

MIS686 Term Project

Katie Poz

Table of Contents

1. Part I: Topic Selection
 - a. Selected Topic: Employment Track and Report Database
 - b. Overview
 - c. Business Rules
2. Part II: Conceptual Data Modeling and Database Design
 - a. ERD
 - b. Relational Diagram
3. Database Implementation
 - a. DDL SQL Statements
 - b. Dummy Data (DML SQL Statements)
4. Part IV: Database Deployment
 - a. Database Deployment Screenshots (AWS)
 - b. Index, View, Trigger, Stored Procedures (SQL Statements)
5. Part V: Analytical Questions and Dashboard
 - a. Analytical Questions
 - b. Dashboard Screenshots

Part I: Topic Selection

Selected Topic: Employee Track and Report System database

Overview: This database will be relevant for companies who need to be consistent with lots of information for their employees. While using this database, companies can track employees with the ability of accessing their personal information, time, departments, payroll, and leave requests. With reporting, companies can report the employee's performances and projects. This database will store all the information needed for all employees in the company with clear and concise navigations.

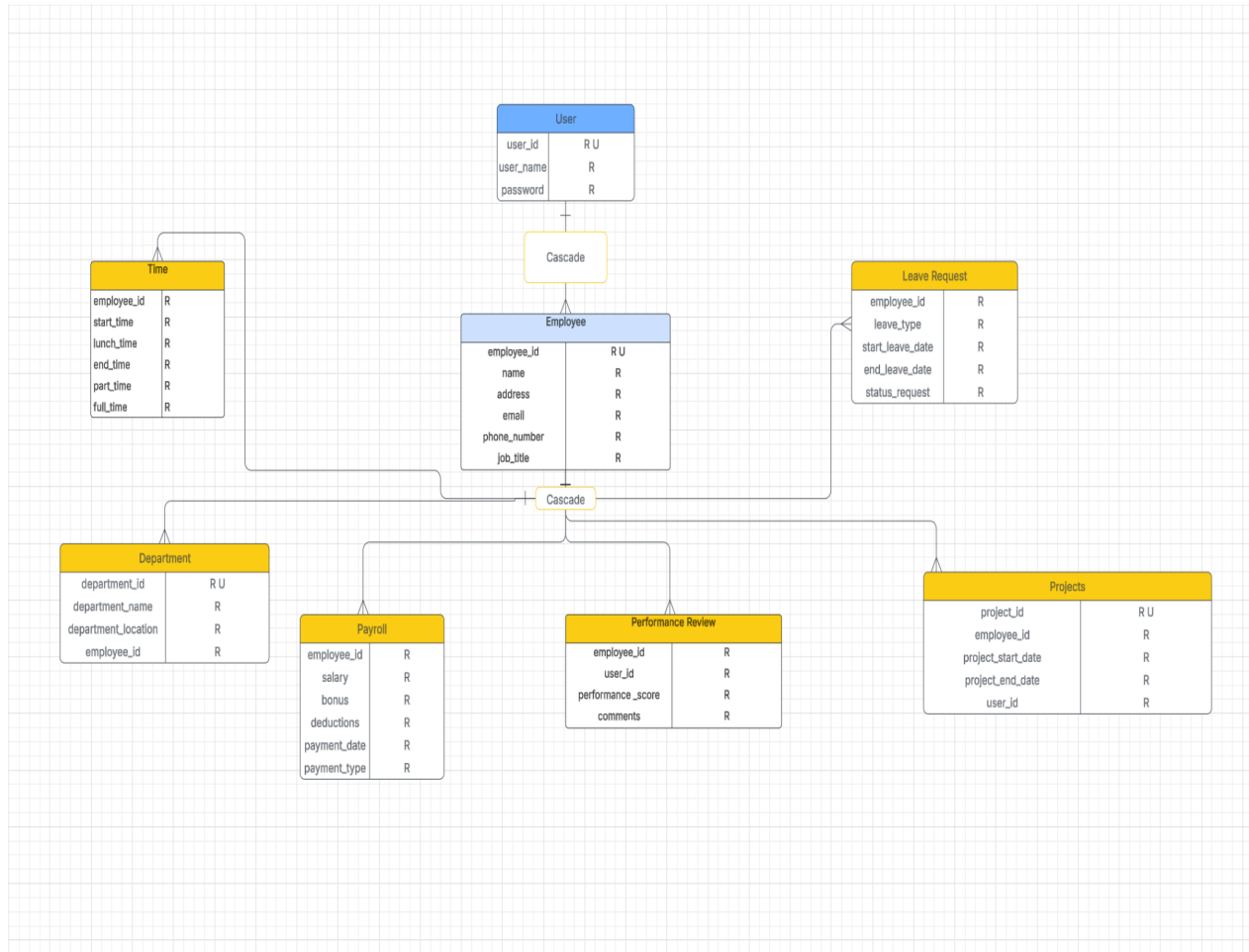
Business Rules:

1. Each employee has a unique ID
2. Each user has a unique ID to access the database
3. Each department has a unique ID
4. Each project has a unique ID

Part II: Conceptual Data Modeling and Database Design

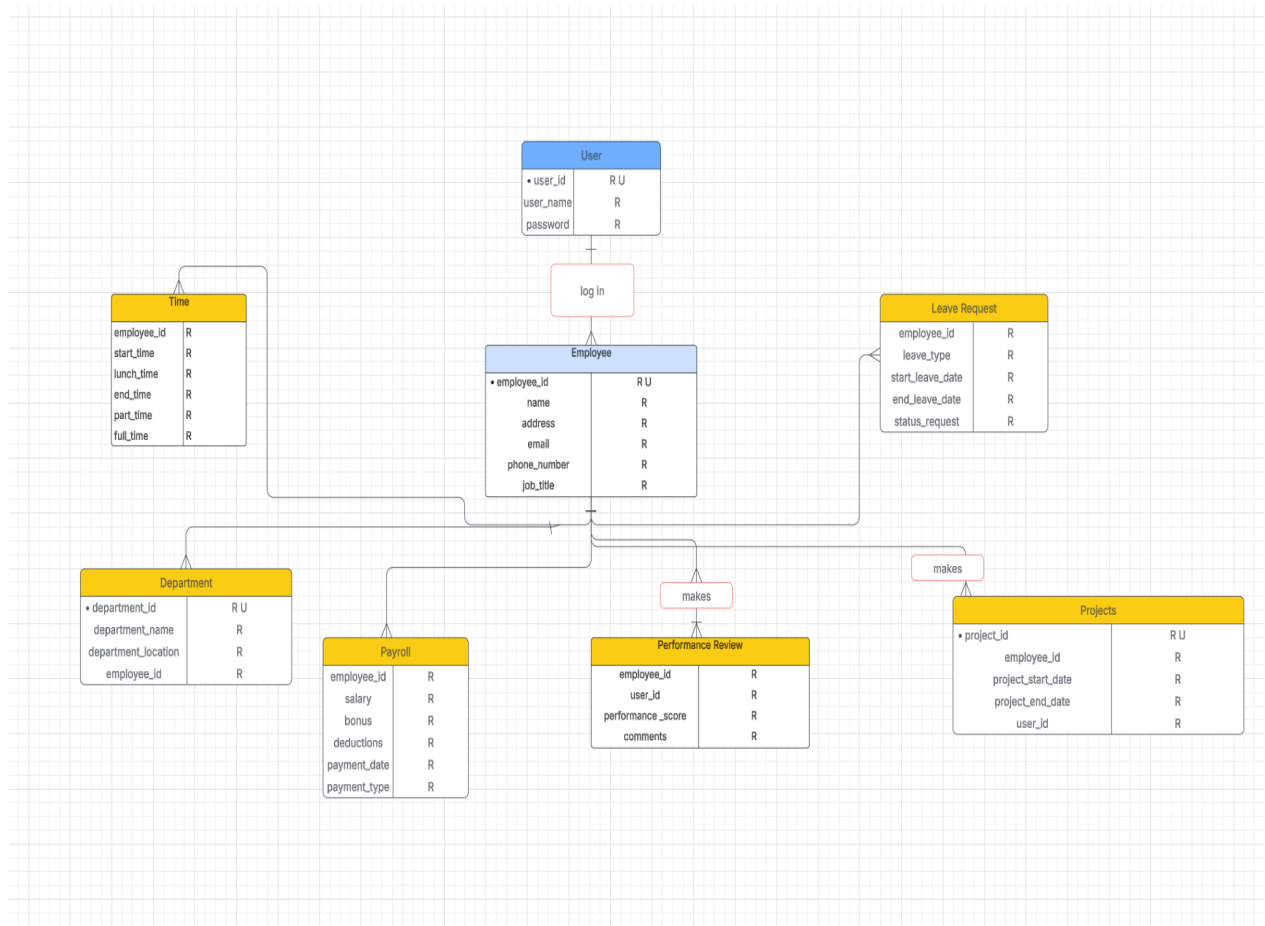
ERD:

My Entity-Relation Diagram for the database demonstrates the way that I will be using CASCADE. Any updates from the user to the employee will affect the database.



Relational Diagram:

This relational diagram demonstrates the Primary keys and foreign keys. In addition, it shows the intentions of the entities.



Part III: Database Implementation

DDL SQL Statements

```
DROP DATABASE IF EXISTS `Employee_Track_and_Report`;
CREATE DATABASE `Employee_Track_and_Report`;
USE `Employee_Track_and_Report`;
```

```
CREATE TABLE User (
user_id INT AUTO_INCREMENT PRIMARY KEY,
user_name VARCHAR(15) NOT NULL,
password VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE Employee (
employee_id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(200) NOT NULL,
address VARCHAR(200) NOT NULL,
email VARCHAR(50) NOT NULL,
phone_number INT NOT NULL,
job_title VARCHAR (50) NOT NULL
);
```

```
CREATE TABLE Time (
employee_id INT NOT NULL,
start_time TIME NOT NULL,
lunch_time TIME NOT NULL,
end_time TIME NOT NULL,
part_time VARCHAR(8) NOT NULL,
full_time VARCHAR (8) NOT NULL,
FOREIGN KEY (employee_id) REFERENCES Employee (employee_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE Department (
department_id INT AUTO_INCREMENT PRIMARY KEY,
employee_id INT NOT NULL ,
department_name VARCHAR(50) NOT NULL,
department_location VARCHAR(50) NOT NULL,
FOREIGN KEY (employee_id) REFERENCES Employee (employee_id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE Payroll (  
employee_id INT NOT NULL,  
salary INT NOT NULL,  
bonus INT NOT NULL,  
deductions INT NOT NULL,  
payment_date DATE NOT NULL,  
payment_type VARCHAR(20) NOT NULL,  
FOREIGN KEY (employee_id) REFERENCES Employee (employee_id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Leave_Request (  
employee_id INT NOT NULL,  
leave_type VARCHAR(20) NOT NULL,  
start_leave_date DATE NOT NULL,  
end_leave_date DATE NOT NULL,  
status_request VARCHAR(20) NOT NULL,  
FOREIGN KEY (employee_id) REFERENCES Employee (employee_id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Performance_Review (  
employee_id INT NOT NULL,  
user_id VARCHAR(200) NOT NULL,  
performance_score INT NOT NULL,  
comments VARCHAR (500) NOT NULL,  
FOREIGN KEY (employee_id) REFERENCES Employee (employee_id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE Projects (  
project_id INT AUTO_INCREMENT PRIMARY KEY,  
employee_id INT NOT NULL,  
user_id INT NOT NULL,  
project_start_date DATE NOT NULL,  
project_end_date DATE NOT NULL,  
FOREIGN KEY (user_id) REFERENCES User(user_id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

Dummy Data Generation:

I used ChatGPT to retrieve this dummy data, receiving 10 results.

```
-- Inserting sample data into the User table
INSERT INTO User (user_name, password)
VALUES
('admin', 'admin123'),
('manager1', 'manager123'),
('hr1', 'hrpassword'),
('it_support', 'itsupport123'),
('employee1', 'emp1234'),
('employee2', 'emp5678'),
('employee3', 'emp9876'),
('employee4', 'emp5432'),
('employee5', 'emp1122'),
('employee6', 'emp3344');

-- Inserting sample data into the Employee table
INSERT INTO Employee (name, address, email, phone_number, job_title)
VALUES
('John Doe', '123 Main St, Cityville', 'john.doe@example.com', 1234567890, 'Software Developer'),
('Jane Smith', '456 Oak St, Townsville', 'jane.smith@example.com', 2345678901, 'HR Manager'),
('Tom Brown', '789 Pine St, Villageburg', 'tom.brown@example.com', 3456789012, 'Project Manager'),
('Sara Green', '321 Maple St, Suburbia', 'sara.green@example.com', 4567890123, 'Designer'),
('Emily White', '654 Birch St, Uptown', 'emily.white@example.com', 5678901234, 'QA Engineer'),
('Michael Blue', '987 Cedar St, Downtown', 'michael.blue@example.com', 6789012345, 'Marketing Specialist'),
('David Black', '123 Elm St, Cityville', 'david.black@example.com', 7890123456, 'Sales Executive'),
('Olivia Gray', '234 Pine St, Suburbia', 'olivia.gray@example.com', 8901234567, 'Business Analyst'),
('Lucas Yellow', '876 Maple St, Villageburg', 'lucas.yellow@example.com', 9012345678, 'HR Specialist'),
('Sophia Red', '432 Oak St, Uptown', 'sophia.red@example.com', 1230984567, 'Content Writer');

-- Inserting sample data into the Time table
INSERT INTO Time (employee_id, start_time, lunch_time, end_time, part_time, full_time)
VALUES
(1, '09:00:00', '12:00:00', '18:00:00', '04:00', '08:00'),
(2, '08:30:00', '12:30:00', '17:30:00', '04:00', '08:00');
```

```
(3, '10:00:00', '13:00:00', '19:00:00', '04:00', '08:00'),
(4, '09:00:00', '12:00:00', '18:00:00', '04:00', '08:00'),
(5, '09:30:00', '13:00:00', '18:30:00', '04:00', '08:00'),
(6, '08:45:00', '12:15:00', '17:45:00', '04:00', '08:00'),
(7, '09:15:00', '12:15:00', '18:15:00', '04:00', '08:00'),
(8, '10:00:00', '13:00:00', '19:00:00', '04:00', '08:00'),
(9, '08:30:00', '12:00:00', '17:30:00', '04:00', '08:00'),
(10, '09:00:00', '12:00:00', '18:00:00', '04:00', '08:00');
```

-- Inserting sample data into the Department table

```
INSERT INTO Department (employee_id, department_name, department_location)
```

```
VALUES
```

```
(1, 'IT', 'Headquarters'),
(2, 'HR', 'Headquarters'),
(3, 'Project Management', 'Suburban Office'),
(4, 'Design', 'Headquarters'),
(5, 'QA', 'Suburban Office'),
(6, 'Marketing', 'Headquarters'),
(7, 'Sales', 'Headquarters'),
(8, 'Business Analysis', 'Suburban Office'),
(9, 'HR', 'Headquarters'),
(10, 'Content Writing', 'Uptown Office');
```

-- Inserting sample data into the Payroll table

```
INSERT INTO Payroll (employee_id, salary, bonus, deductions, payment_date,
```

```
payment_type)
```

```
VALUES
```

```
(1, 60000, 5000, 2000, '2025-03-01', 'Direct Deposit'),
(2, 75000, 3000, 1500, '2025-03-01', 'Cheque'),
(3, 90000, 4000, 2500, '2025-03-01', 'Direct Deposit'),
(4, 50000, 2000, 1500, '2025-03-01', 'Cheque'),
(5, 70000, 3000, 1000, '2025-03-01', 'Direct Deposit'),
(6, 55000, 2500, 1200, '2025-03-01', 'Cheque'),
(7, 65000, 3500, 1800, '2025-03-01', 'Direct Deposit'),
(8, 48000, 2200, 900, '2025-03-01', 'Cheque'),
(9, 52000, 2700, 1000, '2025-03-01', 'Direct Deposit'),
(10, 45000, 1800, 800, '2025-03-01', 'Cheque');
```

-- Inserting sample data into the Leave_Request table

```
INSERT INTO Leave_Request (employee_id, leave_type, start_leave_date, end_leave_date,
status_request)
```

```
VALUES
```

```
(1, 'Sick', '2025-02-01', '2025-02-03', 'Approved'),
(2, 'Vacation', '2025-03-01', '2025-03-07', 'Approved'),
(3, 'Sick', '2025-03-10', '2025-03-12', 'Pending'),
(4, 'Vacation', '2025-03-15', '2025-03-20', 'Approved');
```

```
(5, 'Maternity', '2025-03-05', '2025-04-05', 'Approved'),  
(6, 'Sick', '2025-03-02', '2025-03-04', 'Denied'),  
(7, 'Vacation', '2025-03-20', '2025-03-25', 'Approved'),  
(8, 'Sick', '2025-03-10', '2025-03-12', 'Approved'),  
(9, 'Vacation', '2025-03-10', '2025-03-17', 'Pending'),  
(10, 'Sick', '2025-03-05', '2025-03-06', 'Approved');
```

-- Inserting sample data into the Performance_Review table

```
INSERT INTO Performance_Review (employee_id, user_id, performance_score, comments)  
VALUES
```

```
(1, 'admin', 85, 'Great performance this quarter.'),  
(2, 'hr1', 78, 'Good but can improve in team leadership.'),  
(3, 'manager1', 92, 'Excellent project management skills.'),  
(4, 'admin', 80, 'Solid work but needs better time management.'),  
(5, 'hr1', 88, 'Good attention to detail and quality.'),  
(6, 'it_support', 75, 'Could use more initiative in projects.'),  
(7, 'manager1', 90, 'Strong sales performance.'),  
(8, 'admin', 84, 'Solid work in business analysis, keep it up.'),  
(9, 'it_support', 70, 'Needs improvement in communication skills.'),  
(10, 'admin', 82, 'Great writing skills but needs to focus on deadlines.');
```

-- Inserting sample data into the Projects table

```
INSERT INTO Projects (employee_id, user_id, project_start_date, project_end_date)  
VALUES
```

```
(1, 1, '2025-01-15', '2025-03-15'),  
(2, 2, '2025-02-01', '2025-04-01'),  
(3, 3, '2025-01-10', '2025-04-10'),  
(4, 4, '2025-03-01', '2025-05-01'),  
(5, 5, '2025-02-20', '2025-05-20'),  
(6, 6, '2025-03-01', '2025-04-30'),  
(7, 7, '2025-01-01', '2025-03-31'),  
(8, 8, '2025-03-01', '2025-06-01'),  
(9, 9, '2025-02-10', '2025-04-10'),  
(10, 10, '2025-03-05', '2025-04-05');
```


Part IV: Database Deployment

Database Deployment:

I created the database “database-mis686fp” on AWS under RDS.

The screenshot shows the AWS RDS console for the database instance 'database-mis686fp'. The left sidebar contains navigation links for Amazon RDS, including Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, Recommendations, and Certificate update.

The main content area displays the 'database-mis686fp' instance details. The 'Summary' section shows the DB identifier, CPU usage (5.23%), Status (Available), Role (Instance), Engine (MySQL Community), Region & AZ (us-east-2a), and Recommendations. Below this, the 'Connectivity & security' tab is selected, showing the Endpoint (database-mis686fp.cfuacca2sp3.us-east-2.rds.amazonaws.com), Port (3306), Availability Zone (us-east-2a), VPC (vpc-0d04ac1213185e68a), Subnet group (default-vpc-0d04ac1213185e68a), and Security groups (MIS686VPCsecuritygroup). The 'Security' section shows the VPC security groups, Publicly accessible status (Yes), Certificate authority (Info), Certificate authority date (May 21, 2061, 17:04 UTC-07:00), and DB instance certificate expiration date (March 13, 2026, 17:53 UTC-07:00).

The 'Connected compute resources' section shows a table with columns: Resource identifier, Resource type, Availability Zone, VPC security group, Compute resource security group, and Connected proxy. The table is currently empty, with a message stating 'No connected compute resources'.

The screenshot shows the AWS RDS console for the database instance 'database-mis686fp', specifically the 'Proxies', 'Security group rules', and 'Replication' sections.

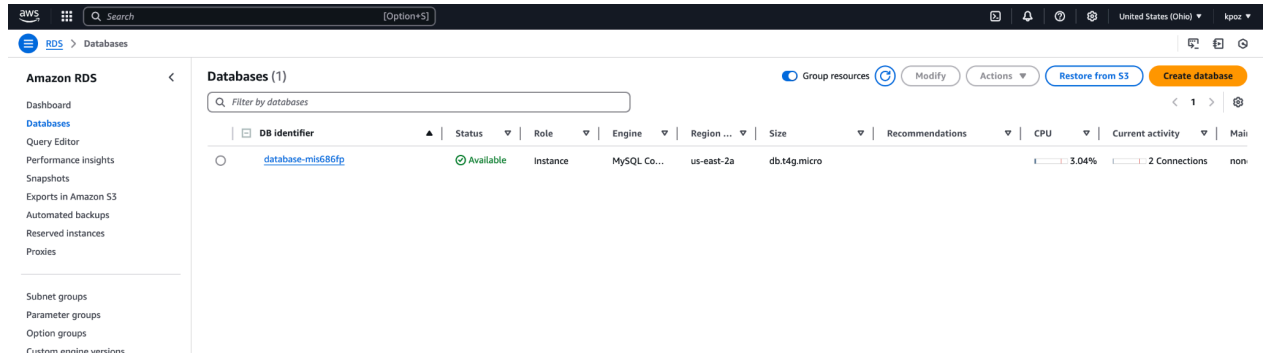
The 'Proxies' section shows a table with columns: Proxy identifier, Status, and Engine family. The table is currently empty, with a message stating 'No proxies' and a 'Create proxy' button.

The 'Security group rules' section shows a table with columns: Security group, Type, and Rule. The table contains two rules:

Security group	Type	Rule
MIS686VPCsecuritygroup (sg-0a8236aa9af47295c)	CIDR/IP - Inbound	0.0.0.0/0
MIS686VPCsecuritygroup (sg-0a8236aa9af47295c)	CIDR/IP - Outbound	0.0.0.0/0

The 'Replication' section shows a table with columns: DB identifier, Role, Region & AZ, Replication source, Replication state, and Lag. The table contains one entry:

DB identifier	Role	Region & AZ	Replication source	Replication state	Lag
database-mis686fp	Instance	us-east-2a	-	-	-



Index/View/Trigger/Stored Procedure:

Index: To get queries quicker since the employee id is used in many tables.

```
CREATE INDEX index_employee_id ON Employee (employee_id);
```

View: To see how these two are merged together.

```
CREATE VIEW Employee_Leave_Request AS
SELECT e.employee_id, e.name, e.job_title, e.email, e.phone_number, r.leave_type,
r.status_request
FROM Employee e
JOIN leave_request r ON e.employee_id = r.employee_id;
```

Trigger: This will track any updates regarding the performance_review table.

```
DELIMITER //

CREATE TRIGGER employee_update_trigger
AFTER UPDATE ON Employee
FOR EACH ROW
BEGIN
    UPDATE performance_review
    SET last_modified = NOW()
    WHERE employee_id = NEW.employee_id;

END //
```

```
DELIMITER ;
```

Stored Procedure: This will give the total wages necessary to view from employees.

```
DELIMITER //
```

```
CREATE PROCEDURE payroll_process(IN emp_id INT, IN salary INT, IN bonus INT, IN
deductions INT)
BEGIN
    DECLARE total_wages INT;

    SET total_wages = salary + bonus - deductions;

    INSERT INTO Payroll (employee_id, salary, bonus, deductions, total_wage, payment_date,
payment_type)
    VALUES (emp_id, salary, bonus, deductions, total_wages, CURDATE(), 'Direct Deposit');
END //

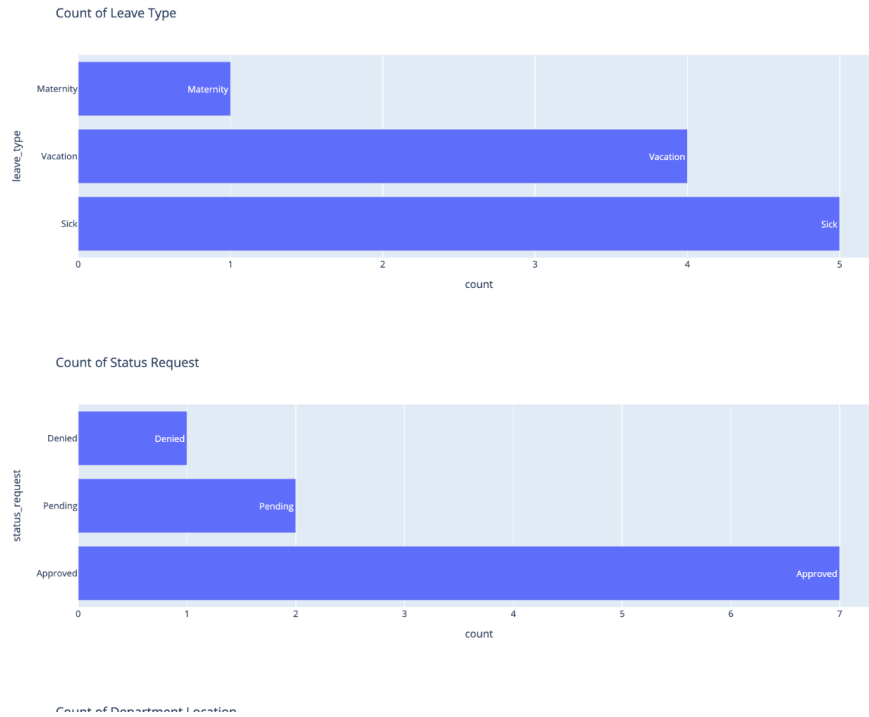
DELIMITER ;
```

Part V: Analytical Questions and Dashboard

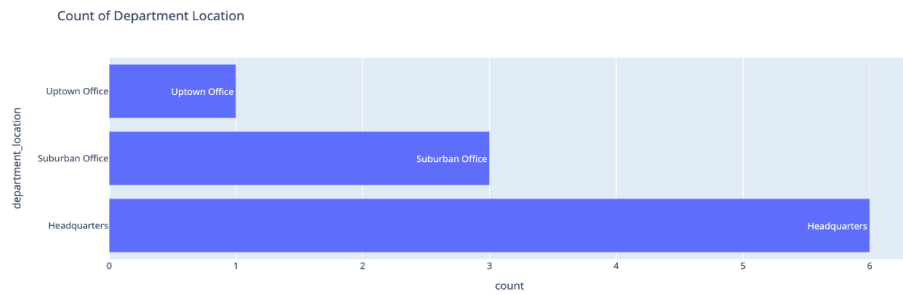
Analytical Questions:

1. What type of leave are employees taking the most?
2. How many leave requests are approved?
3. What are the most common department locations?
4. What is the average score of performance for the employee's?
5. What is the average salary?

Dashboard Screenshots:



These two bar graphs are answering questions #1 and #2. For #1, the type of leave individuals are taking the most is due to being sick. For #2, there are 7 approved leave requests.



1 rows1 columns1 cells			Run SQL Query	Export
AVG(performance_sc				
1	82.4			

1 rows1 columns1 cells			Run SQL Query	Export
AVG(salary)				
1	61000			

This graph and the two tables represents questions #3,#4, and #5. For #3, the most common department location is the headquarters. For #4, the average performance score is 82.4. For #5, the average salary of employees is \$61,000.