13 Oct 2022

# Homework #4

## Queries and their meaning (+ output)

For this homework we are focusing on doing the “hardest task first”. Therefore the following queries are the most difficult ones that are also the centre of our project. This is the current situation of the tables:

```
+---------------------------- +
| Tables_in_group12      |
+----------------------------+
| BOOKS                  |
| Bookable               |
| Bookable_Equipment     |
| Bookable_Facility      |
| College                |
| Event_CO_Hours         |
| Event_Parent           |
| Event_Uni              |
| LIVES_IN               |
| MANAGES                |
| ORGANIZES              |
| PLANS_TO_ATTEND        |
| Residential_Staff      |
| Residential_Staff_RA   |
| Residential_Staff_RM   |
| Student                |
| Student_CO_Member      |
| WORKS_IN               |
+----------------------------+
```

*1. Central in our project is booking equipment and having a clue whether there is something we need available. Therefore, this query shows the names of all the Bookables that are not booked.*

Nested query - bookables that are free (not booked)

```
SELECT Bookable.name
FROM Bookable
WHERE bookable_id
NOT IN (
SELECT bookable_id
FROM BOOKS
);
```

```
MariaDB [group12]> SELECT Bookable.name
    -> FROM Bookable
    -> WHERE bookable_id
    -> NOT IN (
    -> SELECT bookable_id
    -> FROM BOOKS
    -> );
+-------------+
| name        |
+-------------+
| C4_Vacuum3  |
| C4_Football |
+-------------+
```

13 Oct 2022

*2. After that, organisers, and residential staff would like to know which student attends which event, whether for general information or maybe it brings credits. Therefore, this query will give us the name of the student and the event they attend.*

Join - students who attend events

```
SELECT Student.name, Event_Parent.name
FROM Student, Event_Parent,
PLANS_TO_ATTEND
WHERE Student.student_id =
PLANS_TO_ATTEND.student_id
AND Event_Parent.event_id =
PLANS_TO_ATTEND.event_id;
```

```
MariaDB [group12]> SELECT Student.name, Event_Parent.name
    -> FROM Student, Event_Parent, PLANS_TO_ATTEND
    -> WHERE Student.student_id = PLANS_TO_ATTEND.student_id
    -> AND Event_Parent.event_id = PLANS_TO_ATTEND.event_id;
+---------+-------------+
| name    | name        |
+---------+-------------+
| John    | Jacolympix22 |
| Severus | Jacolympix22 |
| Lisa    | Jacolympix22 |
| Bob     | C4_CO_Mon_Ev |
| Severus | C4_CO_Wed_No |
| John    | Town_Hall_1  |
+---------+-------------+
```

*3. Sometimes it is good to know which College Office has less members, or more than needed. This query will give us the COs where the number of students who work in them are bigger than 1. Of course depending on the need the last line of the query can vary.*

Group by + aggregation - see which CO has more than 1 member

```
SELECT C.name
FROM Student S, WORKS_IN W, College C
WHERE S.student_id = W.student_id
AND W.college_id = C.college_id
GROUP BY C.name
HAVING COUNT(*)>1;
```

```
MariaDB [group12]> SELECT C.name
    -> FROM Student S, WORKS_IN W, College C
    -> WHERE S.student_id = W.student_id
    -> AND W.college_id = C.college_id
    -> GROUP BY C.name
    -> HAVING COUNT(*)>1;
+------------+
| name       |
+------------+
| College IV |
+------------+
```

*4. Going back to the bookables, if there is a problem in the college where equipment is missing or a booked room has been damaged, we want to know who booked it. This query gives us the name of the student and what they booked.*

Join - See who booked what

```
SELECT Student.name, Bookable.name
FROM Student, Bookable, BOOKS
WHERE Student.student_id = BOOKS.student_id
AND Bookable.bookable_id = BOOKS.bookable_id;
```

```
MariaDB [group12]> SELECT Student.name, Bookable.name
    -> FROM Student, Bookable, BOOKS
    -> WHERE Student.student_id = BOOKS.student_id
    -> AND Bookable.bookable_id = BOOKS.bookable_id;
+----------+-----------------+
| name     | name            |
+----------+-----------------+
| Karolina | C4_Vacuum2      |
| Leia     | C4_MMR          |
| Leia     | C4_Vacuum1      |
| Dwayne   | TOS             |
| John     | C4_Monopoly     |
| Pit      | C4_Tennis_Racet1 |
+----------+-----------------+
```

*5. Students want to attend CO hours and if there is something they need from a specific CO or they want to visit, they will want to know when they are open. This query returns the name of the event - which CO's hours, and when they occur.*

Join (ISA hierarchy) - See all the CO hours on campus

```
SELECT E.name, E.hours
FROM Event_Parent E, Event_CO_Hours CO
WHERE E.event_id = CO.event_id;
```

```
MariaDB [group12]> SELECT E.name, E.hours
    -> FROM Event_Parent E, Event_CO_Hours C
    -> WHERE E.event_id = CO.event_id;
+--------------+---------------------+
| name         | hours               |
+--------------+---------------------+
| C4_CO_Mon_Ev | 2022-10-10 19:00:00 |
| C4_CO_Tue_Ev | 2022-10-10 19:00:00 |
| C4_CO_Wed_No | 2022-10-10 13:00:00 |
+--------------+---------------------+
```

*6. Students, or residential staff might want to know who works in the CO, therefore they can look them up using this query. This returns the name of the students, their email, and the college they work in, since they are present in the relationship table.*

Join - Show which student works in which CO

```
SELECT S.name, S.email, C.name
FROM Student S, WORKS_IN W, College C
WHERE S.student_id = W.student_id
AND W.college_id = C.college_id;
```

```
MariaDB [group12]> SELECT S.name, S.email, C.name
    -> FROM Student S, WORKS_IN W, College C
    -> WHERE S.student_id = W.student_id
    -> AND W.college_id = C.college_id;
+----------+-----------------------------+------------------+
| name     | email                       | name             |
+----------+-----------------------------+------------------+
| Christie | c.abigail@con-uni.de        | College IV       |
| Severus  | s.snape@jacobs-university.de | College IV       |
| Dwayne   | d.rock@gmail.com            | College IV       |
| Bob      | b.smith@jacobs-university.de | College Morecator |
+----------+-----------------------------+------------------+
```

13 Oct 2022

*7. Residential staff might be interested in how many people live in each college. This query returns the number of students living in each college grouped by the name of the college.*
Group by + aggregation

SELECT COUNT(L.student_id), C.name FROM LIVES_IN L, College C WHERE L.college_id=C.college_id GROUP BY C.name;

```
MariaDB [group12]> SELECT COUNT(L.student_id), C.name  FROM  LIVES_IN L, College C WHERE L.college_id=C.college_id GROUP BY C.name;
+--------------------+--------------------+
| COUNT(L.student_id) | name               |
+--------------------+--------------------+
|                 14 | College IV         |
|                  1 | College Morecator  |
|                  1 | College West-Metall |
+--------------------+--------------------+
3 rows in set (0.001 sec)
```

*8. Students might be interested in what is the email of the residential staff that is managing their college. The query returns the name, surname, email and the college that is being managed by each residential staff.*

SELECT RS.name, RS.surname, RS.email, C.name FROM MANAGES M, College C, Residential_Staff RS WHERE M.residential_staff_id= RS.residential_staff_id AND C.college_id = M.college_id;

```
MariaDB [group12]> SELECT RS.name, RS.surname, RS.email, C.name FROM MANAGES M, College C, Residential_Staff RS WHERE M.residential_staff_id= RS.residential_staff_id AND C.college_id =
M.college_id;
+--------+---------+-----------------------------+------------------+
| name   | surname | email                       | name             |
+--------+---------+-----------------------------+------------------+
| Emilia | Schmidt | e.schmidt@jacobs-university.de | College IV       |
| Ben    | Kenobi  | highground@web.de           | College IV       |
| R2     | D2      | Droid@jacobs-university.de  | College IV       |
| Angela | Merkel  | a.merkel@bund.de            | College IV       |
| Maniel | Derten  | M.derten@jacobs-university.de | College Morecator |
+--------+---------+-----------------------------+------------------+
5 rows in set (0.001 sec)
```

*9. There might be scenarios when the administration of the colleges want to know how many RAs are on duty at the same time. The query returns the number of RA on shift in the entire campus.*

SELECT COUNT(residential_staff_id) FROM Residential_Staff WHERE Residential_Staff.on_shift=1;

```
MariaDB [group12]> SELECT COUNT(residential_staff_id) from Residential_Staff where Residential_Staff.on_shift = 1;
+------------------------------+
| COUNT(residential_staff_id) |
+------------------------------+
|                            2 |
+------------------------------+
1 row in set (0.000 sec)
```

13 Oct 2022

*10. When booking facilities one of the most important information needed by the students is what are the opening hours. This query returns the name of the bookable facility and the opening hours.*

SELECT B.name, BF.opening_hours FROM Bookable B, Bookable_Facility BF WHERE BF.bookable_id=B.bookable_id;

```
MariaDB [group12]> SELECT B.name, BF.opening_hours FROM Bookable B, Bookable_Facility BF WHERE BF.bookable_id=B.bookable_id;
+--------+---------------------+
| name   | opening_hours       |
+--------+---------------------+
| C4_MMR | 2022-10-10 18:00:00 |
| TOS    | 2022-10-10 20:00:00 |
+--------+---------------------+
2 rows in set (0.001 sec)
```

*11. One of the most used information by the student when ordering different products online would be the address of their College. A simple query yet powerful one is returning the name of the college and the location.*

SELECT name, location FROM College;

```
MariaDB [group12]> SELECT name, location FROM College;
+----------------------+-------------+
| name                 | location    |
+----------------------+-------------+
| College IV           | Uni-Ring 6  |
| College West-Metall  | Uni-Ring 3  |
| College Morecator    | Campring 8  |
+----------------------+-------------+
3 rows in set (0.000 sec)
```

*12. Last but not least, students might be interested in who is organising what event. This query returns the name, surname and email of the student-organiser and the event that will take place.*

SELECT S.name, S.surname, S.email, E.name FROM Student S, Event_Parent E, ORGANIZES O WHERE S.student_id = O.student_id AND E.event_id = O.event_id;

```
MariaDB [group12]> SELECT S.name, S.surname, S.email, E.name FROM Student S, Event_Parent E, ORGANIZES O WHERE S.student_id = O.student_id AND E.event_id = O.event_id;
+----------+-----------+-----------------------------------+--------------+
| name     | surname   | email                             | name         |
+----------+-----------+-----------------------------------+--------------+
| Christie | Abigail   | c.abigail@con-uni.de              | C4_CO_Wed_No |
| George   | Washington | g.washington@jacobs-university.de | Town_Hall_1  |
| Bob      | Smith     | b.smith@jacobs-university.de      | Dancestoned  |
| Dwayne   | Rock      | d.rock@gmail.com                  | C4_CO_Tue_Ev |
| Allice   | Wonder    | a.won11@yahoo.com                 | Jacolympix22 |
+----------+-----------+-----------------------------------+--------------+
5 rows in set (0.001 sec)
```