

**Name:** Krish Porwal

**Registration Number:** 231080910

## **Indexing, Triggering and Procedures in DBMS=**

### **Indexing:**

- Index on hostel\_id in Hostel\_Branch table:

```
CREATE INDEX idx_hostel_id ON Hostel_Branch(hostel_id);
```

- Index on room\_number in Rooms table:

```
CREATE INDEX idx_room_number ON Rooms(room_number);
```

- Index on hostel\_id in Rooms table (if frequently joined with Hostel\_Branch):

```
CREATE INDEX idx_hostel_id_rooms ON Rooms(hostel_id);
```

- Index on student\_id in Students table:

```
CREATE INDEX idx_student_id ON Students(student_id);
```

- Index on hostel\_id in Students table (if frequently joined with Hostel\_Branch):

```
CREATE INDEX idx_hostel_id_students ON Students(hostel_id);
```

- Index on student\_id in Student\_room\_assignment table:

```
CREATE INDEX idx_student_id_sra ON Student_room_assignment(student_id);
```

- Index on hostel\_id in Student\_room\_assignment table (if frequently joined with Hostel\_Branch):

```
CREATE INDEX idx_hostel_id_sra ON Student_room_assignment(hostel_id);
```

- Index on student\_id in Dependents table:

```
CREATE INDEX idx_student_id_dependents ON Dependents(student_id);
```

- **Index on student\_id in Fees table:**

```
CREATE INDEX idx_student_id_fees ON Fees(student_id);
```

## Triggering=

- **Trigger to update number\_of\_students in Hostel\_Branch table when a new student is added:**

```
DELIMITER //
```

```
CREATE OR REPLACE TRIGGER update_number_of_students  
AFTER INSERT ON Students  
FOR EACH ROW  
BEGIN  
    UPDATE Hostel_Branch  
    SET number_of_students = number_of_students + 1  
    WHERE hostel_id = NEW.hostel_id;  
END//
```

```
DELIMITER ;
```

- **Trigger to update number\_of\_students in Hostel\_Branch table when a student is removed:**

```
DELIMITER //
```

```
CREATE OR REPLACE TRIGGER reduce_number_of_students  
AFTER DELETE ON Students  
FOR EACH ROW  
BEGIN  
    UPDATE Hostel_Branch  
    SET number_of_students = number_of_students - 1  
    WHERE hostel_id = OLD.hostel_id;  
END//
```

```
DELIMITER ;
```

- **Trigger to update paid\_till\_date and pending columns in Fees table when a new payment is made:**

```
DELIMITER //
```

```
CREATE OR REPLACE TRIGGER update_number_of_employees
AFTER INSERT ON Employees
FOR EACH ROW
BEGIN
    UPDATE Hostel_Branch
    SET number_of_employees = number_of_employees + 1
    WHERE hostel_id = NEW.hostel_id;
END//

DELIMITER ;
```

- **Trigger to update paid\_till\_date and pending columns in Fees table when a payment is deleted:**

```
DELIMITER //

CREATE OR REPLACE TRIGGER reduce_number_of_employees
AFTER DELETE ON Employees
FOR EACH ROW
BEGIN
    UPDATE Hostel_Branch
    SET number_of_employees = number_of_employees - 1
    WHERE hostel_id = OLD.hostel_id;
END//

DELIMITER ;
```

## **Procedure=**

- **Procedure to retrieve all students in a given hostel:**

```
DELIMITER //

CREATE OR REPLACE PROCEDURE get_students_in_hostel(
    hostel_id_param VARCHAR(255)
)
BEGIN
    SELECT *
    FROM Students
    WHERE hostel_id = hostel_id_param;
END //

DELIMITER ;
```

- **Procedure to calculate the total fees paid by a student:**

```

DELIMITER //

CREATE OR REPLACE PROCEDURE calculate_total_fees(
    student_id_param VARCHAR(255),
    total_fees OUT DECIMAL
)
BEGIN
    SELECT SUM(amount) INTO total_fees
    FROM Fees
    WHERE student_id = student_id_param;
END //

DELIMITER ;

```

- **Procedure to assign a room to a student:**

```

DELIMITER //

CREATE OR REPLACE PROCEDURE assign_room_to_student(
    student_id_param VARCHAR(255),
    hostel_id_param VARCHAR(255),
    room_number_param INT
)
BEGIN
    INSERT INTO Student_room_assignment (student_id, hostel_id, room_number)
    VALUES (student_id_param, hostel_id_param, room_number_param);
END //

DELIMITER ;

```

- **Procedure to update the salary of an employee:**

```

DELIMITER //

CREATE OR REPLACE PROCEDURE update_employee_salary(
    employee_id_param VARCHAR(255),
    new_salary_param DECIMAL
)
BEGIN
    UPDATE Employees
    SET salary = new_salary_param
    WHERE employee_id = employee_id_param;
END //

DELIMITER ;

```

**[Gist Link for Output of the above commands](#)**