THE TECHNOLOGICAL UNIVERSITY OF THE SHANNON: MIDLANDS MIDWEST

Course: Relational Database

Student Name: Parimal Sawant

Student ID: A00325752

Date: 11 December 2024

Analytic SQL Project

Introduction

In today's education environment, scholarship supports the aspiring students to help pursue their higher education. Scholarship provides financial assistant to students, allowing them to pursue their studies at higher colleges without the burden of financial part. The scholarships are majorly funded by businesses or organisations, which serves as sponsors to the scholarship. Sponsors are essential to promote education and to find out talent.

Applicants, typically apply for this scholarship during their secondary school to pursue higher education. The selection process may take some measures that are recommended by the sponsor. Most of the scholarships are awarded to deserving and talented students, there are instances where scholarships are unawarded as well. By implementation of database the stake holders will be able to track the scholarships efficiently and manage scholarship awards, to ensure the process remains streamlined.

Entities and Relationship

The database consists of 2 key entities: Sponsor and Scholarship which are structured in a parent child relationship. The sponsor serves as a parent table here while the scholarship serves as a child table. The 2 parent and child tables are merged by a foreign key Sponsor_Id which is primary key in the Sponsor table. This link between the two tables ensures that each scholarship is associated with one sponsor.

Sponsor: **Sponsor_Id (PK)**, Name, Town, County, Email, Business_Type, Num_Emp, Turnover, Costs, Main_Market

Scholarship: **Sponsor_Id (FK)**, **Scholarship_Id (PK)**, Title, Discipline, Annual_Value, Duration, Initial_Payment_Date, Num_Installments, Category, Condition, University

ER Diagram

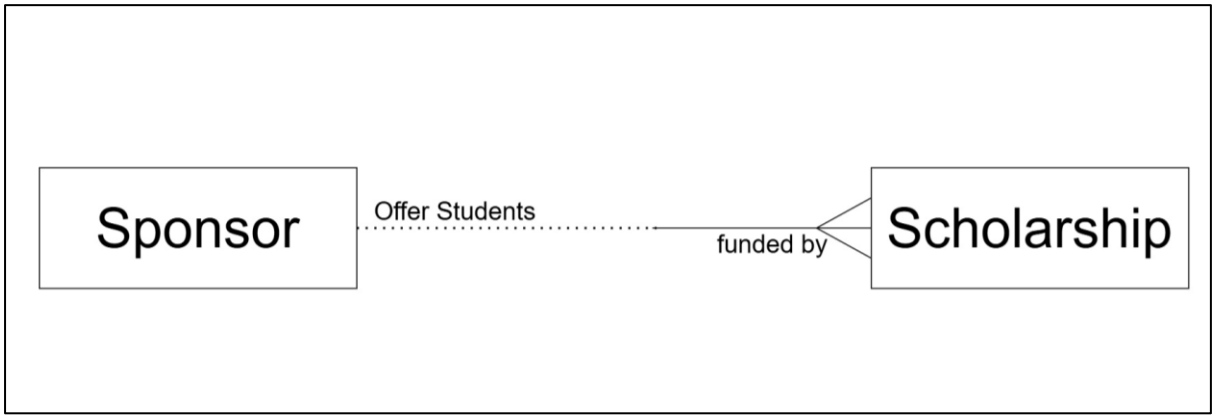


Fig 1.1 ER Diagram without attribute

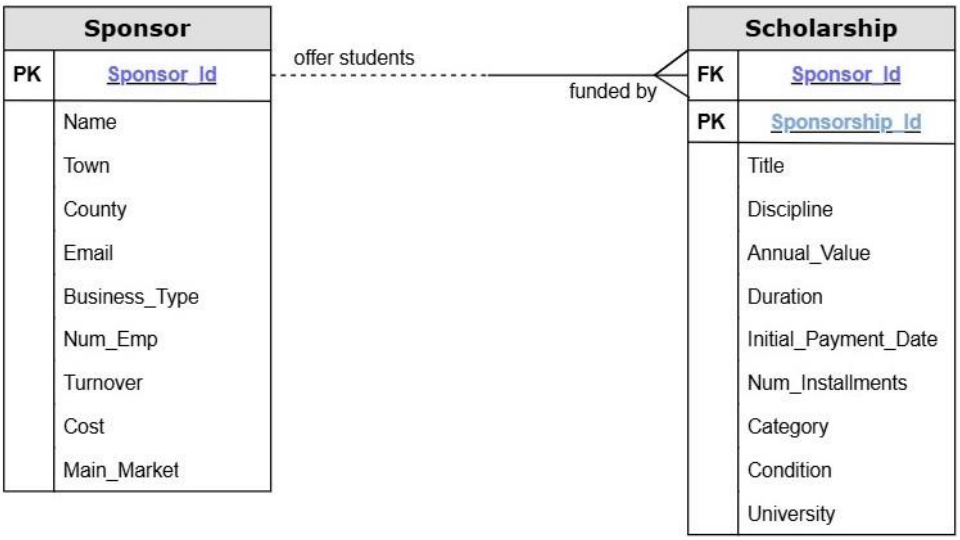


Fig 1.2 ER Diagram with attribute

Create and Insert Query

Create Sponsor Table

```
Create Table Sponsor(
Sponsor_Id          Number(2),
Name                Varchar2(30) Constraint Sponsor_Name_Nn Not Null,
Town                Varchar2(15),
County              Varchar2(11),
Email               Varchar2(35) Constraint Sponsor_Email_Uq Unique,
Business_Type       Varchar2(13),
Num_Emp             Number(5),
Turnover            Number(10),
Costs               Number(9),
Main_Market         Varchar2(7),
Constraint Sponsor_Sponsor_Id_Pk Primary Key (Sponsor_Id),
Constraint Sponsor_Sponsor_Id_Ck Check (Sponsor_Id > 0),
Constraint Sponsor_Town_Initcap_Ck Check (Town = Initcap(Town)),
Constraint Sponsor_County_Initcap_Ck Check (County = Initcap(County)),
Constraint Sponsor_Business_Initcap_Ck Check (Business_Type = Initcap(Business_Type)),
Constraint Sponsor_Business_Type_Ck Check (Business_Type In ('Retail', 'Manufacturing',
'Service', 'Leisure')),
Constraint Sponsor_Email_Ck Check (Email Is Null Or Email LIKE '_%@_%._%'));
```

Insert Data Sample for Sponsor Table

```
Insert Into Sponsor Values(1, 'Eircom', 'Courtmacsherry', 'Dublin 2',
'1stinitialsurname@eircom.ie', 'Service', 8000, 168000000, 23520000, 'Ireland');
```

Create Scholarship Table

```
Create Table Scholarship(
Sponsor_Id              Number(2),
Scholarship_Id          Number(3),
Title                   Varchar2(50) Constraint Scholarship_Scholarship_Id_Nn
                        Not Null,
Discipline              Varchar2(11),
Annual_Value            Number(5),
Duration                Number(1),
Initial_Payment_Date    Date,
Num_Instalments         Number(2),
Category                Varchar2(13),
Condition               Varchar2(32),
University              Varchar2(35),
Constraint Scholarship_Sponsor_Id_Fk Foreign Key (Sponsor_Id) References Sponsor
(Sponsor_Id),
Constraint Scholarship_Scholarship_Id_Pk Primary Key (Scholarship_Id),
Constraint Scholarship_Scholarship_Id_Ck Check (Scholarship_Id > 0),
```

```
Constraint Scholarship_Discipline_Ck Check (Discipline In ('Medical', 'Engineering',
'Psychology', 'Science')),
Constraint Scholarship_Number_Ck Check (Duration > 0 And Annual_Value > 0 And
Num_Instalments > 0),
Constraint Scholarship_Condition_Ck Check (Condition In ('Must get average above 70%',
'Must pass exams', 'Reviewed at the end of each year')));
```

Insert Data Sample Scholarship Table

```
Insert Into Scholarship Values(21, 1, 'Diageo Ireland Student Scholarship', 'Medical', 16500,
3, '19-Apr-2024', 41, 'Undergraduate', 'Must get average above 70%', 'University of
Westminster');
```

SQL Analytic Query

Over Clause

Example 1. List the Name, Business Type and Cost from Sponsor for each individual employee plus the overall total cost for all the Sponsor, only show results where Business Type is Retail and Service. Sort the results by Business Type and Name.

```
Cl scr
Select Name, Business_Type, Costs,
        Sum(Costs) Over() Total_Costs
From Sponsor
Where Business_Type = 'Retail'
Order By Business_Type, Name;
```


```
A00325752PS_SQL>Select Name, Business_Type, Costs,
  2     Sum(Costs) Over() Total_Costs
  3  From Sponsor
  4  Where Business_Type = 'Retail'
  5  Order By Business_Type, Name;

NAME                           BUSINESS_TYPE      COSTS TOTAL_COSTS
------------------------------ ------------- ---------- -----------
Capital Bars                   Retail            672000   210574420
Dunnes Stores                  Retail          38150000   210574420
Grafton Group                  Retail          20944000   210574420
Greencore                      Retail          20626340   210574420
IAWS Group                     Retail          17511480   210574420
Irish Food Processors          Retail          12236000   210574420
Irish Shell                    Retail          12600000   210574420
Musgrave                       Retail          38857000   210574420
Paddy Power                    Retail          12782000   210574420
Primark                        Retail          15946000   210574420
United Drug                    Retail          20249600   210574420

11 rows selected.

A00325752PS_SQL>
```

This SQL query consist of Sum aggregate feature, but it also features OVER clause. Therefore, we know that it is an Analytic function instead of an aggregate function.

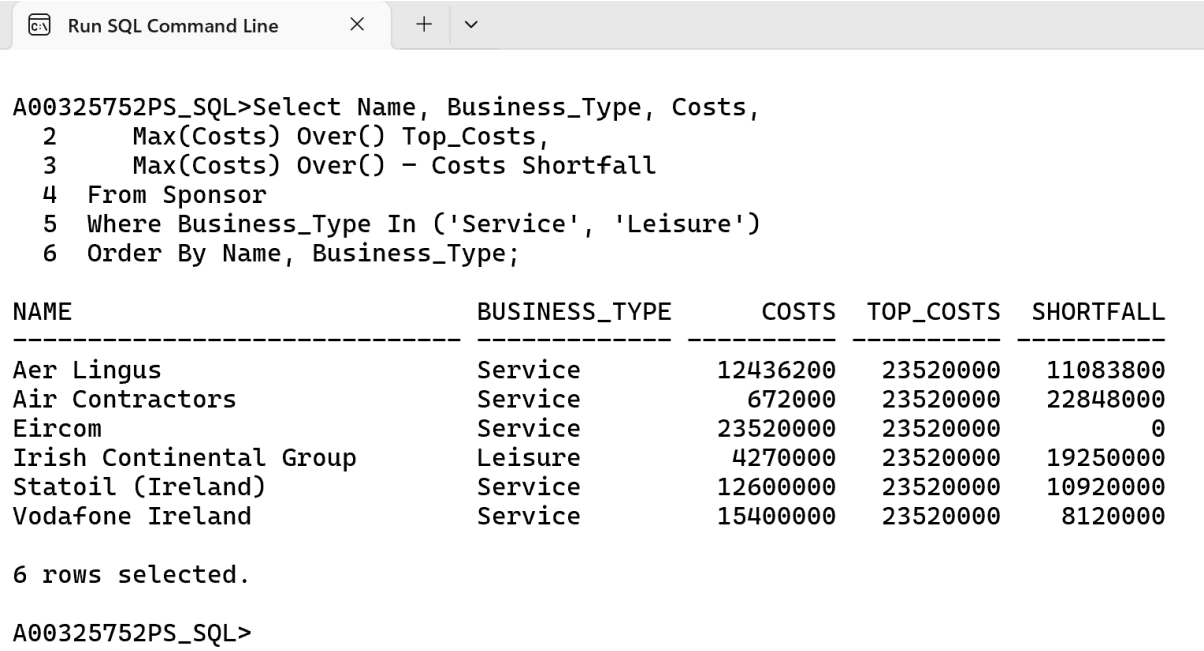
The OVER clause is the basic one as it will add up the total costs and display in the last column which is not feasible with aggregate functions.

Over Clause

Example 2. List the Name, Business_Type, Costs for each sponsor plus the largest costs among the Sponsors. In addition, calculate and display the amount each individual cost falls short of the largest costs. Filter out results where Business Type in Service or Leisure. Sort the results by Name and then by Business Type.

Cl scr

```
Select Name, Business_Type, Costs,
        Max(Costs) Over() Top_Costs,
        Max(Costs) Over() – Costs Shortfall
From Sponsor
Where Business_Type In ('Service', 'Leisure')
Order By Name, Business_Type;
```



In this query we are using Over clause we have used in previous example but on the Shortfall column we are subtracting the actual cost with maximum cost.

Over Clause

Example 3. List the Name, Business_Type, Costs for each sponsor plus the largest and smallest costs among the Sponsors. In addition, calculate and display the amount each individual cost falls short of the largest costs, and the amount each sponsor exceeds the smallest costs. Filter out results where Business Type in Service or Leisure. Sort the results by Name and then by Business Type.

Cl scr

```
Select Name, Business_Type, Costs,
        Max(Costs) Over() Top_Costs,
        Max(Costs) Over() – Costs Shortfall,
        Min(Costs) Over() Worst_Sal,
        Costs - Min(Costs) Over() Excess
From Sponsor
Where Business_Type In ('Service', 'Leisure')
Order By Name, Business_Type;
```

Run SQL Command Line

```
A00325752PS_SQL>Select Name, Business_Type, Costs,
  2     Max(Costs) Over() Top_Costs,
  3     Max(Costs) Over() – Costs Shortfall,
  4     Min(Costs) Over() Worst_Sal,
  5     Costs - Min(Costs) Over() Excess
  6  From Sponsor
  7  Where Business_Type In ('Service', 'Leisure')
  8  Order By Name, Business_Type;

NAME                           BUSINESS_TYPE      COSTS  TOP_COSTS  SHORTFALL  WORST_SAL     EXCESS
------------------------------ ------------- ---------- ---------- ---------- ---------- ----------
Aer Lingus                     Service         12436200   23520000   11083800     672000   11764200
Air Contractors                Service           672000   23520000   22848000     672000          0
Eircom                         Service         23520000   23520000          0     672000   22848000
Irish Continental Group        Leisure          4270000   23520000   19250000     672000    3598000
Statoil (Ireland)              Service         12600000   23520000   10920000     672000   11928000
Vodafone Ireland               Service         15400000   23520000    8120000     672000   14728000

6 rows selected.

A00325752PS_SQL>
```

This query is similar to the previous one but we are using Over clause here to show Minimum cost and the excess by subtracting cost with Minimum cost.

Over Clause with all aggregate functions

Example 4. List the Name, Business_Type and Costs for each individual Sponsor plus the lowest, highest, average and overall total costs for every Business Type except Manufacturing. Sort the results by Name and then by Business Type.

Cl scr

```
Select Name, Business_Type, Costs,
        Max(Costs) Over() Top_Costs,
        Min(Costs) Over() Bottom_Costs,
        Avg(Costs) Over() Average_Sal,
        Sum(Costs) Over() Total_Sal
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Name, Business_Type;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Name, Business_Type, Costs,
  2     Max(Costs) Over() Top_Costs,
  3     Min(Costs) Over() Bottom_Costs,
  4     Avg(Costs) Over() Average_Sal,
  5     Sum(Costs) Over() Total_Sal
  6  From Sponsor
  7  Where Business_Type <> 'Manufacturing'
  8  Order By Name, Business_Type;

NAME                           BUSINESS_TYPE      COSTS  TOP_COSTS BOTTOM_COSTS AVERAGE_SAL  TOTAL_SAL
------------------------------ ------------- ---------- ---------- ------------ ----------- ----------
Aer Lingus                     Service         12436200   38857000       672000  16439565.9  279472620
Air Contractors                Service           672000   38857000       672000  16439565.9  279472620
Capital Bars                   Retail            672000   38857000       672000  16439565.9  279472620
Dunnes Stores                  Retail          38150000   38857000       672000  16439565.9  279472620
Eircom                         Service         23520000   38857000       672000  16439565.9  279472620
Grafton Group                  Retail          20944000   38857000       672000  16439565.9  279472620
Greencore                      Retail          20626340   38857000       672000  16439565.9  279472620
IAWS Group                     Retail          17511480   38857000       672000  16439565.9  279472620
Irish Continental Group        Leisure          4270000   38857000       672000  16439565.9  279472620
Irish Food Processors          Retail          12236000   38857000       672000  16439565.9  279472620
Irish Shell                    Retail          12600000   38857000       672000  16439565.9  279472620
Musgrave                       Retail          38857000   38857000       672000  16439565.9  279472620
Paddy Power                    Retail          12782000   38857000       672000  16439565.9  279472620
Primark                        Retail          15946000   38857000       672000  16439565.9  279472620
Statoil (Ireland)              Service         12600000   38857000       672000  16439565.9  279472620
United Drug                    Retail          20249600   38857000       672000  16439565.9  279472620
Vodafone Ireland               Service         15400000   38857000       672000  16439565.9  279472620

17 rows selected.

A00325752PS_SQL>
```

For this query we are using all the aggregate functions (Sum, Max, Min, Avg) along with the over clause.

Over Clause with all aggregate functions

Example 5. List the Name, Business_Type and Costs for each individual Sponsor plus the lowest, highest, average and overall total costs along with the running total for every Business Type except Manufacturing. Sort the results by Business Type (ascending order) and then by Name.

Cl scr

```
Select Name, Business_Type, Costs,
        Max(Costs) Over(Order By Business_Type) Top_Costs,
        Min(Costs) Over(Order By Business_Type) Bottom_Costs,
        Avg(Costs) Over(Order By Business_Type) Average_Sal,
        Sum(Costs) Over(Order By Business_Type) Total_Sal
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Business_Type, Name;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Name, Business_Type, Costs,
  2     Max(Costs) Over(Order By Business_Type) Top_Costs,
  3     Min(Costs) Over(Order By Business_Type) Bottom_Costs,
  4     Avg(Costs) Over(Order By Business_Type) Average_Sal,
  5     Sum(Costs) Over(Order By Business_Type) Total_Sal
  6  From Sponsor
  7  Where Business_Type <> 'Manufacturing'
  8  Order By Business_Type, Name;

NAME                           BUSINESS_TYPE      COSTS  TOP_COSTS BOTTOM_COSTS AVERAGE_SAL  TOTAL_SAL
------------------------------ ------------- ---------- ---------- ------------ ----------- ----------
Irish Continental Group        Leisure          4270000    4270000      4270000     4270000    4270000
Capital Bars                   Retail            672000   38857000       672000  17903701.7  214844420
Dunnes Stores                  Retail          38150000   38857000       672000  17903701.7  214844420
Grafton Group                  Retail          20944000   38857000       672000  17903701.7  214844420
Greencore                      Retail          20626340   38857000       672000  17903701.7  214844420
IAWS Group                     Retail          17511480   38857000       672000  17903701.7  214844420
Irish Food Processors          Retail          12236000   38857000       672000  17903701.7  214844420
Irish Shell                    Retail          12600000   38857000       672000  17903701.7  214844420
Musgrave                       Retail          38857000   38857000       672000  17903701.7  214844420
Paddy Power                    Retail          12782000   38857000       672000  17903701.7  214844420
Primark                        Retail          15946000   38857000       672000  17903701.7  214844420
United Drug                    Retail          20249600   38857000       672000  17903701.7  214844420
Aer Lingus                     Service         12436200   38857000       672000  16439565.9  279472620
Air Contractors                Service           672000   38857000       672000  16439565.9  279472620
Eircom                         Service         23520000   38857000       672000  16439565.9  279472620
Statoil (Ireland)              Service         12600000   38857000       672000  16439565.9  279472620
Vodafone Ireland               Service         15400000   38857000       672000  16439565.9  279472620

17 rows selected.

A00325752PS_SQL>
```

In this query, the ORDER BY inside the OVER() clause organizes rows by Business_Type for each window function, ensuring the aggregare calculations like MAX, MIN, AVG, and SUM are computed sequentially for all rows ordered by Business_Type. This creates a running total and other aggregate values that adjust dynamically based on the sorted order.

Over Clause with Order By

Example 6. List the Name, Business_Type and Costs for each individual Sponsor plus the lowest, highest, average and overall total costs for every Business Type except Manufacturing. Sort the results by Business Type **(descending order)** and then by Name.

Cl scr

```
Select Name, Business_Type, Costs,
        Max(Costs) Over(Order By Business_Type Desc) Top_Costs,
        Min(Costs) Over(Order By Business_Type Desc) Bottom_Costs,
        Avg(Costs) Over(Order By Business_Type Desc) Average_Sal,
        Sum(Costs) Over(Order By Business_Type Desc) Total_Sal
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Business_Type Desc, Name;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Name, Business_Type, Costs,
  2     Max(Costs) Over(Order By Business_Type Desc) Top_Costs,
  3     Min(Costs) Over(Order By Business_Type Desc) Bottom_Costs,
  4     Avg(Costs) Over(Order By Business_Type Desc) Average_Sal,
  5     Sum(Costs) Over(Order By Business_Type Desc) Total_Sal
  6  From Sponsor
  7  Where Business_Type <> 'Manufacturing'
  8  Order By Business_Type Desc, Name;

NAME                          BUSINESS_TYPE      COSTS  TOP_COSTS BOTTOM_COSTS AVERAGE_SAL  TOTAL_SAL
----------------------------- ------------- ---------- ---------- ------------ ----------- ----------
Aer Lingus                    Service         12436200   23520000       672000    12925640   64628200
Air Contractors               Service           672000   23520000       672000    12925640   64628200
Eircom                        Service         23520000   23520000       672000    12925640   64628200
Statoil (Ireland)             Service         12600000   23520000       672000    12925640   64628200
Vodafone Ireland              Service         15400000   23520000       672000    12925640   64628200
Capital Bars                  Retail            672000   38857000       672000  17200163.8  275202620
Dunnes Stores                 Retail          38150000   38857000       672000  17200163.8  275202620
Grafton Group                 Retail          20944000   38857000       672000  17200163.8  275202620
Greencore                     Retail          20626340   38857000       672000  17200163.8  275202620
IAWS Group                    Retail          17511480   38857000       672000  17200163.8  275202620
Irish Food Processors         Retail          12236000   38857000       672000  17200163.8  275202620
Irish Shell                   Retail          12600000   38857000       672000  17200163.8  275202620
Musgrave                      Retail          38857000   38857000       672000  17200163.8  275202620
Paddy Power                   Retail          12782000   38857000       672000  17200163.8  275202620
Primark                       Retail          15946000   38857000       672000  17200163.8  275202620
United Drug                   Retail          20249600   38857000       672000  17200163.8  275202620
Irish Continental Group       Leisure          4270000   38857000       672000  16439565.9  279472620

17 rows selected.

A00325752PS_SQL>
```

This query is similar to query from **Ex 5.** But since we are using Desc(Descending) the running totals are adjusted dynamically based on the descending order.

Over Clause with Order By and all aggregate functions

Example 7. List the Name, County, Business_Type and Costs for each individual Sponsor plus the lowest, highest, average and overall total costs for every Business Type except Manufacturing. Order By County.

```
Cl scr
Select Name, County, Business_Type, Costs,
        Max(Costs) Over(Order By County) Top_Costs,
        Min(Costs) Over(Order By County) Bottom_Costs,
        Avg(Costs) Over(Order By County) Average_Sal,
        Sum(Costs) Over(Order By County) Total_Sal
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By County;
```

Run SQL Command Line

```
A00325752PS_SQL>Select Name, County, Business_Type, Costs,
  2     Max(Costs) Over(Order By County) Top_Costs,
  3     Min(Costs) Over(Order By County) Bottom_Costs,
  4     Avg(Costs) Over(Order By County) Average_Sal,
  5     Sum(Costs) Over(Order By County) Total_Sal
  6  From Sponsor
  7  Where Business_Type <> 'Manufacturing'
  8  Order By County;

NAME                           COUNTY      BUSINESS_TYPE      COSTS  TOP_COSTS BOTTOM_COSTS AVERAGE_SAL  TOTAL_SAL
------------------------------ ----------- ------------- ---------- ---------- ------------ ----------- ----------
Musgrave                       Cork        Retail          38857000   38857000     38857000    38857000   38857000
Air Contractors                Dublin      Service           672000   38857000       672000  17321733.3   51965200
Aer Lingus                     Dublin      Service         12436200   38857000       672000  17321733.3   51965200
Primark                        Dublin 1    Retail          15946000   38857000       672000    14130200   84781200
Statoil (Ireland)              Dublin 1    Service         12600000   38857000       672000    14130200   84781200
Irish Continental Group        Dublin 1    Leisure          4270000   38857000       672000    14130200   84781200
Vodafone Ireland               Dublin 18   Service         15400000   38857000       672000    15140650  121125200
Grafton Group                  Dublin 18   Retail          20944000   38857000       672000    15140650  121125200
Dunnes Stores                  Dublin 2    Retail          38150000   38857000       672000    17007795  204093540
Capital Bars                   Dublin 2    Retail            672000   38857000       672000    17007795  204093540
Greencore                      Dublin 2    Retail          20626340   38857000       672000    17007795  204093540
Eircom                         Dublin 2    Service         23520000   38857000       672000    17007795  204093540
United Drug                    Dublin 24   Retail          20249600   38857000       672000    16937510  237125140
Paddy Power                    Dublin 24   Retail          12782000   38857000       672000    16937510  237125140
Irish Shell                    Dublin 4    Retail          12600000   38857000       672000  16648342.7  249725140
IAWS Group                     Dublin 8    Retail          17511480   38857000       672000  16702288.8  267236620
Irish Food Processors          Louth       Retail          12236000   38857000       672000  16439565.9  279472620

17 rows selected.

A00325752PS_SQL>
```

In this query is the same as **Ex 6.** we are ordering it by county in order to get a good example of running totals.

Over Clause with Order By with wrong Order By sequence

Example 8. List the Name, County Business_Type and Costs for each individual Sponsor plus the lowest, highest, average and overall total costs for every Business Type except Manufacturing. Order By **Name and Business_Type**.

```
Cl scr
Select Name, County, Business_Type, Costs,
        Max(Costs) Over(Order By County) Top_Costs,
        Min(Costs) Over(Order By County) Bottom_Costs,
        Avg(Costs) Over(Order By County) Average_Sal,
        Sum(Costs) Over(Order By County) Total_Sal
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Name, Business_Type;
```

Run SQL Command Line

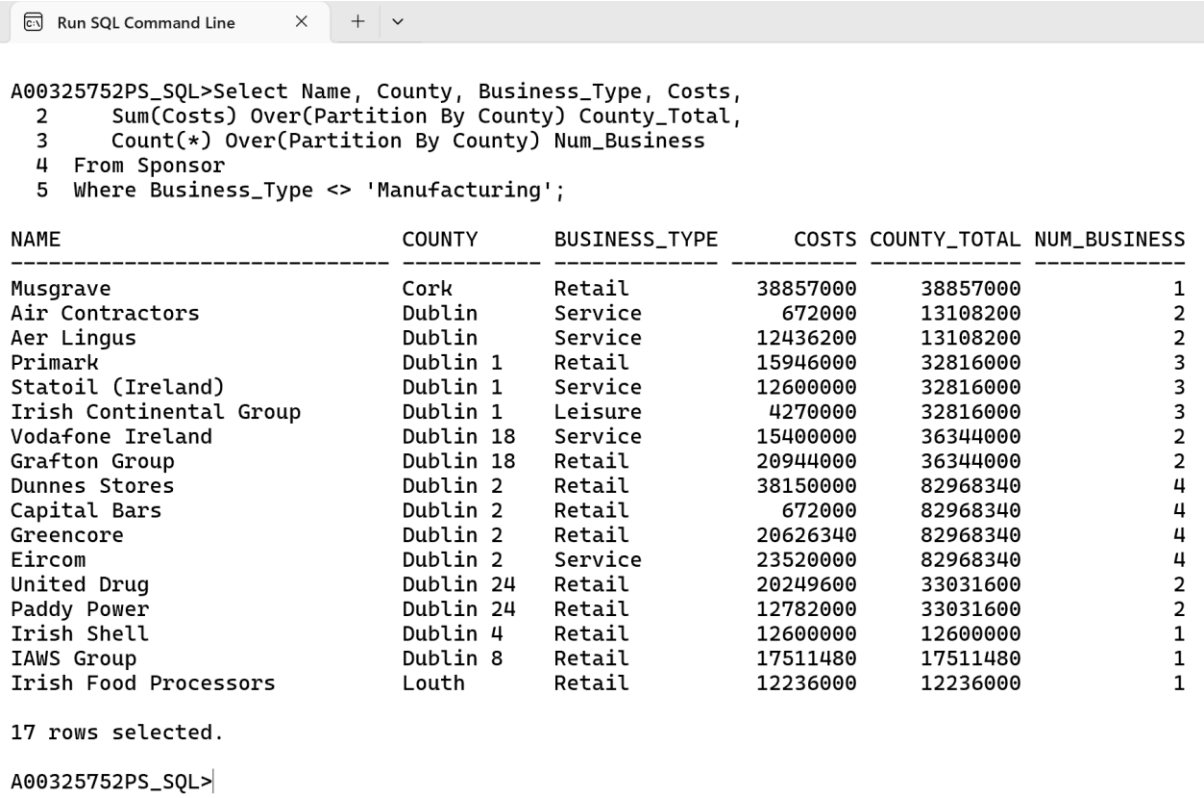
```
A00325752PS_SQL>Select Name, County, Business_Type, Costs,
  2     Max(Costs) Over(Order By County) Top_Costs,
  3     Min(Costs) Over(Order By County) Bottom_Costs,
  4     Avg(Costs) Over(Order By County) Average_Sal,
  5     Sum(Costs) Over(Order By County) Total_Sal
  6  From Sponsor
  7  Where Business_Type <> 'Manufacturing'
  8  Order By Name, Business_Type;

NAME                           COUNTY      BUSINESS_TYPE      COSTS  TOP_COSTS BOTTOM_COSTS AVERAGE_SAL  TOTAL_SAL
------------------------------ ----------- ------------- ---------- ---------- ------------ ----------- ----------
Aer Lingus                     Dublin      Service         12436200   38857000       672000  17321733.3   51965200
Air Contractors                Dublin      Service           672000   38857000       672000  17321733.3   51965200
Capital Bars                   Dublin 2    Retail            672000   38857000       672000    17007795  204093540
Dunnes Stores                  Dublin 2    Retail          38150000   38857000       672000    17007795  204093540
Eircom                         Dublin 2    Service         23520000   38857000       672000    17007795  204093540
Grafton Group                  Dublin 18   Retail          20944000   38857000       672000    15140650  121125200
Greencore                      Dublin 2    Retail          20626340   38857000       672000    17007795  204093540
IAWS Group                     Dublin 8    Retail          17511480   38857000       672000  16702288.8  267236620
Irish Continental Group        Dublin 1    Leisure          4270000   38857000       672000    14130200   84781200
Irish Food Processors          Louth       Retail          12236000   38857000       672000  16439565.9  279472620
Irish Shell                    Dublin 4    Retail          12600000   38857000       672000  16648342.7  249725140
Musgrave                       Cork        Retail          38857000   38857000     38857000    38857000   38857000
Paddy Power                    Dublin 24   Retail          12782000   38857000       672000    16937510  237125140
Primark                        Dublin 1    Retail          15946000   38857000       672000    14130200   84781200
Statoil (Ireland)              Dublin 1    Service         12600000   38857000       672000    14130200   84781200
United Drug                    Dublin 24   Retail          20249600   38857000       672000    16937510  237125140
Vodafone Ireland               Dublin 18   Service         15400000   38857000       672000    15140650  121125200

17 rows selected.

A00325752PS_SQL>
```

In the Over clause we are ordering it by County but in Order By Clause at the end of query we are ordering it by Name, Business Name. This generated query will require a lot of scrutiny before you will be make any sense out of them.

Over Clause with Partition

Example 9. List the Name, County Business Type and Costs for each individual Sponsor, as well as and a running costs total for business in county, plus the companies in each county by business type for every Business Type except Manufacturing.

```
Cl scr
Select Name, County, Business_Type, Costs,
        Sum(Costs) Over(Partition By County) County_Total,
        Count(*) Over(Partition By County) Num_Business
From Sponsor
Where Business_Type <> 'Manufacturing';
```



```
Run SQL Command Line

A00325752PS_SQL>Select Name, County, Business_Type, Costs,
  2     Sum(Costs) Over(Partition By County) County_Total,
  3     Count(*) Over(Partition By County) Num_Business
  4  From Sponsor
  5  Where Business_Type <> 'Manufacturing';

NAME                           COUNTY     BUSINESS_TYPE      COSTS COUNTY_TOTAL NUM_BUSINESS
------------------------------ ---------- ------------- ---------- ------------ ------------
Musgrave                       Cork       Retail          38857000     38857000            1
Air Contractors                Dublin     Service           672000     13108200            2
Aer Lingus                     Dublin     Service         12436200     13108200            2
Primark                        Dublin 1   Retail          15946000     32816000            3
Statoil (Ireland)              Dublin 1   Service         12600000     32816000            3
Irish Continental Group        Dublin 1   Leisure          4270000     32816000            3
Vodafone Ireland               Dublin 18  Service         15400000     36344000            2
Grafton Group                  Dublin 18  Retail          20944000     36344000            2
Dunnes Stores                  Dublin 2   Retail          38150000     82968340            4
Capital Bars                   Dublin 2   Retail            672000     82968340            4
Greencore                      Dublin 2   Retail          20626340     82968340            4
Eircom                         Dublin 2   Service         23520000     82968340            4
United Drug                    Dublin 24  Retail          20249600     33031600            2
Paddy Power                    Dublin 24  Retail          12782000     33031600            2
Irish Shell                    Dublin 4   Retail          12600000     12600000            1
IAWS Group                     Dublin 8   Retail          17511480     17511480            1
Irish Food Processors          Louth      Retail          12236000     12236000            1

17 rows selected.

A00325752PS_SQL>
```

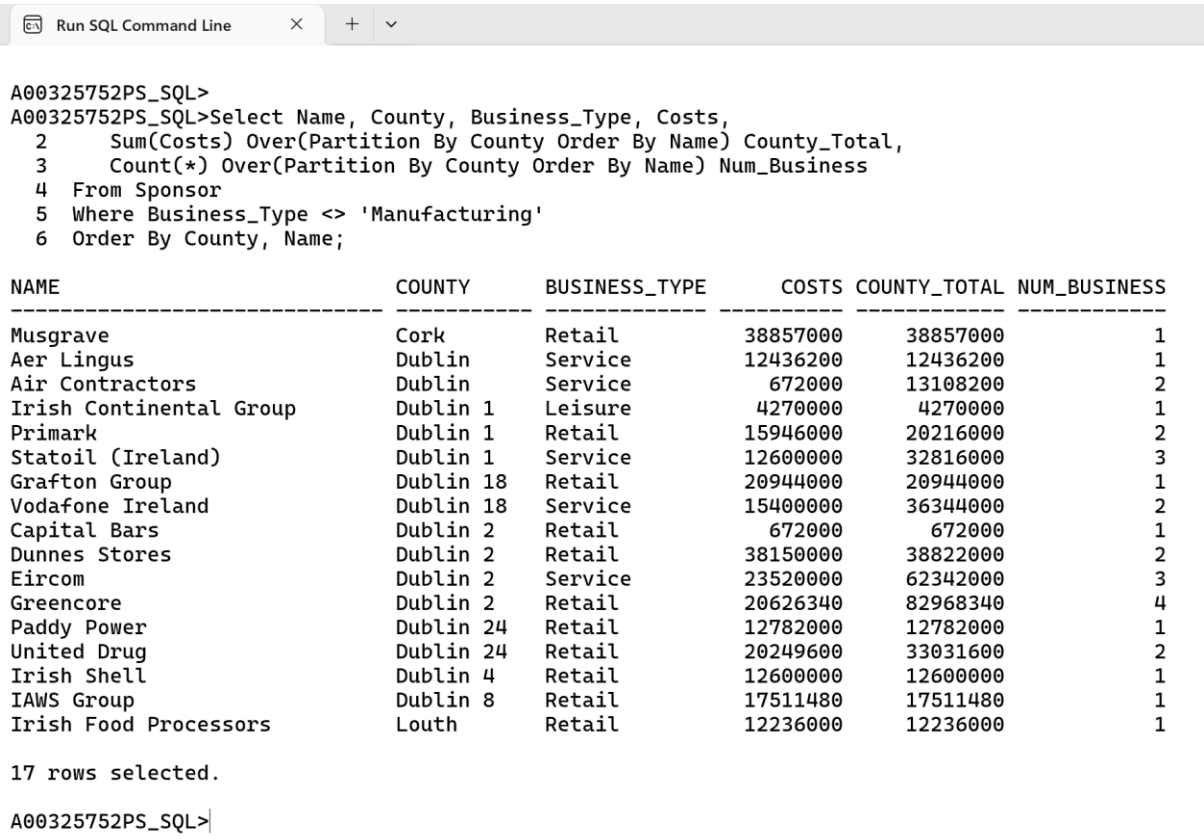
The PARTITION BY clause in the OVER() function here is similar to the Group By function in normal SQL query, this Partition By Clause groups rows by County. This allows aggregate functions like SUM and COUNT to compute totals and counts within each County while retaining row-level details.

Over Clause with Partition and Order By

Example 10. List the Name, County Business Type and Costs for each individual Sponsor, as well as and a running costs total for business in county, plus the companies in each county by business type for every Business Type except Manufacturing. Sort the results by County and then by Name

Cl scr

```
Select Name, County, Business_Type, Costs,
        Sum(Costs) Over(Partition By County Order By Name) County_Total,
        Count(*) Over(Partition By County Order By Name) Num_Business
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By County, Name;
```



```
A00325752PS_SQL>
A00325752PS_SQL>Select Name, County, Business_Type, Costs,
  2     Sum(Costs) Over(Partition By County Order By Name) County_Total,
  3     Count(*) Over(Partition By County Order By Name) Num_Business
  4  From Sponsor
  5  Where Business_Type <> 'Manufacturing'
  6  Order By County, Name;

NAME                           COUNTY      BUSINESS_TYPE      COSTS COUNTY_TOTAL NUM_BUSINESS
------------------------------ ----------- ------------- ---------- ------------ ------------
Musgrave                       Cork        Retail          38857000     38857000            1
Aer Lingus                     Dublin      Service         12436200     12436200            1
Air Contractors                Dublin      Service           672000     13108200            2
Irish Continental Group        Dublin 1    Leisure          4270000      4270000            1
Primark                        Dublin 1    Retail          15946000     20216000            2
Statoil (Ireland)              Dublin 1    Service         12600000     32816000            3
Grafton Group                  Dublin 18   Retail          20944000     20944000            1
Vodafone Ireland               Dublin 18   Service         15400000     36344000            2
Capital Bars                   Dublin 2    Retail            672000       672000            1
Dunnes Stores                  Dublin 2    Retail          38150000     38822000            2
Eircom                         Dublin 2    Service         23520000     62342000            3
Greencore                      Dublin 2    Retail          20626340     82968340            4
Paddy Power                    Dublin 24   Retail          12782000     12782000            1
United Drug                    Dublin 24   Retail          20249600     33031600            2
Irish Shell                    Dublin 4    Retail          12600000     12600000            1
IAWS Group                     Dublin 8    Retail          17511480     17511480            1
Irish Food Processors          Louth       Retail          12236000     12236000            1

17 rows selected.

A00325752PS_SQL>
```

In this query, the PARTITION BY County ORDER BY Name ensures that the running total (SUM) and count (COUNT) are calculated within each County, while also considering the ascending order of Name. This combination provides dynamic aggregates that respect both grouping by County and sorting by Name.

Over Clause with Partition

Example 11. List the Name, Business Type and Costs for each individual Sponsor, as well as and a running costs total, minimum costs and maximum costs for business type, for every Business Type except Manufacturing. Sort the results by Business Type and then by Name

Cl scr

```
Select Name, Business_Type, Costs,
        Sum(Costs) Over(Partition By Business_Type) Costs_Total,
        Min(Costs) Over(Partition By Business_Type) Min_Costs,
        Max(Costs) Over(Partition By Business_Type) Max_Total
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Business_Type, Name;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Name, Business_Type, Costs,
  2     Sum(Costs) Over(Partition By Business_Type) Costs_Total,
  3     Min(Costs) Over(Partition By Business_Type) Min_Costs,
  4     Max(Costs) Over(Partition By Business_Type) Max_Total
  5  From Sponsor
  6  Where Business_Type <> 'Manufacturing'
  7  Order By Business_Type, Name;

NAME                         BUSINESS_TYPE      COSTS COSTS_TOTAL  MIN_COSTS  MAX_TOTAL
---------------------------- ------------ ---------- ----------- ---------- ----------
Irish Continental Group      Leisure         4270000     4270000    4270000    4270000
Capital Bars                 Retail           672000   210574420     672000   38857000
Dunnes Stores                Retail         38150000   210574420     672000   38857000
Grafton Group                Retail         20944000   210574420     672000   38857000
Greencore                    Retail         20626340   210574420     672000   38857000
IAWS Group                   Retail         17511480   210574420     672000   38857000
Irish Food Processors        Retail         12236000   210574420     672000   38857000
Irish Shell                  Retail         12600000   210574420     672000   38857000
Musgrave                     Retail         38857000   210574420     672000   38857000
Paddy Power                  Retail         12782000   210574420     672000   38857000
Primark                      Retail         15946000   210574420     672000   38857000
United Drug                  Retail         20249600   210574420     672000   38857000
Aer Lingus                   Service        12436200    64628200     672000   23520000
Air Contractors              Service          672000    64628200     672000   23520000
Eircom                       Service        23520000    64628200     672000   23520000
Statoil (Ireland)            Service        12600000    64628200     672000   23520000
Vodafone Ireland             Service        15400000    64628200     672000   23520000

17 rows selected.

A00325752PS_SQL>
```

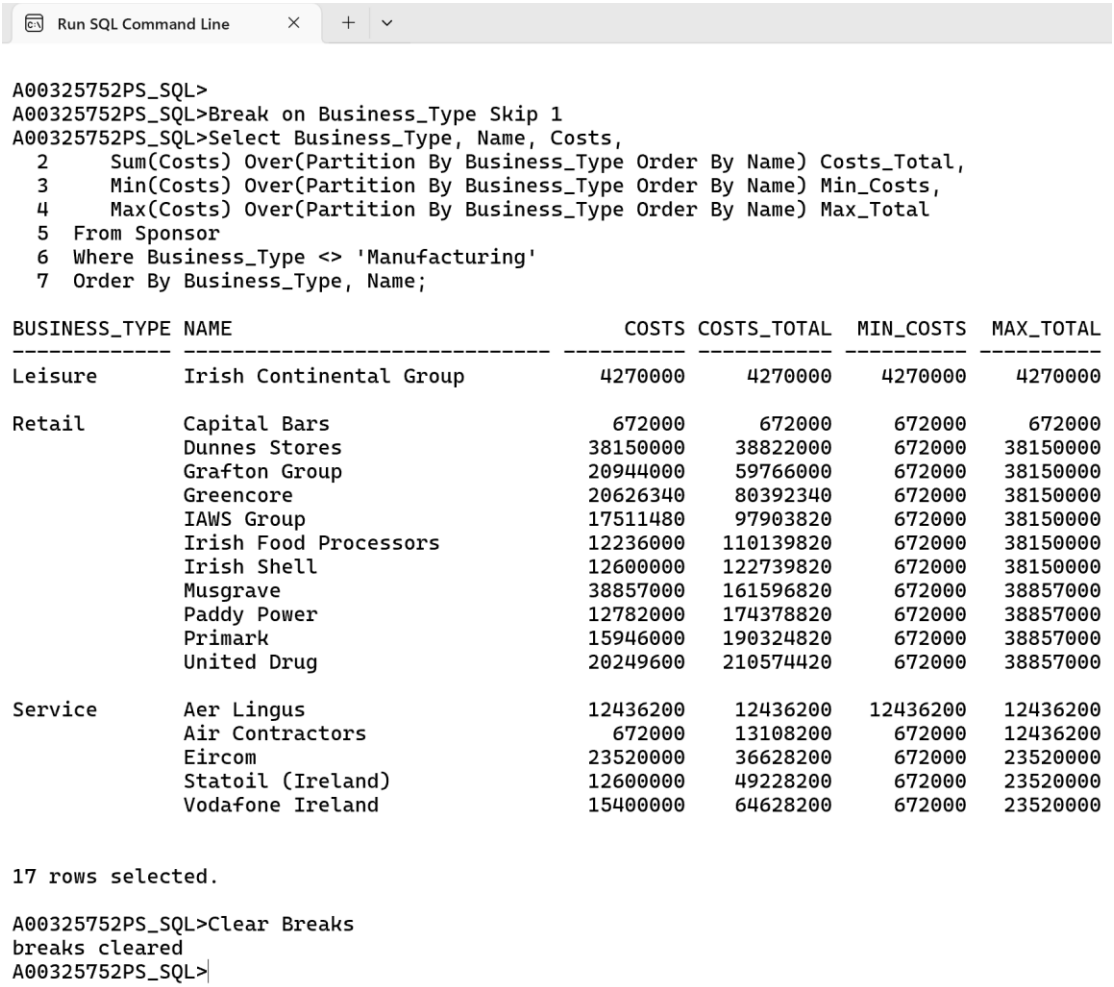
In this query, the PARTITION BY Business_Type groups rows by Business_Type and we are sorting by Business_Type and Name ensures the results are organized for easier interpretation within each category.

Over Clause with Partition and Order By

Example 12. List the Name, Business Type and Costs for each individual Sponsor, as well as and a running costs total, minimum costs and maximum costs for business type, for every Business Type except Manufacturing. Sort the results by Business Type and then by Name

Cl scr

Break on Business_Type Skip 1
Select Business_Type, Name, Costs,
Sum(Costs) Over(Partition By Business_Type Order By Name) **Costs_Total**,
Min(Costs) Over(Partition By Business_Type Order By Name) **Min_Costs**,
Max(Costs) Over(Partition By Business_Type Order By Name) **Max_Total**
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Business_Type, Name;
Clear Breaks



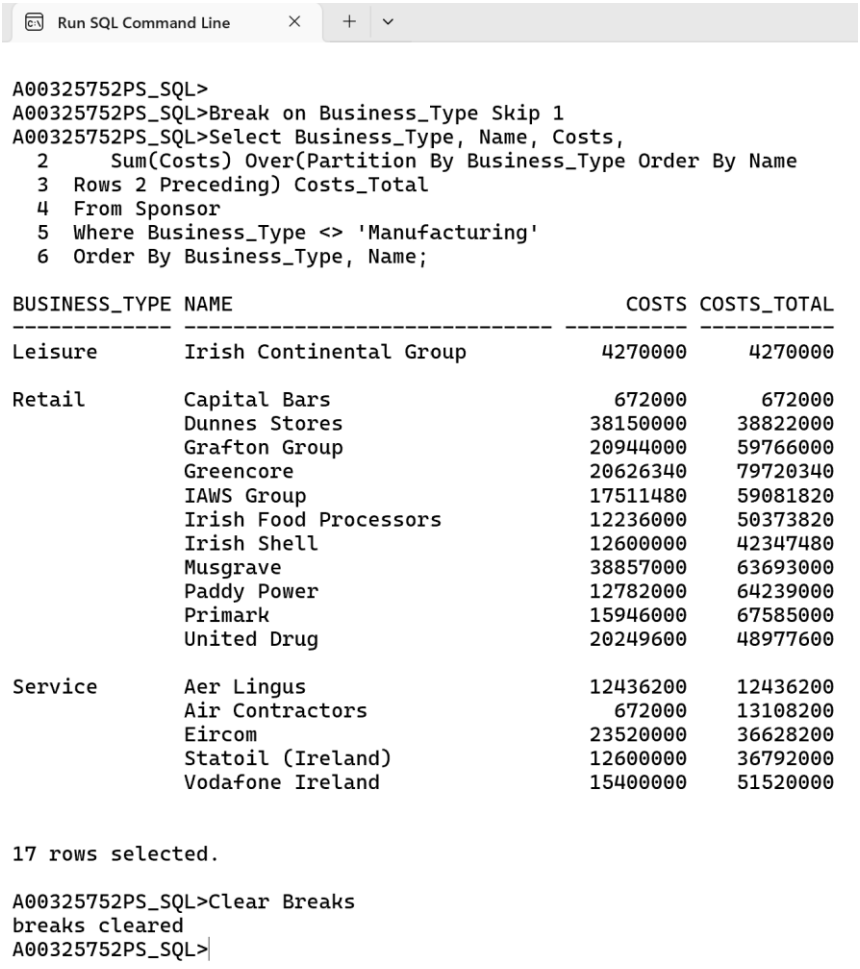

```
A00325752PS_SQL>
A00325752PS_SQL>Break on Business_Type Skip 1
A00325752PS_SQL>Select Business_Type, Name, Costs,
  2     Sum(Costs) Over(Partition By Business_Type Order By Name) Costs_Total,
  3     Min(Costs) Over(Partition By Business_Type Order By Name) Min_Costs,
  4     Max(Costs) Over(Partition By Business_Type Order By Name) Max_Total
  5  From Sponsor
  6  Where Business_Type <> 'Manufacturing'
  7  Order By Business_Type, Name;

BUSINESS_TYPE NAME                           COSTS COSTS_TOTAL  MIN_COSTS  MAX_TOTAL
------------- ------------------------------ ----------- ----------- ---------- ----------
Leisure       Irish Continental Group         4270000     4270000    4270000    4270000

Retail        Capital Bars                     672000      672000     672000     672000
              Dunnes Stores                  38150000    38822000     672000   38150000
              Grafton Group                  20944000    59766000     672000   38150000
              Greencore                      20626340    80392340     672000   38150000
              IAWS Group                     17511480    97903820     672000   38150000
              Irish Food Processors          12236000   110139820     672000   38150000
              Irish Shell                    12600000   122739820     672000   38150000
              Musgrave                       38857000   161596820     672000   38857000
              Paddy Power                    12782000   174378820     672000   38857000
              Primark                        15946000   190324820     672000   38857000
              United Drug                    20249600   210574420     672000   38857000

Service       Aer Lingus                     12436200    12436200   12436200   12436200
              Air Contractors                  672000    13108200     672000   12436200
              Eircom                         23520000    36628200     672000   23520000
              Statoil (Ireland)              12600000    49228200     672000   23520000
              Vodafone Ireland               15400000    64628200     672000   23520000


17 rows selected.

A00325752PS_SQL>Clear Breaks
breaks cleared
A00325752PS_SQL>
```

This query is similar to the one we have done previously but by using Break on Business Type gives a better view for viewing the data.

Over Clause with Partition, Order By and Windowing

Example 13. List the Name, Business Type and Costs for each individual Sponsor, as well as and a running cost for current Name plus the two previous names processed at the same Business type, filter every Business Type except Manufacturing. Sort the results by Business Type and then by Name.

```
Cl scr

Break on Business_Type Skip 1
Select Business_Type, Name, Costs,
        Sum(Costs) Over(Partition By Business_Type Order By Name
        Rows 2 Preceding) Costs_Total
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Business_Type, Name;
Clear Breaks
```



This query we have used Rows 2 Preceding so that the running total will only add up to the values of it's previous 2 rows Partition by Business_Type.

Over Clause with Partition, Order By and Windowing

Example 14. List the Title, Discipline, Payment Date and Costs for each individual Scholarship, as well as and a running annual_value for current scholarship from same discipline whose Payment date is 180 days preceding. Filter the results from 01- Jan 2024 to 31- Dec 2024. Sort the results by Business Type and then by Name.

Cl scr

```
Break on Discipline Skip 1
Select Discipline, Title, Initial_Payment_Date, Annual_Value,
        Sum(Annual_Value) Over(Partition By Discipline Order By Initial_Payment_Date
        Range 180 Preceding) Annual_Value_Total
From Scholarship
Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
And  Discipline Not In ('Medical', 'Psychology')
Order By Discipline, Initial_Payment_Date;
Clear Breaks
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Break on Discipline Skip 1
A00325752PS_SQL>Select Discipline, Title, Initial_Payment_Date, Annual_Value,
  2     Sum(Annual_Value) Over(Partition By Discipline Order By Initial_Payment_Date
  3  Range 180 Preceding) Annual_Value_Total
  4  From Scholarship
  5  Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
  6  And  Discipline Not In ('Medical', 'Psychology')
  7  Order By Discipline, Initial_Payment_Date;

DISCIPLINE  TITLE                                              INITIAL_P ANNUAL_VALUE ANNUAL_VALUE_TOTAL
----------- -------------------------------------------------- --------- ------------ -------------------
Engineering Air Contractors Student Scholarship                24-JAN-24        13600               13600
            Eircom Student Scholarship                         07-FEB-24        14700               28300
            Jefferson Smurfit Group Student Scholarship        04-APR-24        24900               53200
            Primark Student Scholarship                        08-MAY-24        21800               75000
            Glen Dimplex Student Scholarship                   30-MAY-24        19400               94400
            Eircom Student Scholarship                         02-JUN-24        14800              109200
            Glen Dimplex Student Scholarship                   21-JUL-24        23600              132800
            Century Homes Student Scholarship                  14-DEC-24        25400               49000
            Microsoft European Operations Student Scholarship  27-DEC-24        27700               76700

Science     Forest Laboratories Ireland Student Scholarship    15-JAN-24        19900               19900
            Capital Bars Student Scholarship                   14-FEB-24        15900               35800
            Dunnes Stores Student Scholarship                  09-APR-24        21100               56900
            Kerry Group Student Scholarship                    04-JUL-24        21500               78400
            Eircom Student Scholarship                         25-JUL-24        15500               74000
            Statoil (Ireland) Student Scholarship              05-SEP-24        19200               77300
            Kerry Group Student Scholarship                    14-SEP-24        16700               94000
            Diageo Ireland Student Scholarship                 30-OCT-24        15000               87900
            Century Homes Student Scholarship                  27-NOV-24        23200              111100


18 rows selected.

A00325752PS_SQL>Clear Breaks
breaks cleared
A00325752PS_SQL>
```

This query we have used Range 180 Preceding so that the running total will only add up to the annual value only for those whose date is 180 days before the Initial_Payment_Date.

Over Clause with Partition, Order By and Windowing

Example 15. List the Title, Discipline, Payment Date and Costs for each individual Scholarship, as well as and a running annual_value for current scholarship from same discipline whose Payment date is before and after the date in consideration (i.e +90 and -90 days). Filter the results from 01- Jan 2024 to 31- Dec 2024. Sort the results by Business Type and then by Name.

Cl scr

```
Break on Discipline Skip 1
Select Discipline, Title, Initial_Payment_Date, Annual_Value,
        Sum(Annual_Value) Over(Partition By Discipline Order By Initial_Payment_Date
        Range Between 90 Preceding And 90 Following) Annual_Value_Total
From Scholarship
Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
And  Discipline Not In ('Medical', 'Psychology')
Order By Discipline, Initial_Payment_Date;
Clear Breaks
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Break on Discipline Skip 1
A00325752PS_SQL>Select Discipline, Title, Initial_Payment_Date, Annual_Value,
  2     Sum(Annual_Value) Over(Partition By Discipline Order By Initial_Payment_Date
  3  Range Between 90 Preceding And 90 Following) Annual_Value_Total
  4  From Scholarship
  5  Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
  6  And  Discipline Not In ('Medical', 'Psychology')
  7  Order By Discipline, Initial_Payment_Date;

DISCIPLINE  TITLE                                              INITIAL_P ANNUAL_VALUE ANNUAL_VALUE_TOTAL
----------- -------------------------------------------------- --------- ------------ ------------------
Engineering Air Contractors Student Scholarship                24-JAN-24        13600              53200
            Eircom Student Scholarship                         07-FEB-24        14700              53200
            Jefferson Smurfit Group Student Scholarship        04-APR-24        24900             109200
            Primark Student Scholarship                        08-MAY-24        21800             104500
            Glen Dimplex Student Scholarship                   30-MAY-24        19400             104500
            Eircom Student Scholarship                         02-JUN-24        14800             104500
            Glen Dimplex Student Scholarship                   21-JUL-24        23600              79600
            Century Homes Student Scholarship                  14-DEC-24        25400              53100
            Microsoft European Operations Student Scholarship  27-DEC-24        27700              53100

Science     Forest Laboratories Ireland Student Scholarship    15-JAN-24        19900              56900
            Capital Bars Student Scholarship                   14-FEB-24        15900              56900
            Dunnes Stores Student Scholarship                  09-APR-24        21100              78400
            Kerry Group Student Scholarship                    04-JUL-24        21500              94000
            Eircom Student Scholarship                         25-JUL-24        15500              72900
            Statoil (Ireland) Student Scholarship              05-SEP-24        19200             111100
            Kerry Group Student Scholarship                    14-SEP-24        16700             111100
            Diageo Ireland Student Scholarship                 30-OCT-24        15000              74100
            Century Homes Student Scholarship                  27-NOV-24        23200              74100

18 rows selected.

A00325752PS_SQL>Clear Breaks
breaks cleared
A00325752PS_SQL>
```

This query is same as Example 14. But we are using initial_payment_date preceding and following of 90 days to add up into running total for annual payment.

Over Clause with Partition, Order By and Windowing

Example 16. Irrespective of the Department, list the Title, Payment Date and Day before 100 days of Payment Date. Examine the Payment Date for Title under consideration and display the Title and Payment Date of the first Title during the 100-day period that fell for current title (*there may be more than one title whose payment date is in the 100 days window but we are interested in the first of those*). Filter the results from 01- Jan 2024 to 31- Dec 2024 and discipline not equal to Medical or Psychology. Sort the results by Payment_Date

Cl scr

```
Select Title, Initial_Payment_Date, Initial_Payment_Date - 100 Pay_100, Annual_Value,
        First_Value(Title) Over (Order By Initial_Payment_Date Asc
                Range 100 Preceding) Scholarship_First,
        First_Value(Initial_Payment_Date) Over (Order By Initial_Payment_Date Asc
                Range 100 Preceding) Pay_Date_First
From Scholarship
Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
And  Discipline Not In ('Medical', 'Psychology')
Order By Initial_Payment_Date;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Title, Initial_Payment_Date, Initial_Payment_Date - 100 Pay_100, Annual_Value,
  2     First_Value(Title) Over (Order By Initial_Payment_Date Asc
  3  Range 100 Preceding) Scholarship_First,
  4     First_Value(Initial_Payment_Date) Over (Order By Initial_Payment_Date Asc
  5  Range 100 Preceding) Pay_Date_First
  6  From Scholarship
  7  Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
  8  And  Discipline Not In ('Medical', 'Psychology')
  9  Order By Initial_Payment_Date;
```

TITLE	INITIAL_P	PAY_100	ANNUAL_VALUE	SCHOLARSHIP_FIRST	PAY_DATE_
Forest Laboratories Ireland Student Scholarship	15-JAN-24	07-OCT-23	19900	Forest Laboratories Ireland Student Scholarship	15-JAN-24
Air Contractors Student Scholarship	24-JAN-24	16-OCT-23	13600	Forest Laboratories Ireland Student Scholarship	15-JAN-24
Eircom Student Scholarship	07-FEB-24	30-OCT-23	14700	Forest Laboratories Ireland Student Scholarship	15-JAN-24
Capital Bars Student Scholarship	14-FEB-24	06-NOV-23	15900	Forest Laboratories Ireland Student Scholarship	15-JAN-24
Jefferson Smurfit Group Student Scholarship	04-APR-24	26-DEC-23	24900	Forest Laboratories Ireland Student Scholarship	15-JAN-24
Dunnes Stores Student Scholarship	09-APR-24	31-DEC-23	21100	Forest Laboratories Ireland Student Scholarship	15-JAN-24
Primark Student Scholarship	08-MAY-24	29-JAN-24	21800	Eircom Student Scholarship	07-FEB-24
Glen Dimplex Student Scholarship	30-MAY-24	20-FEB-24	19400	Jefferson Smurfit Group Student Scholarship	04-APR-24
Eircom Student Scholarship	02-JUN-24	23-FEB-24	14800	Jefferson Smurfit Group Student Scholarship	04-APR-24
Kerry Group Student Scholarship	04-JUL-24	26-MAR-24	21500	Jefferson Smurfit Group Student Scholarship	04-APR-24
Glen Dimplex Student Scholarship	21-JUL-24	12-APR-24	23600	Primark Student Scholarship	08-MAY-24
Eircom Student Scholarship	25-JUL-24	16-APR-24	15500	Primark Student Scholarship	08-MAY-24
Statoil (Ireland) Student Scholarship	05-SEP-24	28-MAY-24	19200	Glen Dimplex Student Scholarship	30-MAY-24
Kerry Group Student Scholarship	14-SEP-24	06-JUN-24	16700	Kerry Group Student Scholarship	04-JUL-24
Diageo Ireland Student Scholarship	30-OCT-24	22-JUL-24	15000	Eircom Student Scholarship	25-JUL-24
Century Homes Student Scholarship	27-NOV-24	19-AUG-24	23200	Statoil (Ireland) Student Scholarship	05-SEP-24
Century Homes Student Scholarship	14-DEC-24	05-SEP-24	25400	Statoil (Ireland) Student Scholarship	05-SEP-24
Microsoft European Operations Student Scholarship	27-DEC-24	18-SEP-24	27700	Diageo Ireland Student Scholarship	30-OCT-24

```
18 rows selected.

A00325752PS_SQL>
```

This query identifies the first title and payment date within a 100-day window for each title, based on the Initial_Payment_Date. The FIRST_VALUE() function, combined with the ORDER BY and RANGE 100 PRECEDING, retrieves the earliest title and payment date within the rolling 100-day window for the current row

Over Clause with Partition, Order By and Windowing

Example 17. Irrespective of the Department, list the Title, Payment Date and Day before 100 days of Payment Date. Examine the Payment Date for Title under consideration and display the average Annual Value from Title where fell on 30 days or before and likewise show average Annual value for Title which fell on 30 days after the Payment date was initiated. Sort the output by "before Initial Payment Date" or "after Initial Payment Date" . Filter the results from 01- Jan 2024 to 31- Dec 2024 and discipline not equal to Medical or Psychology. Sort the results by Payment_Date

Cl scr

```
Select Title, Initial_Payment_Date,
Initial_Payment_Date - 30 "Date – 30",
Initial_Payment_Date + 30 "Date + 30", Annual_Value,
        Avg(Annual_Value) Over (Order By Initial_Payment_Date Asc
                Range 30 Preceding) Avg_Sal_30Dy_Before,
        Avg(Annual_Value) Over (Order By Initial_Payment_Date Desc
                Range 30 Preceding) Avg_Sal_30Dy_After
From Scholarship
Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
And  Discipline Not In ('Medical', 'Psychology')
Order By Initial_Payment_Date;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Title, Initial_Payment_Date,
  2  Initial_Payment_Date - 30 "Date - 30",
  3  Initial_Payment_Date + 30 "Date + 30", Annual_Value,
  4     Avg(Annual_Value) Over (Order By Initial_Payment_Date Asc
  5  Range 30 Preceding) Avg_Sal_30Dy_Before,
  6     Avg(Annual_Value) Over (Order By Initial_Payment_Date Desc
  7  Range 30 Preceding) Avg_Sal_30Dy_After
  8  From Scholarship
  9  Where Initial_Payment_Date Between '01-JAN-24' And '31-DEC-24'
 10  And  Discipline Not In ('Medical', 'Psychology')
 11  Order By Initial_Payment_Date;
```

TITLE	INITIAL_P	Date - 30	Date + 30	ANNUAL_VALUE	AVG_SAL_30DY_BEFORE	AVG_SAL_30DY_AFTER
Forest Laboratories Ireland Student Scholarship	15-JAN-24	16-DEC-23	14-FEB-24	19900	19900	16025
Air Contractors Student Scholarship	24-JAN-24	25-DEC-23	23-FEB-24	13600	16750	14733.3333
Eircom Student Scholarship	07-FEB-24	08-JAN-24	08-MAR-24	14700	16066.6667	15300
Capital Bars Student Scholarship	14-FEB-24	15-JAN-24	15-MAR-24	15900	16025	15900
Jefferson Smurfit Group Student Scholarship	04-APR-24	05-MAR-24	04-MAY-24	24900	24900	23000
Dunnes Stores Student Scholarship	09-APR-24	10-MAR-24	09-MAY-24	21100	23000	21450
Primark Student Scholarship	08-MAY-24	08-APR-24	07-JUN-24	21800	21450	18666.6667
Glen Dimplex Student Scholarship	30-MAY-24	30-APR-24	29-JUN-24	19400	20600	17100
Eircom Student Scholarship	02-JUN-24	03-MAY-24	02-JUL-24	14800	18666.6667	14800
Kerry Group Student Scholarship	04-JUL-24	04-JUN-24	03-AUG-24	21500	21500	20200
Glen Dimplex Student Scholarship	21-JUL-24	21-JUN-24	20-AUG-24	23600	22550	19550
Eircom Student Scholarship	25-JUL-24	25-JUN-24	24-AUG-24	15500	20200	15500
Statoil (Ireland) Student Scholarship	05-SEP-24	06-AUG-24	05-OCT-24	19200	19200	17950
Kerry Group Student Scholarship	14-SEP-24	15-AUG-24	14-OCT-24	16700	17950	16700
Diageo Ireland Student Scholarship	30-OCT-24	30-SEP-24	29-NOV-24	15000	15000	19100
Century Homes Student Scholarship	27-NOV-24	28-OCT-24	27-DEC-24	23200	19100	25433.3333
Century Homes Student Scholarship	14-DEC-24	14-NOV-24	13-JAN-25	25400	24300	26550
Microsoft European Operations Student Scholarship	27-DEC-24	27-NOV-24	26-JAN-25	27700	25433.3333	27700

```
18 rows selected.

A00325752PS_SQL>
```

This query computes the average Annual_Value for a rolling 30-day window before and after each Initial_Payment_Date, helping analyze trends around payment initiation. Filtering ensures only relevant disciplines and dates are included, and results are sorted chronologically by Initial_Payment_Date.

Over Clause with Partition By

Example 18: List the Name, Business Type, Main Market, Costs and Town for each individual Sponsor, as well as costs total, minimum costs broken by Business Type, Average cost broken by market and maximum costs for business Town, for every Business Type except Manufacturing. Sort the results by Business Type.

Cl scr

```
Select Name, Business_Type, Main_Market, Costs, Town,
        Sum(Costs) Over(Partition By Business_Type) Sum_Business_Types,
        Avg(Costs) Over(Partition By Main_Market) Avg_Market,
        Max(Costs) Over(Partition By Town) Max_Town
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Business_Type;
```

Run SQL Command Line

```
A00325752PS_SQL>
A00325752PS_SQL>Select Name, Business_Type, Main_Market, Costs, Town,
  2     Sum(Costs) Over(Partition By Business_Type) Sum_Business_Types,
  3     Avg(Costs) Over(Partition By Main_Market) Avg_Market,
  4     Max(Costs) Over(Partition By Town) Max_Town
  5  From Sponsor
  6  Where Business_Type <> 'Manufacturing'
  7  Order By Business_Type;

NAME                          BUSINESS_TYPE MAIN_MA      COSTS TOWN            SUM_BUSINESS_TYPES AVG_MARKET   MAX_TOWN
----------------------------- ------------- ------- ---------- --------------- ------------------ ---------- ----------
Irish Continental Group       Leisure       Ireland    4270000 Courtmacsherry             4270000 16429466.7   38150000
Greencore                     Retail        UK        20626340 Courtmacsherry           210574420   22951985   38150000
Dunnes Stores                 Retail        UK        38150000 Courtmacsherry           210574420   22951985   38150000
Musgrave                      Retail        Ireland   38857000 Ballingeary              210574420 16429466.7   38857000
IAWS Group                    Retail        Germany   17511480 Courtmacsherry           210574420    9949870   38150000
Capital Bars                  Retail        Germany     672000 Courtmacsherry           210574420    9949870   38150000
United Drug                   Retail        UK        20249600 Courtmacsherry           210574420   22951985   38150000
Grafton Group                 Retail        Germany   20944000 Courtmacsherry           210574420    9949870   38150000
Irish Food Processors         Retail        Ireland   12236000 Graiguenamanagh          210574420 16429466.7   12436200
Primark                       Retail        Ireland   15946000 Courtmacsherry           210574420 16429466.7   38150000
Paddy Power                   Retail        UK        12782000 Courtmacsherry           210574420   22951985   38150000
Irish Shell                   Retail        Ireland   12600000 Courtmacsherry           210574420 16429466.7   38150000
Vodafone Ireland              Service       Ireland   15400000 Courtmacsherry            64628200 16429466.7   38150000
Statoil (Ireland)             Service       Ireland   12600000 Courtmacsherry            64628200 16429466.7   38150000
Eircom                        Service       Ireland   23520000 Courtmacsherry            64628200 16429466.7   38150000
Aer Lingus                    Service       Ireland   12436200 Graiguenamanagh           64628200 16429466.7   12436200
Air Contractors               Service       Germany     672000 Graiguenamanagh           64628200    9949870   12436200

17 rows selected.

A00325752PS_SQL>
```

This query calculates the total costs by Business_Type, average costs by Main_Market, and maximum costs by Town using the PARTITION BY clause. These aggregations retain row-level details while grouping values logically, excluding 'Manufacturing' from Business_Type and sorting results by Business_Type.

Over Clause with Partition, Order By and Rank Function

Example 19. List the Name from Sponsor and Annual_Value, Discipline from Scholarship as well as the rank for Annual Value partitioned by Discipline. Limit the results to top 3 ranks only and sort the list by Discipline and Annual Value.

```
Cl scr

Break on Discipline Skip 1
Select *
From (Select SP.Name, SC.Annual_Value, SC.Discipline,
        Dense_Rank() Over (Partition By SC.Discipline Order By Annual_Value Desc) As
        Rank
        From Sponsor SP, Scholarship SC
        Where SP.Sponsor_Id = SC.Sponsor_Id)
Where Rank < 4
Order By Discipline, Annual_Value Desc;
Clear Breaks
```

```
A00325752_SQL>
A00325752_SQL>Break on Discipline Skip 1
A00325752_SQL>Select *
  2  From (Select SP.Name, SC.Annual_Value, SC.Discipline,
  3     Dense_Rank() Over (Partition By SC.Discipline Order By Annual_Value Desc) As Rank
  4  From Sponsor SP, Scholarship SC
  5  Where SP.Sponsor_Id = SC.Sponsor_Id)
  6  Where Rank < 4
  7  Order By Discipline, Annual_Value Desc;

NAME                           ANNUAL_VALUE DISCIPLINE        RANK
------------------------------ ------------ ----------- ----------
Microsoft European Operations         27700 Engineering          1
Kerry Group                           27300                      2
Century Homes                         25400                      3

Jefferson Smurfit Group               27700 Medical              1
Irish Food Processors                 25600                      2
Kerry Group                           25300                      3

Paddy Power                           27000 Psychology           1
Swords Laboratories                   26900                      2
Irish Food Processors                 26600                      3
Swords Laboratories                   26600                      3

Analog Devices                        27800 Science              1
Capital Bars                          27600                      2
Jefferson Smurfit Group               27300                      3


13 rows selected.
```

In this query we are using Dense_Rank to Rank the Annual Value based on Discipline. We have used Dense Rank instead of Rank to give a proper ranking without skipping any rank in between. We are limiting the results to top 3 ranks in this list to achieve this we have used in-line views so that we can add a condition on to the rank column.

Example 19 Cont…

Cl scr

```
Select SP.Name, SC.Annual_Value, SC.Discipline,
        Dense_Rank() Over (Partition By SC.Discipline Order By Annual_Value Desc) As
        Rank
From Sponsor SP, Scholarship SC
Where SP.Sponsor_Id = SC.Sponsor_Id
And Rank < 4
Order By Discipline, Annual_Value Desc;
```


```
A00325752_SQL>
A00325752_SQL>Select SP.Name, SC.Annual_Value, SC.Discipline,
  2  Dense_Rank() Over (Partition By SC.Discipline Order By Annual_Value Desc) As Rank
  3  From Sponsor SP, Scholarship SC
  4  Where SP.Sponsor_Id = SC.Sponsor_Id
  5  And Rank < 4
  6  Order By Discipline, Annual_Value Desc;
And Rank < 4
    *
ERROR at line 5:
ORA-00904: "RANK": invalid identifier
```

The above query cannot be done without the inline view since the rank is getting processed inside the select column and Oracle doesn't allow alias from the select column to be listed in the where clause.

Over Clause with Partition, Order By and Rank Function

Example 20. List the Name from Sponsor, Discipline from Scholarship as well as the rank for sum of Annual Value partitioned by Name and Discipline along with the Sum for Annual value partitioned by Name and Discipline. Limit the results to top 1 rank only.

```
Cl scr

Break on Discipline Skip 1
Select *
        From (Select Name, Discipline, Sum_Annual_Value,
        Dense_Rank() Over (Partition By Discipline Order By Sum_Annual_Value Desc) As
        Rank
        From(Select SP.Name, SC.Annual_Value, SC.Discipline,
        Sum(Annual_Value) Over (Partition By SP.Name, SC.Discipline Order By
        Annual_Value Desc) As Sum_Annual_Value
        From Sponsor SP, Scholarship SC
        Where SP.Sponsor_Id = SC.Sponsor_Id))
Where Rank < 2;
Clear Breaks
```

Run SQL Command Line

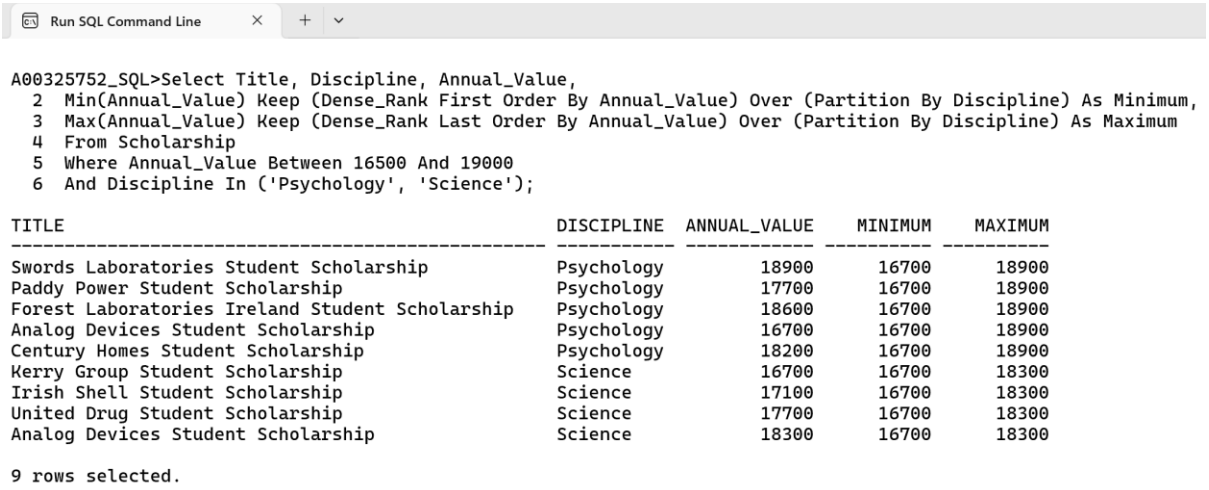
```
A00325752_SQL>
A00325752_SQL>Break on Discipline Skip 1
A00325752_SQL>Select *
  2  From (Select Name, Discipline, Sum_Annual_Value,
  3  Dense_Rank() Over (Partition By Discipline Order By Sum_Annual_Value Desc) As Rank
  4  From(Select SP.Name, SC.Annual_Value, SC.Discipline,
  5  Sum(Annual_Value) Over (Partition By SP.Name, SC.Discipline Order By Annual_Value Desc) As Sum_Annual_Value
  6  From Sponsor SP, Scholarship SC
  7  Where SP.Sponsor_Id = SC.Sponsor_Id))
  8  Where Rank < 2;

NAME                           DISCIPLINE  SUM_ANNUAL_VALUE       RANK
------------------------------ ----------- ---------------- ----------
Eircom                         Engineering            54200          1

Swords Laboratories            Medical                75500          1

Swords Laboratories            Psychology             95100          1

Kerry Group                    Science                61600          1


4 rows selected.

A00325752_SQL>Clear Breaks
breaks cleared
A00325752_SQL>
```

In the above query I am using Rank to rank the top 1 sponsor among various discipline. This query runs on 2 inline views where in first I am showing the Rank and Sum Value and in second I have taken Sum_Annual_Value in second inline query so that I can process it in the first inline view with Order By Sum_Annual Value as Oracle does not process the alias referred in same inline view.

Keep, Over Clause with Partition By and Rank Function

Example 21. From the Scholarship table select title, discipline and select minimum and maximum value for annual value that is partitioned by discipline. Filter the results where annual value between 16500 and 19000 and discipline must be in Psychology or Science.

```
Cl scr
Select Title, Discipline, Annual_Value,
Min(Annual_Value) Keep (Dense_Rank First Order By Annual_Value) Over (Partition By Discipline) As Minimum,
Max(Annual_Value) Keep (Dense_Rank Last Order By Annual_Value) Over (Partition By Discipline) As Maximum
From Scholarship
Where Annual_Value Between 16500 And 19000
And Discipline In ('Psychology', 'Science');
```



In this query the min and maximum summary functions are not used as it uses FIRST, LAST and KEEP that are picking the rows for which the values are being used. The Dense_Rank here will take away confusion if there is a tie between two rows but this won't affect the final results even if it is removed. This can be further simplified as below without the use of Dense_Rank KEEP, FIRST and LAST.

Example 21 Cont…

Cl scr
Select Title, Discipline, Annual_Value,
Min(Annual_Value) Over (Partition By Discipline Order By Annual_Value) As **Minimum**,
Max(Annual_Value) Over (Partition By Discipline) As **Maximum**
From Scholarship
Where Annual_Value Between 16500 And 19000;

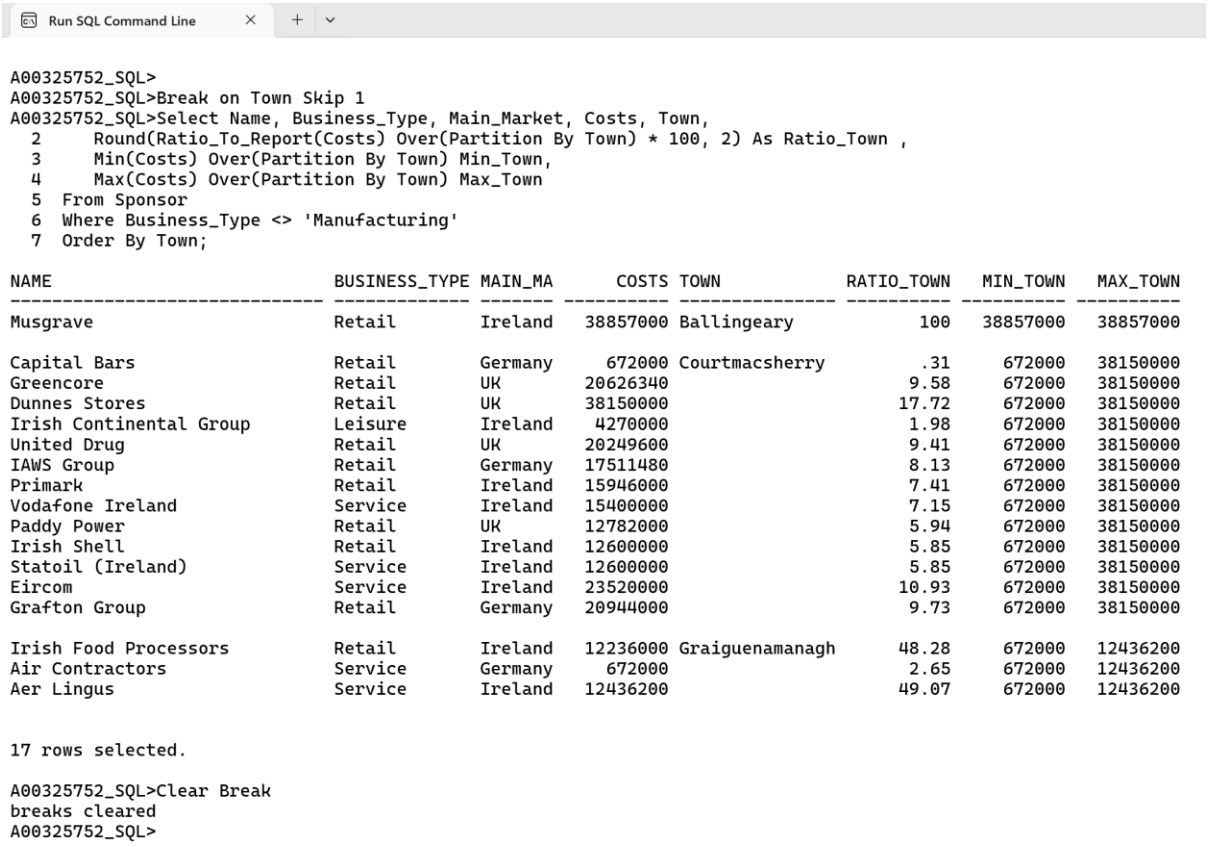

```
A00325752_SQL>Select Title, Discipline, Annual_Value,
  2  Min(Annual_Value) Over (Partition By Discipline Order By Annual_Value) As Minimum,
  3  Max(Annual_Value) Over (Partition By Discipline) As Maximum
  4  From Scholarship
  5  Where Annual_Value Between 16500 And 19000
  6  And Discipline In ('Psychology', 'Science');

TITLE                                              DISCIPLINE  ANNUAL_VALUE    MINIMUM    MAXIMUM
-------------------------------------------------- ----------- ------------ ---------- ----------
Analog Devices Student Scholarship                 Psychology         16700      16700      18900
Paddy Power Student Scholarship                    Psychology         17700      16700      18900
Century Homes Student Scholarship                  Psychology         18200      16700      18900
Forest Laboratories Ireland Student Scholarship    Psychology         18600      16700      18900
Swords Laboratories Student Scholarship            Psychology         18900      16700      18900
Kerry Group Student Scholarship                    Science            16700      16700      18300
Irish Shell Student Scholarship                    Science            17100      16700      18300
United Drug Student Scholarship                    Science            17700      16700      18300
Analog Devices Student Scholarship                 Science            18300      16700      18300

9 rows selected.
```

This will give the same output but this will query will be confused on which minimum and maximum value to take when there is a tie between two rows.

Over Clause with Partition By, Ratio Function and Round Function

Example 22. List the Name, Business Type, Main Market, Costs and Town for each individual Sponsor, as well as ratio of costs, maximum and minimum costs broken by Town, for every Business Type except Manufacturing. Sort the results by Business Type.

```
Cl scr

Break on Town Skip 1
Select Name, Business_Type, Main_Market, Costs, Town,
        Round(Ratio_To_Report(Costs) Over(Partition By Town) * 100, 2) As Ratio_Town ,
        Min(Costs) Over(Partition By Town) Min_Town,
        Max(Costs) Over(Partition By Town) Max_Town
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Town;
Clear Break
```



```
A00325752_SQL>
A00325752_SQL>Break on Town Skip 1
A00325752_SQL>Select Name, Business_Type, Main_Market, Costs, Town,
  2     Round(Ratio_To_Report(Costs) Over(Partition By Town) * 100, 2) As Ratio_Town ,
  3     Min(Costs) Over(Partition By Town) Min_Town,
  4     Max(Costs) Over(Partition By Town) Max_Town
  5  From Sponsor
  6  Where Business_Type <> 'Manufacturing'
  7  Order By Town;

NAME                           BUSINESS_TYPE MAIN_MA      COSTS TOWN            RATIO_TOWN   MIN_TOWN   MAX_TOWN
------------------------------ ------------- ------- ---------- --------------- ---------- ---------- ----------
Musgrave                       Retail        Ireland   38857000 Ballingeary            100   38857000   38857000

Capital Bars                   Retail        Germany     672000 Courtmacsherry         .31     672000   38150000
Greencore                      Retail        UK        20626340                      9.58     672000   38150000
Dunnes Stores                  Retail        UK        38150000                     17.72     672000   38150000
Irish Continental Group        Leisure       Ireland    4270000                      1.98     672000   38150000
United Drug                    Retail        UK        20249600                      9.41     672000   38150000
IAWS Group                     Retail        Germany   17511480                      8.13     672000   38150000
Primark                        Retail        Ireland   15946000                      7.41     672000   38150000
Vodafone Ireland               Service       Ireland   15400000                      7.15     672000   38150000
Paddy Power                    Retail        UK        12782000                      5.94     672000   38150000
Irish Shell                    Retail        Ireland   12600000                      5.85     672000   38150000
Statoil (Ireland)              Service       Ireland   12600000                      5.85     672000   38150000
Eircom                         Service       Ireland   23520000                     10.93     672000   38150000
Grafton Group                  Retail        Germany   20944000                      9.73     672000   38150000

Irish Food Processors          Retail        Ireland   12236000 Graiguenamanagh     48.28     672000   12436200
Air Contractors                Service       Germany     672000                      2.65     672000   12436200
Aer Lingus                     Service       Ireland   12436200                     49.07     672000   12436200


17 rows selected.

A00325752_SQL>Clear Break
breaks cleared
A00325752_SQL>
```

This query will give us total percent of ratio with the use of Ratio to Report function, minimum and maximum cost partitioned by Town. I have used the round function to round the ratio to report function into 2 decimal and multiplying it with 100 because it returns the value between 0 and 1.

Over Clause with Partition By, Order By and Lag Function

Example 23. List the Name, Town and Costs for each individual Sponsor, as well as and the cost for current Name, previous costs and costs difference between previous and current cost partition by Town and Group By Name, filter every Business_Type except Manufacturing. Sort the results by Business Type and then by Name.

Cl scr

```
Break on Town Skip 1
Select Name, Town, Costs,
        Lag(Costs,1,0) Over(Partition By Town Order By Name) Cost_Prev,
        Costs - Lag(Costs,1,0) Over(Partition By Town Order By Name) Cost_Diff
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Town, Name;
Clear Breaks
```

Run SQL Command Line

```
A00325752_SQL>
A00325752_SQL>Break on Town Skip 1
A00325752_SQL>Select Name, Town, Costs,
  2     Lag(Costs,1,0) Over(Partition By Town Order By Name) Cost_Prev,
  3     Costs - Lag(Costs,1,0) Over(Partition By Town Order By Name) Cost_Diff
  4  From Sponsor
  5  Where Business_Type <> 'Manufacturing'
  6  Order By Town, Name;

NAME                           TOWN                COSTS  COST_PREV  COST_DIFF
------------------------------ --------------- ---------- ---------- ----------
Musgrave                       Ballingeary       38857000          0   38857000

Capital Bars                   Courtmacsherry      672000          0     672000
Dunnes Stores                                    38150000     672000   37478000
Eircom                                           23520000   38150000  -14630000
Grafton Group                                    20944000   23520000   -2576000
Greencore                                        20626340   20944000    -317660
IAWS Group                                       17511480   20626340   -3114860
Irish Continental Group                           4270000   17511480  -13241480
Irish Shell                                      12600000    4270000    8330000
Paddy Power                                      12782000   12600000     182000
Primark                                          15946000   12782000    3164000
Statoil (Ireland)                                12600000   15946000   -3346000
United Drug                                      20249600   12600000    7649600
Vodafone Ireland                                 15400000   20249600   -4849600

Aer Lingus                     Graiguenamanagh   12436200          0   12436200
Air Contractors                                    672000   12436200  -11764200
Irish Food Processors                            12236000     672000   11564000


17 rows selected.

A00325752_SQL>Clear Breaks
breaks cleared
A00325752_SQL>
```

In this query we are using Lag function to take the value from previous cost and we are subtracting it from the previous cost to get the difference. Lag(Costs,1,0) The costs here in syntax is column name, 1 is the row we need for lag the output and 0 is the default value if there are no to lag ahead it will take 0 as input.

Over Clause with Partition By, Order By and Lead Function

Example 24. List the Name, Town and Costs for each individual Sponsor, as well as and the cost for current Name, next costs and costs difference between next and current cost partition by Town and Group by Name, filter every Business_Type except Manufacturing. Sort the results by Business Type and then by Name.

```
Cl scr

Break on Town Skip 1
Select Name, Town, Costs,
        Lead(Costs,1,0) Over(Partition By Town Order By Name) Cost_Prev,
        Costs - Lead(Costs,1,0) Over(Partition By Town Order By Name) Cost_Diff
From Sponsor
Where Business_Type <> 'Manufacturing'
Order By Town, Name;
Clear Breaks
```

Run SQL Command Line

```
A00325752_SQL>
A00325752_SQL>Break on Town Skip 1
A00325752_SQL>Select Name, Town, Costs,
  2     Lead(Costs,1,0) Over(Partition By Town Order By Name) Cost_Prev,
  3     Costs - Lead(Costs,1,0) Over(Partition By Town Order By Name) Cost_Diff
  4  From Sponsor
  5  Where Business_Type <> 'Manufacturing'
  6  Order By Town, Name;

NAME                           TOWN                COSTS  COST_PREV  COST_DIFF
------------------------------ --------------- ---------- ---------- ----------
Musgrave                       Ballingeary       38857000          0   38857000

Capital Bars                   Courtmacsherry      672000   38150000  -37478000
Dunnes Stores                                    38150000   23520000   14630000
Eircom                                           23520000   20944000    2576000
Grafton Group                                    20944000   20626340     317660
Greencore                                        20626340   17511480    3114860
IAWS Group                                       17511480    4270000   13241480
Irish Continental Group                           4270000   12600000   -8330000
Irish Shell                                      12600000   12782000    -182000
Paddy Power                                      12782000   15946000   -3164000
Primark                                          15946000   12600000    3346000
Statoil (Ireland)                                12600000   20249600   -7649600
United Drug                                      20249600   15400000    4849600
Vodafone Ireland                                 15400000          0   15400000

Aer Lingus                     Graiguenamanagh   12436200     672000   11764200
Air Contractors                                    672000   12236000  -11564000
Irish Food Processors                            12236000          0   12236000


17 rows selected.

A00325752_SQL>Clear Breaks
breaks cleared
A00325752_SQL>
```

This is the same as previous from Example 23 query but the only difference is we have a lead function here which takes the next value instead of the previous one.

Conclusion

Through Analytic queries in SQL. I have gained a strong foundation of SQL and analytic functions like Dense_Rank(), Rank(), Lag(), Lead(), Min(), Max() and Sum() using Over Clause with Partition By, Order By and Windowing for advanced analysis by applying these skills to real-world scenarios like scholarship tracking.

Youtube Video Link

Link Video 1: https://youtu.be/KhL8P3OeZNI

Link Video 2: https://youtu.be/SaYABtn7b0g