DAY 3 • WORKSHOP

Advanced Data Engineering and Applied Analytics for IT and Developer Teams

Loan Analytics Pipeline

`CSV → Python ETL → MSSQL → Power BI`

โครงวันนี้ — 1 วัน (6 ชม.)

AGENDA

เช้า (3 ชม.)

ETL → MSSQL Star Schema

- สำรวจข้อมูล + ออกแบบ star schema
- ETL: clean → validate → transform
- สร้าง dimensions + fact
- โหลดเข้า MSSQL (de_loan_dw)

บ่าย (3 ชม.)

Power BI Dashboard

- ต่อ Power BI → MSSQL
- สร้าง model (star) + relationships
- DAX measures: Default Rate %, Total Profit
- Dashboard: Loan Analytics

โจทย์: Lending Club Loan Data

Lending Club = แพลตฟอร์มสินเชื่อบุคคล (P2P) รายใหญ่ของสหรัฐฯ

เปิดข้อมูลสินเชื่อย้อนหลังจริง — แต่ละแถว = 1 สินเชื่อ (วงเงิน · ดอกเบี้ย · เกรด A–G · สถานะ)

เราใช้จำลองงาน DE จริง: เปลี่ยน **ข้อมูลดิบ → ตอบคำถามธุรกิจ** ได้

คำถามธุรกิจที่ dashboard จะตอบ:

- 🔴 เกรดไหน **เบี้ยวหนี้สูง** เมื่อเทียบกับดอกเบี้ยที่คิด? → ตั้งราคาถูกไหม
- 💰 สินเชื่อกลุ่มไหน **กำไร/ขาดทุน**?
- 📈 แนวโน้ม **ยอดปล่อยกู้ + อัตราเบี้ยวหนี้** เป็นยังไง?

บริบท

ข้อมูลตัวอย่าง
`loan_sample.csv`

5,020
แถว × 22 คอลัมน์

ตัวอย่างคอลัมน์ที่ต้องล้าง:
`int_rate "13.56%" · term " 36 months"`

ภาพรวม: จากไฟล์ดิบ → dashboard

ARCHITECTURE

ทุกอย่างรันบน **JupyterHub** → โหลดเข้า **MSSQL (sql1)** → Power BI ต่อผ่าน host

CSV
ข้อมูลดิบ (loan)

→

Python ETL
clean + transform

→

Star Schema
fact + 7 dims

→

MSSQL
de_loan_dw

→

Power BI
dashboard

💡 โค้ด ETL = แพ็กเกจ `loan_etl` — ชุดเดียวใช้ได้ทั้ง notebook / CLI / API

ETL = Extract ดึงข้อมูลออกมา → **Transform** แปลง/ล้าง → **Load** โหลดเข้าปลายทาง

🌅 MORNING SESSION

เช้า — ETL → MSSQL Star Schema

จาก CSV ดิบ → ข้อมูลสะอาดใน data warehouse

เครื่องมือ + Setup

เช้า · ขั้นเตรียม

JupyterHub

รัน notebook ทำ ETL บน browser — ไม่ต้องติดตั้งอะไรในเครื่อง

loan_etl

แพ็กเกจ ETL พร้อมใช้ — clean / validate / transform / load

github

คลิกลิงก์เดียว ดึง repo + notebook มาไว้บน hub

🔑 เข้าใช้ JupyterHub — ทุกคนใช้บัญชีกลางนี้

URL `jupyter-hub.minddatatech.com` user `deadmin` pass `deadmin@2026`

▶ ดึง repo เข้า hub

สำรวจข้อมูล — ของจริงไม่เคยสะอาด

Notebook 01

คอลัมน์เก็บมาเป็น “ข้อความปนสัญลักษณ์” — คำนวณไม่ได้ทันที:

คอลัมน์	ดิบ (ข้อความ)	หลังแก้
`int_rate`	`"13.56%"`	13.56 (ตัวเลข)
`term`	`" 36 months"`	36
`emp_length`	`"10+ years"`	10
`issue_d`	`"Dec-2018"`	วันที่จริง

🎯 **เป้าหมาย**

ปั้นข้อมูลให้เป็น **Star Schema**

- 1 fact (fact_loan)
- 7 dimensions (date, grade, purpose, status, geo, ...)

🧪 LAB 1 · ลงมือทำ

⏱ 20 นาที

ตั้งสภาพแวดล้อม + สำรวจข้อมูลจริง

▶ คลิกเดียว — เปิด hub + ดึง repo อัตโนมัติ

🔗 GitHub repo

1. login ด้วยบัญชีกลาง (กล่องขวา →)
2. กดปุ่มเขียวด้านบน — ดึง repo อัตโนมัติ
3. รัน 00_setup — ตั้ง TEAM_ID ของทีม
4. รัน 01_explore — ดูข้อมูลดิบที่ต้องล้าง

🔑 บัญชีกลาง — ใช้ร่วมกัน

URL	`jupyter-hub.minddatatech.com`
user	`deadmin`
pass	`deadmin@2026`

✅ **เสร็จแล้วได้:** repo อยู่บน hub · เห็นปัญหาในข้อมูล · พร้อมลงมือ ETL

Data Model — ทำไมต้อง Star Schema?

เช้า · ออกแบบก่อนทำ

ก่อนเขียน ETL ออกแบบ **“ปลายทาง”** ก่อน — ข้อมูลควรอยู่รูปไหนถึงตอบคำถามธุรกิจได้ง่ายที่สุด

fact (ตารางกลาง)

เก็บ “ตัวเลขวัดผล” — วงเงิน · ดอกเบี้ย
is_default · profit
1 แถว = 1 สินเชื่อ + คีย์ไปหา dim

dimension (ตารางรอบ)

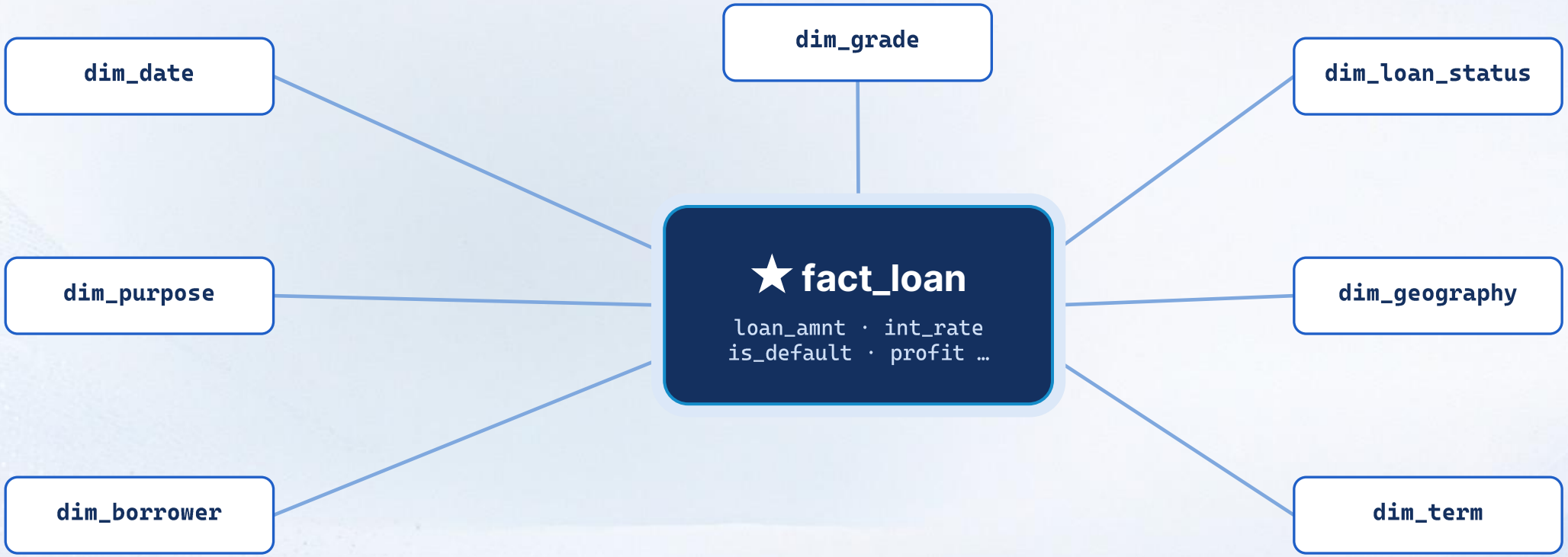
เก็บ “มุมมอง” — เกรด · วันที่ · รัฐ
วัตถุประสงค์
ใช้ filter / group เวลาวิเคราะห์

ทำไม star schema?

- ตอบคำถามแบบ **“X ต่อ Y”** ได้ทันที — default rate ต่อเกรด · กำไรต่อวัตถุประสงค์
- Power BI ทำงานกับ star schema ได้เร็ว + relationship เข้าใจง่าย

Star Schema — fact + 7 dimensions

เช้า · Data Model



📐 **Grain = "1 แถวของ fact แทนอะไร"** → ที่นี่ = 1 สินเชื่อ · กำหนดให้ชัดเป็นขั้นแรกของการออกแบบ schema

📖 CONCEPT

ETL ① Clean — แก้ format ให้คำนวณได้

Notebook 02

เปลี่ยนข้อความปนสัญลักษณ์ → ตัวเลข/วันที่จริง

- ลบ % / ตัดคำว่า months / years
- แปลงเป็น numeric / datetime
- ตัดช่องว่าง (strip) ข้อความ

```
# int_rate "13.56%" -> 13.56
df['int_rate'] = pd.to_numeric(
    df['int_rate'].str
      .replace('%','').str.strip())

# term " 36 months" -> 36
df['term_months'] = (df['term']
    .str.extract(r'(\d+)'))

# issue_d "Dec-2018" -> datetime
df['issue_date'] = pd.to_datetime(
    df['issue_d'], format='%b-%Y')
```

ETL ② Validate + Quarantine

Notebook 02

ตรวจ **ทุกแถว** ตามกฎ business — แถวไม่ผ่าน “ไม่ลบ” แต่เก็บใน quarantine พร้อมเหตุผล (audit)

```
# รวมทุกกฎเป็น mask เดียว
mask = ( (df['loan_amnt'] > 0)
  & (df['funded_amnt'] > 0)
  & (df['annual_inc'] >= 0)
  & df['term_months'].isin([36,60])
  & df['loan_status'].isin(VALID) )
good = df[mask]      # ผ่าน
bad  = df[~mask]     # ไม่ผ่าน
bad['reject_reason'] = why(bad)
```

แต่ละบรรทัดทำอะไร

`mask` เงื่อนไขทุกข้อรวมด้วย & (AND)

`df[mask]` เก็บเฉพาะแถวที่ผ่าน → good

`df[~mask]` แถวไม่ผ่าน (~ = NOT) → bad

`reject_reason` ใส่เหตุผลที่ถูกคัด → audit

ผล: good 5,000 · quarantine 20 แถว

ETL ③ Transform — สร้างความหมายธุรกิจ

Notebook 02

เพิ่มคอลัมน์ที่ **ธุรกิจใช้จริง** จากข้อมูลที่สะอาดแล้ว — ขั้นนี้คือ ~70% ของงาน DE

```
def derive(df):
  df['is_default'] = (df['loan_status']
      .isin(BAD).astype(int))
  df['fico_avg'] = df[['fico_range_low',
      'fico_range_high']].mean(axis=1)
  df['profit'] = (df['total_pymnt']
      - df['funded_amnt'])
  return df
```

คอลัมน์ที่ได้ (ใช้ทำ KPI)

- **`is_default`** 1 = เบี้ยวหนี้ → คำนวณ default rate
- **`fico_avg`** คะแนนเครดิตเฉลี่ย (low+high)/2
- **`profit`** กำไร = เงินได้คืน − เงินปล่อยกู้
- **`status_group`** จัดกลุ่มสถานะ Good / Bad

🧪 LAB 2 · ลงมือทำ

⏱ 30 นาที

ETL: clean → validate → transform

เปิด: `notebooks/02_etl`

🔗 GitHub repo

1. รัน clean — แปลง %, term, วันที่ ให้คำนวณได้
2. รัน validate — แยก good / quarantine + เหตุผล
3. รัน transform — สร้าง is_default · profit · fico_avg
4. ตรวจจำนวนแถว good / quarantine ว่าตรงไหม

✅ **เสร็จแล้วได้:** DataFrame สะอาด + คอลัมน์ธุรกิจ พร้อมขึ้น star schema

โหลดเข้า MSSQL — จบ ETL

Notebook 03

Database
`de_loan_dw`

Login (ร่วมกัน)
`deadmin`

ชื่อตาราง
`<TEAM_ID>_fact_loan`

Reconcile
`row + sum` ตรงกัน

✅ **ทุกคนใช้ `deadmin`** เหมือนกัน — กันชนด้วย prefix ชื่อตาราง (TEAM_ID) → ไม่ต้องสร้าง schema รายทีม

🧪 LAB 3 · ลงมือทำ

⏱ 25 นาที

โหลด Star Schema เข้า MSSQL

เปิด: `notebooks/03_star_schema_load`

🔗 GitHub repo

1. สร้าง dimensions + fact จากข้อมูลที่ transform แล้ว
2. เขียนลง de_loan_dw → ตาราง <TEAM_ID>_*
3. รัน reconcile — row count + ยอดรวมต้องตรง
4. เปิด Power BI Desktop ค้างไว้ ใช้ต่อช่วงบ่าย

✅ **เสร็จแล้วได้:** fact_loan + 7 dimensions อยู่ใน MSSQL — จบกระบวนการ ETL

AFTERNOON SESSION

บ่าย — Power BI Dashboard

ต่อ MSSQL → สร้าง model + DAX → dashboard เล่าเรื่อง

ติดตั้ง Power BI Desktop (Windows)

บ่าย · เตรียม

ฟรี · ใช้ได้เฉพาะ Windows — เลือกติดตั้งได้ 2 ทาง

1 Microsoft Store

แนะนำ

- เปิด Microsoft Store
- ค้นหา "Power BI Desktop"
- กด Get / Install
- อัปเดตอัตโนมัติ

2 ดาวน์โหลด Installer

- **Microsoft Power BI Desktop**
- Download → ได้ไฟล์ .exe
- รันไฟล์ → Next → Install
- เหมาะกับเครื่องที่ปิด Store

⚠ **Mac / Linux:** ไม่มี Power BI Desktop — ใช้ **Power BI Service (เว็บ)** หรือรันผ่าน Windows VM

Power BI: Connect → Model → DAX

บ่าย

1 Connect

Get Data → SQL Server
mssql.minddatatech.com / de_loan_dw
เลือกตาราง <TEAM_ID>_*

2 Model

เชื่อม relationships
fact → dim (star)
mark dim_date เป็น date table

3 DAX

Default Rate %
Total Profit, Total Funded
Avg Int Rate

ตัวอย่าง DAX:

```
Default Rate % = DIVIDE( SUM(is_default), COUNTROWS(fact_loan) ) * 100
```

🧪 LAB 4 · ลงมือทำ

⏱ 45 นาที

ติดตั้ง Power BI + ต่อ MSSQL

เปิด: `Power BI Desktop → SQL Server`

🔗 GitHub repo

1. ติดตั้ง Power BI Desktop (Store หรือ .exe)
2. Get Data → SQL Server → mssql.minddatatech.com / de_loan_dw
3. เลือกตาราง <TEAM_ID>_* ของทีมเรา
4. Model view: เชื่อม fact → dim + เพิ่ม DAX measures

✅ **เสร็จแล้วได้:** model แบบ star + measures (Default Rate %, ROI %) พร้อมใช้

Loan Analytics Dashboard — Layout

บ่าย · ผลลัพธ์

Total Funded	Total Loans	Default Rate %	Avg Int Rate	Total Profit
00.0	00.0	00.0	00.0	00.0

Default Rate % vs Int Rate · by Grade

Total Funded · by Month

Total Profit · by Purpose

Default Rate % · by State

choropleth ตามรัฐ

🧪 LAB 5 · STEP 0

⏱ 5 นาที

ตรวจความพร้อมก่อนเริ่ม

เปิด: `powerbi/LAB5.md — STEP 0`

🔗 GitHub repo

1. โหลด 8 ตาราง: fact_loan + 7 dim → rename ตัด teamXX_ ออก
2. Relationships ครบ 7 เส้น fact → dim (เกิน = ลบ auto-detect)
3. ไม่ต้องใช้ quarantine_loan / measures.dax — เดี๋ยวเขียน DAX เอง
4. ribbon View → ☑ Gridlines ☑ Snap to grid (Page view = Fit to page อยู่แล้ว)

✅ พร้อมแล้ว — ถัดไป: สร้าง DAX measures 5 ตัวให้ KPI cards

DAX Measures สำหรับ KPI Cards

บ่าย · DAX

measure 1 ตัว = KPI 1 ค่า — สร้างที่ **Modeling → New measure (ตาราง fact_loan)**

นับ & รวมยอด
Total Loans · Total Funded

เฉลี่ย & กำไร
Avg Int Rate · Total Profit

อัตราเบี้ยวหนี้
Default Rate % (DIVIDE กันหารศูนย์)

```
Total Loans = COUNTROWS(fact_loan)

Total Funded = SUM(fact_loan[funded_amnt])

Avg Int Rate = AVERAGE(fact_loan[int_rate])

Total Profit = SUM(fact_loan[profit])

Default Rate % =
    DIVIDE(SUM(fact_loan[is_default]),
           COUNTROWS(fact_loan), 0)
```

🧪 LAB 5 · A1

⏱ 5 นาที

KPI Cards — 5 ค่าในการ์ดเดียว

เปิด: `powerbi/LAB5.md – A1`

🔗 GitHub repo

1. Visualizations → Card (new) → คลิกวางบน canvas
2. Data pane → ติ๊ก measures ทั้ง 5 → ลงช่อง Value ให้เอง
3. layout จัดให้เท่ากันอัตโนมัติ — ลากการ์ดยาวเต็มหน้าได้
4. Format: Callout value · Label · ลากสลับลำดับใน Value well

✅ **เสร็จแล้วได้: แถว KPI 5 ค่า ในการ์ดเดียว — Loans · Funded · Int Rate · Profit · Default %**

🧪 LAB 5 · A2

⏱️ 10 นาที

Combo: Default % vs Int Rate

เปิด: `powerbi/LAB5.md — A2`

🔗 GitHub repo

1. Visualizations → Line and clustered column chart
2. X: dim_grade[grade] · Column y: Default Rate % · Line y: Avg Int Rate
3. เรียงเกรด: ... → Sort axis → grade → Ascending (A→G)

✅ **เสร็จแล้วได้:** กราฟตอบคำถาม — ตั้งราคาตามความเสี่ยงถูกไหม

🧪 LAB 5 · A3 + A4

⏱ 16 นาที

Line รายเดือน + Bar by Purpose

เปิด: `powerbi/LAB5.md — A3 · A4`

🔗 GitHub repo

1. Line chart: X = dim_date[year_month] · Y = Total Funded
2. Sort ตามเวลา (Ascending) + เปิด Markers ให้อ่านง่าย
3. คลิกที่ว่าง → Visualizations → Clustered bar chart (แท่งแนวนอน)
4. Y: dim_purpose[purpose] · X: Total Profit → Sort มาก→น้อย = กำไร/ขาดทุน

✅ **เสร็จแล้วได้:** แนวโน้มยอดปล่อยกู้รายเดือน + กำไรรายวัตถุประสงค์

🧪 LAB 5 · A5

⏱ 8 นาที

Map รายรัฐ

เปิด: `powerbi/LAB5.md — A5`

🔗 GitHub repo

1. ตั้งก่อน: dim_geography[addr_state] → Data category = State or Province
2. Filled map: Location = dim_geography[addr_state] · Tooltips = Default Rate %
3. ผูกสี: Fill colors → ปุ่ม fx → Gradient → field = Default Rate %
4. เฉด 3 จุด: #FFEDA0 → #FD8D3C → #BD0026 (เหลืองอ่อน→ส้ม→แดงเข้ม)

✅ เสร็จแล้วได้: หน้า Analytics ครบ — การ์ด KPI 5 ค่า + 4 กราฟ

BOI STEM++
Development Programs

IRIS
BrighterBee

สิ่งที่คุณทำได้หลังวันนี้

สร้าง pipeline จริง CSV → Power BI

- เขียน Python ETL: clean / validate / quarantine / transform
- ออกแบบ star schema → โหลดเข้า MSSQL (de_loan_dw)
- ต่อ Power BI → สร้าง dashboard Loan Analytics

Next steps: dbt · Apache Airflow · Cloud DW (BigQuery/Synapse) · Power BI Advanced · CI/CD

ขอบคุณ — Q & A

amornpan/advanced-de-applied-analytics