Created database:

```
mysql> create database payroll;
Query OK, 1 row affected (0.01 sec)

mysql> use payroll;
Database changed
```

CREATED TABLES:

```
mysql> create table employee(
    -> employee_id int(6),
    -> first_name varchar(25),
    -> last_name varchar(25),
    -> hire_date date,
    -> city varchar(25),
    -> state varchar(25),
    -> constraint employee_pk primary key(employee_id));
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

```
mysql> create table department(
    -> department_id int,
    -> department_name varchar(30),
    -> constraint deparmtent_pk primary key(department_id));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table account_details(
    -> account_id int,
    -> bank_name varchar(50),
    -> account_number varchar(50),
    -> employee_id int,
    -> constraint account_pk primary key(account_id),
    -> foreign key(employee_id) references employee(employee_id));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table project(
    -> project_id int,
    -> project_name varchar(50),
    -> project_description varchar(50),
    -> constraint project_pk primary key(project_id));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table salary(
    -> salary_id int,
    -> gross_salary int,
    -> hourly_pay int,
    -> state_tax int,
    -> federal_tax int,
    -> account_id int,
    -> constraint salary_pk primary key(salary_id),
    -> foreign key(account_id) references account_details(account_id));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table department_project(
    -> department_id int,
    -> project_id int,
    -> constraint deptproject_pk primary key (department_id),
    -> foreign key(department_id) references department(department_id),
    -> foreign key(project_id) references project(project_id));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> create table education(
    -> education_id int,
    -> employee_id int,
    -> degree varchar(30),
    -> graduation_year int(4),
    -> constraint location_pk primary key (education_id),
    -> foreign key (employee_id) references employee(employee_id));
Query OK, 0 rows affected, 1 warning (0.03 sec)
```

```
mysql> create table leaves(
    -> leave_id int,
    -> employee_id int,
    -> leave_date date,
    -> constraint leave_pk primary key (leave_id),
    -> foreign key (employee_id) references employee(employee_id));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table attendance(
    -> attendance_id int,
    -> hours_worked int,
    -> constraint attendance_pk primary key(attendance_id));
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> create table employee_attendance(
    -> employee_id int,
    -> attendance_id int,
    -> constraint departmentproject_pk primary key (employee_id,attendance_id),
    -> foreign key (employee_id) references employee(employee_id),
    -> foreign key (attendance_id) references attendance(attendance_id));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> create table work_location(
    -> location_id int,
    -> location varchar(25),
    -> number_of_employees int,
    -> city varchar(25),
    -> state varchar(25),
    -> constraint loc_pk primary key (location_id));
Query OK, 0 rows affected (0.02 sec)
```

ALL THE TABLES LISTED UP:

```
mysql> show tables;
+---------------------+
| Tables_in_payroll   |
+---------------------+
| account_details     |
| attendance          |
| department          |
| department_project  |
| education           |
| employee            |
| employee_attendance |
| leaves              |
| project             |
| salary              |
| work_location       |
+---------------------+
11 rows in set (0.00 sec)
```

ALTER EXISTING TABLE:

```
mysql> ALTER TABLE Employee ADD COLUMN department_id INT;
Query OK, 0 rows affected (0.17 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> UPDATE Employee
    -> SET department_id = 1
    -> WHERE employee_id = 101;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> UPDATE Employee
    -> SET department_id = 2
    -> WHERE employee_id = 102;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

1. Retrieve Employee Information:

```
mysql> select * from employee;
+-------------+------------+-----------+------------+---------------+---------------+---------------+
| employee_id | first_name | last_name | hire_date  | city          | state         | department_id |
+-------------+------------+-----------+------------+---------------+---------------+---------------+
|         101 | Ojas       | Phansekar | 2016-04-14 | New York City | New York      |             1 |
|         102 | Vrushali   | Patil     | 2018-06-21 | Boston        | Massachusetts |             2 |
|         103 | Pratik     | Parija    | 2019-09-13 | Chicago       | Illinois      |             3 |
|         104 | Chetan     | Mistry    | 2011-04-12 | Miami         | Florida       |             4 |
|         105 | Anugraha   | Varkey    | 2017-08-16 | Atlanta       | Georgia       |             5 |
|         106 | Rasagnya   | Reddy     | 2018-07-25 | San Mateo     | California    |             6 |
|         107 | Aishwarya  | Boralkar  | 2010-12-18 | San Francisco | California    |             7 |
|         108 | Shantanu   | Savant    | 2015-11-27 | Seattle       | Washington    |             8 |
|         109 | Kalpita    | Malvankar | 2016-04-24 | Boston        | Massachusetts |             8 |
|         110 | Saylee     | Bhagat    | 2014-05-21 | San Francisco | California    |             7 |
+-------------+------------+-----------+------------+---------------+---------------+---------------+
```

```
mysql> SELECT employee_id, first_name, last_name, hire_date, city, state
    -> FROM Employee;
+-------------+------------+-----------+------------+---------------+---------------+
| employee_id | first_name | last_name | hire_date  | city          | state         |
+-------------+------------+-----------+------------+---------------+---------------+
|         101 | Ojas       | Phansekar | 2016-04-14 | New York City | New York      |
|         102 | Vrushali   | Patil     | 2018-06-21 | Boston        | Massachusetts |
|         103 | Pratik     | Parija    | 2019-09-13 | Chicago       | Illinois      |
|         104 | Chetan     | Mistry    | 2011-04-12 | Miami         | Florida       |
|         105 | Anugraha   | Varkey    | 2017-08-16 | Atlanta       | Georgia       |
|         106 | Rasagnya   | Reddy     | 2018-07-25 | San Mateo     | California    |
|         107 | Aishwarya  | Boralkar  | 2010-12-18 | San Francisco | California    |
|         108 | Shantanu   | Savant    | 2015-11-27 | Seattle       | Washington    |
|         109 | Kalpita    | Malvankar | 2016-04-24 | Boston        | Massachusetts |
|         110 | Saylee     | Bhagat    | 2014-05-21 | San Francisco | California    |
+-------------+------------+-----------+------------+---------------+---------------+
10 rows in set (0.00 sec)
```

2. List Employees in a Specific Department:

```
mysql> SELECT first_name, last_name
    -> FROM Employee
    -> WHERE department_id = 2; -- Change 2 to the desired department ID
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Vrushali   | Patil     |
+------------+-----------+
1 row in set (0.00 sec)
```

3. Calculate Total Salary Expenses:

```
mysql> SELECT SUM(gross_salary) AS total_salary_expenses
    -> FROM Salary;
+-----------------------+
| total_salary_expenses |
+-----------------------+
|                825600 |
+-----------------------+
1 row in set (0.01 sec)
```

4. Find Employees Who Took Leave on a Specific Date:

```
mysql> SELECT first_name, last_name, leave_date
    -> FROM Employee
    -> JOIN leaves ON Employee.employee_id = leaves.employee_id
    -> WHERE leave_date = '2019-12-05'; -- Change the date as needed
+------------+-----------+------------+
| first_name | last_name | leave_date |
+------------+-----------+------------+
| Rasagnya   | Reddy     | 2019-12-05 |
+------------+-----------+------------+
1 row in set (0.01 sec)
```

5. Retrieve Employee Education Details:

```
mysql> SELECT first_name, last_name, degree, graduation_year
    -> FROM Employee
    -> JOIN Education ON Employee.employee_id = Education.employee_id;
+------------+-----------+----------+-----------------+
| first_name | last_name | degree   | graduation_year |
+------------+-----------+----------+-----------------+
| Ojas       | Phansekar | MS       |            2017 |
| Vrushali   | Patil     | MS       |            2019 |
| Chetan     | Mistry    | MS       |            2011 |
| Shantanu   | Savant    | MS       |            2015 |
| Kalpita    | Malvankar | Bachelor |            2013 |
| Aishwarya  | Boralkar  | Bachelor |            2008 |
| Rasagnya   | Reddy     | Bachelor |            2007 |
+------------+-----------+----------+-----------------+
7 rows in set (0.01 sec)
```

6. List Departments and Their Project Names:

```
mysql> SELECT department_name, project_name
    -> FROM Department
    -> LEFT JOIN department_project ON Department.department_id = department_project.department_id
    -> LEFT JOIN Project ON department_project.project_id = Project.project_id;
+-----------------------+--------------+
| department_name       | project_name |
+-----------------------+--------------+
| Human Resources       | Dev          |
| Software Development  | Prod         |
| Data Analysis         | Test         |
| Data Science          | Nothing      |
| Business Intelligence | Research     |
| Data Engineering      | Next Steps   |
| Manufacturing         | Dev          |
| Quality Control       | Nothing      |
+-----------------------+--------------+
8 rows in set (0.00 sec)
```

7. Calculate Average Hours Worked by Employees:

```
mysql> SELECT AVG(hours_worked) AS average_hours_worked
    -> FROM Attendance;
+----------------------+
| average_hours_worked |
+----------------------+
|              37.0000 |
+----------------------+
1 row in set (0.00 sec)
```

8. Find Employees and Their Work Locations:

```
mysql> SELECT first_name, last_name, location
    -> FROM Employee
    -> JOIN Work_Location ON Employee.employee_id = Work_Location.location_id;
Empty set (0.00 sec)
```

9. Calculate Employee Turnover Rate:

```
mysql> SELECT COUNT(leave_id) / COUNT(employee_id) AS turnover_rate
    -> FROM leaves;
+---------------+
| turnover_rate |
+---------------+
|        1.0000 |
+---------------+
1 row in set (0.00 sec)
```

10. List Employees with No Leave Records:

```
mysql> SELECT first_name, last_name
    -> FROM Employee
    -> LEFT JOIN leaves ON Employee.employee_id = leaves.employee_id
    -> WHERE leave_id IS NULL;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Ojas       | Phansekar |
| Vrushali   | Patil     |
| Pratik     | Parija    |
| Anugraha   | Varkey    |
| Saylee     | Bhagat    |
+------------+-----------+
5 rows in set (0.00 sec)
```

11. Find Employees and Their Departments:

```
mysql> SELECT Employee.employee_id, Employee.first_name, Employee.last_name, Department.department_name
    -> FROM Employee
    -> JOIN Department ON Employee.department_id = Department.department_id;
+-------------+------------+-----------+-----------------------+
| employee_id | first_name | last_name | department_name       |
+-------------+------------+-----------+-----------------------+
|         101 | Ojas       | Phansekar | Human Resources       |
|         102 | Vrushali   | Patil     | Software Development  |
|         103 | Pratik     | Parija    | Data Analysis         |
|         104 | Chetan     | Mistry    | Data Science          |
|         105 | Anugraha   | Varkey    | Business Intelligence |
|         106 | Rasagnya   | Reddy     | Data Engineering      |
|         107 | Aishwarya  | Boralkar  | Manufacturing         |
|         108 | Shantanu   | Savant    | Quality Control       |
|         109 | Kalpita    | Malvankar | Quality Control       |
|         110 | Saylee     | Bhagat    | Manufacturing         |
+-------------+------------+-----------+-----------------------+
10 rows in set (0.00 sec)
```

12. Find Employees Who Took Leaves in December 2019:

```
mysql> SELECT Employee.employee_id, Employee.first_name, Employee.last_name, Leaves.leave_date
    -> FROM Employee
    -> JOIN Leaves ON Employee.employee_id = Leaves.employee_id
    -> WHERE Leaves.leave_date BETWEEN '2019-12-01' AND '2019-12-31';
+-------------+------------+-----------+------------+
| employee_id | first_name | last_name | leave_date |
+-------------+------------+-----------+------------+
|         104 | Chetan     | Mistry    | 2019-12-01 |
|         108 | Shantanu   | Savant    | 2019-12-02 |
|         109 | Kalpita    | Malvankar | 2019-12-03 |
|         107 | Aishwarya  | Boralkar  | 2019-12-04 |
|         106 | Rasagnya   | Reddy     | 2019-12-05 |
|         104 | Chetan     | Mistry    | 2019-12-06 |
|         108 | Shantanu   | Savant    | 2019-12-07 |
|         109 | Kalpita    | Malvankar | 2019-12-07 |
|         107 | Aishwarya  | Boralkar  | 2019-12-08 |
|         106 | Rasagnya   | Reddy     | 2019-12-09 |
+-------------+------------+-----------+------------+
10 rows in set (0.00 sec)
```

13. Inline views:

```
mysql> SELECT Department_Name, COUNT(*) AS Count,
    -> CONCAT(FORMAT((COUNT(*) / No_of_Employees.cnt) * 100, 2), '%') AS Percentages
    -> FROM Department
    -> JOIN Employee ON Department.Department_Id = Employee.Department_Id
    -> CROSS JOIN (SELECT COUNT(*) AS cnt FROM Employee) AS No_of_Employees
    -> GROUP BY Department_Name, No_of_Employees.cnt;
+-----------------------+-------+-------------+
| Department_Name       | Count | Percentages |
+-----------------------+-------+-------------+
| Human Resources       |     1 | 10.00%      |
| Software Development  |     1 | 10.00%      |
| Data Analysis         |     1 | 10.00%      |
| Data Science          |     1 | 10.00%      |
| Business Intelligence |     1 | 10.00%      |
| Data Engineering      |     1 | 10.00%      |
| Manufacturing         |     2 | 20.00%      |
| Quality Control       |     2 | 20.00%      |
+-----------------------+-------+-------------+
8 rows in set (0.00 sec)
```