SQL Code Examples with Bangladeshi Names

1. Create a Database

```
CREATE DATABASE bangladeshi_employees;
```

2. Use the Database

```
USE bangladeshi_employees;
```

3. Create an Employee Table

```
CREATE TABLE employees (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age INT NOT NULL,
    department VARCHAR(50),
    salary DECIMAL(10, 2),
    city VARCHAR(50)
);
```

4. Insert Bangladeshi Names into the Table

```
INSERT INTO employees (name, age, department, salary, city)
VALUES
('Md. Rakib Hasan', 30, 'IT', 45000.00, 'Dhaka'),
('Farzana Rahman', 28, 'Finance', 50000.00, 'Chittagong'),
('Shakib Al Mamun', 35, 'HR', 40000.00, 'Sylhet'),
('Tahmina Akter', 25, 'IT', 55000.00, 'Rajshahi'),
('Abdul Karim', 40, 'Management', 60000.00, 'Barisal');
```

5. View All Data

```
SELECT * FROM employees;
```

6. Filter Employees from Dhaka

```
SELECT * FROM employees
WHERE city = 'Dhaka';
```

7. Find Employees Older than 30

```
SELECT name, age
FROM employees
WHERE age > 30;
```

8. Order Employees by Salary

```
SELECT name, salary
FROM employees
ORDER BY salary DESC;
```

9. Update Salary for an Employee

```
UPDATE employees
SET salary = 48000.00
WHERE name = 'Md. Rakib Hasan';
```

10. Add a New Column

```
ALTER TABLE employees
ADD COLUMN join_date DATE;
```

11. Insert Join Dates

```
UPDATE employees
SET join_date = '2022-01-15'
WHERE name = 'Md. Rakib Hasan';
```

12. Delete an Employee

```
DELETE FROM employees
WHERE name = 'Abdul Karim';
```

13. Count Employees in Each Department

```
SELECT department, COUNT(*) AS total_employees
FROM employees
GROUP BY department;
```

14. Find the Average Salary

```
SELECT AVG(salary) AS avg_salary FROM employees;
```

15. Find the Maximum Salary

```
SELECT MAX(salary) AS max_salary FROM employees;
```

16. Create a Table for Departments

```
CREATE TABLE departments (
  department_id INT PRIMARY KEY AUTO_INCREMENT,
  department_name VARCHAR(50) UNIQUE
);
```

17. Insert Departments

```
INSERT INTO departments (department_name)
VALUES
('IT'), ('Finance'), ('HR'), ('Management');
```

18. Join Employees with Departments

```
SELECT employees.name, employees.salary, departments.department_name
FROM employees
INNER JOIN departments
ON employees.department = departments.department_name;
```

19. Find Employees Earning Above Average

```
SELECT name, salary
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

20. Employees Who Joined in 2022

```
SELECT name
FROM employees
WHERE YEAR(join_date) = 2022;
```

21. Employees with IT Background in Sylhet

```
SELECT name, salary
FROM employees
WHERE department = 'IT' AND city = 'Sylhet';
```

22. Group Employees by City

```
SELECT city, COUNT(*) AS employee_count
FROM employees
GROUP BY city;
```

23. Rename Table

```
RENAME TABLE employees TO employee_records;
```

24. Employee Names Ending with 'Rahman'

```
SELECT name
FROM employee_records
WHERE name LIKE '%Rahman';
```

25. Employees with Salary Between 40,000 and 60,000

```
SELECT name, salary
FROM employee_records
WHERE salary BETWEEN 40000 AND 60000;
```

26. Drop a Column

```
ALTER TABLE employee_records
DROP COLUMN join_date;
```

27. Find Distinct Departments

```
SELECT DISTINCT department
FROM employee_records;
```

28. Top 3 Highest Salaries

```
SELECT name, salary
FROM employee_records
ORDER BY salary DESC
LIMIT 3;
```

29. Add a New Table for Projects

```
CREATE TABLE projects (
    project_id INT AUTO_INCREMENT PRIMARY KEY,
    project_name VARCHAR(100),
    employee_id INT,
    FOREIGN KEY (employee_id) REFERENCES employee_records(id)
);
```

30. Assign Projects

```
INSERT INTO projects (project_name, employee_id)
VALUES
('E-Commerce Website', 1),
('Banking App', 2),
('HR Management System', 3);
```

31. View Projects with Employees

```
SELECT projects.project_name, employee_records.name
FROM projects
INNER JOIN employee_records
ON projects.employee_id = employee_records.id;
```

32. Employees Without a Project

```
SELECT name
FROM employee_records
WHERE id NOT IN (SELECT employee_id FROM projects);
```

33. Employees with Specific Names

```
SELECT *
FROM employee_records
WHERE name IN ('Tahmina Akter', 'Shakib Al Mamun');
```

34. Group Employees by Age

```
SELECT age, COUNT(*) AS employee_count
FROM employee_records
GROUP BY age;
```

35. Create an Index

```
CREATE INDEX idx_department ON employee_records(department);
```

36. Drop the Index

```
DROP INDEX idx_department ON employee_records;
```

37. Average Salary by City

```
SELECT city, AVG(salary) AS average_salary
FROM employee_records
GROUP BY city;
```

38. Create a Table for Employee Logins

```
CREATE TABLE employee_logins (
    login_id INT AUTO_INCREMENT PRIMARY KEY,
    employee_id INT,
    login_time DATETIME,
    FOREIGN KEY (employee_id) REFERENCES employee_records(id)
);
```

39. Insert Login Records

```
INSERT INTO employee_logins (employee_id, login_time)
VALUES
(1, '2025-01-06 09:00:00'),
(2, '2025-01-06 09:15:00');
```

40. Find Recent Logins

```
SELECT employee_id, login_time
FROM employee_logins
WHERE login_time > '2025-01-05 00:00:00';
```

41. Drop the Projects Table

```
DROP TABLE projects;
```

42. Find Salaries Above 50K in Each City

```
SELECT city, COUNT(*) AS high_earners
FROM employee_records
WHERE salary > 50000
GROUP BY city;
```

43. Rename a Column

```
ALTER TABLE employee_records
CHANGE COLUMN city location VARCHAR(50);
```

44. Find Employees with No Department

```
SELECT name
FROM employee_records
WHERE department IS NULL;
```

45. Employee Salaries by Department

```
SELECT department, SUM(salary) AS total_salary
FROM employee_records
GROUP BY department;
```

46. Find the Second Highest Salary

```
SELECT MAX(salary) AS second_highest
FROM employee_records
WHERE salary < (SELECT MAX(salary) FROM employee_records);
```

47. Delete Employees from Barisal

```
DELETE FROM employee_records
WHERE location = 'Barisal';
```

48. Employees with Names Containing 'Md.'

```
SELECT *
FROM employee_records
WHERE name LIKE 'Md.%';
```

49. Find Total Employees

```
SELECT COUNT(*) AS total_employees
FROM employee_records;
```

50. Drop the Database

```
DROP DATABASE bangladeshi_employees;
```