



# SQL

## Hexavarsity



# Objective

- Introduction to Subquery
- Subquery example





# Introduction to Subquery

# Introduction to Subquery



- A subquery is a query nested within another query such as SELECT, INSERT, UPDATE or DELETE. Also, a subquery can be nested within another subquery.
- A subquery is called an inner query while the query that contains the subquery is called an outer query.
- A subquery can be used anywhere that expression is used and must be closed in parentheses.
- You can use a subquery in many places such as:
  - With the IN or NOT IN operator
  - With comparison operators
  - With the EXISTS or NOT EXISTS operator
  - With the ANY or ALL operator
  - In the FROM clause
  - In the SELECT clause

Outer Query

Subquery or Inner Query

```
SELECT lastname, firstname
FROM employees
WHERE officeCode IN (SELECT officeCode
                     FROM offices
                     WHERE country = 'USA')
```

# Guidelines



- Subqueries must be enclosed within parentheses.
- A subquery can have only one column in the SELECT clause, unless multiple columns are in the main query for the subquery to compare its selected columns.
- An ORDER BY command cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP BY command can be used to perform the same function as the ORDER BY in a subquery.
- Subqueries that return more than one row can only be used with multiple value operators such as the IN operator.
- The SELECT list cannot include any references to values that evaluate to a BLOB, ARRAY, CLOB, or NCLOB.
- A subquery cannot be immediately enclosed in a set function.
- The BETWEEN operator cannot be used with a subquery. However, the BETWEEN operator can be used within the subquery.

# SQL Subquery With IN and NOT IN



finds the names of all mountain bikes and road bikes products that the Bike Stores sell.

➤ *SQL subquery with the IN operator*

```
SELECT product_id, product_name
FROM production.products
WHERE category_id
IN (
    SELECT category_id
    FROM production.categories
    WHERE category_name = 'Mountain Bikes'
    OR category_name = 'Road Bikes'
);
```

product_id	product_name
1	Trek 820 - 2016
2	Ritchey Timberwolf Frameset - 2016
3	Surly Wednesday Frameset - 2016
4	Trek Fuel EX 8 29 - 2016
5	Heller Shagamaw Frame - 2016
6	Surly Ice Cream Truck Frameset - 2016
7	Trek Slash 8 27.5 - 2016
8	Trek Remedy 29 Carbon Frameset - 2016
27	Surly Big Dummy Frameset - 2017
28	Surly Karate Monkey 27.5+ Frameset - 2017
29	Trek X-Caliber 8 - 2017
30	Surly Ice Cream Truck Frameset - 2017
31	Surly Wednesday - 2017



# SQL Subquery with ANY operator



finds the products whose list prices are greater than or equal to the average list price of any product brand.

➤ *SQL subquery with the ANY operator*

```
SELECT product_name, list_price
FROM production.products
WHERE list_price >=
ANY (
    SELECT AVG (list_price)
    FROM production.products
    GROUP BY brand_id
);
```

product_name	list_price
Trek 820 - 2016	379.99
Ritchey Timberwolf Frameset - 2016	749.99
Surly Wednesday Frameset - 2016	999.99
Trek Fuel EX 8 29 - 2016	2899.99
Heller Shagamaw Frame - 2016	1320.99
Surly Ice Cream Truck Frameset - 2016	469.99
Trek Slash 8 27.5 - 2016	3999.99
Trek Remedy 29 Carbon Frameset - 2016	1799.99
Trek Conduit+ - 2016	2999.99
Surly Straggler - 2016	1549.00
Surly Straggler 650b - 2016	1680.99
Electra Townie Original 21D - 2016	549.99
Electra Cruiser 1 (24-Inch) - 2016	269.99
Electra Girl's Hawaii 1 (16-inch) - 2015/2016	269.99
Electra Moto 1 - 2016	529.99

- For each brand, the subquery finds the maximum list price. The outer query uses these max prices and determines which individual product's list price is greater than or equal to any brand's maximum list price.
- The ANY operator is a logical operator that compares a value with a set of values returned by a subquery.
- If the subquery returns no row, the condition evaluates to false.

# SQL Subquery with ALL operator

finds the products whose list price is greater than or equal to the average list price returned by the subquery

➤ *SQL subquery with the All operator*

```
SELECT product_name, list_price
FROM production.products
WHERE list_price >=
ALL (
    SELECT AVG (list_price)
    FROM production.products
    GROUP BY brand_id
)
```

product_name	list_price
Trek Fuel EX 8 29 - 2016	2899.99
Trek Slash 8 27.5 - 2016	3999.99
Trek Conduit+ - 2016	2999.99
Trek Fuel EX 9.8 29 - 2017	4999.99
Trek Fuel EX 9.8 27.5 Plus - 2017	5299.99
Trek Remedy 9.8 - 2017	5299.99
Trek Domane SL 6 - 2017	3499.99
Trek Silque SLR 7 Women's - 2017	5999.99
Trek Silque SLR 8 Women's - 2017	6499.99
Trek Domane SL Disc Frameset - 2017	3199.99
Trek Domane S 6 - 2017	2699.99
Trek Domane SLR 6 Disc - 2017	5499.99
Trek Madone 9.2 - 2017	4999.99
Trek Domane S 5 Disc - 2017	2599.99
Trek Powerfly 8 FS Plus - 2017	4999.99

Note:- The ALL operator returns TRUE if all comparison pairs (scalar\_expression, vi) evaluate to TRUE; otherwise, it returns FALSE.



# SQL Subquery with EXISTS and NOT EXISTS operator



finds the customers who bought products in 2017:

➤ *SQL subquery with the EXISTS operator*

```
SELECT customer_id, first_name, last_name, city
FROM sales.customers c
WHERE EXISTS (
    SELECT customer_id
    FROM sales.orders o
    WHERE o.customer_id = c.customer_id
    AND YEAR (order_date) = 2017
) ORDER BY first_name, last_name;
```

customer_id	first_name	last_name	city
75	Abby	Gamble	Amityville
1224	Abram	Copeland	Harlingen
673	Adam	Henderson	Los Banos
1023	Adena	Blake	Ballston Spa
1412	Adrien	Hunter	Rego Park
769	Agatha	Melton	Springfield Gardens
771	Agnes	Sims	Buffalo
1181	Agustina	Lawrence	Brooklyn
735	Aide	Franco	Atwater
384	Aimee	Meritt	Flushing
1093	Alejandrina	Hodges	Deer Park
534	Alejandro	Haney	Wantagh
562	Alejandro	Norman	Upland

find the customers who did not buy any products in 2017

➤ *SQL subquery with the NOT EXISTS operator*

```
SELECT customer_id, first_name, last_name, city
FROM sales.customers c
WHERE NOT EXISTS (
    SELECT customer_id
    FROM sales.orders o
    WHERE o.customer_id = c.customer_id
    AND YEAR (order_date) = 2017
)
```

customer_id	first_name	last_name	city
1174	Aaron	Knapp	Yonkers
338	Abbey	Pugh	Forest Hills
1085	Adam	Thornton	Central Islip
195	Addie	Hahn	Franklin Square
1261	Adelaida	Hancock	San Pablo
22	Adelle	Larsen	East Northport
574	Adriene	Rivera	Encino
1252	Adriene	Rollins	Plainview
527	Afton	Juarez	Coram
1353	Agatha	Daniels	South El Monte
1322	Ai	Forbes	Franklin Square
937	Aida	Koch	West Hempstead

# SQL Subquery with FROM clause



find all employees whose salaries are greater than or equal to the highest salary of every department.

➤ *SQL subquery with the ANY operator*

```
SELECT
AVG(order_count) average_order_count_by_staff
FROM
(
  SELECT
  staff_id, COUNT(order_id) order_count
  FROM sales.orders
  GROUP BY staff_id
) t;
```

average_order_count_by_staff
269

- The query that you place in the FROM clause must have a table alias. In this example, we used the t as the table alias for the subquery. To come up with the final result, SQL Server carries the following steps:
  - Execute the subquery in the FROM clause.
  - Use the result of the subquery and execute the outer query.



# Demo





# Quiz





1

The \_\_\_\_ operator cannot be used with the sub query, but within it.

IN  
INTO  
BETWEEN  
JOIN

BETWEEN

2

Which of the following clause cannot be used in SQL sub queries?

GROUP BY  
ORDER BY  
DELETE  
FROM

ORDER BY



The \_\_\_\_\_ construct returns true if a given tuple is present in the subquery.

- A. not exists
- B. present
- C. not present
- D. exists

exists



4

What is a correlated sub-query?

- A. An independent query that uses the correlation name of another independent query.
- B. A sub-query that uses the correlation name of an outer query
- C. A sub-query that substitutes the names of the outer query
- D. A sub-query that does not depend on its outer query's correlation names

A sub-query that uses the correlation name of an outer query

5

Consider two statements about outer and inner queries in context of SQL sub-queries?

- i. The inner queries can get data from only one table
  - ii. The inner queries can get data from more than one table
- Which of the above statements are true?

- A. Only i
- B. Only ii
- C. Both i and ii
- D. None of the above

Only ii



# References



1. <https://powerbidocs.com/2019/12/25/sql-keys/>
2. <https://www.boardinfinity.com/blog/a-quick-guide-to-entities-in-dbms/>
3. <https://www.sqltutorial.org/>
4. <https://www.w3schools.com/mysql>
5. <https://www.w3resource.com/mysql>



# Thank you

*Innovative Services*



*Passionate Employees*



*Delighted Customers*

