```
-- ============================================================
-- BEGINNER'S GUIDE TO WINDOW FUNCTIONS USING CUSTOMERS TABLE
-- ============================================================

USE MyDatabase;

/*
    Assumed Customers table columns:
    - id
    - first_name
    - country
    - score

    WINDOW FUNCTIONS:
    - Perform calculations across rows related to the current row.
    - Unlike GROUP BY, all rows remain in the result.
    - Useful for totals, averages, rankings, row numbers, etc.
*/

-- ============================================================
-- 1. SUM WINDOW FUNCTION
-- ============================================================
/*
    Goal: Calculate the total score for each country while keeping all customer
rows.
    SUM() OVER(PARTITION BY country) calculates the sum per country.
*/

SELECT
    id,
    first_name,
    country,
    score,
    SUM(score) OVER (PARTITION BY country) AS total_score_per_country
FROM Customers
ORDER BY country, id;

-- Explanation:
-- Each row still exists.
-- total_score_per_country shows the sum of scores for all customers in the same
country.

-- ============================================================
-- 2. AVG WINDOW FUNCTION
-- ============================================================
/*
    Goal: Calculate the average score per country.
    AVG() OVER(PARTITION BY country) calculates average without collapsing rows.
*/

SELECT
    id,
    first_name,
    country,
    score,
    AVG(score) OVER (PARTITION BY country) AS avg_score_per_country
FROM Customers
ORDER BY country, id;
```

```
-- ============================================================
-- 3. RANK WINDOW FUNCTION
-- ============================================================
/*
    Goal: Rank customers within their country by score (highest score = rank 1)
    RANK() OVER(PARTITION BY country ORDER BY score DESC) assigns ranks.
*/

SELECT
    id,
    first_name,
    country,
    score,
    RANK() OVER (PARTITION BY country ORDER BY score DESC) AS rank_in_country
FROM Customers
ORDER BY country, rank_in_country;

-- ============================================================
-- 4. DENSE_RANK WINDOW FUNCTION
-- ============================================================
/*
    Goal: Similar to RANK, but no gaps in ranking if scores are tied.
*/

SELECT
    id,
    first_name,
    country,
    score,
    DENSE_RANK() OVER (PARTITION BY country ORDER BY score DESC) AS
dense_rank_in_country
FROM Customers
ORDER BY country, dense_rank_in_country;

-- ============================================================
-- 5. ROW_NUMBER WINDOW FUNCTION
-- ============================================================
/*
    Goal: Assign a unique row number to each customer within a country.
    ROW_NUMBER() OVER(PARTITION BY country ORDER BY score DESC) ensures unique
numbering.
*/

SELECT
    id,
    first_name,
    country,
    score,
    ROW_NUMBER() OVER (PARTITION BY country ORDER BY score DESC) AS
row_num_in_country
FROM Customers
ORDER BY country, row_num_in_country;

-- ============================================================
-- 6. CUMULATIVE SUM WINDOW FUNCTION
-- ============================================================
/*
```

```
    Goal: Calculate running total of scores per country.
    SUM(score) OVER(PARTITION BY country ORDER BY score DESC ROWS BETWEEN UNBOUNDED
PRECEDING AND CURRENT ROW)
*/

SELECT
    id,
    first_name,
    country,
    score,
    SUM(score) OVER(
        PARTITION BY country
        ORDER BY score DESC
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS running_total_score
FROM Customers
ORDER BY country, running_total_score DESC;
```