IIT Guwahati

Advanced Certification Programme in Data Science Business Analytics

Week 7

Aggregation Functions

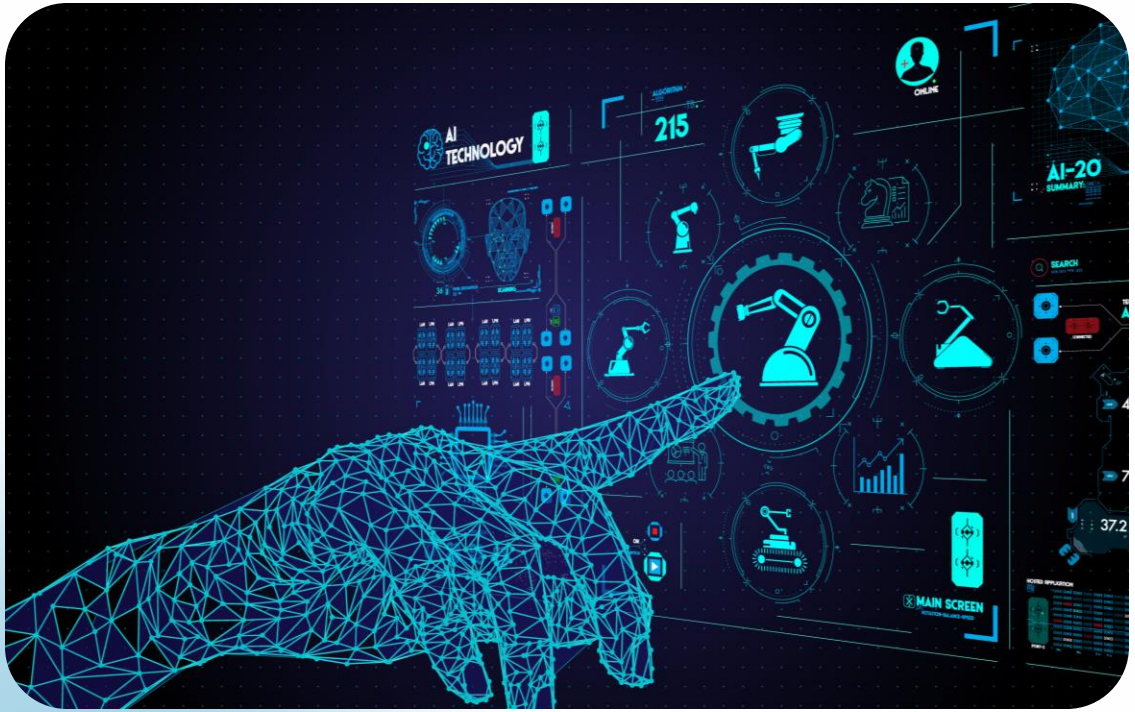
Topics Covered

- Aggregate Functions
- Case Study to Show Aggregate Functions
- Q and A

Aggregation Functions

Understanding Aggregation Functions

Key Tools to Summarise Data and Drive Insights



- **Purpose:**
  Summarise multiple values into a single value for insights

- **Examples:**
  SUM, AVG, COUNT, MAX, MIN, GROUP BY, HAVING

- **Importance:**
  Key in analytics, data mining, machine learning and decision-making

Key Aggregate Functions in SQL

Simplifying Data Analysis With Core Functions

Function	Purpose	Example
SUM	Adds all values in a column	SELECT SUM(sales_amount) AS TotalSales FROM orders;
AVG	Returns average of numeric column	SELECT AVG(sales_amount) AS AverageSales FROM orders;
COUNT	Counts rows or specific values	SELECT COUNT(order_id) AS TotalOrders FROM orders;
MAX	Finds maximum value in a column	SELECT MAX(sales_amount) AS HighestSale FROM orders;
MIN	Finds minimum value in a column	SELECT MIN(sales_amount) AS LowestSale FROM orders;

Understanding the WHERE Clause

Filter Data with Specific Conditions in SQL

- **Purpose**: Filters records based on specified conditions
- **Use**: Retrieves only rows that meet given criteria
- **Benefit**: Makes queries more specific and efficient

Understanding the HAVING Clause

Filter Grouped Data Using Conditions in SQL

SQL **HAVING** clause Example


```sql
SELECT Category, SUM(Price) 'Total Price'
    FROM dbo.Books
    WHERE Category IN ('Computers', 'Science', 'Programming')
    GROUP BY Category
    HAVING SUM(Price) > 50 AND SUM(Price) < 200
```

- Filters records after grouping data
- Applies conditions to groups of rows
- WHERE filters rows before grouping
- HAVING filters groups after grouping

WHERE vs HAVING Clause

Key Differences Between WHERE and HAVING Clauses in SQL

Feature	WHERE clause	HAVING clause
Filters by	Each row	Each group
Processing order	Before any grouping	After any grouping
Aggregate functions	Cannot have aggregate functions	Can have aggregate functions
Usage	Used in SELECT, INSERT, UPDATE, DELETE statements	Only used in SELECT statements
Placement in query	Written before GROUP BY clause	Written after GROUP BY clause
Example	SELECT * FROM table WHERE column1 >= condition;	SELECT * FROM table GROUP BY column2 HAVING MIN(column1) >= condition;

Understanding the GROUP BY Clause

Organise and Summarise Data Using SQL Grouping

Syntax:

```
select column1, aggregate_function
from table_name
where filetr_condition
group by column1
having condition_With_aggreagte_fucntion
order by column1 asc ;
```

- Groups rows based on one or more columns
- Used to organise and summarise data
- Works with aggregate functions like SUM, COUNT, AVG, MAX, MIN
- Helps analyse data by categories or groups

CASE STUDY TO SHOW AGGREGATE FUNCTIONS

Case Study: Aggregate Functions in Action

Applying Data Summarisation Techniques

SaleID	Product	Quantity	Price	SaleDate	Region
1	Laptop	5	800	01-01-2025	North
2	Laptop	3	820	02-01-2025	South
3	Smartphone	10	600	01-01-2025	North
4	Smartphone	12	620	03-01-2025	South
5	Tablet	8	400	02-01-2025	West
6	Laptop	2	830	03-01-2025	East
7	Smartphone	15	590	04-01-2025	West
8	Tablet	5	450	05-01-2025	South
9	Tablet	10	420	01-01-2025	North
10	Laptop	7	810	05-01-2025	West

Understanding Product Revenue

Identify Top Performers and Optimise Resources

Problem statement:

- Identify each product's revenue contribution to total sales
- Optimise resource allocation and marketing efforts

Why it's important?

- Helps prioritise inventory and promotions
- Supports better investment in high-performing products

Solution:

```
SELECT
    Product,
    SUM(Quantity * Price) AS TotalRevenue
FROM
    Sales
GROUP BY
    Product;
```

Regional Sales Analysis

Identify Demand Patterns and Optimise Distribution

Problem statement:

- Identify product distribution across regions
- Assess market penetration and high-demand areas

Why it's important?

- Helps target marketing and optimise logistics
- Supports better resource allocation in high-demand regions

Solution:

```
SELECT
    Region,
    SUM(Quantity) AS TotalQuantity
FROM
    Sales
GROUP BY
    Region;
```

Average Product Price Analysis by Region

Identify Pricing Trends and Ensure Profitability

Problem statement:

- Evaluate average product prices across regions
- Identify pricing variations and regional trends

Why it's important?

- Helps optimise pricing strategies
- Ensures consistent profitability across regions

Solution:

```
SELECT
    Region,
    Product,
    AVG(Price) AS AveragePrice
FROM
    Sales
GROUP BY
    Region, Product;
```

Identifying High-Revenue Regions

Focus on Regions With Sales Over $5,000 For Resource Allocation

Problem statement:

- Identify the most profitable regions with revenue exceeding $5,000
- Focus on areas with consistently strong sales performance

Why it's important?

- Helps prioritise resources and strategic investments
- Supports data-driven regional growth planning

Solution:

```
SELECT
    Region,
    SUM(Quantity * Price) AS TotalRevenue
FROM
    Sales
GROUP BY
    Region
HAVING
    SUM(Quantity * Price) > 5000;
```

Total Products Sold Per Sale Date

Track Daily Sales Trends For Better Decisions

Problem statement:

- Analyse daily sales trends
- Identify peak days & seasonal patterns

Why it's important?

- Helps manage inventory and staffing
- Optimises marketing for high-demand periods

Solution:

```
SELECT
    SaleDate,
    SUM(Quantity) AS TotalQuantity
FROM
    Sales
GROUP BY
    SaleDate;
```

Average Revenue Per Sale by Region

Analyse Sales Efficiency Across Different Regions

Problem statement:

- Calculate average revenue per transaction in each region
- Helps understand regional sales efficiency

Why it's important?

- Assesses sales performance across locations
- Identifies areas for sales strategy improvements

Solution:

```
SELECT
    Region,
    AVG(Quantity * Price) AS
     AverageRevenuePerSale
FROM
    Sales
GROUP BY
    Region;
```

Sales Performance of High-Value Products

Analyse Sales of Products Priced Over $600

Problem statement:

- Measure total quantity sold for high-value products
- Assess their contribution to overall sales

Why it's important?

- High-priced products drive higher profit margins
- Helps refine premium product strategies

Solution:

```
SELECT
    SUM(Quantity) AS TotalQuantity
FROM
    Sales
WHERE
    Price > 600;
```

Track sales where quantity sold exceeds 15

Identifying High-Demand Products & Regions

Problem statement:

- Identify regions and products with consistently high demand
- Helps optimise marketing and stock management

Why it's important?

- Ensures inventory meets demand levels
- Reduces risk of stockouts or overstocking

Solution:

```
SELECT
    Region,
    Product,
    SUM(Quantity) AS TotalQuantity
FROM
    Sales
GROUP BY
    Region, Product
HAVING
    SUM(Quantity) > 15;
```

Monthly Revenue Analysis

Track Total Revenue For Each Product – January 2025

Problem statement:

- Analyse total revenue per product for January 2025
- Supports monthly financial performance review

Why it's important?

- Helps track revenue trends and seasonal variations
- Aids strategic planning for upcoming months

Solution:

```
SELECT
    Product,
    SUM(Quantity * Price) AS
     TotalRevenue
FROM
    Sales
WHERE
    SaleDate BETWEEN '2025-01-01' AND
     '2025-01-31'
GROUP BY
    Product;
```

Q & A

Thank you