



AERIESCON
S p r i n g 2 0 2 4

950 –

SQL Query Intermediate

Enhancing your SQL Skills

Presented By:

Sarah Walpole |
Integration Engineer



Information shared throughout this presentation is the property of Aeries Software. Information or images may not be reproduced, duplicated, or shared without the prior written consent of Aeries Software.

Other SQL Resources

1

Brent Ozar
(<https://brentozar.com>)



2

SQLZoo
(<https://sqlzoo.net/>)



3

ChatGPT
(<https://chat.openai.com>)



AERIESCON

ICEBREAKER



1. Write a word or phrase on your paper that describes your feelings about SQL Queries. (e.g. “Love”, “Meh”, “Terror”, “I’m a boss”, etc.)
2. Think of the best way to get your paper up here. (e.g. Crumple and Throw it like a basketball, Paper Airplane, Walk it up here, Kick it, Balance on your head, ANYTHING)
3. When I say Go... get those papers up here to me.



Agenda

01 Basic SQL Review

02 Subqueries

03 Joins

04 Variables

05 Rollup

06 Aggregate Functions

07 Case Statements

08 Fun Aeries SQL Tricks



SQL Cheat Sheet

SQL CHEAT SHEET <http://www.sqltutorial.org>



QUERYING DATA FROM A TABLE

SELECT c1, c2 FROM t;

Query data in columns c1, c2 from a table

SELECT * FROM t;

Query all rows and columns from a table

SELECT c1, c2 FROM t

WHERE condition;

Query data and filter rows with a condition

SELECT DISTINCT c1 FROM t

WHERE condition;

Query distinct rows from a table

SELECT c1, c2 FROM t

ORDER BY c1 ASC [DESC];

Sort the result set in ascending or descending order

SELECT c1, c2 FROM t

ORDER BY c1

LIMIT n OFFSET offset;

Skip *offset* of rows and return the next *n* rows

SELECT c1, aggregate(c2)

FROM t

GROUP BY c1;

Group rows using an aggregate function

SELECT c1, aggregate(c2)

FROM t

GROUP BY c1

HAVING condition;

Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

SELECT c1, c2

FROM t1

INNER JOIN t2 ON condition;

Inner join t1 and t2

SELECT c1, c2

FROM t1

LEFT JOIN t2 ON condition;

Left join t1 and t2

SELECT c1, c2

FROM t1

RIGHT JOIN t2 ON condition;

Right join t1 and t2

SELECT c1, c2

FROM t1

FULL OUTER JOIN t2 ON condition;

Perform full outer join

SELECT c1, c2

FROM t1

CROSS JOIN t2;

Produce a Cartesian product of rows in tables

SELECT c1, c2

FROM t1, t2;

Another way to perform cross join

SELECT c1, c2

FROM t1 A

INNER JOIN t2 B ON condition;

Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

SELECT c1, c2 FROM t1

UNION [ALL]

SELECT c1, c2 FROM t2;

Combine rows from two queries

SELECT c1, c2 FROM t1

INTERSECT

SELECT c1, c2 FROM t2;

Return the intersection of two queries

SELECT c1, c2 FROM t1

MINUS

SELECT c1, c2 FROM t2;

Subtract a result set from another result set

SELECT c1, c2 FROM t1

WHERE c1 [NOT] LIKE pattern;

Query rows using pattern matching %, _

SELECT c1, c2 FROM t

WHERE c1 [NOT] IN value_list;

Query rows in a list

SELECT c1, c2 FROM t

WHERE c1 BETWEEN low AND high;

Query rows between two values

SELECT c1, c2 FROM t

WHERE c1 IS [NOT] NULL;

Check if values in a table is NULL or not



Subqueries

```
SELECT column_name1, column_name2, ....  
FROM table1
```

Main Query
(Outer Query)

```
WHERE column_name EXPRESSION OPERATOR (
```

```
    SELECT column_name  
    FROM table2  
    [WHERE]...
```

SubQuery (Inner
Query)

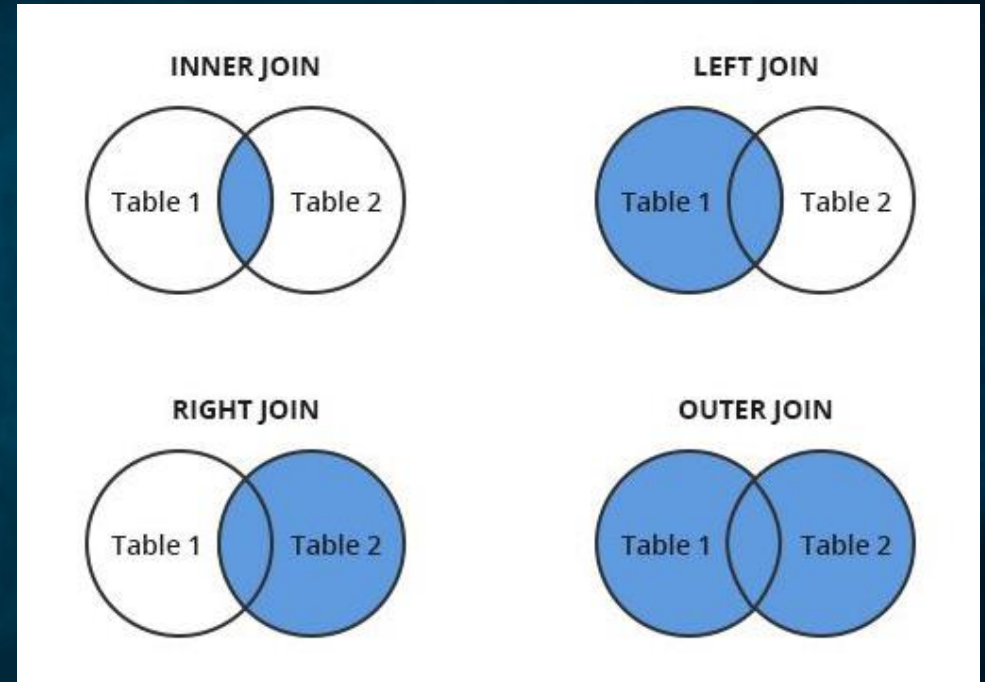
```
)
```

- A subquery is a query within another query. The outer query is called the **main query** and the inner query is called a **subquery**.
- A subquery can have only one column in the SELECT clause.
- If the subquery returns more than one record, it can only be used with IN operator.
- The subquery executes before the main query executes.
- A subquery can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the =, <, >, >=, <=, IN, BETWEEN, etc. operators.
- A Subquery must be enclosed in parentheses.

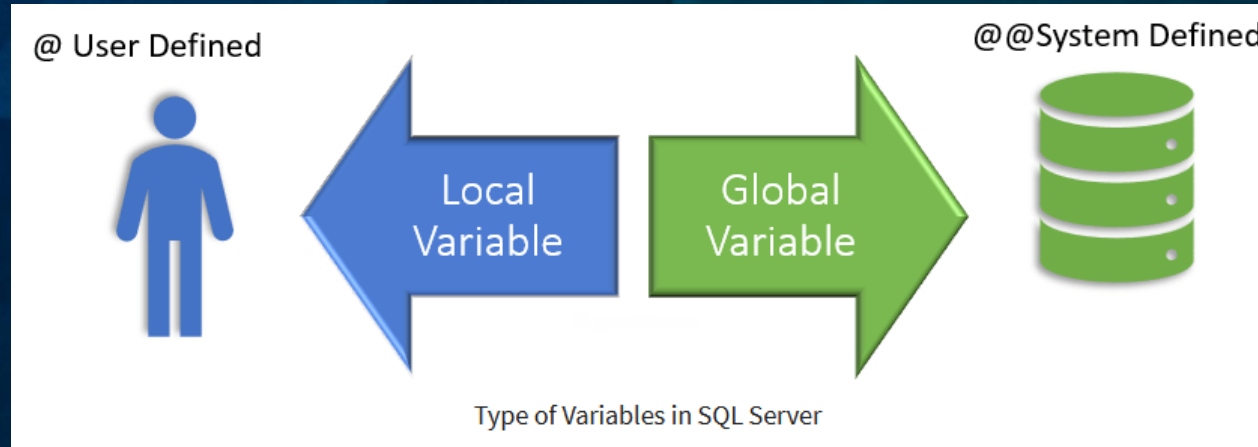


Joins

- **INNER JOIN**
 - Returns rows with matching values in both tables.
 - Discards non-matching rows.
- **LEFT JOIN (or LEFT OUTER JOIN)**
 - Returns all rows from the left table and matching rows from the right table.
 - Non-matching rows from the left table have NULL values from the right table.
- **RIGHT JOIN (or RIGHT OUTER JOIN)**
 - Returns all rows from the right table and matching rows from the left table.
 - Non-matching rows from the right table have NULL values from the left table.
- **FULL JOIN (or FULL OUTER JOIN)**
 - Returns all rows when there is a match in either the left or the right table.
 - Non-matching rows have NULL values in the columns from the non-matching table.



Variables



- There are 2 types of Variables
 - Local
 - A user declares the local variable.
 - By default, a local variable starts with @.
 - Every local variable scope has the restriction to the current batch or procedure within any given session.
 - Global
 - The system maintains the global variable. A user cannot declare them.
 - The global variable starts with @@
 - It stores session related information.
 - <https://www.codeproject.com/Articles/39131/Global-Variables-in-SQL-Server>



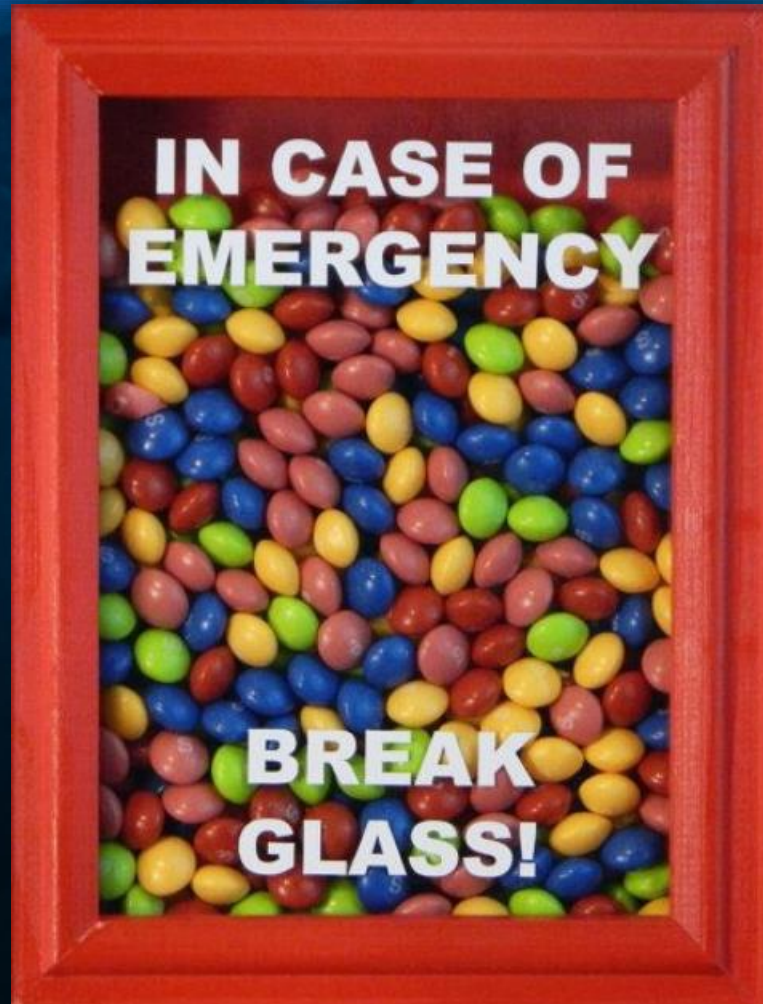
Rollup like a Boss



- ROLLUP is an extension of the GROUP BY clause.
- The ROLLUP option allows you to include extra rows that represent the subtotals.
- The subtotal rows are commonly referred to as super-aggregate rows, along with the grand total row.
- By using the ROLLUP option, you can use a single query to generate multiple grouping sets.
- Rollup is not advisable when querying data for a reporting tool as you can not differentiate the subtotal rows from regular data rows.

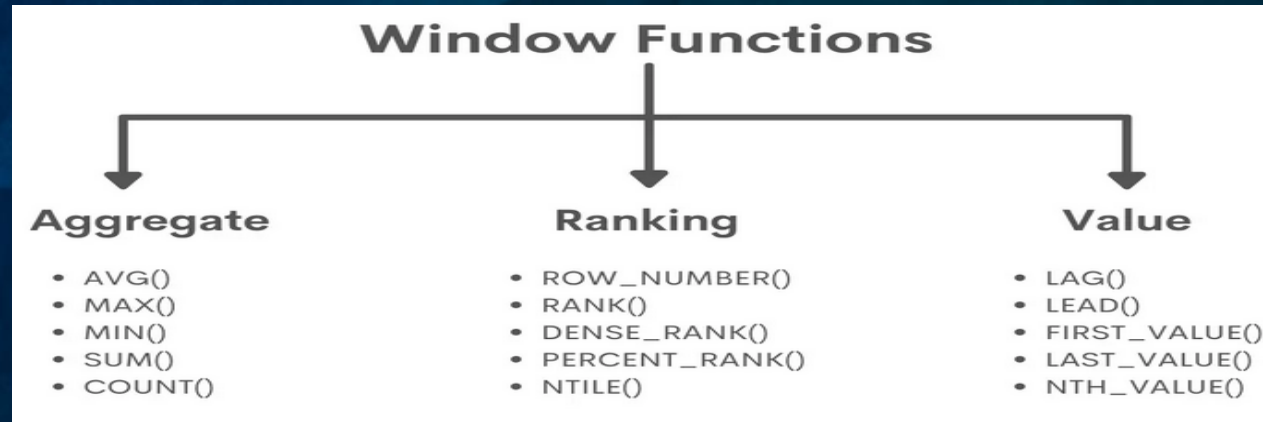


Wait, Wait Candy Break



AERIESCON


Aggregate/Window Functions



- Window functions apply aggregate and ranking functions over a particular window (set of rows).
- The OVER clause is used with window functions to define that window. OVER clause does two things :
 - Partitions rows into a formed set of rows. (PARTITION BY clause is used)
 - Orders rows within those partitions into a particular order. (ORDER BY clause is used)
- Window functions can be called in the SELECT statement or in the ORDER BY clause. However, they can never be used in the WHERE clause.
- Various aggregate functions such as SUM(), COUNT(), AVERAGE(), MAX(), MIN() applied over a particular window (set of rows) are called aggregate window functions.



Case Statements – Just in Case



```
SELECT
    column_name(s)
    CASE condition_field
        WHEN condition_field_value_1 THEN result_1
        WHEN condition_field_value_2 THEN result_2
        ...
    ELSE
    END AS
FROM
    table_name;
```

- The CASE statement always goes in the SELECT clause
- CASE must include the following components: WHEN, THEN, and END.
- ELSE is an optional component.
- You can make any conditional statement using any conditional operator (like WHERE) between WHEN and THEN. This includes stringing together multiple conditional statements using AND and OR.
- You can include multiple WHEN statements, as well as an ELSE statement to deal with any unaddressed conditions.



Fun SQL Tricks for Aeries



- Graduation Year
- View Logged in users
- Schema column information
- Get_Description vs. Get_Description_New –
- <https://support.aeries.com/en/support/solutions/articles/14000073620-using-the-get-description-function-in-sql-server>
- Active Permissions
- Aeries DBA Tools in Freshdesk -
<https://support.aeries.com/en/support/solutions/folders/14000112684>

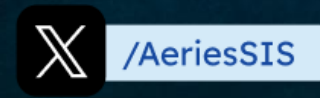
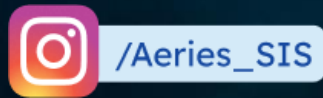
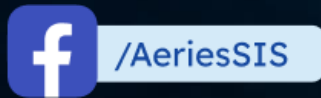


SQL Server Performance Tuning



AERIESCON

THANK YOU!



AERIESCON

Share your Feedback:



Please take a moment to complete our session survey.

<http://aeries.AeriesCon-Session-Feedback-Survey-Spring-2024.alchemer.com/s3/>



AERIESCON