1. **Get by Name**
   - **Task:** Create and call a procedure

```
CREATE PROCEDURE GetCustomerByName(IN cust_name VARCHAR(20))
BEGIN
  SELECT * FROM CUSTOMERS WHERE NAME = cust_name;
END;
```

   - **Verify:**

```
CALL GetCustomerByName('Ramesh');
```

---

2. **Insert New Customer**
   - **Task:** Write a procedure `AddCustomer` with four IN parameters (`p_name`, `p_age`, `p_address`, `p_salary`) that inserts one row into `CUSTOMERS`.
   - **Verify:** Call it twice with different data, then `SELECT * FROM CUSTOMERS;` to see the new rows.

---

3. **Update Salary by %**
   - **Task:** Build `RaiseSalary` (IN `p_id INT`, IN `pct DECIMAL(5,2)`) that increases a customer's salary by `pct` percent.
   - **Verify:**

```
CALL RaiseSalary(3, 15.0);
SELECT SALARY FROM CUSTOMERS WHERE ID = 3;
```

---

4. **Count by Age (OUT)**
   - **Task:** Create `CountByAge` with IN `p_age INT`, OUT `total_count INT` that returns how many customers have that age.
   - **Verify:**

```
CALL CountByAge(25, @cnt);
SELECT @cnt;
```

---

5. **Swap Salaries (INOUT)**
   - **Task:** Write `SwapSalaries` (INOUT `id1 INT`, INOUT `id2 INT`) that swaps the salary values of two customer IDs.
   - **Verify:** Compare salaries before and after calling:

```
SET @a=1; SET @b=2;
CALL SwapSalaries(@a, @b);
```

---

6. **Delete by Age**

- **Task:** Create `DeleteByAge` (IN `p_age INT`) to delete all customers of that age.
- **Verify:** Call for age you know exists, then `SELECT * FROM CUSTOMERS WHERE AGE = p_age;` should return zero rows.