

Exercise:

Use the correct function to return the number of records that have the **Price** value set to 18.

```
SELECT count(*)
FROM Products
where Price = 18;
```

Submit Answer »

d)
 select post_id, count(*) from Comment
 where email <> 'lara@tvm.de'
 group by post_id having count(*)

SQL Comparison Operators

Operator	Description	Example
=	Equal to	Try It
>	Greater than	Try It
<	Less than	Try It
>=	Greater than or equal to	Try It
<=	Less than or equal to	Try It
<>	Not equal to	Try It

<> nicht gleich

primary key =

Primärschlüssel =
 Durch Angabe des Primärschlüssels ist ein Tupel in der Tabelle eindeutig bestimmt

User		Post	
id	password	id	title
1	12345678	1	Post 1
2	87654321	2	Post 2
3	11223344	3	Post 3
4	55667788	4	Post 4

b) select email from User u
 where
 (select count(*) from Post p where p.email = u.email) > 1
 select email from Post
 group by email having count(*) > 1

- (c) Number of comments for each post having at least one comment.
- (d) Number of comments for each post having at least one comment, without taking into account Lara's comments.
- (e) Average number of comments per post.

~~c) select p.id from Post p
 where (select count(*) from Comment c where c.post_id = p.id) >= 1
 group~~

c) select post_id, count(*) from Comment
 group by post_id having count(*) >= 1

e) select avg(x.numComments) from (select post_id, count(*) as numComments from Comment group by post_id) X;

Selektiere alle Autoren

select * from User where author = 'true'

select nickname from User
 select password, nickname from User

a) select nickname from User u where
 (select count(*) from Comment c where c.email = u.email) >= 1
 select nickname from User where u.email in (select c.email from Comment c where u.email = c.mail)

In

- in returns true, if an element belongs to a set, and it returns false otherwise.
- not negates the resulting boolean value.
- Examples

A		B	
id	a	id	b
1	2	1	3
2	1	2	4
3	4	3	1
4	3	4	2

```
select A.id  

from A  

where a in (select b from B);
```

select * from A where a not in (select b from B)

id	a
1	2
3	4

select * from A where id not in (select b from B)

id	a
2	1
4	3

Nested Queries

- "Sub-query" or "inner query" and "outer query".
- Sub-queries that return a single tuple versus sub-queries that return multiple tuples
- Sub-queries in select-clause, from-clause, where-clause
- Correlated sub-queries versus non-correlated sub-queries

$avg(X.num(Comments))$ from (select post_id, count(*) as \sum num(Comments) group by post-id) X;

select * from A where ...

id	a
2	1
4	3

Nested Queries

- "Sub-query" or "inner query" and "outer query".
- Sub-queries that return a single tuple versus sub-queries that return multiple tuples
- Sub-queries in select-clause, from-clause, where-clause
- Correlated sub-queries versus non-correlated sub-queries
- Examples:
 - Sub-query returns single tuple, is in where-clause, not correlated.

A		B	
id	a	id	b
1	2	a	1
2	7	b	7

```

select *
from A
where A.a = (select max(B.b) from B);
    
```

id	a
2	7

select * from A where A.id = (select min(b) from B)

id	a
1	2

Homework 8.1: SQL — University — **

Referring to the schema in Tutorial 8.1, provide SQL queries that do the following:

1. Determine the nickname of the users who didn't like their own comments (1P)
2. Determine the id and title of the posts where the author didn't like any of the comments. (1P)
3. Determine the post id and the total number of tags of the posts with the largest number of tags. (1P)

You can download the database *SQL.db.sqlite3* from Moodle in the section from Homework 08, import it in IntelliJ and test your SQL queries.

To import the database in IntelliJ:

1. open the "Database" tab in the top-right corner of IntelliJ,
2. drag-and-drop *SQL.db.sqlite3* in the "Database" tool window,
3. in the wizard, click on "Download missing driver files" (needed only for the first import)
4. click on "OK",
5. you can now test your queries in the "console" tab.

- Sub-query returns single tuple, is in select-clause, is correlated

A		B		
id		id	b	a
1		a	1	1
2		b	7	1
		c	2	2
		d	5	2

```

select A.id, (select sum(B.b) from B where B.a = A.id) as "s"
from A;
    
```

id	s
1	8
2	7

select A.id, (select max(B.b) from B) as m

id	m
1	7
2	5

select A.id, (select avg(B.b) ...) as avg

id	avg
1	4
2	3,5

- Sub-query returns multiple tuples, is in from-clause, not correlated

```

select A.id, X.b
from A, (select b from B) X;
    
```

A		B		
id		id	b	a
1		a	1	1
2		b	7	1
		c	2	2
		d	5	2

id	b
1	1
1	7
1	2
1	5
2	1
2	7
2	2
2	5

$A = \{1, 2, 3\}$
 $B = \{4, 5\}$
 $A \times B = \{(a, b) \mid a \in A, b \in B\}$
 $= \{(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)\}$