CASE STUDY FOR SQL

Ischool

INTRODUCTION

A complex query is a query that uses more than one parameter and may comprise a combination of several joins across multiple tables and quite a few nested subqueries. Complex queries also frequently involve heavy use of **AND** and **OR** clauses.There are many reasons why you might want to use a complex query in MySQL. For example, you might want to:

- Search for data that meets multiple criteria
- Join data from multiple tables
- Nested subqueries

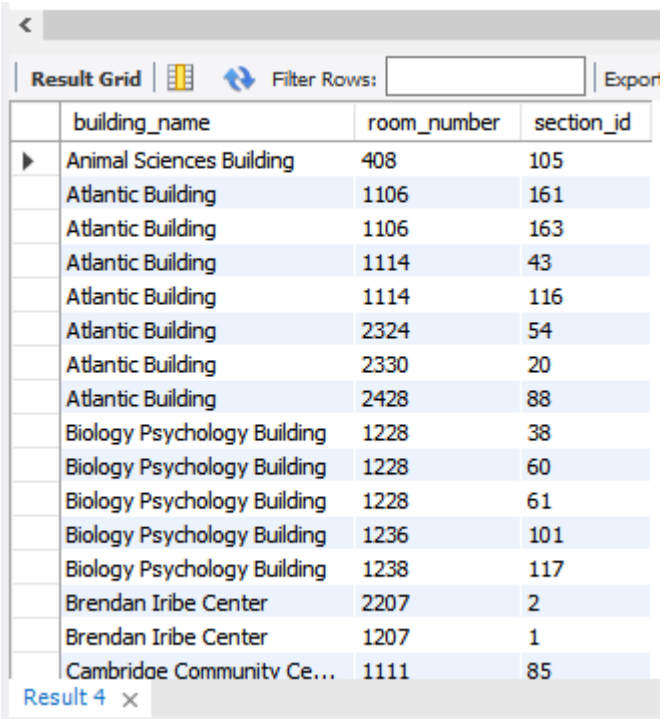
CASE STUDY:

Consider as a Database Administrator at the prestigious learning center called ischool , renowned for its extensive and diverse course offerings. The university operates a MySQL database named “ischool” that stores a wealth of information, from course details to student records.

CASE STUDY QUESTIONS AND ANSWERS

(1) Show all class locations

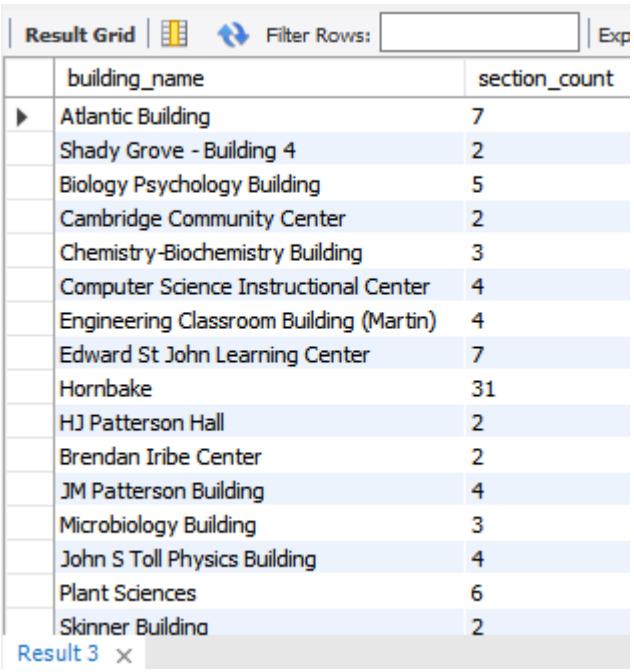
```
SELECT L.building_name,L.room_number,C.section_id
FROM locations L
JOIN course_sections C ON L.location_id=C.location_id
ORDER BY L.building_name
```



building_name	room_number	section_id
Animal Sciences Building	408	105
Atlantic Building	1106	161
Atlantic Building	1106	163
Atlantic Building	1114	43
Atlantic Building	1114	116
Atlantic Building	2324	54
Atlantic Building	2330	20
Atlantic Building	2428	88
Biology Psychology Building	1228	38
Biology Psychology Building	1228	60
Biology Psychology Building	1228	61
Biology Psychology Building	1236	101
Biology Psychology Building	1238	117
Brendan Iribe Center	2207	2
Brendan Iribe Center	1207	1
Cambridge Community Ce...	1111	85

(2) Aggregate sections by location

```
SELECT L.building_name, COUNT(C.section_id) AS section_count
FROM locations L
JOIN course_sections C ON L.location_id=C.location_id
GROUP  BY L.building_name
HAVING COUNT(C.section_id)>1;
```



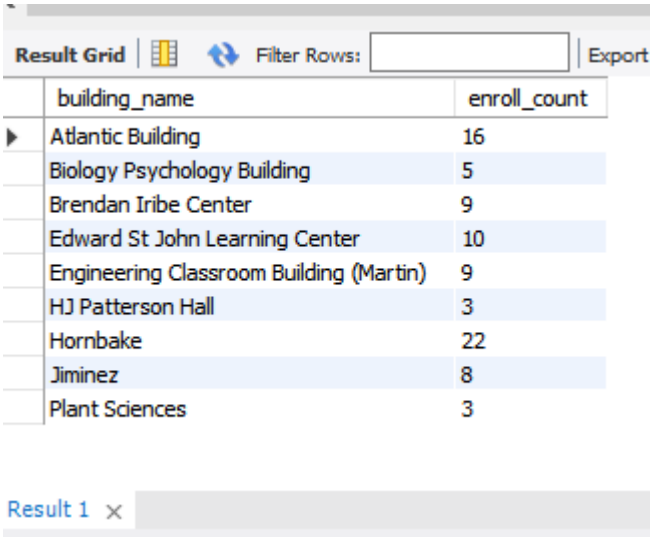
Result Grid | Filter Rows: | Exp

building_name	section_count
Atlantic Building	7
Shady Grove - Building 4	2
Biology Psychology Building	5
Cambridge Community Center	2
Chemistry-Biochemistry Building	3
Computer Science Instructional Center	4
Engineering Classroom Building (Martin)	4
Edward St John Learning Center	7
Hornbake	31
HJ Patterson Hall	2
Brendan Iribe Center	2
JM Patterson Building	4
Microbiology Building	3
John S Toll Physics Building	4
Plant Sciences	6
Skinner Building	2

Result 3

(3) Aggregate enrollment by location

```
SELECT L.building_name, COUNT(E.person_id) AS enroll_count
FROM locations L
JOIN course_sections C ON L.location_id=C.location_id
JOIN enrollments E ON C.section_id= E.section_id
GROUP   BY L.building_name
ORDER BY L.building_name;
```



Result Grid | Filter Rows: | Export

building_name	enroll_count
Atlantic Building	16
Biology Psychology Building	5
Brendan Iribe Center	9
Edward St John Learning Center	10
Engineering Classroom Building (Martin)	9
HJ Patterson Hall	3
Hornbake	22
Jiminez	8
Plant Sciences	3

Result 1 ×

(4) **Aggregate both sections and enrollment by location, using a subquery or CTE**

```
WITH section_counts AS (
  SELECT L.building_name, COUNT(DISTINCT C.section_id) AS
section_count
  FROM locations L
  JOIN course_sections C ON L.location_id = C.location_id
  GROUP BY L.building_name
  HAVING COUNT(DISTINCT C.section_id) > 1
),
enrollment_counts AS (
  SELECT L.building_name, COUNT(E.person_id) AS enroll_count
  FROM locations L
```

```
  JOIN course_sections C ON L.location_id = C.location_id
  JOIN enrollments E ON C.section_id = E.section_id
  GROUP BY L.building_name
)
SELECT S.building_name, S.section_count, E.enroll_count
FROM section_counts S
LEFT JOIN enrollment_counts E ON S.building_name = E.building_name
ORDER BY S.building_name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

building_name	section_count	enroll_count
Atlantic Building	7	16
Biology Psychology Building	5	5
Brendan Iribe Center	2	9
Cambridge Community Center	2	NULL
Chemistry-Biochemistry Building	3	NULL
Computer Science Instructional Center	4	NULL
Edward St John Learning Center	7	10
Engineering Classroom Building (Martin)	4	9
HJ Patterson Hall	2	3
Hornbake	31	22
JM Patterson Building	4	NULL
John S Toll Physics Building	4	NULL
Microbiology Building	3	NULL
Plant Sciences	6	3
Shady Grove - Building 4	2	NULL
Skinner Building	2	NULL

Result 2

(5) Retrieve the 'building_name' and 'room_number' from the 'building_info' table where the 'section_id' is 105

```
SELECT building_name , room_number
FROM building_info
WHERE section_id = 105;
```

Result Grid | Filter Rows: | Expor

building_name	room_number	section_id
Animal Sciences Building	408	105
Atlantic Building	1106	161
Atlantic Building	1106	163
Atlantic Building	1114	43

(6) Retrieve all rows from the building_info table where the room_number is the same for different section_ids within the same building_name.

```
SELECT building_name, room_number, section_id

FROM building_info

WHERE room_number IN (

   SELECT room_number

   FROM building_info

   GROUP BY room_number, building_name

   HAVING COUNT(DISTINCT section_id) > 1 )
```

Biology Psychology Building	1228	38
Biology Psychology Building	1228	60
Biology Psychology Building	1228	61
Biology Psychology Building	1236	101
Biology Psychology Building	1238	117

CONCLUSION:

This article has explored the concept of complex queries in MySQL and provided a case study of a complex query that can be used to analyze ischool data. Complex queries can be a powerful tool for data analysis.