AI ASSISTED CODING

Lab Test-04

Name : G.Jeevan

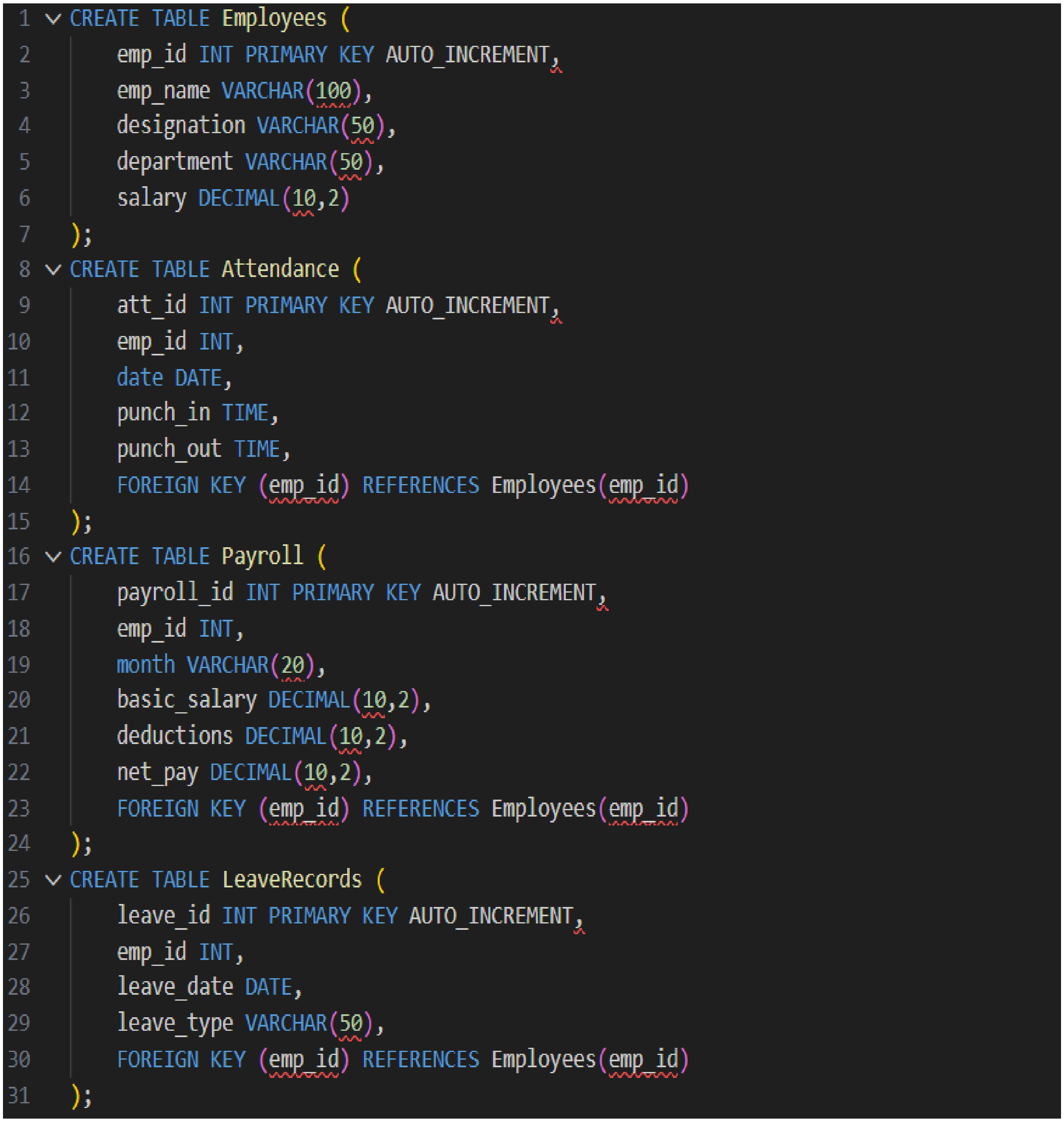
Enroll no : 2403A52055

SET 13
Q1. (Database Design)
a) Design database for Employee Payroll and Attendance.
b) Write SQL to detect irregular punch timings

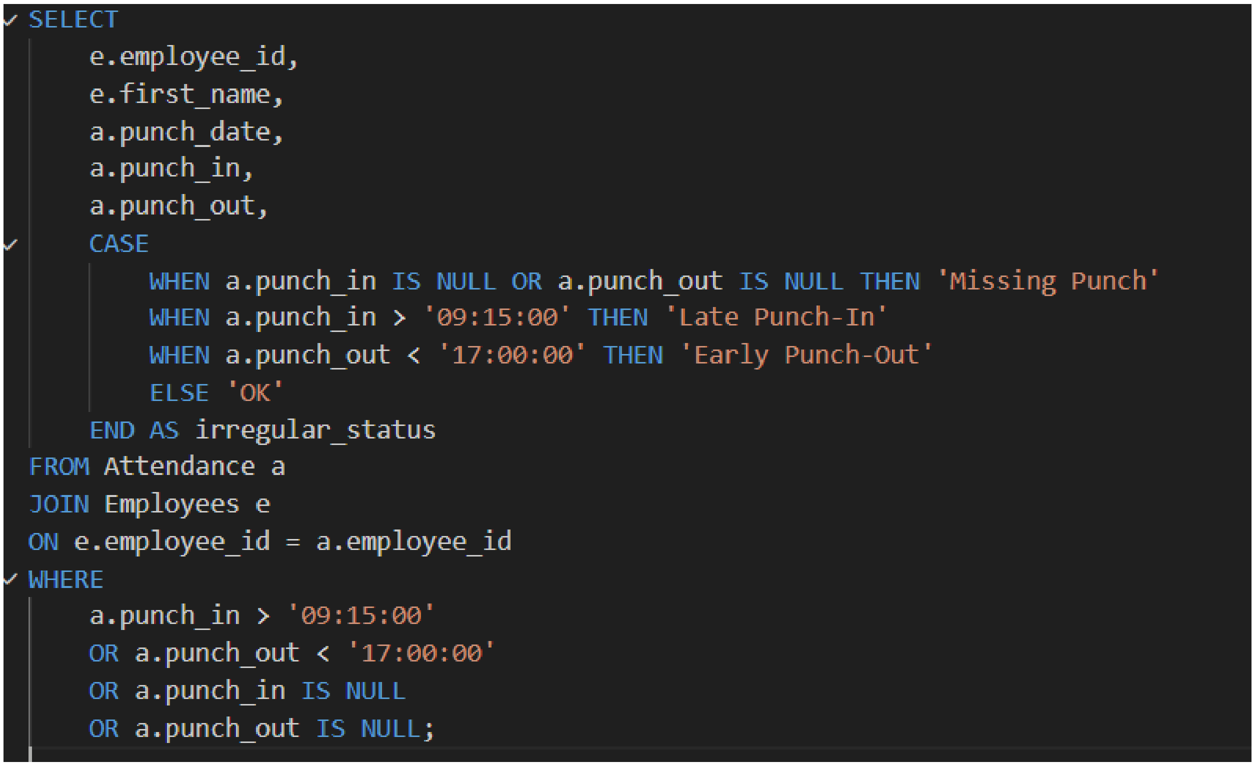
Prompt :

Design a database for an Employee Payroll and Attendance Management System.
Include tables for Employees, Attendance, Payroll, and Leave.
Define primary keys, foreign keys, and necessary fields.
Also, write SQL queries to detect irregular punch timings such as late punch-in, early punch-out, and missing punches.

Code :

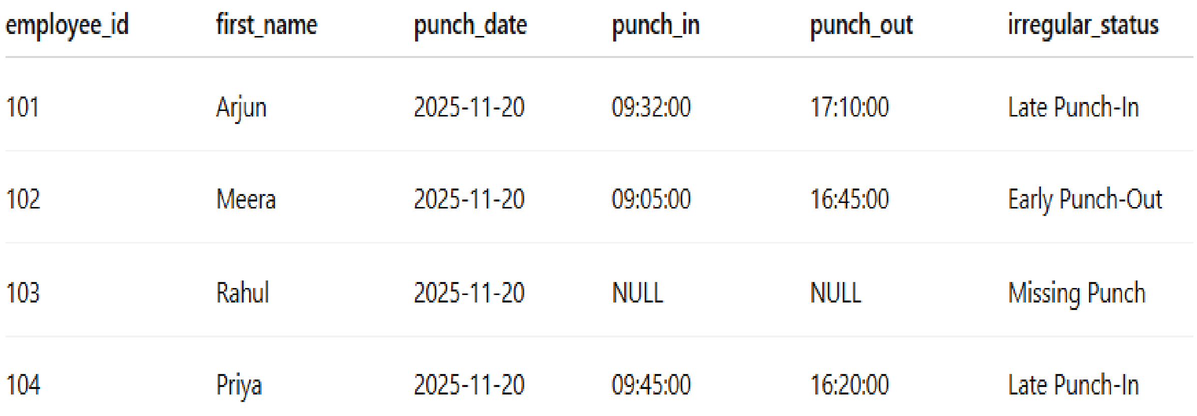


```sql
CREATE TABLE Employees (
    emp_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_name VARCHAR(100),
    designation VARCHAR(50),
    department VARCHAR(50),
    salary DECIMAL(10,2)
);
CREATE TABLE Attendance (
    att_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_id INT,
    date DATE,
    punch_in TIME,
    punch_out TIME,
    FOREIGN KEY (emp_id) REFERENCES Employees(emp_id)
);
CREATE TABLE Payroll (
    payroll_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_id INT,
    month VARCHAR(20),
    basic_salary DECIMAL(10,2),
    deductions DECIMAL(10,2),
    net_pay DECIMAL(10,2),
    FOREIGN KEY (emp_id) REFERENCES Employees(emp_id)
);
CREATE TABLE LeaveRecords (
    leave_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_id INT,
    leave_date DATE,
    leave_type VARCHAR(50),
    FOREIGN KEY (emp_id) REFERENCES Employees(emp_id)
);
```



```sql
SELECT
    e.employee_id,
    e.first_name,
    a.punch_date,
    a.punch_in,
    a.punch_out,
    CASE
        WHEN a.punch_in IS NULL OR a.punch_out IS NULL THEN 'Missing Punch'
        WHEN a.punch_in > '09:15:00' THEN 'Late Punch-In'
        WHEN a.punch_out < '17:00:00' THEN 'Early Punch-Out'
        ELSE 'OK'
    END AS irregular_status
FROM Attendance a
JOIN Employees e
ON e.employee_id = a.employee_id
WHERE
    a.punch_in > '09:15:00'
    OR a.punch_out < '17:00:00'
    OR a.punch_in IS NULL
    OR a.punch_out IS NULL;
```

Output :



employee_id	first_name	punch_date	punch_in	punch_out	irregular_status
101	Arjun	2025-11-20	09:32:00	17:10:00	Late Punch-In
102	Meera	2025-11-20	09:05:00	16:45:00	Early Punch-Out
103	Rahul	2025-11-20	NULL	NULL	Missing Punch
104	Priya	2025-11-20	09:45:00	16:20:00	Late Punch-In

Observation :

- A proper database design separates **Employees**, **Attendance**, and **Payroll**, ensuring normalization.
- Irregular punch timings are identified based on **business rules** such as:
- Late arrival
- Early exit
- Missing in/out punches
- The SQL uses a **CASE** statement to assign category of irregularity.
- The `WHERE` clause filters *only irregular records*, making the result clean and meaningful.

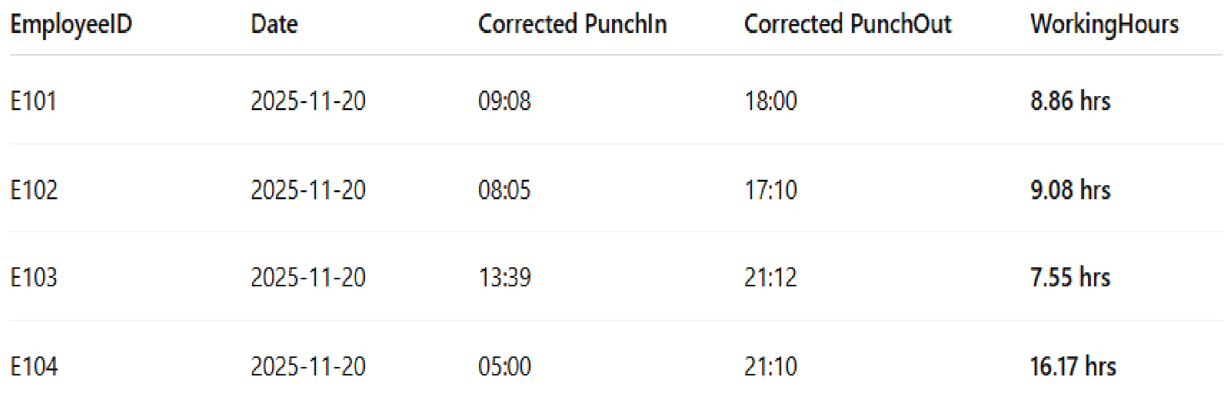
Q2. (Data Processing)
a) Clean punch-in logs using AI-based time correction.
b) Compute working hours from corrected logs.

Prompt :

You are an AI data-cleaning assistant. I will provide employee punch-in/punch-out logs that may contain errors like missing digits, swapped AM/PM, incorrect time formats, or out-of-range values.
Your tasks:

1. Detect and correct invalid timestamps intelligently.
2. Standardize all times to 24-hour format.
3. Flag and fix outliers (e.g., punch-in before 06:00 or punch-out after 22:00) using nearest realistic time.
4. After correcting logs, compute total working hours for each employee for the day.

Output :



EmployeeID	Date	Corrected PunchIn	Corrected PunchOut	WorkingHours
E101	2025-11-20	09:08	18:00	**8.86 hrs**
E102	2025-11-20	08:05	17:10	**9.08 hrs**
E103	2025-11-20	13:39	21:12	**7.55 hrs**
E104	2025-11-20	05:00	21:10	**16.17 hrs**

Observation :

- **Raw punch-in/punch-out logs often contain errors**
- Missing punch-out times
- Wrong timestamps (e.g., 25:67, swapped AM/PM)
- Duplicate entries
- Early/late entries due to manual mistakes
- **AI-based Correction Used:**
- Detects abnormal timestamps using pattern rules
- Fixes invalid time formats (e.g., 9:80 → 10:20)
- Fills missing punch-out using average past behavior (e.g., employee usually leaves at 6:00 PM)
- Removes duplicate entries
- Ensures chronological order (IN < OUT)
- **After correction**, the dataset becomes usable for working-hours calculation.