# Analysis Plan

Here’s a detailed plan to build your project based on the provided schema and your goals:

**Phase 1: Setting Up the Environment**

1. **Create the Database in Oracle Cloud Autonomous Data Warehouse (ADW):**
   * Set up an Oracle Cloud ADW instance.
   * Define users and roles with permissions for the schemas RAW and PROD.
2. **Create Schemas:**
   * RAW: For raw, unprocessed data.
   * PROD: For transformed, analytics-ready data.
3. **Prepare the Dataset:**
   * Download and inspect the Kaggle dataset.
   * Understand data structure and its alignment with the schema provided in the uploaded PDF.

**Phase 2: Raw Data Ingestion**

1. **Data Loading:**
   * Use SQL Developer or Oracle Data Load to import the CSV files into tables in the RAW schema.
   * Ensure a proper mapping between CSV columns and database table fields.
   * Verify data integrity post-import using SQL queries.
2. **Validation:**
   * Run exploratory SQL queries on RAW schema tables to identify missing values, duplicates, and inconsistencies.

**Phase 3: Data Transformation Using Apache Hop**

1. **ETL Workflow Design:**
   * **Extract:** Read data from the RAW schema.
   * **Transform:**
     + **Granularity Fixes:** Standardize time-based data to uniform intervals (e.g., hourly or daily).
     + **Merging Tables:**
       - Combine complementary tables (e.g., user details with daily activity logs).
       - Create a wide table for consolidated analysis.
     + Handle NULL values by imputation or removal.
     + Standardize units for consistency (e.g., steps, calories, etc.).
     + Derive new features (e.g., weekly average steps, step-to-calorie ratios).
   * **Load:** Write the transformed data to the PROD schema.
2. **Test and Deploy:**
   * Validate the ETL pipeline in Apache Hop.
   * Schedule regular refreshes if required.

**Phase 4: Data Analysis**

1. **Data Exploration:**
   * Write SQL queries to analyze user trends, such as:
     + Average daily steps per user.
     + Correlation between calories burned and active minutes.
     + Patterns in sleep data across age groups or demographics.
     + Seasonal or daily patterns in physical activity.
2. **Create Views and Aggregates:**
   * Build summary views in the PROD schema to simplify Tableau integration.

**Phase 5: Data Visualization Using Tableau**

1. **Connect Tableau to Oracle ADW:**
   * Establish a secure connection to the PROD schema.
2. **Create Dashboards:**
   * **User Activity Overview:** Steps, calories burned, and active minutes.
   * **Sleep Patterns Analysis:** Average sleep duration and sleep efficiency.
   * **Health Correlations:** Active minutes vs. calories burned.
   * **Demographic Insights:** Activity trends by age group, gender, etc.
3. **Advanced Visualizations:**
   * Heatmaps for activity across time.
   * Line graphs to show trends over weeks/months.
   * Comparative analysis (e.g., top 10 most active users).

**Phase 6: Insights and Storytelling**

1. **Analyze Data Insights:**
   * Identify anomalies (e.g., users with extreme values).
   * Highlight trends that indicate health improvements or risks.
   * Create a narrative around how fitness trackers impact health.
2. **Create Reports:**
   * Use Tableau Stories to present findings in an interactive format.

**Phase 7: Project Documentation and Deployment**

1. **Portfolio Website:**
   * Include:
     + Project objective.
     + Architecture diagram of the data pipeline.
     + Key insights and dashboards.
     + Link to Tableau Public (if applicable).
2. **Technical Documentation:**
   * Include ETL pipeline details, schema definitions, and Tableau workbook details.

**Additional Tools and Skills:**

* **Data Profiling:** Use Python or R for initial data profiling.
* **Version Control:** Use Git to manage ETL pipeline scripts and SQL queries.
* **Optimization:** Indexing and partitioning tables in Oracle ADW to improve performance.

This project will showcase your expertise in data engineering (ETL pipeline design), data analytics (insight generation), and data visualization (interactive dashboards).

# Setting up New Schema

In Oracle Autonomous Data Warehouse (ADW), schemas are associated with users. To implement the multi-tiered table design approach, you can create a new user to act as the schema owner and then assign appropriate permissions to this user and others. Below are the steps:

**Step 1: Create a New Schema (User)**

1. **Login to ADMIN Schema:**
   * Connect to your database using SQL Developer, SQL\*Plus, or another client with ADMIN credentials.
2. **Create a New User:**
3. CREATE USER FITNESS\_DATA IDENTIFIED BY StrongPassword;

Replace StrongPassword with a secure password.

1. **Grant Necessary Privileges:**
2. GRANT CREATE SESSION TO FITNESS\_DATA; -- Allows the user to log in
3. GRANT CREATE TABLE TO FITNESS\_DATA; -- Allows the user to create tables
4. GRANT CREATE VIEW TO FITNESS\_DATA; -- Allows the user to create views
5. GRANT CREATE SEQUENCE TO FITNESS\_DATA; -- Allows the user to create sequences
6. GRANT UNLIMITED TABLESPACE TO FITNESS\_DATA; -- Provides ample storage for the user

**Step 2: Configure Multi-Tiered Tables in the Schema**

1. **Define Table Prefixes:**
   * RAW\_: For raw data (directly imported CSV files).
   * STG\_: For intermediate or staging tables (used during transformations).
   * PROD\_: For production-ready, analytics-focused tables.
2. **Example Table Creation:**
3. -- Raw table
4. CREATE TABLE FITNESS\_DATA.RAW\_ACTIVITY (
5. USER\_ID NUMBER,
6. ACTIVITY\_DATE DATE,
7. STEPS NUMBER,
8. CALORIES NUMBER,
9. ACTIVE\_MINUTES NUMBER
10. );
11. -- Staging table
12. CREATE TABLE FITNESS\_DATA.STG\_ACTIVITY\_DAILY (
13. USER\_ID NUMBER,
14. ACTIVITY\_DATE DATE,
15. DAILY\_STEPS NUMBER,
16. DAILY\_CALORIES NUMBER,
17. ACTIVE\_MINUTES\_TOTAL NUMBER
18. );
19. -- Production table
20. CREATE TABLE FITNESS\_DATA.PROD\_ACTIVITY\_SUMMARY (
21. USER\_ID NUMBER,
22. ACTIVITY\_WEEK\_START DATE,
23. AVG\_DAILY\_STEPS NUMBER,
24. AVG\_DAILY\_CALORIES NUMBER,
25. TOTAL\_ACTIVE\_MINUTES NUMBER
26. );

**Step 3: Assign Permissions to Other Users**

If you plan to allow other users or tools (like Apache Hop or Tableau) to access the schema:

1. **Create an ETL User (for Apache Hop):**
2. CREATE USER ETL\_USER IDENTIFIED BY AnotherSecurePassword;
3. GRANT CREATE SESSION TO ETL\_USER; -- Allows login
4. GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES TO ETL\_USER; -- Grants table access
5. **Create a Visualization User (for Tableau):**
6. CREATE USER VISUAL\_USER IDENTIFIED BY AnotherSecurePassword;
7. GRANT CREATE SESSION TO VISUAL\_USER; -- Allows login
8. GRANT SELECT ON FITNESS\_DATA.PROD\_ACTIVITY\_SUMMARY TO VISUAL\_USER; -- Read-only access
9. **Optional: Provide Access to ADMIN:**
   * You can give the ADMIN user selective access for troubleshooting or full schema access:
   * GRANT SELECT ON FITNESS\_DATA.\* TO ADMIN; -- Read-only for all tables

**Step 4: Automate Granting Future Table Permissions**

Use a procedure to automatically grant privileges on future tables:

1. **Create Grant Procedure:**
2. CREATE OR REPLACE PROCEDURE GRANT\_TABLE\_ACCESS IS
3. BEGIN
4. FOR tbl IN (SELECT table\_name FROM user\_tables) LOOP
5. EXECUTE IMMEDIATE 'GRANT SELECT, INSERT, UPDATE, DELETE ON ' || tbl.table\_name || ' TO ETL\_USER';
6. EXECUTE IMMEDIATE 'GRANT SELECT ON ' || tbl.table\_name || ' TO VISUAL\_USER';
7. END LOOP;
8. END;
9. **Run After Creating New Tables:**
10. BEGIN
11. GRANT\_TABLE\_ACCESS;
12. END;

**Step 5: Test and Validate Permissions**

1. **Login as ETL\_USER:** Verify the ability to query, insert, and update data in RAW\_ and STG\_ tables.
2. **Login as VISUAL\_USER:** Ensure only SELECT access to PROD\_ tables.

**Best Practices**

* Use strong passwords for all users.
* Regularly audit user access and table privileges.
* Automate granting privileges if tables are frequently added or modified.
* Consider role-based access control (RBAC) for scalability in larger projects.

This setup ensures a secure and organized implementation of the multi-tiered table design.

**1. ETL Transformation Plan for Staging and PROD Tables**

**Overview:**

The goal is to:

* **Staging (STG\_):** Clean and standardize the raw data while retaining individual table structures.
* **Production (PROD\_):** Create consolidated tables optimized for analysis and visualization.

**Transformation Steps using Apache Hop:**

**Step 1: Import Raw Tables into STG\_**

* **Input Tables:** RAW\_ tables like RAW\_DAILYACTIVITY\_MERGED\_APR\_MAY, RAW\_HEARTRATE\_SECONDS\_MERGED\_APR\_MAY.
* **Transformations:**
  + **Standardize Data Types:**
    - Convert VARCHAR2 timestamp fields (e.g., ACTIVITYDAY, TIME) to proper DATE or TIMESTAMP types.
    - Normalize units for consistency (e.g., distance in km, weight in kg).
  + **Clean Data:**
    - Handle missing values: Impute, fill, or mark as NULL.
    - Remove duplicates using deduplication nodes.
  + **Field Renaming:**
    - Rename columns for clarity (e.g., VALUE → HEARTRATE\_VALUE in heart rate data).
* **Output:** Create STG\_ tables mirroring the raw data schema but standardized.

**Step 2: Merge and Aggregate in PROD\_**

* **Input Tables:** Staging (STG\_) tables.
* **Transformations:**
  + **Merge Related Data:**
    - Combine DAILYACTIVITY, DAILYCALORIES, and DAILYSTEPS tables into PROD\_DAILY\_SUMMARY.
    - Join WEIGHTLOGINFO with SLEEPDAY to analyze weight and sleep relationships.
  + **Aggregate Data:**
    - Hourly Data: Sum hourly steps, calories, and intensities into PROD\_HOURLY\_ACTIVITY.
    - Daily Data: Compute averages and totals (e.g., total steps, active minutes).
  + **Feature Engineering:**
    - Calculate derived metrics such as:
      * Steps per Calorie = TOTALSTEPS / CALORIES.
      * Sleep Efficiency = TOTALMINUTESASLEEP / TOTALTIMEINBED.
      * BMI Change Rate over weeks.
* **Output:** Production tables (PROD\_).

**2. Analysis Plan and Insights**

**Key Analysis Objectives:**

1. **User Activity Patterns:**
   * Analyze trends in daily and hourly steps, calories burned, and active minutes.
   * Identify periods of high/low activity.
2. **Health Metrics:**
   * Correlation between sleep duration and calories burned.
   * Weight changes over time and their relationship with activity levels.
3. **Comparative Analysis:**
   * Compare activity metrics across months (March-April vs. April-May).
   * Segment users into activity tiers (e.g., sedentary, moderately active).
4. **Advanced Insights:**
   * Identify users at risk based on sedentary behavior or irregular heart rates.
   * Track adherence to fitness goals (e.g., steps > 10,000/day).

**3. Visualization Plan Using Tableau**

**Dashboard Types:**

1. **Interactive Dashboards:**
   * **Daily Activity Overview:**
     + KPIs: Steps, calories, active minutes.
     + Line charts showing daily trends.
     + Filters for user selection.
   * **Sleep vs. Activity:**
     + Scatter plots for TOTALTIMEINBED vs. TOTALMINUTESASLEEP.
     + Heatmaps showing sleep patterns by time.
   * **Heart Rate Analysis:**
     + Box plots of heart rate variability.
     + Histograms for heart rate distributions.
2. **Static Dashboards:**
   * Monthly summaries showing:
     + Top active users.
     + Average calories burned.
     + Most common activity periods.

**Storyboards:**

* **User Health Journey:**
  + Slide 1: Overview of user activity and health metrics.
  + Slide 2: Deep dive into sleep and weight trends.
  + Slide 3: Recommendations and insights.

**Interactive Features:**

* Filters for selecting:
  + Date ranges.
  + Specific users.
  + Activity types (steps, sleep, calories).
* Drill-down capabilities:
  + Click on KPIs to explore detailed trends.

This plan integrates raw data transformation into insights-ready tables and provides compelling visualizations to showcase key findings interactively. Let me know if you want further assistance in building any of these components!