
Real-World Simulation: Subscription-Based SaaS Platform

Dataset 1 - subscriptions.csv

```
SubscriptionID,UserID,PlanType,StartDate,EndDate,PriceUSD,IsActive,AutoRenew
SUB001,U001,Basic,2024-01-01,2024-04-01,30.0,true,true
SUB002,U002,Pro,2024-02-15,2024-05-15,90.0,true,false
SUB003,U003,Pro,2024-03-10,2024-06-10,90.0,false,false
SUB004,U001,Premium,2024-04-05,2024-07-05,120.0,true,true
SUB005,U004,Basic,2024-01-20,2024-04-20,30.0,false,false
```

Dataset 2 - user_activity.csv

```
UserID,EventTime,EventType,FeatureUsed
U001,2024-04-07 10:22:00,login,Dashboard
U002,2024-04-08 11:10:00,upload,Reports
U003,2024-04-09 09:45:00,download,Analytics
U001,2024-04-10 16:00:00,logout,Dashboard
U004,2024-04-11 12:00:00,login,Dashboard
```

Creative Exercises

A. Subscription Engagement Score (Real Metric Modeling)

- Combine both datasets.
- Calculate:
 - `active_days = EndDate - StartDate`
 - `events_per_user = count(EventType) grouped by UserID`
- Create a score: `engagement_score = (events_per_user / active_days) * PriceUSD`

B. Anomaly Detection via SQL

- Identify users with:
 - Subscription inactive **but recent activity**
 - AutoRenew is true **but no events in 30 days**

Use SQL views to expose this logic.

C. Delta Lake + Merge Simulation

- Imagine a billing fix needs to be applied:
 - For all Pro plans in March, increase price by \$5 retroactively.
- Use `MERGE INTO` on Delta table to apply the change.

▮ D. Time Travel Debugging

- Show `describe history` of the table before and after the billing fix.
- Query using `VERSION AS OF` to prove the issue existed.

▮ E. Build Tier Migration Table

- Identify users who upgraded:
 - From Basic → Pro → Premium
- Use PySpark with `lag()` function to model this.

▮ F. Power Users Detection

- Define a power user as:
 - Used ≥ 2 features
 - Logged in ≥ 3 times
- Create a separate Delta table `power_users`

▮ G. Session Replay View

- Build a user session trace table using:

```
Window.partitionBy("UserID").orderBy("EventTime")
```

- Show how long each user spent between login and logout events.
-