### ****1. Purpose of the Query****

The goal is to find athletes who have achieved medals in both the Summer and Winter Olympic Games. The query not only identifies them but also counts how many medals they won in each season and overall.

### ****2. Removing Any Old Temporary Table****

Before creating a new temporary table, the script first deletes any table with the same name that might already exist in the session. This ensures that the new data isn’t blocked by an existing structure and that the results will be based only on the latest execution.

### ****3. Creating the Temporary Table****

A temporary table is then created to store the final results. A temporary table exists only during the current database session — it allows the user to work with intermediate results without affecting permanent data. This table will contain one record for each competitor who meets the conditions, along with counts of their medals by season.

### ****4. Gathering and Combining Data from Multiple Tables****

The main query collects information by joining several tables from the Olympic database:

* The **person** table holds the athletes’ personal information, such as their IDs and full names.
* The **competitor\_event** table links athletes to the specific events in which they competed.
* The **event** table contains details about each Olympic event, including which edition of the Games it belonged to.
* The **games** table indicates the year, location, and most importantly, whether the Games were Summer or Winter.
* The **medal** table lists all medals awarded, enabling the query to determine which competitors won and in what circumstances.

By joining these tables on matching IDs, the query builds a dataset where each row represents a single instance of an athlete winning a medal in a specific event.

### ****5. Grouping the Results by Competitor****

After assembling this combined dataset, the query groups all the rows by competitor. Grouping means that all of a person’s medal records are collected together under one row. This allows the query to calculate totals and perform conditional counts for each individual athlete.

### ****6. Counting Medals for Each Season****

Within each competitor’s group, the query counts how many medals belong to the Summer and Winter Games separately. It does this by evaluating the season information for each medal record:

* If the medal came from a Summer Games edition, it contributes to the Summer count.
* If it came from a Winter edition, it contributes to the Winter count.

These two separate counts reveal whether the competitor has medals in both seasons and how many in each.

### ****7. Calculating the Total Medal Count****

In addition to the seasonal counts, the query also counts the total number of medals across all Games. This provides an overall measure of the competitor’s achievements.

### ****8. Filtering for Dual-Season Medalists****

Once all counts are computed, a filter is applied to include only those athletes who have at least one medal in the Summer category **and** at least one medal in the Winter category.  
Competitors who won medals in only one type of Games are excluded at this stage. The filter is crucial because it ensures that the final table contains only the exceptional group of athletes who succeeded in both environments.

### ****9. Storing the Results****

The resulting dataset — containing each qualifying athlete’s ID, name, number of Summer medals, number of Winter medals, and total medals — is saved into the temporary table. This makes it easy to use or analyze further without re-running the whole query.

### ****10. Displaying the Results****

Finally, the contents of the temporary table are displayed. The output typically lists each dual-season medalist, showing their full name and the counts of medals they earned in Summer and Winter Olympics, along with the combined total.  
For better readability, the results are ordered by total medal count, so the most decorated athletes appear first.