

**BT ID :- BT23CSE175**

**NAME:-YENNE TANOOJ KUMAR**

```
create database hostelDB;
```

```
use hostelDB;
```

```
CREATE TABLE Students (  
    student_id INT PRIMARY KEY,  
    student_name VARCHAR(50),  
    gender VARCHAR(10),  
    course VARCHAR(50),  
    year INT  
);
```

```
CREATE TABLE Hostels (  
    hostel_id INT PRIMARY KEY,  
    hostel_name VARCHAR(50),  
    capacity INT  
);
```

```
CREATE TABLE Rooms (  
    room_id INT PRIMARY KEY,  
    hostel_id INT,  
    room_number VARCHAR(10),  
    capacity INT,  
    FOREIGN KEY (hostel_id) REFERENCES Hostels(hostel_id)  
);
```

```
CREATE TABLE Room_Allocation (  
    allocation_id INT PRIMARY KEY,  
    student_id INT,  
    room_id INT,  
    allocation_date DATE,  
    FOREIGN KEY (student_id) REFERENCES Students(student_id),
```

```
FOREIGN KEY (room_id) REFERENCES Rooms(room_id)
);
```

```
INSERT INTO Students VALUES
(1, 'Alice', 'Female', 'B.Tech', 2),
(2, 'Bob', 'Male', 'B.Sc', 1),
(3, 'Charlie', 'Male', 'B.Tech', 2),
(4, 'Diana', 'Female', 'BBA', 3),
(5, 'Ethan', 'Male', 'BCA', 1),
(6, 'Fiona', 'Female', 'B.Sc', 1),
(7, 'George', 'Male', 'BBA', 3),
(8, 'Hannah', 'Female', 'B.Tech', 2);
```

```
INSERT INTO Hostels VALUES
(1, 'Sunrise Hostel', 10),
(2, 'Moonlight Hostel', 8);
```

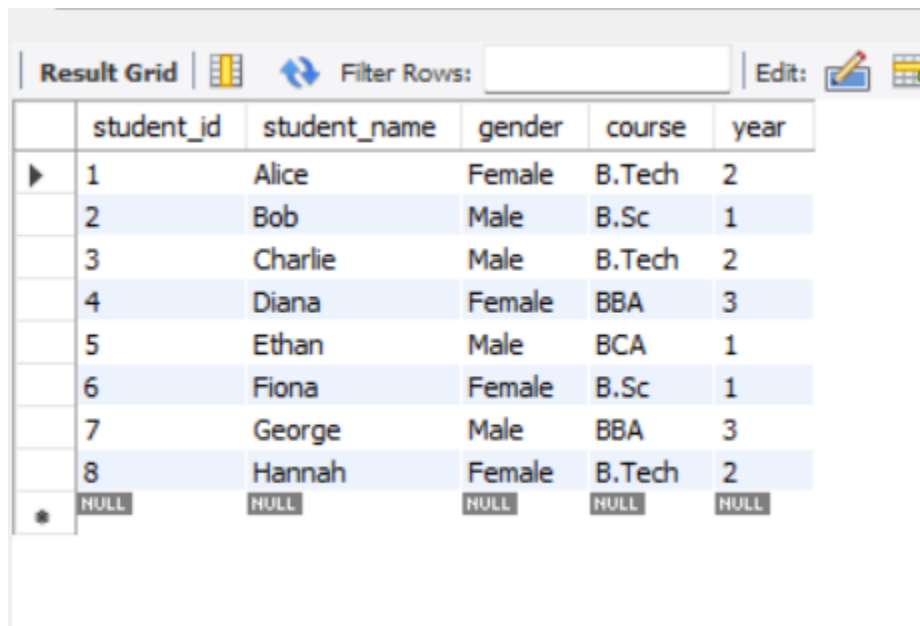
```
INSERT INTO Rooms VALUES
(1, 1, '101', 2),
(2, 1, '102', 2),
(3, 1, '103', 2),
(4, 2, '201', 2),
(5, 2, '202', 2),
(6, 2, '203', 2);
```

```
INSERT INTO Room_Allocation VALUES
(1, 1, 1, '2025-01-10'),
(2, 2, 1, '2025-01-11'),
(3, 3, 2, '2025-01-12'),
(4, 4, 2, '2025-01-13'),
(5, 5, 3, '2025-01-14'),
(6, 6, 3, '2025-01-15'),
(7, 7, 4, '2025-01-16'),
```

(8, 8, 5, '2025-01-17');

1.-- List all students

SELECT \* FROM Students;



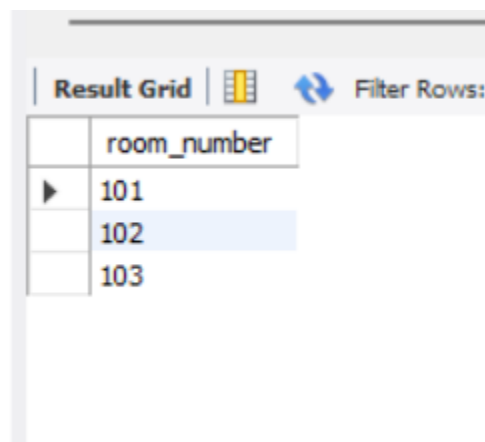
The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query. The columns are 'student\_id', 'student\_name', 'gender', 'course', and 'year'. There are 8 rows of data, each representing a student. The first row is Alice (Female, B.Tech, 2), the second is Bob (Male, B.Sc, 1), the third is Charlie (Male, B.Tech, 2), the fourth is Diana (Female, BBA, 3), the fifth is Ethan (Male, BCA, 1), the sixth is Fiona (Female, B.Sc, 1), the seventh is George (Male, BBA, 3), and the eighth is Hannah (Female, B.Tech, 2). Below the data rows, there is a row with all NULL values. The interface also includes a 'Filter Rows' field and an 'Edit' button.

	student_id	student_name	gender	course	year
▶	1	Alice	Female	B.Tech	2
	2	Bob	Male	B.Sc	1
	3	Charlie	Male	B.Tech	2
	4	Diana	Female	BBA	3
	5	Ethan	Male	BCA	1
	6	Fiona	Female	B.Sc	1
	7	George	Male	BBA	3
	8	Hannah	Female	B.Tech	2
*	NULL	NULL	NULL	NULL	NULL

2. -- Find all rooms in Sunrise Hostel

SELECT room\_number FROM Rooms

WHERE hostel\_id = (SELECT hostel\_id FROM Hostels WHERE  
hostel\_name='Sunrise Hostel');

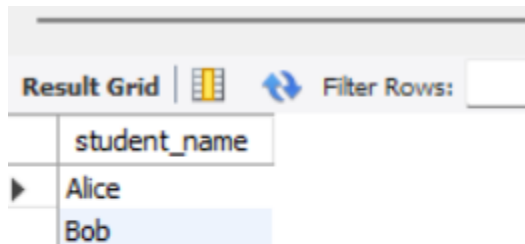


The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query. The column is 'room\_number'. There are 3 rows of data, each representing a room number. The first row is 101, the second is 102, and the third is 103. The interface also includes a 'Filter Rows' field.

	room_number
▶	101
	102
	103

3. -- : Show students allocated to room 101

```
SELECT student_name FROM Students  
WHERE student_id IN (SELECT student_id FROM Room_Allocation WHERE  
room_id=1);
```

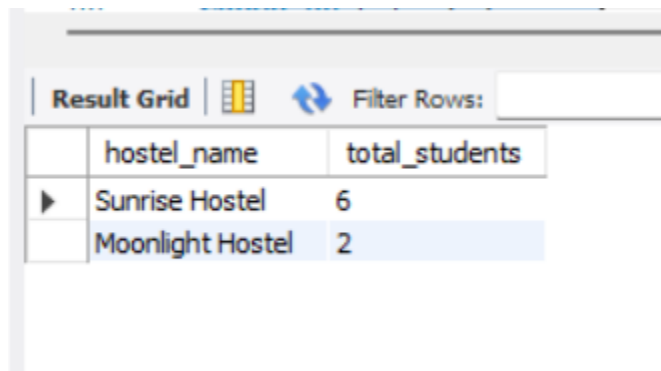


The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query. The first column is labeled 'student\_name'. There are two rows of data: 'Alice' and 'Bob'. The 'Bob' row is highlighted in blue.

student_name
Alice
Bob

4.--: Count total students in each hostel

```
SELECT h.hostel_name, COUNT(ra.student_id) AS total_students  
FROM Hostels h  
JOIN Rooms r ON h.hostel_id = r.hostel_id  
JOIN Room_Allocation ra ON r.room_id = ra.room_id  
GROUP BY h.hostel_name;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a query. The first column is labeled 'hostel\_name' and the second column is labeled 'total\_students'. There are two rows of data: 'Sunrise Hostel' with a count of 6, and 'Moonlight Hostel' with a count of 2. The 'Moonlight Hostel' row is highlighted in blue.

hostel_name	total_students
Sunrise Hostel	6
Moonlight Hostel	2

5.-- List unallocated students

```
SELECT student_name FROM Students  
WHERE student_id NOT IN (SELECT student_id FROM Room_Allocation);
```

Result Grid	
	student_name

6.--: Find rooms with remaining capacity  
 SELECT r.room\_number, (r.capacity - COUNT(ra.student\_id)) AS  
 remaining\_capacity  
 FROM Rooms r  
 LEFT JOIN Room\_Allocation ra ON r.room\_id = ra.room\_id  
 GROUP BY r.room\_id, r.room\_number, r.capacity  
 HAVING remaining\_capacity > 0;

	room_number	remaining_capacity
▶	201	1
	202	1
	203	2

7.-- : Get allocation date for each student  
 SELECT s.student\_name, ra.allocation\_date  
 FROM Students s  
 JOIN Room\_Allocation ra ON s.student\_id = ra.student\_id;

Result Grid			Filter Rows:
	student_name	allocation_date	
▶	Alice	2025-01-10	
	Bob	2025-01-11	
	Charlie	2025-01-12	
	Diana	2025-01-13	
	Ethan	2025-01-14	
	Fiona	2025-01-15	
	George	2025-01-16	
	Hannah	2025-01-17	

Result 8 x

8.--: List male students in Moonlight Hostel  
 SELECT s.student\_name FROM Students s  
 JOIN Room\_Allocation ra ON s.student\_id = ra.student\_id  
 JOIN Rooms r ON ra.room\_id = r.room\_id  
 JOIN Hostels h ON r.hostel\_id = h.hostel\_id  
 WHERE s.gender='Male' AND h.hostel\_name='Moonlight Hostel';

Result Grid		Filter Rows:
	student_name	
▶	George	

9.--: Find hostels with more than 1 student allocated  
 SELECT h.hostel\_name FROM Hostels h  
 JOIN Rooms r ON h.hostel\_id = r.hostel\_id

```

JOIN Room_Allocation ra ON r.room_id = ra.room_id
GROUP BY h.hostel_name
HAVING COUNT(ra.student_id) > 1;

```

Result Grid	
	hostel_name
▶	Sunrise Hostel
	Moonlight Hostel

10.--: Show students allocated in alphabetical order

```

SELECT s.student_name FROM Students s
JOIN Room_Allocation ra ON s.student_id = ra.student_id
ORDER BY s.student_name;

```

Result Grid	
	student_name
▶	Alice
	Bob
	Charlie
	Diana
	Ethan
	Fiona
	George
	Hannah

11.-- : Display hostel capacity vs allocated students

```

SELECT h.hostel_name, h.capacity, COUNT(ra.student_id) AS
allocated_students
FROM Hostels h
LEFT JOIN Rooms r ON h.hostel_id = r.hostel_id
LEFT JOIN Room_Allocation ra ON r.room_id = ra.room_id
GROUP BY h.hostel_name, h.capacity;

```

	hostel_name	capacity	allocated_students
▶	Sunrise Hostel	10	6
	Moonlight Hostel	8	2

12. --: List students allocated after a specific date



```
SELECT s.student_name, ra.allocation_date FROM Students s
JOIN Room_Allocation ra ON s.student_id = ra.student_id
WHERE ra.allocation_date > '2025-01-10';
```

Result Grid		Filter Rows:
	student_name	allocation_date
▶	Bob	2025-01-11
	Charlie	2025-01-12
	Diana	2025-01-13
	Ethan	2025-01-14
	Fiona	2025-01-15
	George	2025-01-16
	Hannah	2025-01-17

13.--: Find students sharing the same room

```
SELECT r.room_number, GROUP_CONCAT(s.student_name) AS students
FROM Rooms r
JOIN Room_Allocation ra ON r.room_id = ra.room_id
JOIN Students s ON ra.student_id = s.student_id
GROUP BY r.room_number
HAVING COUNT(s.student_id) > 1;
```



Result Grid     Filter Rows: <input type="text"/>		
	room_number	students
▶	101	Alice,Bob
	102	Charlie,Diana
	103	Ethan,Fiona



14.-- : Count students by course in hostels

```
SELECT s.course, COUNT(ra.student_id) AS total_students
```

```
FROM Students s
```

```
LEFT JOIN Room_Allocation ra ON s.student_id = ra.student_id
```

```
GROUP BY s.course;
```

Result Grid     Filter Rows: <input type="text"/>		
	course	total_students
▶	B.Tech	3
	B.Sc	2
	BBA	2
	BCA	1



15.--: Find rooms fully occupied

```
SELECT r.room_number FROM Rooms r
```

```
JOIN Room_Allocation ra ON r.room_id = ra.room_id
```

```
GROUP BY r.room_id, r.capacity, r.room_number
```

```
HAVING COUNT(ra.student_id) = r.capacity;
```

Result Grid			 Filter Row
	room_number		
▶	101		
	102		
	103		