Project Title: IT Service Delivery & SLA Performance Analytics)

Objective: Simulate an end-to-end helpdesk data pipeline to optimize system uptime and reduce ticket backlog.

Phase 1: Data Generation (Python & Faker)

Goal: Create a realistic, relational database schema that mimics a banking IT environment.
Tools: Python (Faker, pandas, numpy, random)

We will generate 4 linked datasets (tables) based on your request. We will generate roughly **10,000 tickets** spanning a 1-2 year period to show trends.

1. Ticket Metadata (The Fact Table)

- **Primary Key:** ticket_id
- **Columns:**
  - customer_id (Employee/Internal User)
  - subject (e.g., "Login Failure - Core Banking")
  - body (Text description)
  - created_at (Timestamp)
  - closed_at (Timestamp)
  - status (Open, Closed, Pending, Resolved)
  - priority (Low, Medium, High, Critical)
  - category (Hardware, Software, Network, Access)
  - system (e.g., "Core Banking", "Mobile App", "ATM Switch")
  - channel (Email, Phone, Portal)

2. SLA & Performance Metrics

- **Foreign Key:** ticket_id
- **Columns:**
  - sla_limit_hours (Derived from Priority, e.g., Critical = 4hrs)
  - resolution_hours (Diff between created/closed)

    - first_response_minutes
    - sla_breached (Boolean: 1 if resolution > sla_limit)
    - uptime_impact_minutes (How long the system was down)

3. Root Cause & Ownership

- **Foreign Key:** ticket_id
- **Columns:**
    - assigned_team (e.g., "Database Admin", "Network Ops")
    - assigned_engineer
    - root_cause_category (e.g., "Code Bug", "Server Overload", "User Error")
    - fix_type (Permanent, Patch, Workaround)
    - is_repeat_incident (Boolean)

4. Business Impact (Transactions)

- **Foreign Key:** ticket_id (Only for tickets involving transaction failures)
- **Columns:**
    - failed_transaction_count (Number of failed txns during incident)
    - estimated_financial_loss (Currency amount)
    - affected_region (Branch location or Digital)

Phase 2: Processing, EDA & Machine Learning (Python)

Goal: Clean data and derive "Senior-level" insights. **Tools:** pandas, matplotlib/seaborn, scikit-learn (optional but recommended for your level).

1. Data Cleaning

- Handle null values (e.g., open tickets won't have a closed_at date).
- Datatype conversion (Datetime objects).

2. Exploratory Data Analysis (EDA)

We will answer these specific business questions:

- **Pareto Analysis (80/20 Rule):** Which 20% of systems cause 80% of the tickets?
- **Hourly/Daily Trends:** identifying "Peak Load" times for the helpdesk.
- **MTTR (Mean Time To Resolve):** by Team and by System.

3. Advanced Analytics (The "Data Science" Hook)

Since you have a Master's, simply summing up tickets is too easy. Let's add a predictive element:

- **SLA Breach Prediction:** Use a simple Logistic Regression or Random Forest model to predict *probability of SLA breach* based on priority, system, and time_of_day.
    - *Why?* This allows proactive alerting, aligning with your resume point about "identifying patterns."

Phase 3: Data Warehousing (SQL)

Goal: Move processed data into a relational database to simulate a production environment. **Tools:** MySQL (or PostgreSQL/SQLite).

1. Schema Design

We will normalize the data slightly or use a Star Schema:

- **Fact Table:** fact_tickets
- **Dim Tables:** dim_systems, dim_teams, dim_dates

2. Analytical Queries (Resume "Power" Points)

We won't just do SELECT *. We will write queries to demonstrate advanced SQL skills:

- **Window Functions:** Calculate a "Rolling 7-Day Average" of ticket volume to smooth out noise.
- **CTEs (Common Table Expressions):** To break down complex logic (e.g., calculating "Repeat Incidents" by finding tickets from the same user on the same system within 24 hours).
- **Stored Procedures:** A procedure to flag "Chronic Issues" (Systems that breach SLA > 10 times a month).

Phase 4: Visualization (Tableau)

Goal: Build the "Cockpit" for IT Management.

Dashboard 1: Service Overview (The "Executive View")

- **KPIs:** Total Tickets, % SLA Breached, % Automatable (Repeat Incidents).
- **Visuals:** Trend line of Ticket Volume vs. Closed Tickets (Backlog analysis).
- **Filter:** Date Range, Region.

Dashboard 2: System Performance (The "Technical View")

- **KPIs:** System Uptime %, Critical Incidents Count.
- **Visuals:**
  - Heatmap: Systems vs. Root Cause (Where are the bugs coming from?).
  - Bar Chart: Top 5 Unstable Systems by Down Time.

Dashboard 3: Operational Efficiency (The "Team View")

- **KPIs:** Average MTTR, First Response Time.
- **Visuals:**
  - Engineer Performance: Resolution time vs. Ticket Volume (Scatter plot).
  - Fix Type Distribution: Permanent Fix vs. Workaround (Are we solving problems or just patching them?).