

Java • SQL • Spring Boot Practice

This repository contains my **daily practice** covering **Core Java logic**, **SQL queries**, and **Spring Boot REST APIs**. It is structured to strengthen fundamentals required for a **Java Developer** role.

Core Java Practice

' Count Vowels in a String

```
```java
String str = "Geetanjali";
int count = 0;

for (int i = 0; i < str.length(); i++) {
 char ch = str.charAt(i);
 if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'
 || ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
 count++;
 }
}
System.out.println("Vowels count = " + count);
```
```

Explanation:

- * Traversed the string character by character
- * Checked vowels using conditional statements

' Count Consonants in a String

```
```java
String str = "Roshni";
int count = 0;

for (int i = 0; i < str.length(); i++) {
 char ch = str.charAt(i);
 if (Character.isLetter(ch) &&
```

```
!(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'
|| ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')) {
count++;
}
}
System.out.println("Consonants count = " + count);

```

## SQL Practice

### 1p Find Highest Salary

```
```sql
SELECT MAX(salary) FROM employees;
---
```

2p Find Second Highest Salary

```
```sql
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);

```

### 3p Count Employees Department-wise

```
```sql
SELECT department, COUNT(*)
FROM employees
GROUP BY department;
---
```

What is GROUP BY?

`GROUP BY` groups rows that have the same values in specified columns and is commonly used with aggregate functions like `COUNT`, `SUM`, `AVG`, `MAX`, and `MIN`.

4p Employee with Minimum Salary

```
```sql
SELECT *
FROM employees
WHERE salary = (SELECT MIN(salary) FROM employees);
```
```

Explanation:

- * Used a subquery to fetch the minimum salary
- * Retrieved complete employee details

Spring Boot – REST API (CRUD Start)

Entity

```
```java
@Entity
public class Employee {

 @Id
 @GeneratedValue(strategy = GenerationType.IDENTITY)
 private int id;
 private String name;
 private int salary;

 // Getters and Setters
}
```
```

Repository

```
```java
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {
}
```
```

🔗 Controller (GET + POST)

```
```java
@RestController
@RequestMapping("/employees")
public class EmployeeController {

 @Autowired
 private EmployeeRepository repo;

 @GetMapping
 public List<Employee> getAll() {
 return repo.findAll();
 }

 @PostMapping
 public Employee save(@RequestBody Employee emp) {
 return repo.save(emp);
 }
}
```
```

🔗 GET Employee by ID API

```
```java
@GetMapping("/{id}")
public Employee getEmployeeByID(@PathVariable int id) {
 return repo.findById(id).orElse(null);
}
```
```

URL:

GET http://localhost:8888/employees/1

Explanation:

* Fetches employee data using ID

* Uses Spring Data JPA `findById()` method

'(Tech Stack

* Java (Core + OOP)

* SQL

* Spring Boot

* Spring Data JPA

* REST APIs

Ø=Ü This repository reflects my continuous learning and hands-on practice toward becoming a strong Java Backend Developer. Ø=Ü^a