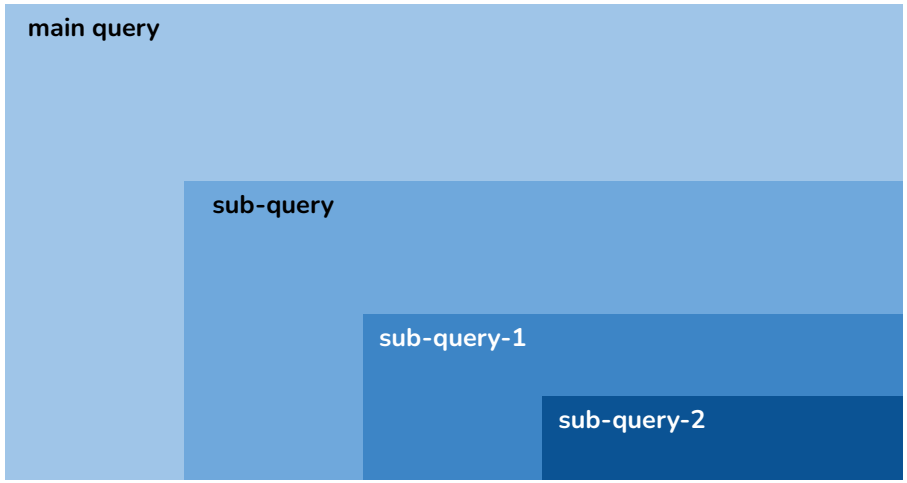
Subqueries in SQL - How they work? with examples

Write queries within queries to find answers to complex questions!



Examples	Concept
Example-1	Single Row Subquery
Example-2	Multiple Row Subquery
Example-3	Multiple Column Subquery
Example-4	Nested Subquery
Example-5	Subquery using Aggregare Function
Example-6	Subquery with Joins
Example-7	Subquery in FROM Clause
Example-8	Subquery using EXISTS

1 *Find a list of students who scored higher than average.*

stud_details		
stud_id	**name**	**marks**
1001	John	30
1002	Marcus	45
1003	Robert	60
1004	Luke	70
1005	Ryan	50
1006	Chris	100

Let's first find out what the average is.

59

What are the rows where the student has scored above-average marks?

stud_details		
stud_id	**name**	**marks**
1001	John	30
1002	Marcus	45
1003	Robert	60
1004	Luke	70
1005	Ryan	50
1006	Chris	100

How can I write a query to get this answer programmatically?

[SUBQUERY]

SELECT AVG(marks) FROM stud_details;

59

The query above can be part of an outer query to filter for above-average rows.

SELECT * FROM stud_details WHERE marks > [SUBQUERY]

SELECT * FROM stud_details WHERE marks > *SELECT AVG(marks) FROM stud_details*

SELECT * FROM stud_details WHERE marks > (59)

stud_id	name	marks
1003	Robert	60
1004	Luke	70
1006	Chris	100

2 *Find the average age of students who have scored above 75 in math.*

stud_details		
stud_id	name	age
1001	John	9
1002	Marcus	11
1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

stud_marks		
stud_id	subject	marks
1001	math	50
1001	science	80
1002	math	60
1002	science	30
1003	math	80
1003	science	40
1004	math	85
1004	science	100
1005	math	90
1005	science	45
1006	math	95
1006	science	90

Let's first find the student IDs for those who have scored above 75 in math.

stud_marks		
stud_id	subject	marks
1001	math	50
1001	science	80
1002	math	60
1002	science	30
1003	math	80
1003	science	40
1004	math	85
1004	science	100
1005	math	90
1005	science	45
1006	math	95
1006	science	90

Now, let's find the average age for those student IDs who have scored more than 75 in math.

stud_details		
stud_id	name	age
1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

average_age	10

How can I write a query to get this answer programmatically?

[SUBQUERY]

SELECT stud_id FROM stud_marks WHERE marks > 75 AND subject = 'math'

stud_details		
stud_id	name	age
1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

The query above can be part of an outer query to filter student ids above 75 in math.

SELECT AVG(age) as average_age FROM stud_details WHERE stud_id IN [SUBQUERY]

SELECT AVG(age) as average_age FROM stud_details WHERE stud_id IN (SELECT stud_id FROM stud_marks WHERE marks > 75 AND subject = 'math')

SELECT AVG(age) as average_age FROM stud_details WHERE stud_id IN (1003,1004,1005,1006)

stud_details		
stud_id	name	age
1003	Robert	10
1004	Luke	10

1005	Ryan	9
1006	Chris	11

average_age	10

3 *Find the transactions of the products with the lowest price, within each category.*

transaction				
id	category	product	cost	price
1001	X	A	45	50
1002	X	B	50	60
1003	X	A	40	43
1004	Y	C	56	60
1005	Y	D	70	85
1006	Z	D	70	90

Let's first find the categories with the lowest prices.

category	price
X	43
Y	60
Z	90

Let's return the transaction details of those records that have a matching category and price.

transaction				
id	category	product	cost	price
1003	X	A	40	43
1004	Y	C	56	60
1006	Z	D	70	90

Find the transactions with the lowest price for each category.

transaction				
id	category	product	cost	price
1001	X	A	45	50
1002	X	B	50	60
1003	X	A	40	43
1004	Y	C	56	60
1005	Y	D	70	85
1006	Z	D	70	90

[SUBQUERY]

SELECT Category, MIN(Price) as MinPrice FROM Transaction GROUP BY Category

The above query will give me this table:

category	price
X	43
Y	60
Z	90

The query above can be part of an outer query to filter unique categories along with their lowest price.

SELECT * FROM transaction WHERE (category, price) IN [SUBQUERY]

```
SELECT * FROM transaction
WHERE (category, min_price) IN (SELECT category, MIN(price) as min_price
                                FROM transaction
                                GROUP BY category)
```

transaction				
id	category	product	cost	price
1003	X	A	40	43

1004	Y	C	56	60
1006	Z	D	70	90

4 *What is the average age of students who have scored below average in science?*

stud_details		
stud_id	name	age
1001	John	9
1002	Marcus	11
1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

stud_marks		
stud_id	subject	marks
1001	math	50
1001	science	80
1002	math	60
1002	science	30
1003	math	80
1003	science	40
1004	math	85
1004	science	100
1005	math	90
1005	science	45
1006	math	95
1006	science	90

Let's first find the average marks in science.

stud_marks		
stud_id	subject	marks
1001	math	50
1001	science	80
1002	math	60
1002	science	30
1003	math	80
1003	science	40
1004	math	85
1004	science	100
1005	math	90

1005	science	45
1006	math	95
1006	science	90

64

Now, find the student IDs whose science grades are below the national average.

stud_id	marks
1002	30
1003	40
1005	45

Now, let's retrieve the average age of the rows that match the above student IDs.

stud_details		
stud_id	name	age
1001	John	9
1002	Marcus	11
1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

Average_Age	10

What are the rows where the student has scored below-average marks?

stud_details		
stud_id	name	Age
1001	John	9
1002	Marcus	11

stud_marks		
stud_id	subject	marks
1001	math	50
1001	science	80

1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

1002	math	60
1002	science	30
1003	math	80
1003	science	40
1004	math	85
1004	science	100
1005	math	90
1005	science	45
1006	math	95
1006	science	90

How can I write a query to get this answer programmatically?

[SUBQUERY-1]

```
SELECT AVG(marks) FROM stud_marks WHERE subject = 'science'
```

59

The query above can be part of an outer query to filter for below-average marks in science.

[SUBQUERY-2]

```
SELECT stud_id FROM stud_marks WHERE marks < ([SUBQUERY-1]) AND subject = 'science'
SELECT stud_id FROM stud_marks WHERE marks < (SELECT AVG(marks) FROM stud_marks WHERE subject = 'science') AND subject = 'science'
```

stud_id	marks
1002	30
1003	40
1005	45

```
SELECT AVG(age) as Average_Age FROM stud_details WHERE stud_id IN ([SUBQUERY-2])
```

```
SELECT AVG(age) as Average_Age
FROM stud_details
WHERE stud_id IN (SELECT stud_id FROM stud_marks WHERE marks < ([SUBQUERY-1]) AND subject = 'science')
```

```
SELECT AVG(age) as Average_Age
FROM stud_details
WHERE stud_id IN (SELECT stud_id
                  FROM stud_marks
                  WHERE marks < (SELECT AVG(marks) FROM stud_marks WHERE subject = 'science')
                  AND subject = 'science')
```

Average_Age	10

5 *Find the products where the average cost of each product is more than the maximum cost of product A.*

transaction				
id	category	product	cost	price
1001	X	A	45	50
1002	X	B	50	60
1003	X	A	40	43
1004	Y	C	56	60
1005	Y	D	70	85
1006	Z	D	70	90

Let's first find the maximum cost of product A across all categories.

45

Let us now look for a product with an average cost greater than 45.

product	avg_cost
B	50
C	56
D	70

How can I write a query to get this answer programmatically?

[SUBQUERY]

```
SELECT MAX(cost) FROM transaction WHERE product = 'A';
```

45

The query above can be part of an outer query to filter for above-average rows.

```
SELECT product, AVG(cost) as avg_cost
FROM transaction
GROUP BY product
HAVING AVG(cost) > [SUBQUERY]
```

```
SELECT product, AVG(cost) as avg_cost
FROM transaction
GROUP BY product
HAVING AVG(cost) > (SELECT MAX(cost) FROM transaction WHERE product='A')
```

product	avg_cost
B	50
C	56
D	70

6 *Find student information and scores for students over the age of nine.*

stud_details		
stud_id	name	age
1001	John	9
1002	Marcus	11
1003	Robert	10
1004	Luke	10
1005	Ryan	9
1006	Chris	11

stud_marks		
stud_id	subject	marks
1001	math	50
1001	science	80
1002	math	60
1002	science	30
1003	math	80
1003	science	40
1004	math	85
1004	science	100
1005	math	90
1005	science	45
1006	math	95
1006	science	90

Let's first find all the students who are more than 9 years old.

stud_id	name	age
1002	Marcus	11
1003	Robert	10
1004	Luke	10
1006	Chris	11

Now, let's join the above data with stud_marks to find their respective marks.

stud_id	name	age	subject	marks
1002	Marcus	11	math	60
1002	Marcus	11	science	30
1003	Robert	10	math	80
1003	Robert	10	science	40
1004	Luke	10	math	85
1004	Luke	10	science	100
1006	Chris	11	math	95
1006	Chris	11	science	90

How can I write a query to get this answer programmatically?

[SUBQUERY]

SELECT * FROM stud_details WHERE age > 9

stud_id	name	age
1002	Marcus	11
1003	Robert	10
1004	Luke	10
1006	Chris	11

The query above can be part of an outer query to join with another table.

```
SELECT age9.*, mrks.subject, mrks.marks
FROM stud_marks as mrks
INNER JOIN ([SUBQUERY]) as age9
ON mrks.stud_id=age9.stud_id
```

```
SELECT age9.*, mrks.subject, M.marks
FROM stud_marks as mrks
INNER JOIN (SELECT * FROM stud_details WHERE age > 9) as age9
ON mrks.stud_id=age9.stud_id
```

stud_id	name	age	subject	marks
1002	Marcus	11	math	60
1002	Marcus	11	science	30
1003	Robert	10	math	80
1003	Robert	10	science	40
1004	Luke	10	math	85
1004	Luke	10	science	100
1006	Chris	11	math	95
1006	Chris	11	science	90

7 *Find all the instructors whose salary is greater than the average budget of all departments.*

instructor_details			
id	name	department_id	salary
1001	Richard	1	45000
1002	Michael	1	50000
1003	Rob	3	60000
1004	Marcus	2	45000
1005	Mason	2	40000
1006	Luke	3	55000

departments		
id	department	budget
1	Business	50000
2	Arts	45000
3	Science	55000

Let's first find the average budget for department.

50000

Let's look for the instructor whose salary is more than $50,000.

id	name	department_id	salary
1003	Rob	3	60000
1006	Luke	3	55000

How can I write a query to get this answer programmatically?

[SUBQUERY]

```
SELECT AVG(Budget) FROM Department
```

50000

The query above can be part of an outer query to join with another table.

```
SELECT ins.id, ins.name, ins.salary
FROM [SUBQUERY], instructor as ins
WHERE ins.salary > SUBQ
```

```
SELECT I.id, I.name, I.department_id, I.salary
FROM (SELECT AVG(Budget) FROM Department) as B, Instructor as I
WHERE I.Salary > B.AVG(Budget)
```

id	name	department_id	salary
1003	Rob	3	60000
1006	Luke	3	55000

8 *Find all the details for students who passed in science.*

stud_details		
stud_id	name	marks
1001	John	30
1002	Marcus	45
1003	Robert	60
1004	Luke	70
1005	Ryan	50
1006	Chris	100

stud_marks		
stud_id	subject	result
1001	math	pass
1001	science	pass
1002	math	pass
1002	science	fail
1003	math	pass
1003	science	fail
1004	math	pass
1004	science	pass
1005	math	pass
1005	science	pass
1006	math	pass
1006	science	pass

Let's first find all the student IDs who have passed in science.

stud_marks		
stud_id	subject	result
1001	math	pass
1001	science	pass
1002	math	pass
1002	science	fail
1003	math	pass
1003	science	fail

1004	math	pass
1004	science	pass
1005	math	pass
1005	science	pass
1006	math	pass
1006	science	pass

Now, let's find the student details, which have student IDs as highlighted above.

stud_id	name	marks
1001	John	30
1004	Luke	70
1005	Ryan	50
1006	Chris	100

How can I write a query to get this answer programmatically?

[SUBQUERY]

SELECT stud_id FROM stud_marks WHERE result = 'Pass' and subject = 'science'

This query can be used to check if these student IDs "exist" as part of a table in an outer query.

SELECT * FROM stud_details WHERE EXISTS *[SUBQUERY]*

SELECT * FROM stud_details WHERE EXISTS (SELECT stud_id FROM stud_marks WHERE result='Pass' and subject = 'science')

stud_id	name	marks
1001	John	30
1004	Luke	70
1005	Ryan	50
1006	Chris	100