DESIGN AND IMPLEMENTATION OF RAPIDO CAPTAIN PRODUCTIVITY TRACKING USING ORACLE SQL

1.Introduction:

This project simulates a Rapido-like system to track captain (driver) productivity. It stores information about rides, login/logout sessions, and generates daily performance reports.

The system helps Rapido:

a) Calculate working hours (active time).
b) Track rides completed vs cancelled.
c) Compute earnings.
d) Analyze captain ratings.
e) Provide both historical fixed reports (via tables) and real-time analytics (via views).

2.Objectives:

a) Create a database for captains, rides, and login sessions.
b) Track earnings, ratings, and online hours.
c) Automate daily productivity report generation.
d) Provide a real-time summary view for management.

3.Entities & relationships:

Captains → own rides & sessions.

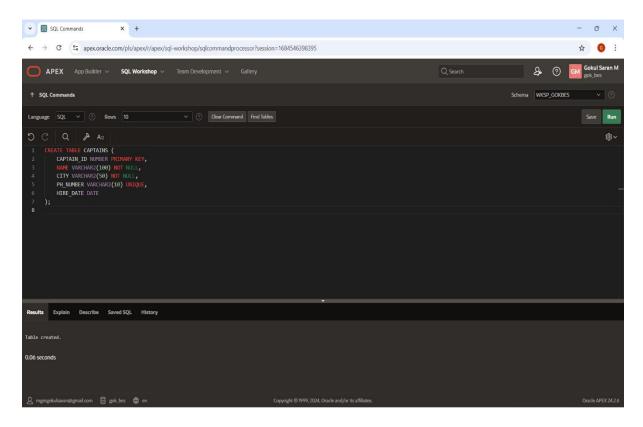Rides → linked to captains.

Login_Sessions → linked to captains (tracks online hours).
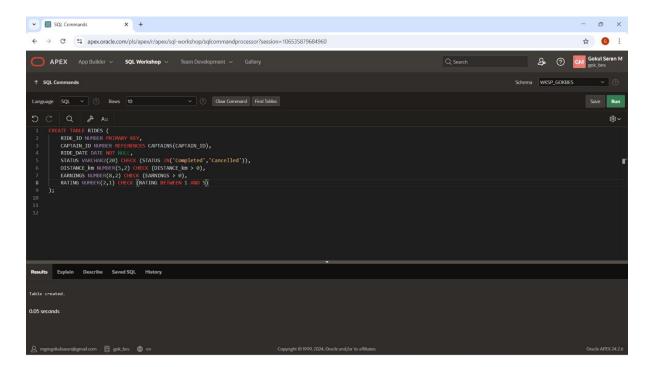
Daily_Reports → summarized data per captain per day.

4.Database Schema (DDL):

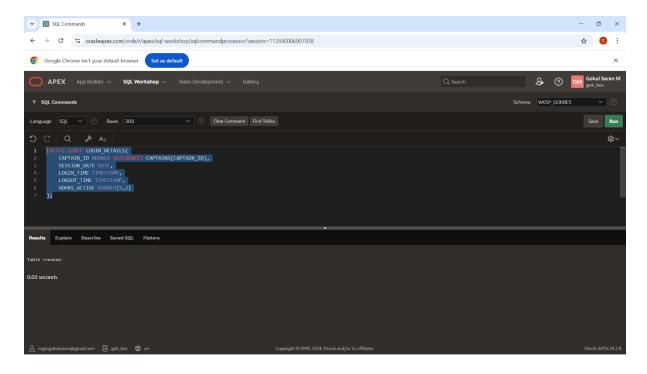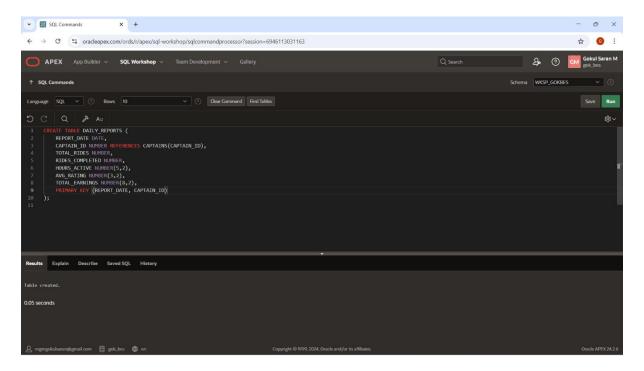### (a)Captains
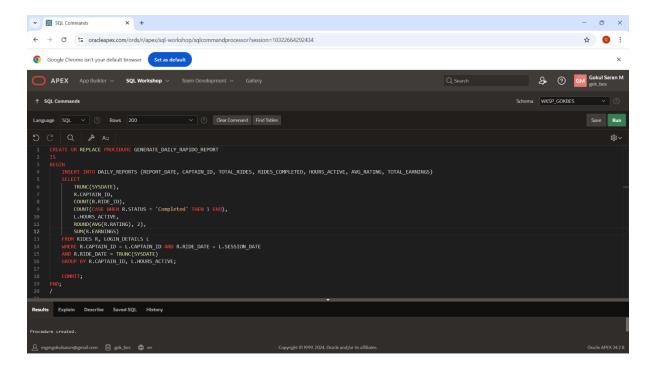

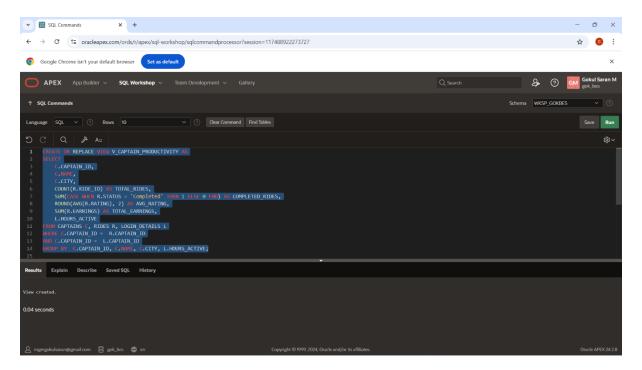
(b)Rides

(c)Login Sessions

(d)Daily Reports

5. Stored Procedure (Generate Daily Reports):


```sql
CREATE OR REPLACE PROCEDURE GENERATE_DAILY_RAPIDO_REPORT
IS
BEGIN
    INSERT INTO DAILY_REPORTS (REPORT_DATE, CAPTAIN_ID, TOTAL_RIDES, RIDES_COMPLETED, HOURS_ACTIVE, AVG_RATING, TOTAL_EARNINGS)
    SELECT
        TRUNC(SYSDATE),
        R.CAPTAIN_ID,
        COUNT(R.RIDE_ID),
        COUNT(CASE WHEN R.STATUS = 'Completed' THEN 1 END),
        L.HOURS_ACTIVE,
        ROUND(AVG(R.RATING), 2),
        SUM(R.EARNINGS)
    FROM RIDES R, LOGIN_DETAILS L
    WHERE R.CAPTAIN_ID = L.CAPTAIN_ID AND R.RIDE_DATE = L.SESSION_DATE
    AND R.RIDE_DATE = TRUNC(SYSDATE)
    GROUP BY R.CAPTAIN_ID, L.HOURS_ACTIVE;

    COMMIT;
END;
/
```

Procedure created.

6.View (Real-time Summary):


```sql
CREATE OR REPLACE VIEW V_CAPTAIN_PRODUCTIVITY AS
SELECT
    C.CAPTAIN_ID,
    C.NAME,
    C.CITY,
    COUNT(R.RIDE_ID) AS TOTAL_RIDES,
    SUM(CASE WHEN R.STATUS = 'Completed' THEN 1 ELSE 0 END) AS COMPLETED_RIDES,
    ROUND(AVG(R.RATING), 2) AS AVG_RATING,
    SUM(R.EARNINGS) AS TOTAL_EARNINGS,
    L.HOURS_ACTIVE
FROM CAPTAINS C, RIDES R, LOGIN_DETAILS L
WHERE C.CAPTAIN_ID = R.CAPTAIN_ID
AND C.CAPTAIN_ID = L.CAPTAIN_ID
GROUP BY C.CAPTAIN_ID, C.NAME, C.CITY, L.HOURS_ACTIVE;
```

View created.

0.04 seconds

7. Sample Queries:

(a)Insert Data into Captain Table

```
8
9   INSERT INTO CAPTAINS(CAPTAIN_ID, NAME, CITY, PH_NUMBER, HIRE_DATE) VALUES(101, 'SARAN', 'MADURAI','9865138551','12/2/2022');
0
```

(b)Insert Data into Rides Table

```
1   INSERT INTO RIDES (RIDE_ID, CAPTAIN_ID, RIDE_DATE, STATUS, DISTANCE_km, EARNINGS, RATING)
2   VALUES (15, 101, TRUNC(SYSDATE), 'Completed', 9, 135, 5);
```

(c)Insert Data into Login_Details Table

```
1   INSERT INTO LOGIN_DETAILS (CAPTAIN_ID, SESSION_DATE, LOGIN_TIME, LOGOUT_TIME, HOURS_ACTIVE)
2   VALUES (
3       107,
4       TRUNC(SYSDATE),
5       TIMESTAMP '2025-08-21 09:00:00',
6       TIMESTAMP '2025-08-21 17:30:00',
7       ROUND((CAST(TIMESTAMP '2025-08-21 17:30:00' AS DATE)
8           - CAST(TIMESTAMP '2025-08-21 09:00:00' AS DATE)) * 24, 2)
9   );
```

(d) Generate Daily Report (Procedure Run)

```
1   BEGIN
2       generate_daily_rapido_report;
3   END;
```
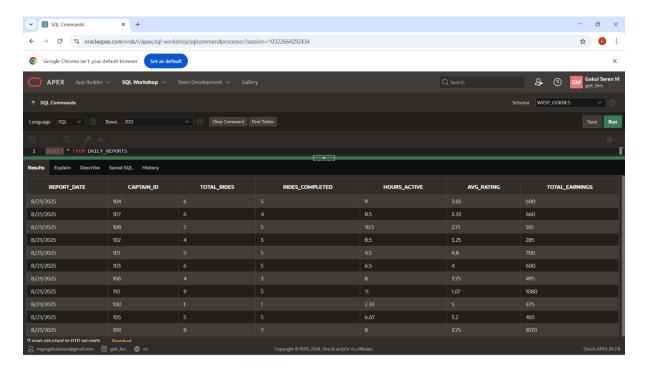
(e) View Fixed Reports

```
1   SELECT * FROM DAILY_REPORTS
```
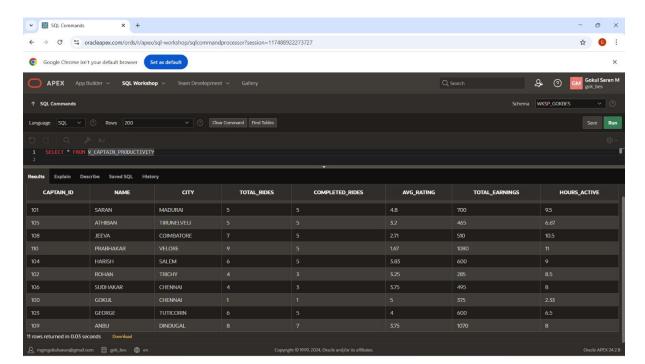
(f) View Real-Time Summary

```
SELECT * FROM V_CAPTAIN_PRODUCTIVITY
```

8. Result:

(a) View Daily_Reports

(b) View Real-time Summary

9. Conclusion:

This project successfully:

1. Captures ride & login details.
2. Tracks earnings, ratings, and productivity.
3. Provides both frozen daily snapshots (via daily_reports) and real-time insights (via V_Captain_Productivity_Summary view).