surveys/popest/datasets/2020-2024/state/asrh/sc-est2024-agesex-civ.csv` (812 KB, HTTP 200)

กรอง `SEX=0` และ `AGE>=65` (ตัด `AGE=999` ซึ่งเป็นแถวผลรวมออก) แล้วรวมตาม `NAME` ได้ครบ 50 รัฐ + DC + ผลรวมประเทศ ทดสอบแล้วได้ยอดทั้งประเทศ **61.2 ล้านคน** ซึ่งสมเหตุสมผล

9.2 S5 — ตาราง lookup ของ CMS

`NH_CitationDescriptions` (643 แถว) แปล รหัส ข้อ บกพร่อง เป็น คำ อธิบาย และ หมวด หมู่ ส่วน `NH_StateUSAverages` (54 แถว × 51 คอลัมน์) ให้ค่าเฉลี่ยรายรัฐและ ระดับประเทศ ใช้เป็น *เส้นอ้างอิง* บนกราฟ เพื่อให้ผู้อ่านรู้ว่าค่าที่เห็นดีหรือแย่กว่ามาตรฐาน

10 ตารางสืบย้อน — แหล่งไหนตอบคำถามไหน

ตารางนี้คือคำตอบรวบยอดของคำถาม "ข้อมูลมาจากไหน มีอะไร ใช้ทำอะไร" และเป็นสิ่งที่ควรคัดลอกไปใส่รายงานฉบับส่งโดยตรง

#	แหล่ง	รูปแบบ	ให้ข้อมูลอะไร	ป้อนตารางใด	ตอบข้อใด
S1	CMS Archived Snapshots	ZIP หลายไฟล์ หลายช่วงเวลา	โปรไฟล์ คะแนน ชั่วโมงพยาบาล ค่าปรับ ย้อนหลัง 7 ปี	Fact 1, Fact 2, Dim ทั้งหมด	BQ1–BQ8
S2	CMS Provider Data API	JSON ผ่าน HTTP	รายการ snapshot และแถวข้อมูลปัจจุบัน	ไม่ป้อนตารางโดยตรง แต่ขับ ETL และใช้ตรวจสอบ	ทุกข้อ (ทางอ้อม)
S3	SNF VBP Facility Performance	CSV รายปีงบประมาณ	ตัวคูณเงินจูงใจและคะแนนสี่ด้าน	แอตทริบิวต์ระดับปีของ Fact 1	BQ2, BQ3
S4	Census ACS API	Nested JSON	ประชากรอายุ 65+ รายรัฐ	Dim_Geography	BQ1
S5	CMS lookup tables	CSV เล็ก	คำอธิบายรหัส และค่าเฉลี่ยอ้างอิง	ตาราง lookup และเส้นอ้างอิงบนกราฟ	BQ1, BQ5, BQ7

ตารางที่ 8: ตารางสืบย้อนแหล่งข้อมูล อ่านจากซ้ายไปขวาจะได้ประโยคเต็มว่า แหล่งนี้มาจากไหน หน้าตาอย่างไร ให้อะไร ไปอยู่ตรงไหน และตอบคำถามข้อใด ทุกคำถามธุรกิจมีแหล่งรองรับครบ

ภาคที่ 3

คุณภาพข้อมูล — อะไรพังบ้าง

ภาคนี้ตอบคำถามว่า: ข้อมูลเปิดของรัฐบาลสะอาดพอจะใช้เลยไหม (คำตอบคือไม่)

โจทย์กำหนดให้มีปัญหาคุณภาพข้อมูลอย่างน้อย **3 ประเภท** เอกสารนี้ระบุไว้ **8 ประเภท** ทุกข้อมีทั้งวิธีตรวจจับและวิธีแก้เขียนไว้ล่วงหน้า เพื่อให้ขั้น Clean ทำงานอัตโนมัติได้จริง

11 ปัญหาแปดประเภทและวิธีรับมือ

#	ประเภท	ปัญหาที่คาดว่าจะเจอ	วิธีตรวจจับและแก้
Q1	คีย์ไม่ตรงกัน	CCN เป็นรหัส 6 หลักที่ **มีศูนย์นำหน้า** เช่น `015009` [ตรวจแล้ว] ถ้าอ่านเป็นตัวเลขจะเหลือ `15009` แล้วเชื่อมตารางไม่ติด	บังคับอ่านเป็นข้อความทุกไฟล์ แล้วเติมศูนย์ให้ครบ 6 หลัก ตรวจด้วยเงื่อนไขว่าอัตราการเชื่อมสำเร็จต้องเกิน 99% ก่อนโหลดเข้าคลัง
Q2	ระเบียนซ้ำ	สถานที่เดียวกันปรากฏใน *ทุก* snapshot และค่าปรับรายการเดิม ถูกรายงานซ้ำข้ามหลาย snapshot (แฟ้ม Penalties เป็นหน้าต่างหมุน 3 ปี [ตรวจแล้ว] ค่าปรับหนึ่งรายการจึงปรากฏซ้ำได้ถึง 36 งวด)	Fact 1 ตั้งใจให้ซ้ำ (เป็นสถานะรายงวด) จึงใส่ snapshot ไว้ในคีย์หลัก ส่วน Fact 2 ต้องกำจัดซ้ำจริง ด้วยคีย์ **สองระบบแยกตามยุค** เพราะ `Fine ID` มีเฉพาะงวด 202606 เป็นต้นไป [ตรวจแล้ว] (ดูกล่องแดงท้ายหัวข้อนี้)
Q3	ค่าว่าง	คะแนนดาวว่างสำหรับสถานที่เปิดใหม่ และ CMS ใช้คอลัมน์ footnote คู่กัน เพื่อบอกเหตุผลที่ค่าถูกระงับ (มีอย่างน้อย 9 คอลัมน์) [ตรวจแล้ว]	ไม่เติมค่าเดา แต่เก็บรหัส footnote ไว้ แล้ว **กันแถวที่ถูกระงับออกจากตัวหาร** ตอนคำนวณค่าเฉลี่ย
Q4	ค่าที่เป็นไปไม่ได้	ผู้พักอาศัยเฉลี่ยมากกว่าจำนวนเตียง ทำให้อัตราการเข้าพักเกิน 100% — เกิดขึ้นจริง **128 แห่ง** (0.88%) และค่าสูงสุดคือ **1414%** ซึ่งเป็นข้อผิดพลาดแน่นอน [ตรวจแล้ว]	เขียนกฎตรวจเป็นฟังก์ชัน: อัตราการเข้าพักต้องอยู่ใน [0, 1.2] อัตราลาออกต้องอยู่ใน (0, 200] แถวที่ตกกฎ **ติดธง ไม่ลบทิ้ง**
Q5	ชื่อหมวดไม่สอดคล้อง	ประเภทการถือครองมีหลายค่า เช่น `For profit - Corporation` [ตรวจแล้ว] คาดว่าสะกดต่างกันข้ามปี และชื่อเครือมีปัญหาตัวพิมพ์และช่องว่างเกิน	ทำตารางจับคู่แบบระบุชัด *ไม่ใช้การเดาความคล้าย* จากค่าดิบไปเป็นกลุ่มการถือครอง 3 กลุ่ม ส่วนชื่อเครือให้ตัดช่องว่างและปรับตัวพิมพ์
Q6	รูปแบบวันที่ต่างกัน	ตรวจปลายทางทั้งสองฝั่งแล้ว: ทั้ง ม.ค. 2562 และ มิ.ย. 2569 ใช้ `YYYY-MM-DD` เหมือนกัน [ตรวจแล้ว] สมมติฐานเดิมว่ารูปแบบเปลี่ยน **ยังไม่พบหลักฐาน** แต่ยังไม่ได้ตรวจงวดกลาง	คงกฎเดิมไว้เป็นตาข่ายกันพลาด: แปลงด้วยรายการรูปแบบที่ระบุชัดเรียงลำดับลอง ถ้าไม่เข้าเลยให้บันทึกไว้ **ห้ามปล่อยให้เดารูปแบบเอง** เพราะสลับวันกับเดือนได้เงียบ ๆ
Q7	โครงสร้างคอลัมน์เปลี่ยน	**หนักกว่าที่คาดมาก** [ตรวจแล้ว] ปี 2562 มี **78** คอลัมน์ ปี 2569 มี **99** และ *ชื่อคอลัมน์เปลี่ยนทั้งระบบ*: `PROVNUM` → `CMS Certification Number (CCN)` คอลัมน์อัตราลาออก (5) ชั่วโมงสุดสัปดาห์ (4) เครือ (7) พิกัด (1) ไม่มีในปี 2562 เลย	ทำตารางจับคู่ชื่อคอลัมน์แยกตามรุ่นของไฟล์ คอลัมน์ที่ยังไม่มีให้เป็นค่าว่าง และบันทึกลงตารางบันทึกการทำงานว่างวดใดขาดคอลัมน์ใด ใช้ Table 16 ของ `NH_Data_Dictionary.pdf` เป็นแหล่งอ้างอิงวันที่เปลี่ยน
Q8	หน่วยวัดไม่ตรงกัน	งวด มิ.ย. 2569 เก็บเป็น **เปอร์เซ็นต์** (มัธยฐาน 45.30 ช่วง 4–100) [ตรวจแล้ว] เทียบกับปี 2562 ไม่ได้เพราะยังไม่มีคอลัมน์นี้ (ดู Q7) จึงยัง **เทียบข้ามยุคไม่ได้** [ยังไม่ตรวจ]	ตรวจจากค่ามัธยฐานของแต่ละงวด ถ้าน้อยกว่า 1.5 ให้คูณ 100 แล้วบันทึกการแปลงลงตารางบันทึกการทำงาน กฎนี้ยังจำเป็น เพราะต้องรันกับงวดที่ยังไม่ได้เปิดดูอีกหลายสิบงวด

ตารางที่ 9: ปัญหาคุณภาพข้อมูลแปดประเภท ครอบคลุมทุกหัวข้อที่โจทย์ยกตัวอย่างไว้ คอลัมน์ขวาสุดเขียนเป็นภาษาที่แปลงเป็นโค้ดได้ทันที ซึ่งเป็นเจตนา — แผนที่แปลงเป็นโค้ดไม่ได้ คือแผนที่ยังคิดไม่จบ

อย่าอ่าน CCN เป็นตัวเลข แม้แต่ครั้งเดียว

`015009` เป็นรหัสประจำสถานพยาบาล ไม่ใช่จำนวน การอ่านเป็นตัวเลขทำให้กลายเป็น `15009` ซึ่งจะ*เชื่อมกับตารางอื่นไม่ติดแบบเงียบ ๆ* — ไม่มี error มีแต่แถวที่หายไปจากผลลัพธ์

รหัสไปรษณีย์ก็เป็นแบบเดียวกัน (`01234` ในรัฐทางตะวันออกเฉียงเหนือ) กฎที่ปลอดภัยคือ **อ่านทุกคอลัมน์เป็นข้อความก่อนเสมอ** แล้วค่อยแปลงเฉพาะคอลัมน์ที่เป็นตัวเลขจริง

`Fine ID` ใช้เป็นคีย์กำจัดซ้ำได้ไม่ตลอดช่วง — ต้องมีสองระบบ

เอกสารฉบับก่อนวางคีย์ของ Fact 2 ไว้ที่ (ccn, วันที่, ประเภท, fine_id) ตรวจจริงแล้วพบปัญหาสามชั้น [ตรวจแล้ว]

หนึ่ง — คอลัมน์นี้เพิ่งมี Table 16 ของคู่มือระบุว่า *"202606 Penalties: Added Fine ID column after Penalty Type"* แปลว่างวดก่อน มิ.ย. 2569 **ไม่มีคอลัมน์นี้เลย** ซึ่งคือเกือบทั้งโครงงาน

สอง — ว่างทั้งหมดบนรายการระงับการจ่ายเงิน ในงวด มิ.ย. 2569 `Fine ID` เป็นค่าว่างครบทั้ง **2,470** แถวที่เป็น Payment Denial (ไม่ว่างเลยทั้ง 13,710 แถวที่เป็น Fine) จึงกำจัดซ้ำได้เฉพาะครึ่งเดียวของตาราง

สาม — คีย์ธรรมชาติทดแทนไม่ปลอดภัย เมื่อมี `Fine ID` ค่าไม่ซ้ำเลยแม้แต่รายการเดียว (13,710 ค่าต่อ 13,710 แถว) แต่ถ้าใช้ (ccn, วันที่, ประเภท, จำนวนเงิน) แทน จะเหลือเพียง 13,424 กลุ่ม คือ **กลืนค่าปรับจริงหายไป 286 รายการ (2.09%) คิดเป็นเงิน 2.66 ล้านดอลลาร์ (0.58%)** เพราะสถานที่เดียวกันโดนปรับหลายรายการในวันเดียวด้วยจำนวนเงินเท่ากันได้จริง เช่น CCN `015060` วันที่ 2025-05-09 มีค่าปรับ $5,782 *สามรายการ* (Fine ID 134643, 134644, 134645)

ข้อสรุปเชิงออกแบบ: Fact 2 ต้องมีคอลัมน์ `fine_id_source` บอกว่าแถวนั้นได้คีย์มาจากไหน แล้วกำจัดซ้ำแบบสองระบบ — งวดตั้งแต่ 202606 ใช้ `Fine ID` ตรง ๆ ส่วนงวดก่อนหน้าใช้คีย์ธรรมชาติ **บวกลำดับที่ภายในกลุ่ม** (row number ภายใน ccn+วันที่+ประเภท+จำนวนเงิน) เพื่อไม่ให้รายการที่เหมือนกันถูกยุบรวม และต้องรายงานตัวเลข 2.09% นี้ ไว้ในข้อจำกัดของโครงงาน

ติดธง ดีกว่าลบทิ้ง

ทุกกฎในตารางข้างบนที่เจอค่าผิดปกติ จะ **ติดธง `is_suspect`** ไว้ที่แถวนั้น ไม่ใช่ลบแถวทิ้ง เหตุผลสองข้อ

หนึ่ง — ค่าผิดปกติบางตัวคือของจริง เช่น อัตราการเข้าพัก 105% เกิดได้จริงเมื่อมีเตียงเสริมชั่วคราว

สอง — แดชบอร์ดจะมีสวิตช์ให้ผู้ใช้เลือกเองว่าจะรวมหรือไม่รวมแถวที่น่าสงสัย ซึ่งกลายเป็นตัวกรองที่หนึ่งในห้าตัวไปในตัว และทำให้ผู้ใช้เห็นว่าข้อมูลไม่สมบูรณ์แบบ แทนที่จะซ่อนไว้

ภาคที่ 4

คลังข้อมูล — Grain และ Star Schema

ภาคนี้ตอบคำถามว่า: หนึ่งแถวในตาราง Fact หมายความว่าอะไรกันแน่

12 สี่ขั้นตอนของ Kimball

การออกแบบเดินตามลำดับสี่ขั้นตอนมาตรฐาน ห้ามข้ามลำดับ เพราะขั้นหลังพึ่งขั้นก่อนหน้าทั้งหมด

12.1 ขั้นที่ 1 — เลือกกระบวนการทางธุรกิจ

เลือก **สองกระบวนการ** เพราะทั้งคู่เป็นหัวใจของคำถาม และมี Grain ต่างกัน จึงต้องแยกตาราง Fact (ซึ่งโจทย์ระบุว่าได้คะแนนพิเศษ)

1. **การติดตามผลการดำเนินงานรายงวด** — CMS “ถ่ายรูป” สถานะของทุกแห่งทุกเดือน ไม่ใช่ธุรกรรม แต่เป็นสถานะ ณ เวลาหนึ่ง
2. **การถูกลงโทษทางกฎระเบียบ** — เกิดเป็นครั้ง ๆ มีวันที่ชัดเจนและมีมูลค่าเป็นเงิน

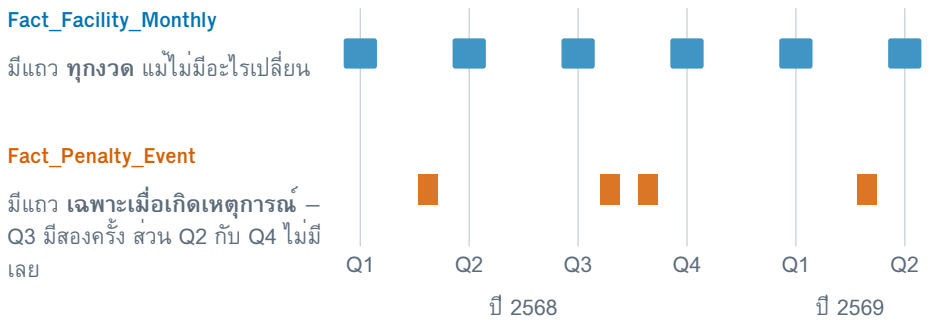
13 ขั้นที่ 2 — ประกาศ Grain

> **Grain คือประโยคเดียวที่ต้องเขียนให้ได้ก่อนทำอย่างอื่น**
>
> Grain คือคำตอบของคำถาม *“หนึ่งแถวในตารางนี้แทนอะไรในโลกจริง”* ถ้าตอบเป็นประโยคเดียวไม่ได้ แปลว่ายังออกแบบไม่เสร็จ และทุกอย่างที่สร้างต่อจากนั้นจะพังตามมา

ตาราง Fact	หนึ่งแถวหมายถึงอะไร	ประเภทและเหตุผล
Fact_Facility_Monthly	**หนึ่งแถว = สถานพยาบาลหนึ่งแห่ง (หนึ่ง CCN) ณ วันที่ CMS เผยแพร่ข้อมูลหนึ่งครั้ง** *ตัวอย่าง:* "CCN 015009 ตามข้อมูลที่เผยแพร่วันที่ 24 มิ.ย. 2569"	Periodic Snapshot — บันทึกสถานะ ไม่ใช่เหตุการณ์ แถวถูกสร้างทุกงวดแม้ไม่มีอะไรเปลี่ยน ซึ่งเป็นสิ่งที่ต้องการ เพราะทำให้เส้นกราฟต่อเนื่องไม่ขาด
Fact_Penalty_Event	**หนึ่งแถว = บทลงโทษหนึ่งรายการ (ค่าปรับหนึ่งครั้ง หรือการระงับการจ่ายเงินหนึ่งครั้ง) ที่ CMS สั่งกับสถานพยาบาลหนึ่งแห่ง ในวันที่กำหนดโทษหนึ่งวัน** *ตัวอย่าง:* "ค่าปรับ 12,350 ดอลลาร์ กับ CCN 015009 วันที่ 3 มี.ค. 2568"	Transaction — เป็นเหตุการณ์จริง ณ จุดเวลา ไม่มีแถวถ้าไม่มีเหตุการณ์ ทำให้การรวมยอดค่าปรับเป็นการบวกที่ถูกต้องแท้จริง

ตารางที่ 10: การประกาศ Grain ของทั้งสองตาราง Fact ประโยคตัวหนาคือสิ่งที่ต้องท่องได้ ส่วนตัวอย่างข้างใต้มีไว้เพื่อให้เห็นภาพว่าแถวจริงหน้าตาเป็นอย่างไร

13.1 ทำไมสองตารางนี้ถึงรวมกันไม่ได้



นี่คือเหตุผลที่ต้องแยกเป็นสองตาราง

ถ้ายัดไว้ตารางเดียว งวดที่ไม่มีค่าปรับจะต้องใส่ศูนย์ และงวดที่มีค่าปรับสองครั้งจะต้องทำสำเนาแถวสถานะขึ้นมาสองแถว ทำให้ชั่วโมงพยาบาลและจำนวนเตียงถูกนับซ้ำทันที

รูปที่ 4: Grain ที่ต่างกันของสอง Fact เมื่อวางบนแกนเวลาเดียวกัน แถบน้ำเงินเกิดสม่ำเสมอตามงวดที่ CMS เผยแพร่ ส่วนแถบส้มเกิดกระจัดกระจายตามเหตุการณ์จริง สองจังหวะนี้อยู่ในตารางเดียวกันไม่ได้

ProviderInfo มีคอลัมน์ค่าปรับอยู่แล้ว แต่ใช้แทน Fact 2 ไม่ได้

คอลัมน์ `Total Amount of Fines in Dollars` ใน ProviderInfo คือยอดรวมของ **หน้าต่างหมุน 3 ปี** [ตรวจแล้ว] ยืนยันสองทาง — คู่มือระบุว่าแฟ้ม Penalties คือ *"a list of the fines...in the last three years"* และเมื่อรวมยอดจากแฟ้ม Penalties เองแล้วเทียบกลับ ได้ตรงกัน **ครบทั้ง 6,563 แห่ง** ทั้งจำนวนเงินและจำนวนรายการ ส่วนวันที่ในแฟ้มก็อยู่ในช่วง 2023-06-17 ถึง 2026-05-13 เทียบกับงวด 2026-06-01 พอดี

มันจึงรวมข้ามสถานที่ได้ แต่ **รวมข้ามงวดไม่ได้** เพราะจะนับซ้ำ คำถาม BQ7 ที่ถามว่า "ค่าปรับกระจุกในช่วงเวลาใด" จึงตอบจากคอลัมน์นี้ไม่ได้เลย — นี่คือเหตุผลเชิงออกแบบของ Fact 2 ไม่ใช่การเพิ่มตารางเพื่อเอาคะแนนพิเศษ

หน้าต่างหมุนอันตรายกว่ายอดสะสม — ค่าปรับ “หายไป” ได้เอง

ตอนแรกเอกสารนี้เขียนว่าคอลัมน์นั้นเป็น “ยอดสะสม” ซึ่งผิด และความต่างนี้สำคัญกว่าที่เห็น ยอดสะสมจะโตขึ้นอย่างเดียว แต่ **หน้าต่างหมุนลดลงได้** ค่าปรับที่เก่าเกิน 3 ปีจะ *หลุดออกจากแฟ้มไปเงียบ ๆ* ทำให้ยอดของสถานที่หนึ่ง ลดลงระหว่างสองงวดได้ทั้งที่ไม่มีอะไรผิดพลาด
ผลที่ตามมาสองข้อ: หนึ่ง — ห้ามตีความยอดที่ลดลงว่าเป็น “ข้อมูลพัง” สอง — Fact 2 ที่สะสมค่าปรับจากทุกงวดจะเก็บประวัติได้ **ยาวกว่า** ที่แฟ้มงวดใดงวดเดียวจะให้ได้ ซึ่งเป็นคุณค่าที่แท้จริงของการทำคลังข้อมูล ไม่ใช่แค่คัดลอกไฟล์มาเก็บ

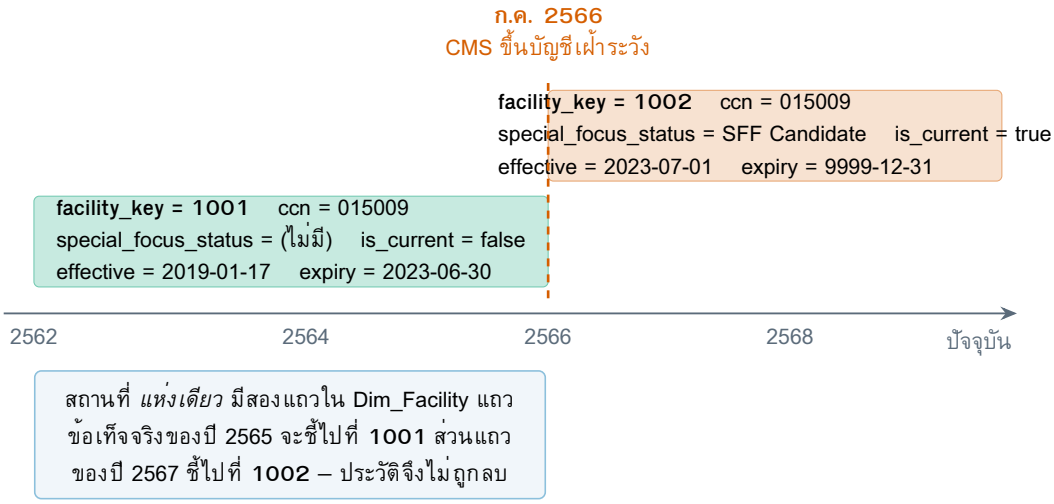
14 ขั้นที่ 3 — เลือก Dimensions

ได้ **หกตาราง** เกินเกณฑ์ขั้นต่ำห้าตาราง โดยห้าตัวแรกเป็น *Conformed Dimension* ที่ใช้ร่วมกันทั้งสอง Fact ซึ่งเป็นสิ่งที่ทำให้ถามข้ามตารางได้ เช่น “ค่าปรับเฉลี่ยต่อเตียง เทียบกับ คะแนนดาวเฉลี่ย ในรัฐเดียวกัน”

Dimension	ชนิด	ใช้กับ	เหตุผล
Dim_Date	—	ทั้งสอง	โจทย์บังคับให้มี ทำที่ระดับวันเพื่อรองรับ Fact 2 และใช้ได้หลายบทบาท: วันเผยแพร่ วันกำหนดโทษ วันตรวจ วันรับรองครั้งแรก
Dim_Facility	**SCD ชนิดที่ 2**	ทั้งสอง	ต้องเป็นชนิดที่ 2 เพราะสถานะ Special Focus และไอคอนการทารุณกรรม *เปลี่ยนไปมาตามเวลา* ถ้าเขียนทับจะตอบ BQ6 ไม่ได้ว่าเคยติดธงแดงหรือไม่
Dim_Geography	ชนิดที่ 1	ทั้งสอง	รวมรหัสไปรษณีย์ เมือง เคาน์ตี รัฐ ภูมิภาค ไว้ตารางเดียว เป็น star ไม่ใช่ snowflake และเป็นจุดเชื่อมกับข้อมูลประชากร
Dim_Ownership	ชนิดที่ 1	ทั้งสอง	แยกออกมาเพราะ BQ2 ต้องการแบ่งกลุ่มด้วยรูปแบบการถือครองโดยตรง และค่าดิบมีหลายค่าที่ต้องยุบเหลือสามกลุ่ม
Dim_Chain	ชนิดที่ 1	ทั้งสอง	รองรับ BQ8 เก็บรหัสเครือ ขนาดเครือ และธงว่าเป็นอิสระหรือไม่
Dim_Penalty_Type	ชนิดที่ 1	Fact 2	แยก “ค่าปรับเป็นเงิน” ออกจาก “ระงับการจ่ายเงิน” ซึ่งมีนัยทางการเงินต่างกันมาก

ตารางที่ 11: หก Dimension และเหตุผลที่ต้องมีแต่ละตัว ตัวที่ต้องระวังที่สุดคือ Dim_Facility เพราะเป็นชนิดที่ 2 ตัวเดียวในผัง

14.1 SCD ชนิดที่ 2 ทำงานอย่างไร



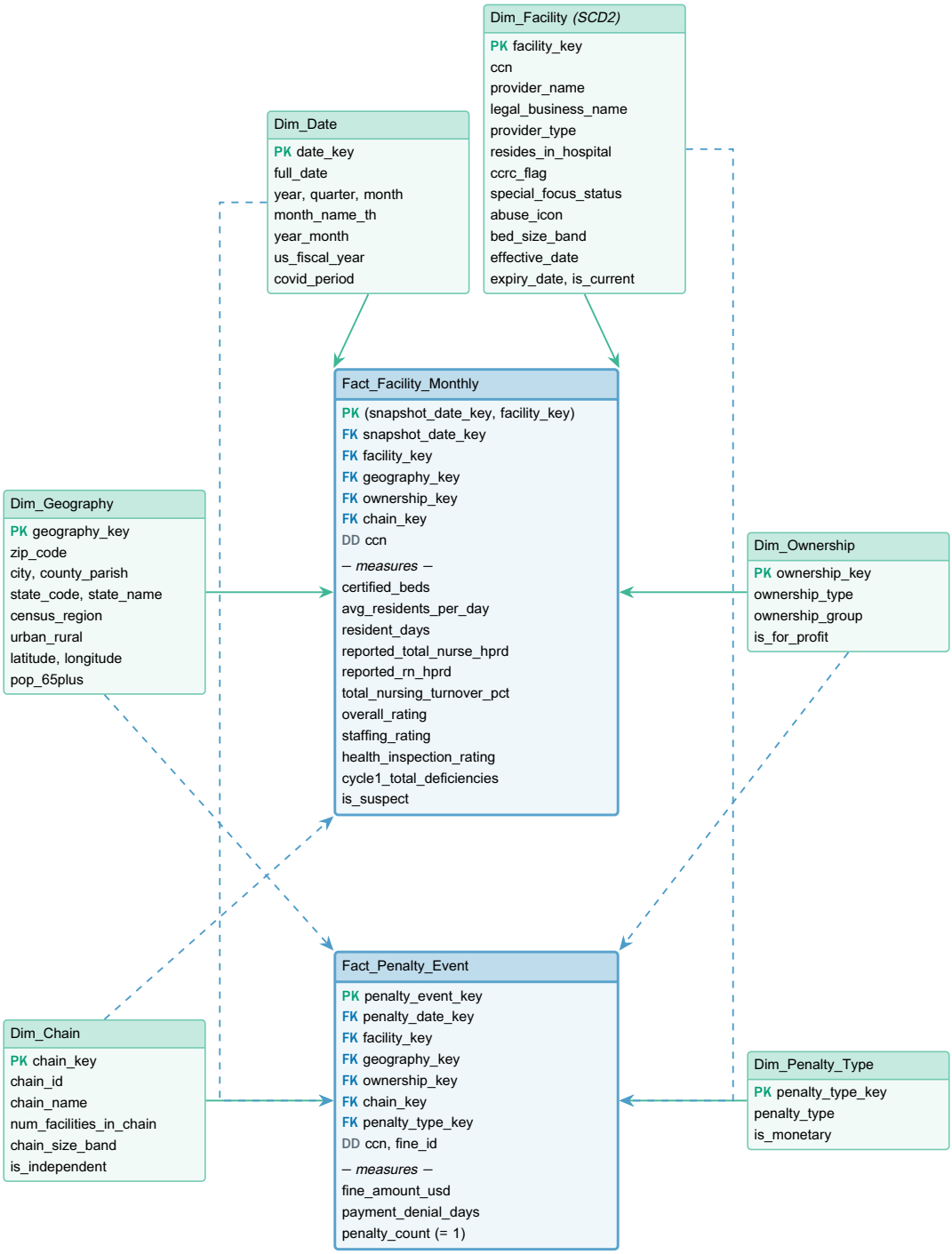
รูปที่ 5: การทำงานของ SCD ชนิดที่ 2 บน Dim_Facility เมื่อสถานะเฝ้าระวังเปลี่ยน ระบบสร้างแถวใหม่แทนการเขียนทับ ทำให้คำถามอย่าง คะแนนตกหลังติดบัญชีเฝ้าระวังหรือไม่ ตอบได้

ผลข้างเคียงของ SCD ชนิดที่ 2 ที่ลืมกันบ่อย

เมื่อ `facility_key` ไม่ใช่หนึ่งต่อหนึ่งกับ CCN อีกต่อไป การเชื่อมตารางด้วย CCN เฉย ๆ จะได้แถวเกินมาเป็นทวีคูณ

ต้องเลือกเวอร์ชันที่ตรงช่วงเวลาเสมอ นั่นคือ `effective_date` $\leq$ วันที่ของข้อเท็จจริง $<$ `expiry_date` กฎข้อนี้ต้องเขียนไว้ในโค้ด ETL เป็นฟังก์ชันเดียวที่ทุกที่เรียกใช้ ไม่ใช่เขียนเงื่อนไขซ้ำในทุก query

15 ขั้นที่ 4 — ผัง Star Schema



รูปที่ 6: ผัง Star Schema สองตาราง Fact และหก Dimension เส้นทึบเขียวคือความสัมพันธ์หลัก เส้นประน้ำเงินคือ conformed dimension ที่ใช้ร่วมกันทั้งสอง Fact — PK คือคีย์หลัก FK คือคีย์นอก DD คือ degenerate dimension ที่เก็บรหัสจากระบบต้นทางไว้ในตาราง Fact เอง

ทำไมวางผังแบบนี้

ผัง Star Schema ที่มีสอง Fact มักถูกวาดกว้าง จนต้องหมุนหน้ากระดาษเป็นแนวนอน เอกสารนี้เลี่ยงด้วยการ **วาง Fact ซ้อนกันแนวตั้ง** แล้ววาง Dimension ที่ใช้ร่วมกันไว้สองข้าง ผังจึงสูงกว่ากว้างและพอดีหน้า A4 แนวตั้ง และได้ผลพลอยได้คือ ดูออกทันทีว่า Dimension ตัวใดใช้ร่วมกัน — ตัวที่มีเส้นออกทั้งขึ้นและลง

16 ข้อควรระวังของผังนี้

1. **Dim_Date ต้องย้อนไปไกลกว่าที่คิด** — คอลัมน์วันที่ได้รับรองครั้งแรก อาจย้อนไปถึงทศวรรษ 2500 จึงสร้างช่วง **1960-01-01 ถึง 2027-12-31** (ราว 24,800 แถว ซึ่งเล็กมาก) พร้อมแถวพิเศษ `date_key = -1` สำหรับค่าที่ไม่ทราบ
2. **ccn ถูกเก็บเป็น degenerate dimension ในทั้งสอง Fact** — เพื่อให้ตรวจย้อนกลับ ไปยังไฟล์ดิบได้โดยไม่ต้องเชื่อมตาราง และทำให้ตรวจสอบการกำจัดข้อมูลซ้ำง่ายขึ้น
3. **ค่าเฉลี่ยของเครือจะไม่เก็บใน Dim_Chain** — เพราะ CMS คำนวณใหม่ทุกเดือน ถ้าเก็บไว้ในตาราง Dimension แบบเขียนทับ กราฟย้อนหลังจะใช้ค่าปัจจุบันทั้งหมดและผิด จะคำนวณจากตาราง Fact ตอนสอบถามแทน
4. **is_suspect ติดอยู่กับแถว Fact ไม่ใช่ Dimension** — เพราะเป็นคุณสมบัติของ *การวัดครั้งนั้น* ไม่ใช่ของสถานที่

ภาคที่ 5

ETL และ Dashboard — แผนงานถัดไป

ภาคนี้ตอบคำถามว่า: ขั้นต่อไปต้องสร้างอะไรบ้าง (ยังไม่ได้เริ่ม)

17 กระบวนการ ETL ห้าขั้น

ขั้น	สิ่งที่ทำ
Extract	เรียก API ปลายทาง A เพื่อได้รายการ snapshot ทั้งหมด เลือกเฉพาะรายไตรมาส ดาวน์โหลด ZIP โดยข้ามไฟล์ที่มีอยู่แล้ว แล้วคลายเฉพาะ 6 CSV ที่ใช้จริง
Clean	ใช้กฎ Q1–Q8 จากภาค 3 ทุกกฎเขียนเป็นฟังก์ชันแยก และเขียนผลลงตารางบันทึกการทำงาน ว่างวดใดแก้อะไรไปกี่แถว
Transform	สร้าง surrogate key สร้าง Dim_Date ทำ SCD ชนิดที่ 2 ให้ Dim_Facility และคำนวณ measure ที่หาได้จากตัวอื่น
Integrate	เชื่อม CMS กับ VBP กับข้อมูลประชากร ด้วย CCN และรหัสรัฐ พร้อมรายงานอัตราการเชื่อมสำเร็จ
Load	เขียนลงฐานข้อมูลไฟล์เดียว แล้วรันการตรวจสอบหลังโหลด: คีย์นอกครบทุกตัว ไม่มีคีย์หลักซ้ำ และจำนวนแถวตรงกับที่ดึงมา

ตารางที่ 12: ห้าขั้นของ ETL ตามที่โจทย์กำหนด หลักการที่ยึดคือรันซ้ำได้โดยไม่ต้องแก้มือ และรันซ้ำแล้วต้องได้ผลเหมือนเดิม

18 แผนแดชบอร์ด

โจทย์กำหนด: measure สรุปอย่างน้อย 3 ตัว กราฟอย่างน้อย 5 กราฟ กราฟตามช่วงเวลาอย่างน้อย 1 กราฟ เปรียบเทียบอย่างน้อย 1 ตัวกรองอย่างน้อย 2 และข้อเสนอแนะอย่างน้อย 5 ข้อ

#	กราฟ	ชนิด	ตอบข้อใด
C1	การ์ดสรุป: อัตราการเข้าพัก คะแนนเฉลี่ย ค่าปรับรวม ชั่วโมงพยาบาล	ตัวเลขสรุป	ภาพรวม
C2	อัตราการเข้าพักและคะแนนเฉลี่ยรายไตรมาส 2562–2569	เส้น (**ตามช่วงเวลา**)	BQ5
C3	เปรียบเทียบคุณภาพและค่าปรับตามรูปแบบการถือครอง	แท่งกลุ่ม (**เปรียบเทียบ**)	BQ2
C4	ชั่วโมงพยาบาล เทียบกับ คะแนนดาว (จุดละหนึ่งแห่ง)	กระจาย + เส้นแนวโน้ม	BQ3
C5	เตียงต่อผู้สูงอายุพันคน รายรัฐ	แผนที่หรือแท่งเรียง	BQ1
C6	มูลค่าค่าปรับรายไตรมาส แยกตามประเภทบทลงโทษ	แท่งซ้อน	BQ7
C7	ตารางเฝ้าระวัง: สถานที่ที่คะแนนลดต่อเนื่อง	ตาราง + แถบสี	BQ6
C8	อัตราลาออก เทียบกับ คะแนน แบ่งตามขนาดเครือ	กล่อง	BQ4, BQ8

ตารางที่ 13: แปดกราฟและคำถามที่แต่ละกราฟตอบ ไม่มีกราฟใดที่ไม่ผูกกับคำถามธุรกิจ ซึ่งเป็นข้อกำหนดของโจทย์โดยตรง

ตัวกรองห้าตัว: ช่วงเวลา · รัฐ · รูปแบบการถือครอง · ขนาดเครือ · สวิตช์รวมหรือไม่รวมแถวที่ติดธง `is_suspect`

19 คะแนนพิเศษที่ตั้งเป้า

1. **สองตาราง Fact** — โจทย์ระบุว่าได้คะแนนพิเศษโดยตรง และมีเหตุผลเชิงออกแบบรองรับ
2. **การใช้ API** — ปลายทาง JSON ของ CMS [ตรวจแล้ว]
3. **ไปป์ไลน์อัตโนมัติและการโหลดแบบเพิ่มเติม** — ข้าม snapshot ที่โหลดแล้ว
4. **การตรวจคุณภาพข้อมูลอัตโนมัติ** — กฎ Q1–Q8 พร้อมตารางบันทึกการทำงาน
5. **แดชบอร์ดด้วยเครื่องมือสมัยใหม่** — Streamlit บนฐานข้อมูล DuckDB
6. **ชุดข้อมูลสำหรับต่อยอดการพยากรณ์** — ตารางแบนสำหรับทำนายว่า สถานที่ใดจะถูกปรับในไตรมาสถัดไป

20 สิ่งที่ต้องทำก่อนเริ่มพัฒนา

สี่ข้อแรกตรวจเสร็จแล้ว สรุปผลไว้ที่นี่ พร้อมชี้ว่าคำตอบไปเปลี่ยนอะไรในเอกสาร

#	สิ่งที่สงสัย	ผลการตรวจ	กระทบอะไร
1	ยอดค่าปรับใน ProviderInfo สะสมกี่ปี	**3 ปีแบบหน้าต่างหมุน** ไม่ใช่สะสมตลอดกาล [ตรวจแล้ว] คู่มือระบุว่า Penalties คือ *"fines...in the last three years"* และพิสูจน์ซ้ำด้วยข้อมูลจริง: ยอดใน ProviderInfo ตรงกับผลรวมจากแฟ้ม Penalties **ครบทั้ง 6,563 แห่ง (100%)** ทั้งจำนวนเงินและจำนวนรายการ	เหตุผลของ Fact 2 **แข็งขึ้น** แต่ต้องเปลี่ยนถ้อยคำ (ดูภาคผนวก ข)
2	ปี 2562 ขาดคอลัมน์อะไรบ้าง	ขาดมากกว่าที่คิด: **78 vs 99 คอลัมน์** และเปลี่ยนระบบตั้งชื่อทั้งหมด [ตรวจแล้ว] อัตราลาออก ชั่วโมงสุดสัปดาห์ เครือ พิกัด ไม่มีเลย	Q7 ยกระดับความรุนแรง และเพิ่มงานทำตารางจับคู่ชื่อคอลัมน์
3	Census API ไม่คืนค่า	**ต้องใช้ API key** [ตรวจแล้ว] เซิร์ฟเวอร์ตอบ 302 ไปหน้า `missing_key.html` จึงได้ body ว่างโดยไม่มี error ทางถอยแบบไม่ต้องใช้คีย์ทดสอบแล้วว่าใช้ได้	BQ1 **ไม่ถูกบล็อกอีกต่อไป** (ดูหัวข้อแหล่งเสริม)
4	`Fine ID` ซ้ำข้าม snapshot ไหม	ไม่ซ้ำเลยภายในงวด (13,710 ค่า/13,710 แถว) **แต่ใช้ไม่ได้ทั้งช่วง** [ตรวจแล้ว] เพราะเพิ่งมีตั้งแต่งวด 202606 และว่างทั้งหมดบนรายการระงับการจ่ายเงิน	คีย์ของ Fact 2 ต้องออกแบบใหม่เป็นสองระบบ (ดูกล่องแดงในภาค 3)

ตารางที่ 14: ผลการตรวจสี่ข้อที่ค้างไว้ ทุกข้อได้คำตอบจากข้อมูลจริงหรือคู่มือฉบับทางการ ไม่มีข้อใดที่ยังต้องเดา

ที่ยังเหลือ

1. ขอ Census API key (ฟรี ใช้เวลาไม่กี่นาที) หรือยืนยันว่าจะใช้ทางถอย
2. เปิด snapshot งวดกลาง (ราวปี 2565) เพื่อปิดช่องว่างของ Q6 และ Q8 ซึ่งตอนนี้ยืนยันได้แค่สองปลาย

ภาคผนวก

A อภิธานศัพท์ ไทย–อังกฤษ

ไทย	อังกฤษ	ความหมายย่อ
คลังข้อมูล	data warehouse	ฐานข้อมูลที่จัดโครงสร้างไว้เพื่อการวิเคราะห์ ไม่ใช่เพื่อรับรายการ
ความละเอียดของข้อมูล	grain	หนึ่งแถวในตาราง Fact แทนอะไรในโลกจริง
ตารางข้อเท็จจริง	fact table	ตารางกลางที่เก็บตัวเลขที่ต้องการวัด
ตารางมิติ	dimension table	ตารางรอบข้างที่บอกว่าตัวเลขนั้นเป็นของใคร ที่ไหน เมื่อไร
มิติร่วม	conformed dimension	ตารางมิติที่ใช้ร่วมกันได้หลายตาราง Fact
มิติเสื่อม	degenerate dimension	รหัสจากระบบต้นทางที่เก็บไว้ในตาราง Fact เอง
คีย์แทน	surrogate key	คีย์ที่ระบบสร้างเอง ไม่ใช่รหัสจากต้นทาง
มิติที่เปลี่ยนช้า ชนิดที่ 2	slowly changing dimension type 2	เก็บประวัติด้วยการเพิ่มแถวใหม่ ไม่เขียนทับ
สถานะรายงวด	periodic snapshot	ตาราง Fact ที่บันทึกสถานะทุกงวด แม้ไม่มีการเปลี่ยนแปลง
ธุรกรรม	transaction fact	ตาราง Fact ที่มีแถวเฉพาะเมื่อเกิดเหตุการณ์
การรวมค่าได้	additivity	บอกว่า measure นั้นเอาไปบวกกันตรง ๆ ได้หรือไม่
อัตราการเข้าพัก	occupancy rate	ผู้พักอาศัยเฉลี่ยต่อวัน หารด้วย จำนวนเตียงที่รับรอง
ชั่วโมงพยาบาลต่อคนต่อวัน	hours per resident day (HPRD)	ชั่วโมงงานพยาบาลทั้งหมด หารด้วย จำนวนผู้พักอาศัย
อัตราการลาออก	turnover rate	สัดส่วนพนักงานที่ลาออกในรอบหนึ่งปี
การโหลดแบบเพิ่มเติม	incremental load	โหลดเฉพาะข้อมูลใหม่ ไม่โหลดซ้ำทั้งหมด
รหัสรับรองสถานพยาบาล	CMS Certification Number (CCN)	รหัสหกหลักประจำสถานพยาบาล มีศูนย์นำหน้าได้
สถานะเฝ้าระวังพิเศษ	Special Focus Facility (SFF)	สถานะที่ CMS ให้กับสถานที่ที่มีปัญหาคุณภาพซ้ำซาก
การจัดซื้อตามคุณค่า	value-based purchasing (VBP)	โครงการที่ผูกเงินที่จ่ายให้เข้ากับผลลัพธ์ด้านคุณภาพ

B คอลัมน์สำคัญของ NH_ProviderInfo ที่จะใช้จริง

จาก 99 คอลัมน์ โครงงานนี้ใช้จริงราว 30 คอลัมน์ ที่เหลือเป็นเชิงอรรถและรายละเอียด การตรวจรายรอบที่ไม่ได้ใช้

คอลัมน์ต้นทาง	ไปเป็นอะไร	หมายเหตุ
CMS Certification Number (CCN)	คีย์ธรรมชาติ	ต้องอ่านเป็นข้อความ (ดูกฎ Q1)
Provider Name, Legal Business Name	Dim_Facility	เปลี่ยนได้เมื่อเปลี่ยนเจ้าของ
City/Town, State, ZIP Code, County/ Parish, Urban	Dim_Geography	
Latitude, Longitude	Dim_Geography	ใช้ทำแผนที่
Ownership Type	Dim_Ownership	ต้องยุบเหลือสามกลุ่ม (กฎ Q5)
Chain Name, Chain ID, Number of Facilities in Chain	Dim_Chain	
Special Focus Status, Abuse Icon	Dim_Facility (SCD2)	เหตุผลที่ต้องใช้ชนิดที่ 2
Number of Certified Beds	measure	ตัวหารของอัตราการเข้าพัก
Average Number of Residents per Day	measure	ตัวตั้งของอัตราการเข้าพัก
Reported Total Nurse Staffing HPRD	measure M3	
Reported RN Staffing HPRD	measure	แยกดูสัดส่วนพยาบาลวิชาชีพ
Total Nursing Staff Turnover	measure M4	ระวังหน่วยวัด (กฎ Q8)
Overall Rating และคะแนนย่อยอีก 3 ตัว	measure M7	มีคอลัมน์ footnote คู่กัน
Rating Cycle 1 Total Number of Health Deficiencies	measure M8	
Total Amount of Fines in Dollars	*ไม่ใช้*	ยอดรวม **หน้าต่างหมุน 3 ปี** ไม่ใช่ยอดสะสมตลอดกาล ใช้ Fact 2 แทน [ตรวจแล้ว]
Processing Date	snapshot_date_-key	วันที่ของ Grain

ตารางที่ 16: การจับคู่คอลัมน์ต้นทางกับปลายทาง แถวสุดท้ายที่ระบุว่าไม่ใช้ คือการตัดสินใจที่สำคัญที่สุดในตารางนี้ และเป็นเหตุผลที่มีตาราง Fact ที่สอง

C คำสั่งที่ใช้ตรวจสอบตัวเลขในเอกสารนี้

ตัวเลขทุกตัวที่กำกับ [ตรวจแล้ว] มาจากการรันคำสั่งเหล่านี้จริง ถ้า CMS ออกข้อมูลใหม่ แล้วรันซ้ำได้ผลต่างไป ตัวเลขในเนื้อหาต้องแก้ตาม

1. **รายการ snapshot ทั้งหมด**

   ```
   curl -sS "https://data.cms.gov/provider-data/api/1/archive/
   aggregate/theme/nursing-homes/relative"
   ```

2. **ดาวน์โหลดและดูรายชื่อไฟล์ใน snapshot**

```
curl -sS -o nh.zip "https://data.cms.gov/provider-data/sites/
default/files/dataset-archives/theme/nursing-homes/
nursing-homes_2026-06-24.zip" && unzip -l nh.zip
```

3. **นับ แถว และ คอลัมน์ ของ CSV** — คลาย ZIP แล้ว อ่าน ด้วย pandas โดย กำหนด `dtype=str` และ `encoding='latin-1'` (ไฟล์ไม่ใช่ UTF-8 ล้วน)
4. **ดึงแถวตัวอย่างจาก API**
   ```
   curl -sS "https://data.cms.gov/provider-data/api/1/datastore/
   query/4pq5-n9py/0?limit=1"
   ```
5. **หาสาเหตุที่ Census คืนค่าว่าง** — ดู redirect แทนที่จะดู body
   ```
   curl -sS -o /dev/null -w "%{http_code} %{redirect_url}" \
   "https://api.census.gov/data/2023/acs/acs5?get=NAME,
   B01001_001E&for=state:*"
   ```
   คืน `302 https://api.census.gov/data/missing_key.html`
6. **ยืนยันว่ายอดค่าปรับเป็นหน้าต่าง 3 ปี** — รวม `Fine Amount` จาก Penalties ตาม CCN แล้วเทียบกับ `Total Amount of Fines in Dollars` ใน ProviderInfo ต้องตรงกัน 100%
7. **ยืนยันว่า Fine ID จำเป็น** — นับ `nunique()` ของ Fine ID เทียบกับจำนวนกลุ่มของ (CCN, Penalty Date, Penalty Type, Fine Amount) ผลต่างคือจำนวนค่าปรับที่คีย์ธรรมชาติจะกลืนหาย
8. **เทียบ โครงสร้าง คอลัมน์ ข้าม ยุค** — คลาย `ProviderInfo_Download.csv` จาก `nursing-homes_-2019-01-17.zip` แล้วเทียบ `set(columns)` กับงวดปัจจุบัน

D แหล่งอ้างอิง

แหล่ง	เกี่ยวข้องกับเอกสารนี้อย่างไร
CMS Provider Data Catalog — Nursing Homes `data.cms.gov/provider-data`	แหล่งข้อมูลหลักทั้งหมด **ต้องอ้างอิงในรายงานฉบับส่งและในวิดีโอ**
CMS Nursing Home Five-Star Quality Rating System — Technical Users' Guide	นิยามของคะแนนดาวและวิธีคำนวณ ต้องใช้ตอนอธิบายว่า M7 หมายถึงอะไร
NH_Data_Dictionary.pdf (อยู่ใน ZIP)	คำจำกัดความของทุกคอลัมน์ **ต้องเปิดก่อนเริ่ม ETL** เพื่อยืนยันข้อสงสัยเรื่องยอดสะสมของค่าปรับ
Kimball & Ross — The Data Warehouse Toolkit	ที่มาของสี่ขั้นตอนการออกแบบ และของคำว่า periodic snapshot กับ transaction fact ซึ่งเป็นแกนของภาค 4

- `group project/69-1-Mini DW & Dashboard-1.pdf` — โจทย์ต้นฉบับ พร้อมเกณฑ์ให้คะแนนหกหมวด