



TUS

**THE TECHNOLOGICAL UNIVERSITY OF THE
SHANNON: MIDLANDS MIDWEST**

Student Name:

Yanikepalli Sri Sekhar

Student Id Number:

A000325769

Course:

Masters in Data Analytics

Title of Assignment:

Relational Database Report

Date:

11 December 2024

Declaration

I hereby certify that the material, which is submitted in this report towards the award of MSc. in Data Analytics, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above named award.

Signed Y.Sri Sekhar

Date 11/12/2024

Table of Contents

Heading	Page no
1. Introduction	4
2. Entities and Attributes	5
3. Entity Relationship Diagram (ERD)	6
4. Creating Queries (Creating the Tables)	7-8
5. Insert Queries (Inserting the values into tables)	9-10
6. Analytic Functions	11-33
7. You tube Links	34

Introduction

The dataset contains two primary entities **Hotel** and **Inspection** providing a clear view of hotel operations, facilities and their compliance with health and safety standards. These tables are connected with a “Hotel_Id” attribute which is the common identifier in both **Hotel** and **Inspection** tables, “Hotel_Id” is the primary key (PK) in **Hotel** table and foreign key (FK) in **Inspection** table.

Hotel Table

The **Hotel** table contains some attributes which defines the critical information about the hotel. The attributes in which **Hotel** table contains are:

(**Hotel_Id (PK)**, Name, Town, County, Web_Site, Year_Opened, Grade (3 Star, 4 Star, 5 Star), Num_Bedrooms, Cost, Specialism (one of Weddings, Concerts, Exhibitions), Num_Events (total number of events in a typical year - Weddings, Concerts etc) Market_Value)

Inspection Table

The **Inspection** table contains some attributes which defines the critical information about the hotel inspection. The attributes in which **Inspection** table contains are:

(**Inspection_Id (FK)**, Hotel_Id, Visit_Date, Overall_Score (value between 1 and 100), Outcome (One of Pass or Fail), Storage (i.e Quality of Storage Facilities for food – one of Poor, Fair, Adequate, Superb), Fine_Imposed (the amount of money the hotel was fined for failing a test, otherwise Null), Food_Prep_Area (Poor, Good, Excellent), Hygiene (Very Dirty, Satisfactory, Very Clean), Atmosphere (Poor, Good, Excellent), Staff_Expertise (i.e. how well the hotel staff understand Health and Safety one of - Fair, Satisfactory, Excellent), Hotel_Id)

Purpose of the Dataset

The dataset gives the result by combining operational data with inspection outcomes. The outcomes may useful to stakeholders can make informed decisions about investments, regulatory interventions, and service improvements in the hospitality industry.

Entities and Attributes

Entities

There is only two entities in these two table that is **Hotel and Inspection.**

Attributes

There are total 12 attributes in **Hotel** table, they are

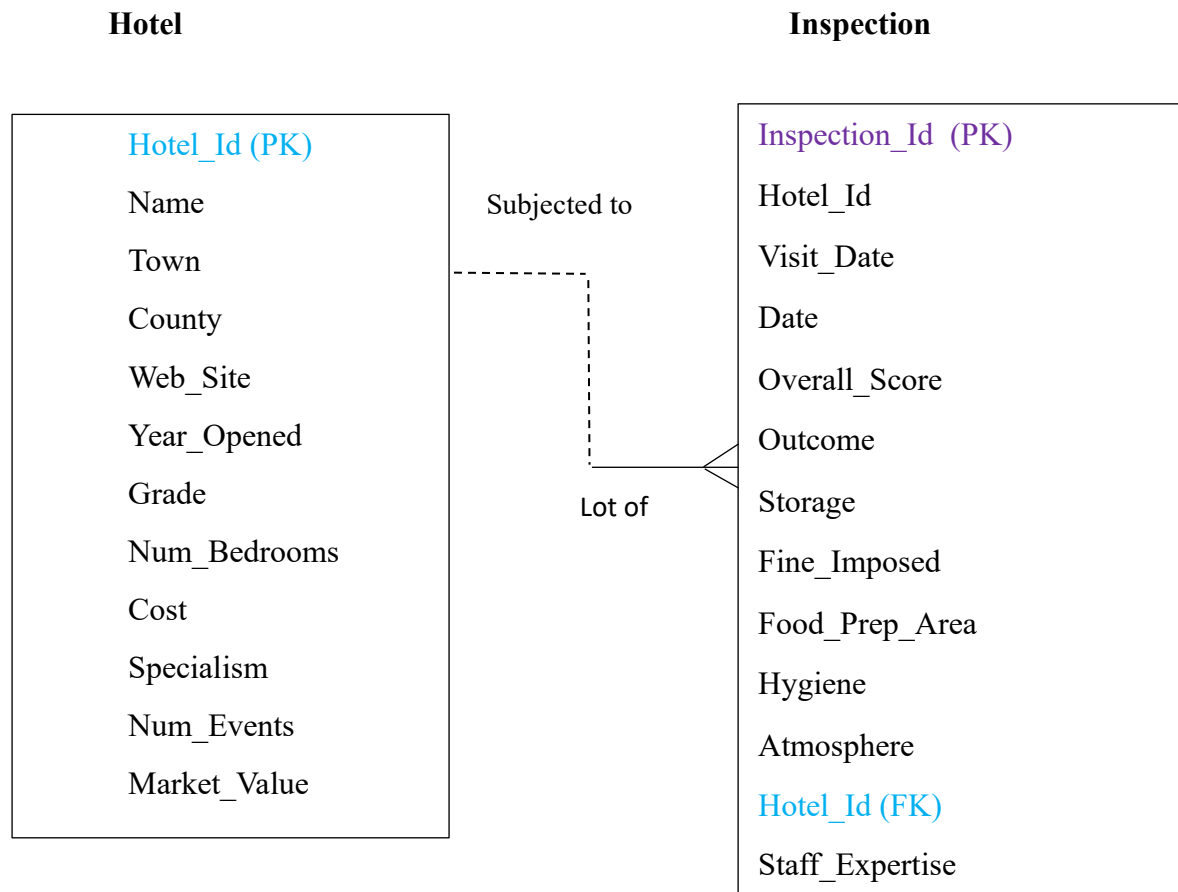
Hotel_Id	Number(3)
Name	Varchar(35)
Town	Varchar(15)
County	Varchar(9)
Web_Site	Varchar(131)
Year_Opened	Number(4)
Grade	Varchar(26)
Num_Bedrooms	Number(2)
Cost	Number(6)
Specialism	Varchar(11)
Num_Events	Number(1)
Market_Value	Varchar2(12)

There are total 11 attributes in **Inspection** table, they are

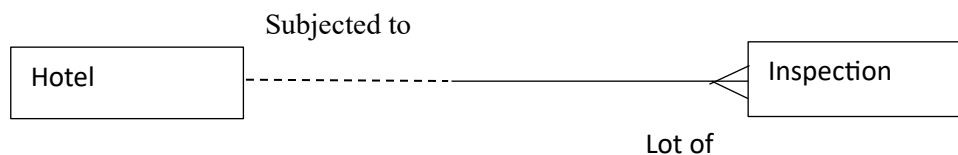
Inspection_Id	Number(4)
Hotel_Id	Number(3)
Visit_Date	Date
Overall_Score	Number(2)
Outcome	Varchar(4)
Storage	Varchar(8)
Fine_Imposed	Number(4)
Food_Prep_Area	Varchar(9)
Hygiene	Varchar(12)
Atmosphere	Varchar(9)
Staff_Expertise	Varchar(12)

Entity Relationship Diagrams (ERD)

Entity Relationship Diagram with attributes



Entity Relationship Diagram without attributes



Creating Queries (Creating the table)

Set SQLPrompt A00325769_SQL>

Set Pagesize 60

Set Linesize 180

Set Echo On

Set Feedback On

Set Termout On

Set Verify On

Create Table Hotel(

Hotel_Id Number(3), **Hotel_Id (PK)**

Name Varchar(35),

Town Varchar(15),

County Varchar(9),

Web_Site Varchar(131),

Year_Opened Number(4),

Grade Varchar(26),

Num_Bedrooms Number(2),

Cost Number(6),

Specialism Varchar(11),

Num_Events Number(1),

Market_Value Varchar2(12),

Constraint Hotel_Hotel_Id_PK Primary Key (Hotel_Id),

Constraint Hotel_Grade Check(Grade In('3 Star', '4 Star', '5 Star')),

Constraint Hotel_Specialism Check(Specialism In('Weddings', 'Concerts', 'Exhibitions')));

```
Run SQL Command Line × + ▾
A00325769_SQL>Create Table      Hotel(
 2 Hotel_Id          Number(3),
 3 Name              Varchar(35),
 4 Town              Varchar(15),
 5 County            Varchar(9),
 6 Web_Site          Varchar(131),
 7 Year_Opened       Number(4),
 8 Grade             Varchar(26),
 9 Num_Bedrooms      Number(2),
10 Cost              Number(6),
11 Specialism        Varchar(11),
12 Num_Events        Number(1),
13 Market_Value      Varchar2(12),
14 Constraint Hotel_Hotel_Id_PK Primary Key (Hotel_Id),
15 Constraint Hotel_Grade Check(Grade In( '3 Star', '4 Star', '5 Star')),
16 Constraint Hotel_Specialism Check(Specialism In('Weddings', 'Concerts', 'Exhibitions')));

Table created.

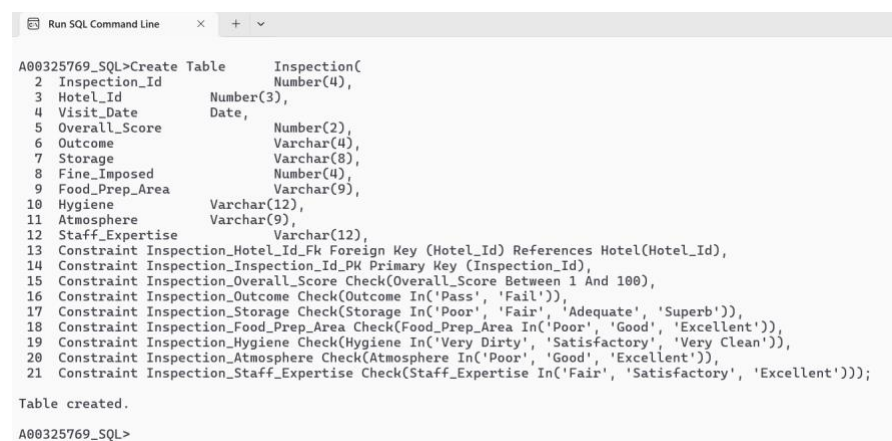
A00325769_SQL>
```

```

Create Table  Inspection(
Inspection_Id      Number(4), Inspection_Id(PK)
Hotel_Id           Number(3),
Visit_Date         Date,
Overall_Score      Number(2),
Outcome            Varchar(4),
Storage            Varchar(8),
Fine_Imposed       Number(4),
Food_Prep_Area     Varchar(9),
Hygiene            Varchar(12),
Atmosphere         Varchar(9),
Staff_Expertise    Varchar(12),

Constraint Inspection_Hotel_Id_Fk Foreign Key (Hotel_Id) References Hotel(Hotel_Id),
Constraint Inspection_Inspection_Id_PK Primary Key (Inspection_Id),
Constraint Inspection_Overall_Score Check(Overall_Score Between 1 And 100),
Constraint Inspection_Outcome Check(Outcome In('Pass', 'Fail')),
Constraint Inspection_Storage Check(Storage In('Poor', 'Fair', 'Adequate', 'Superb')),
Constraint Inspection_Food_Prep_Area Check(Food_Prep_Area In('Poor', 'Good', 'Excellent')),
Constraint Inspection_Hygiene Check(Hygiene In('Very Dirty', 'Satisfactory', 'Very Clean')),
Constraint Inspection_Atmosphere Check(Atmosphere In('Poor', 'Good', 'Excellent')),
Constraint Inspection_Staff_Expertise Check(Staff_Expertise In('Fair', 'Satisfactory', 'Excellent')));

```



```

Run SQL Command Line
A00325769_SQL>Create Table      Inspection(
2  Inspection_Id      Number(4),
3  Hotel_Id           Number(3),
4  Visit_Date         Date,
5  Overall_Score      Number(2),
6  Outcome            Varchar(4),
7  Storage            Varchar(8),
8  Fine_Imposed       Number(4),
9  Food_Prep_Area     Varchar(9),
10 Hygiene            Varchar(12),
11 Atmosphere         Varchar(9),
12 Staff_Expertise    Varchar(12),
13 Constraint Inspection_Hotel_Id_Fk Foreign Key (Hotel_Id) References Hotel(Hotel_Id),
14 Constraint Inspection_Inspection_Id_PK Primary Key (Inspection_Id),
15 Constraint Inspection_Overall_Score Check(Overall_Score Between 1 And 100),
16 Constraint Inspection_Outcome Check(Outcome In('Pass', 'Fail')),
17 Constraint Inspection_Storage Check(Storage In('Poor', 'Fair', 'Adequate', 'Superb')),
18 Constraint Inspection_Food_Prep_Area Check(Food_Prep_Area In('Poor', 'Good', 'Excellent')),
19 Constraint Inspection_Hygiene Check(Hygiene In('Very Dirty', 'Satisfactory', 'Very Clean')),
20 Constraint Inspection_Atmosphere Check(Atmosphere In('Poor', 'Good', 'Excellent')),
21 Constraint Inspection_Staff_Expertise Check(Staff_Expertise In('Fair', 'Satisfactory', 'Excellent')));

Table created.
A00325769_SQL>

```


Insert Queries (Inserting the values into tables)

insert into Hotel values(100,'Step House
Hotel','Graiguenamanagh','Carlow','https://www.stephousehotel.ie/',2007,'4
Star',20,170.59,'Weddings',5,'1.6 million');

insert into Hotel values(101,'Woodford Dolmen
Hotel','Clonegal','Carlow','https://www.woodforddolmenhotel.ie/',1976,'4
Star',81,81,'Weddings',3,'1.5 million');

insert into Hotel values(102,'Borrow View Bread and
Breakfast','Borris','Carlow','http://www.barrowviewbedandbreakfast.com/',2005,'4
Star',10,86,'Concerts',3,'1.8 million');

same insert Queries for 30 entries to Hotel Table

HOTEL_ID	NAME	TOWN	COUNTY	WEB_SITE	YEAR_OPENED	GRADE	NUM_BEDROOMS	COST	SPECIALISM	NUM_EVENTS	MARKET_VALUE
123	Park House Hotel	Carraroe	Galway	https://www.parkhousehotel.ie/	1999	4 Star	18	275	Concerts	4	2.5 million
124	Salthill Hotel	Athenry	Galway	https://www.salthillhotel.com/	2000	4 Star	20	187	Weddings	8	2.9 million
125	Sheraton Athlone Hotel	Castlepollard	Westmeath	https://www.marriott.com/en-us/hotels/gwysi-sheraton-athlone-hotel/overview/?scid=f2ae0541-1279-4f24-b197-a979c79310b0	1997	4 Star	15	181	Weddings	4	1.8 million
126	Glasson Lakehouse	Collinstown	Westmeath	https://glassonlakehouse.ie/	2002	4 Star					
127	Athlone Springs Hotel	Athlone	Westmeath	https://www.athlonespringshotel.com/	2003	4 Star	20	141	Weddings	4	1.8 million
128	Hodson Bay Hotel	Clonmellon	Westmeath	https://www.hodsonbayhotel.com/	2005	4 Star	25	136	Exhibitions	2	2.2 million
129	Radisson Blu Hotel	Athlone	Westmeath	https://www.radissonhotels.com/en-us/hotels/radisson-blu-athlone?cid=a:se+b:gmb+c:emea+i:local+e:rdb+d:ukirwe+h:IEATE1	1999	4 Star	30	162	Weddings	5	2.8 million

30 rows selected.
A00325769_SQL>

Insert Into Inspection Values(3935,100,'17-Jul-2024',81,'Fail','Adequate',200,'Excellent','Very Clean','Good','Satisfactory');

Insert Into Inspection Values(9763,101,'05-Jun-2024',78,'Pass','Poor',300,'Good','Very Dirty','Poor','Excellent');

Insert Into Inspection Values(6426,102,'15-Apr-2024',65,'Pass','Poor',100,'Good','Very Clean','Excellent','Excellent');

Insert Into Inspection Values(1566,103,'17-Oct-2024',53,'Pass','Fair',null,'Good','Very Clean','Poor','Fair');

Insert Into Inspection Values(5029,104,'10-Mar-2024',82,'Pass','Poor',200,'Good','Satisfactory','Good','Excellent');

same insert Queries for 70 entries to Inspection Table

Run SQL Command Line										
6261	106	14-NOV-24	82	Fail	Poor	200	Excellent	Very Clean	Excellent	Excellent
2228	107	12-DEC-23	64	Pass	Poor	200	Good	Satisfactory	Excellent	Fair
3151	108	28-JUL-24	72	Fail	Fair	300	Excellent	Very Dirty	Poor	Excellent
2881	109	13-NOV-24	53	Fail	Adequate		Excellent	Satisfactory	Excellent	Fair
3747	110	09-SEP-24	88	Fail	Adequate		Excellent	Very Clean	Excellent	Satisfactory
7025	111	09-MAY-24	58	Fail	Adequate	100	Poor	Satisfactory	Poor	Fair
3532	112	05-NOV-24	87	Fail	Poor	500	Excellent	Satisfactory	Good	Fair
5132	113	26-NOV-23	77	Fail	Adequate	100	Good	Very Clean	Excellent	Excellent
1149	114	09-MAY-24	93	Fail	Poor		Poor	Very Clean	Poor	Excellent
9241	115	23-APR-24	68	Pass	Poor		Excellent	Very Dirty	Excellent	Fair
1364	116	29-MAY-24	96	Pass	Fair		Excellent	Very Dirty	Excellent	Satisfactory
8887	117	03-APR-24	54	Fail	Adequate	500	Good	Very Dirty	Excellent	Satisfactory
3600	118	02-DEC-23	90	Fail	Adequate		Good	Very Clean	Good	Fair
5387	119	10-AUG-24	79	Fail	Poor		Excellent	Very Clean	Excellent	Satisfactory
1785	120	19-MAR-24	70	Pass	Adequate	100	Excellent	Very Dirty	Excellent	Fair
6638	121	05-DEC-23	61	Fail	Poor		Excellent	Very Clean	Good	Fair
6865	122	16-AUG-24	96	Pass	Poor	300	Excellent	Very Clean	Good	Satisfactory
9781	123	04-FEB-24	85	Fail	Fair	500	Poor	Very Clean	Poor	Satisfactory
9072	124	11-SEP-24	50	Fail	Poor	100	Poor	Very Clean	Poor	Satisfactory
6337	125	22-OCT-24	57	Pass	Adequate		Excellent	Very Clean	Excellent	Excellent
4089	126	09-AUG-24	80	Pass	Poor	100	Good	Very Clean	Excellent	Satisfactory
INSPECTION_ID	HOTEL_ID	VISIT_DAT	OVERALL_SCORE	OUTC	STORAGE	FINE_IMPOSED	FOOD_PREP	HYGIENE	ATMOSPHER	STAFF_EXPERT
8744	127	15-SEP-24	58	Fail	Poor	200	Poor	Satisfactory	Good	Excellent
1506	128	13-JUL-24	75	Pass	Adequate	100	Poor	Satisfactory	Excellent	Fair
3100	129	24-AUG-24	77	Pass	Fair	100	Poor	Satisfactory	Poor	Satisfactory
1595	100	04-AUG-24	66	Fail	Fair	200	Excellent	Very Clean	Poor	Excellent
7499	101	27-OCT-24	89	Pass	Adequate		Good	Satisfactory	Good	Satisfactory
4831	102	08-FEB-24	51	Fail	Fair		Excellent	Very Clean	Excellent	Excellent
5350	103	13-MAR-24	53	Fail	Poor	100	Poor	Very Dirty	Poor	Satisfactory
5542	104	26-APR-24	66	Pass	Adequate	500	Poor	Very Clean	Good	Satisfactory
4670	105	25-JAN-24	89	Fail	Fair		Good	Very Clean	Poor	Excellent
9005	106	14-FEB-24	58	Fail	Fair	100	Excellent	Very Dirty	Excellent	Fair
2966	107	27-JAN-24	55	Pass	Adequate	300	Good	Very Dirty	Good	Satisfactory
6661	108	08-JAN-24	66	Pass	Fair		Poor	Very Dirty	Poor	Satisfactory
5274	109	17-JUN-24	83	Pass	Poor	100	Excellent	Very Dirty	Good	Satisfactory
70 rows selected.										
A00325769_SQL>										

Oracle SQL_Analytic Functions

1. List the name, hotel id, and cost for each individual hotel plus the overall total cost for all hotels. Where specialism is weddings. Sort the results by hotel name, hotel id.

Cl scr

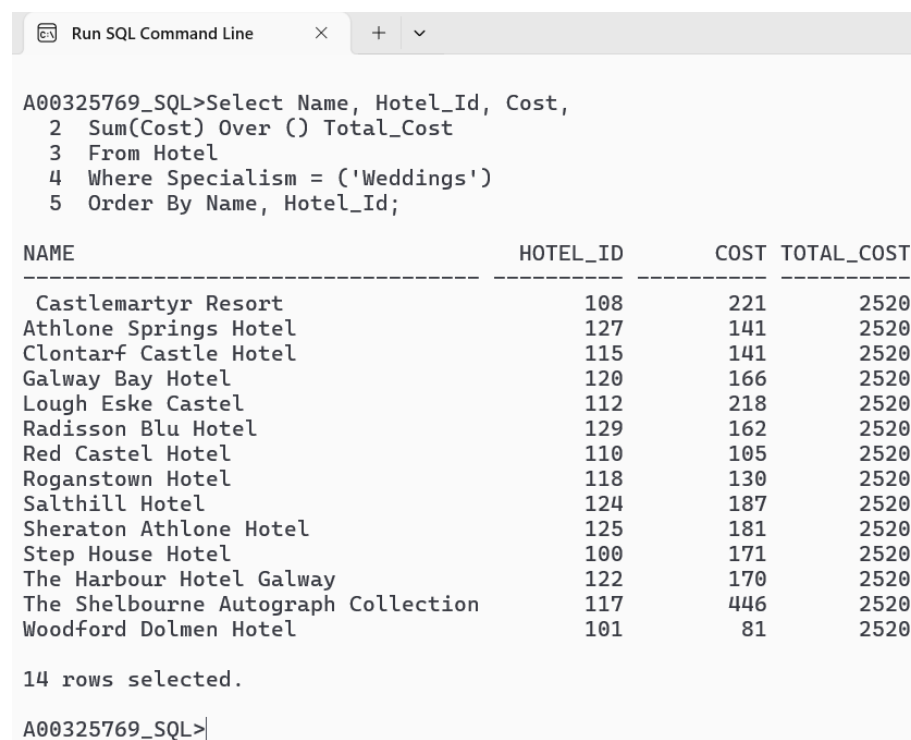
Select Name, Hotel_Id, Cost,

Sum(Cost) Over () Total_Cost

From Hotel

Where Specialism = ('Weddings')

Order By Name, Hotel_Id;



The screenshot shows a terminal window titled "Run SQL Command Line" with a query and its results. The query is:
A00325769_SQL>Select Name, Hotel_Id, Cost,
2 Sum(Cost) Over () Total_Cost
3 From Hotel
4 Where Specialism = ('Weddings')
5 Order By Name, Hotel_Id;
The results are displayed in a table with four columns: NAME, HOTEL_ID, COST, and TOTAL_COST. There are 14 rows of data, all with a TOTAL_COST of 2520. The text "14 rows selected." appears below the table. The prompt "A00325769_SQL>" is at the bottom.

NAME	HOTEL_ID	COST	TOTAL_COST
Castlemartyr Resort	108	221	2520
Athlone Springs Hotel	127	141	2520
Clontarf Castle Hotel	115	141	2520
Galway Bay Hotel	120	166	2520
Lough Eske Castel	112	218	2520
Radisson Blu Hotel	129	162	2520
Red Castel Hotel	110	105	2520
Roganstown Hotel	118	130	2520
Salthill Hotel	124	187	2520
Sheraton Athlone Hotel	125	181	2520
Step House Hotel	100	171	2520
The Harbour Hotel Galway	122	170	2520
The Shelbourne Autograph Collection	117	446	2520
Woodford Dolmen Hotel	101	81	2520

14 rows selected.

A00325769_SQL>

2. List the name, Hotel id, and cost for each individual hotel plus the largest cost among the hotels where specialism is in concert. In addition, calculate and display the amount each individual cost falls short of the largest cost. Sort the results by hotel id and name.

Cl scr

Select Name, Hotel_Id, Cost,

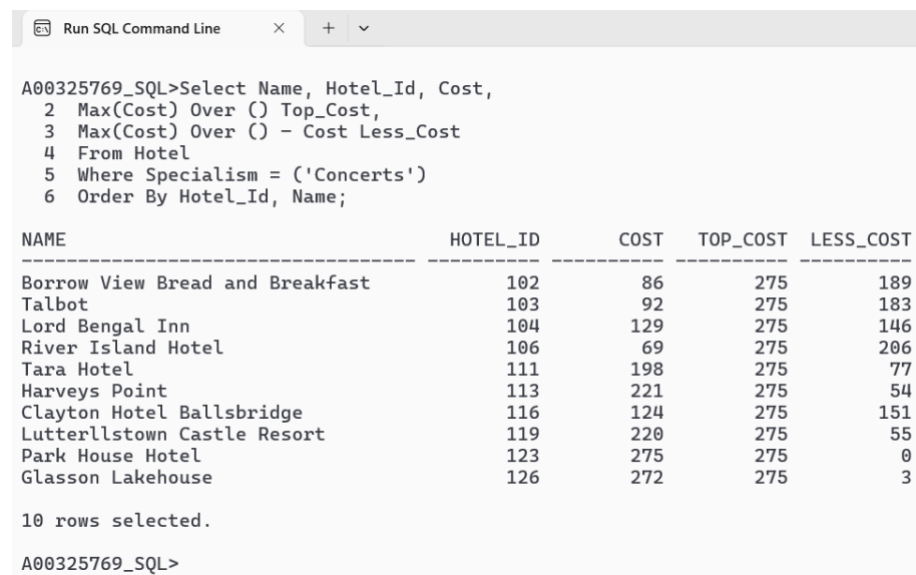
Max(Cost) Over () Top_Cost,

Max(Cost) Over () – Cost Less_Cost

From Hotel

Where Specialism = ('Concerts')

Order By Hotel_Id, Name;



The screenshot shows a SQL command line window titled "Run SQL Command Line". The query entered is:

```
A00325769_SQL>Select Name, Hotel_Id, Cost,
2 Max(Cost) Over () Top_Cost,
3 Max(Cost) Over () – Cost Less_Cost
4 From Hotel
5 Where Specialism = ('Concerts')
6 Order By Hotel_Id, Name;
```

The results are displayed in a table with the following columns: NAME, HOTEL_ID, COST, TOP_COST, and LESS_COST. The data is sorted by HOTEL_ID and then NAME.

NAME	HOTEL_ID	COST	TOP_COST	LESS_COST
Borrow View Bread and Breakfast	102	86	275	189
Talbot	103	92	275	183
Lord Bengal Inn	104	129	275	146
River Island Hotel	106	69	275	206
Tara Hotel	111	198	275	77
Harveys Point	113	221	275	54
Clayton Hotel Ballsbridge	116	124	275	151
Lutterllstown Castle Resort	119	220	275	55
Park House Hotel	123	275	275	0
Glasson Lakehouse	126	272	275	3

10 rows selected.

A00325769_SQL>

3. List the name and cost for each individual hotel plus the largest and smallest costs among hotels. [In addition, calculate and display the amount each individual cost falls Short of the largest cost, and the amount each individual cost exceeds the smallest cost. Whose hotel name starts with L, Sort the results by hotel id, and then by hotel name.

Cl scr

Select Name, Cost,

Max(Cost) Over () Top_Cost,

Min(Cost) Over () Worst_Cost,

Cost - Min(Cost) Over () Excess

From Hotel

Where Name Like 'L%'

Order By Hotel_Id, Name;

```
Run SQL Command Line × + v
A00325769_SQL>Select Name, Cost,
2 Max(Cost) Over () Top_Cost,
3 Min(Cost) Over () Worst_Cost,
4 Cost - Min(Cost) Over () Excess
5 From Hotel
6 Where Name Like 'L%'
7 Order By Hotel_Id, Name;
```

NAME	COST	TOP_COST	WORST_COST	EXCESS
Lord Bengal Inn	129	220	129	0
Lough Eske Castel	218	220	129	89
Lutterllstown Castle Resort	220	220	129	91

```
3 rows selected.
A00325769_SQL>
```

4. List the name, hotel id, and cost for each individual hotel plus the lowest, highest, average, and overall total cost for all hotels and the county is cork. Sort the results by hotel id, and then by hotel name.

Cl scr

Select Name, Hotel_Id, Cost,

Max(Cost) Over () Top_Cost,

Min(Cost) Over () Bottom_Cost,

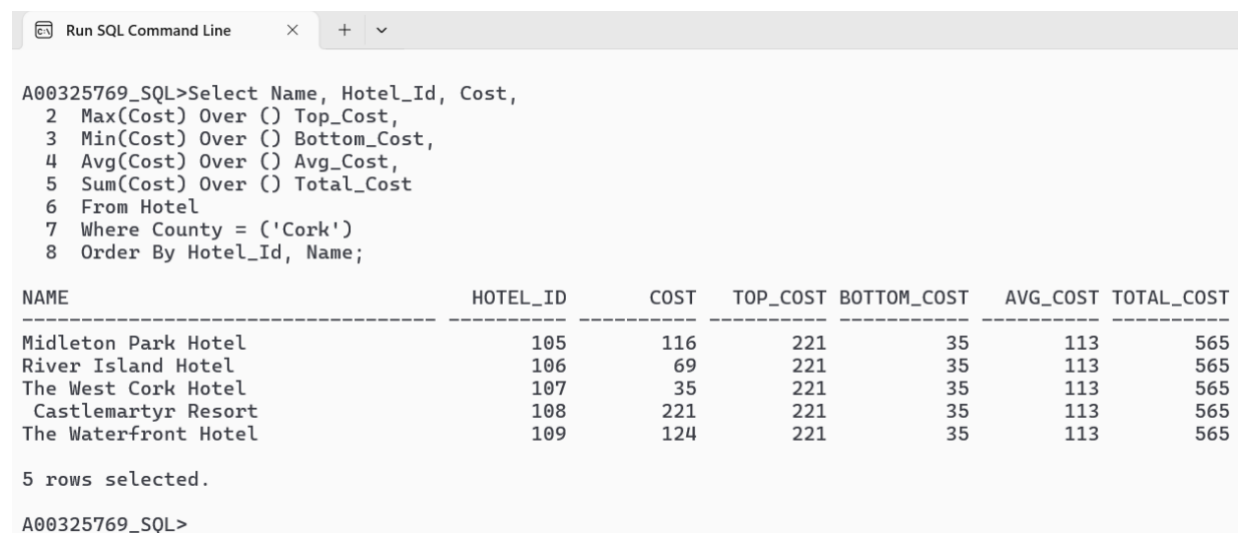
Avg(Cost) Over () Avg_Cost,

Sum(Cost) Over () Total_Cost

From Hotel

Where County = ('Cork')

Order By Hotel_Id, Name;



The screenshot shows a SQL command line window titled "Run SQL Command Line". The query entered is:

```
A00325769_SQL>Select Name, Hotel_Id, Cost,  
2 Max(Cost) Over () Top_Cost,  
3 Min(Cost) Over () Bottom_Cost,  
4 Avg(Cost) Over () Avg_Cost,  
5 Sum(Cost) Over () Total_Cost  
6 From Hotel  
7 Where County = ('Cork')  
8 Order By Hotel_Id, Name;
```

The results are displayed in a table with the following columns: NAME, HOTEL_ID, COST, TOP_COST, BOTTOM_COST, AVG_COST, and TOTAL_COST. The data is sorted by HOTEL_ID and then by NAME.

NAME	HOTEL_ID	COST	TOP_COST	BOTTOM_COST	AVG_COST	TOTAL_COST
Midleton Park Hotel	105	116	221	35	113	565
River Island Hotel	106	69	221	35	113	565
The West Cork Hotel	107	35	221	35	113	565
Castlemartyr Resort	108	221	221	35	113	565
The Waterfront Hotel	109	124	221	35	113	565

5 rows selected.

A00325769_SQL>

5. List name, hotel id, and cost for each 'individual hotel plus the lowest, highest, average, and cost for hotels and the hotel opened between 1999 and 2005. Sort the results by hotel id (ascending order) and then by hotel name.

Cl scr

Select Name, Hotel_Id, Cost,

Max(Cost) Over (Order By Hotel_Id) Top_Cost,

Min(Cost) Over (Order By Hotel_Id) Bottom_Cost,

Avg(Cost) Over (Order By Hotel_Id) Avg_Cost,

Sum(Cost) Over (Order By Hotel_Id) Total_Cost

From Hotel

Where Year_Opened Between 1999 And 2005

Order By Hotel_Id, Name;

Run SQL Command Line

```
A00325769_SQL>Select Name, Hotel_Id, Cost ,
2 Max(Cost) Over (Order By Hotel_Id) Top_Cost,
3 Min(Cost) Over (Order By Hotel_Id) Bottom_Cost,
4 Avg(Cost) Over (Order By Hotel_Id) Avg_Cost,
5 Sum(Cost) Over (Order By Hotel_Id) Total_Cost
6 From Hotel
7 Where Year_Opened Between 1999 And 2005
8 Order By Hotel_Id, Name;
```

NAME	HOTEL_ID	COST	TOP_COST	BOTTOM_COST	AVG_COST	TOTAL_COST
Borrow View Bread and Breakfast	102	86	86	86	86	86
Tara Hotel	111	198	198	86	142	284
Mill Park Hotel	114	130	198	86	138	414
Clayton Hotel Ballsbridge	116	124	198	86	134.5	538
Roganstown Hotel	118	130	198	86	133.6	668
Galway Bay Hotel	120	166	198	86	139	834
The Ardilaun Hotel	121	189	198	86	146.142857	1023
The Harbour Hotel Galway	122	170	198	86	149.125	1193
Park House Hotel	123	275	275	86	163.111111	1468
Salthill Hotel	124	187	275	86	165.5	1655
Glasson Lakehouse	126	272	275	86	175.181818	1927
Athlone Springs Hotel	127	141	275	86	172.333333	2068
Hodson Bay Hotel	128	136	275	86	169.538462	2204
Radisson Blu Hotel	129	162	275	86	169	2366

14 rows selected.

A00325769_SQL>

6. List name, hotel id, and cost for each individual hotel plus the lowest, highest, average, and cost for hotels and the hotel opened between 1999 and 2005. Sort the results by hotel id (descending order) and then by hotel name.

Cl scr

```
Select Name, Hotel_Id, Cost,
Max(Cost) Over (Order By Hotel_Id Desc) Top_Cost,
Min(Cost) Over (Order By Hotel_Id Desc) Bottom_Cost,
Avg(Cost) Over (Order By Hotel_Id Desc) Avg_Cost,
Sum(Cost) Over (Order By Hotel_Id Desc) Total_Cost
From Hotel
Where Year_Opened Between 1999 And 2005
Order By Hotel_Id Desc, Name;
```

Run SQL Command Line

```
A00325769_SQL>Select Name, Hotel_Id, Cost ,
2 Max(Cost) Over (Order By Hotel_Id Desc) Top_Cost,
3 Min(Cost) Over (Order By Hotel_Id Desc) Bottom_Cost,
4 Avg(Cost) Over (Order By Hotel_Id Desc) Avg_Cost,
5 Sum(Cost) Over (Order By Hotel_Id Desc) Total_Cost
6 From Hotel
7 Where Year_Opened Between 1999 And 2005
8 Order By Hotel_Id Desc, Name;
```

NAME	HOTEL_ID	COST	TOP_COST	BOTTOM_COST	AVG_COST	TOTAL_COST
Radisson Blu Hotel	129	162	162	162	162	162
Hodson Bay Hotel	128	136	162	136	149	298
Athlone Springs Hotel	127	141	162	136	146.333333	439
Glasson Lakehouse	126	272	272	136	177.75	711
Salthill Hotel	124	187	272	136	179.6	898
Park House Hotel	123	275	275	136	195.5	1173
The Harbour Hotel Galway	122	170	275	136	191.857143	1343
The Ardilaun Hotel	121	189	275	136	191.5	1532
Galway Bay Hotel	120	166	275	136	188.666667	1698
Roganstown Hotel	118	130	275	130	182.8	1828
Clayton Hotel Ballsbridge	116	124	275	124	177.454545	1952
Mill Park Hotel	114	130	275	124	173.5	2082
Tara Hotel	111	198	275	124	175.384615	2280
Borrow View Bread and Breakfast	102	86	275	86	169	2366

14 rows selected.

A00325769_SQL>

7. List name, hotel id, and cost for each individual hotel plus the lowest, highest, average, and total cost for hotels and name starts with T. Sort the results by Name.

Cl scr

Select Name, Hotel_Id, Cost,

Max(Cost) Over (Order By Name) Top_Cost,

Min(Cost) Over (Order By Name) Bottom_Cost,

Avg(Cost) Over (Order By Name) Avg_Cost,

Sum(Cost) Over (Order By Name) Total_Cost

From Hotel

Where Name Like 'T%'

Order By Name;

Run SQL Command Line						
A00325769_SQL>Select Name, Hotel_Id, Cost , 2 Max(Cost) Over (Order By Name) Top_Cost, 3 Min(Cost) Over (Order By Name) Bottom_Cost, 4 Avg(Cost) Over (Order By Name) Avg_Cost, 5 Sum(Cost) Over (Order By Name) Total_Cost 6 From Hotel 7 Where Name Like 'T%' 8 Order By Name;						
NAME	HOTEL_ID	COST	TOP_COST	BOTTOM_COST	AVG_COST	TOTAL_COST
Talbot	103	92	92	92	92	92
Tara Hotel	111	198	198	92	145	290
The Ardilaun Hotel	121	189	198	92	159.666667	479
The Harbour Hotel Galway	122	170	198	92	162.25	649
The Shelbourne Autograph Collection	117	446	446	92	219	1095
The Waterfront Hotel	109	124	446	92	203.166667	1219
The West Cork Hotel	107	35	446	35	179.142857	1254
7 rows selected.						
A00325769_SQL>						

8. List name, hotel id, and cost for each individual hotel plus the lowest, highest, average, and total cost for hotels and bedrooms must be 10 . Sort the results by Hotel id and Name

Cl scr

Select Name, Hotel_Id, Cost,

Max(Cost) Over (Order By Name) Top_Cost,

Min(Cost) Over (Order By Name) Bottom_Cost,

Avg(Cost) Over (Order By Name) Avg_Cost,

Sum(Cost) Over (Order By Name) Total_Cost

From Hotel

Where Num_Bedrooms = 10

Order By Hotel_Id, Name;

Run SQL Command Line						
A00325769_SQL>Select Name, Hotel_Id, Cost , 2 Max(Cost) Over (Order By Name) Top_Cost, 3 Min(Cost) Over (Order By Name) Bottom_Cost, 4 Avg(Cost) Over (Order By Name) Avg_Cost, 5 Sum(Cost) Over (Order By Name) Total_Cost 6 From Hotel 7 Where Num_Bedrooms = 10 8 Order By Hotel_Id, Name;						
NAME	HOTEL_ID	COST	TOP_COST	BOTTOM_COST	AVG_COST	TOTAL_COST
Borrow View Bread and Breakfast	102	86	86	86	86	86
River Island Hotel	106	69	221	69	129.25	517
Tara Hotel	111	198	221	69	143	715
Harveys Point	113	221	221	86	149.333333	448
Clontarf Castle Hotel	115	141	141	86	113.5	227
5 rows selected.						
A00325769_SQL>						

9. List name, hotel id, and cost for each individual hotel as well as the number of bedrooms in each hotel and cumulative running cost total for hotels at the county cork and carlow. Sort the results by hotel id and then by hotel name.

Cl scr

```
Select Name, Hotel_Id, Cost,
Sum(Cost) Over (Partition By Specialism) Specialism_Total,
Count(*) Over (Partition By Specialism) Num_Bedrooms
From Hotel
Where County In ('Cork', 'Carlow')
Order By Hotel_Id, Name;
```

Run SQL Command Line				
A00325769_SQL>Select Name, Hotel_Id, Cost, 2 Sum(Cost) Over (Partition By Specialism) Specialism_Total, 3 Count(*) Over (Partition By Specialism) Num_Bedrooms 4 From Hotel 5 Where County In ('Cork', 'Carlow') 6 Order By Hotel_Id, Name;				
NAME	HOTEL_ID	COST	SPECIALISM_TOTAL	NUM_BEDROOMS
Step House Hotel	100	171	473	3
Woodford Dolmen Hotel	101	81	473	3
Borrow View Bread and Breakfast	102	86	376	4
Talbot	103	92	376	4
Lord Bengal Inn	104	129	376	4
Midleton Park Hotel	105	116	275	3
River Island Hotel	106	69	376	4
The West Cork Hotel	107	35	275	3
Castlemartyr Resort	108	221	473	3
The Waterfront Hotel	109	124	275	3
10 rows selected.				
A00325769_SQL>				

10. List name, hotel id, and cost for each individual hotel as well as the number of bedrooms in each hotel and cumulative running cost total for hotels at the county cork and carlow. Sort the results by hotel id and then by hotel name.

Cl scr

Select Name, Hotel_Id, Cost,

Sum(Cost) Over (Partition By Specialism Order By Hotel_Id, Name) Specialism_Total,

Count(*) Over (Partition By Specialism Order By Hotel_Id, Name) Num_Bedrooms

From Hotel

Where County In ('Cork', 'Carlow')

Order By Hotel_Id, Name;

```
Run SQL Command Line
A00325769_SQL>Select Name, Hotel_Id, Cost,
2 Sum(Cost) Over (Partition By Specialism Order By Hotel_Id, Name) Specialism_Total,
3 Count(*) Over (Partition By Specialism Order By Hotel_Id, Name) Num_Bedrooms
4 From Hotel
5 Where County In ('Cork', 'Carlow')
6 Order By Hotel_Id, Name;
```

NAME	HOTEL_ID	COST	SPECIALISM_TOTAL	NUM_BEDROOMS
Step House Hotel	100	171	171	1
Woodford Dolmen Hotel	101	81	252	2
Borrow View Bread and Breakfast	102	86	86	1
Talbot	103	92	178	2
Lord Bengal Inn	104	129	307	3
Middleton Park Hotel	105	116	116	1
River Island Hotel	106	69	376	4
The West Cork Hotel	107	35	151	2
Castlemartyr Resort	108	221	473	3
The Waterfront Hotel	109	124	275	3

```
10 rows selected.
A00325769_SQL>
```

11. List name, Grade, hotel id, and cost for each individual hotel as well as a cumulative running cost total for hotels at the county cork and dublin. Sort the results by hotel id and then by hotel name

Cl scr

Select Name, Grade, Hotel_Id, Cost,

Sum(Cost) Over (Partition By Specialism) Specialism_Total,

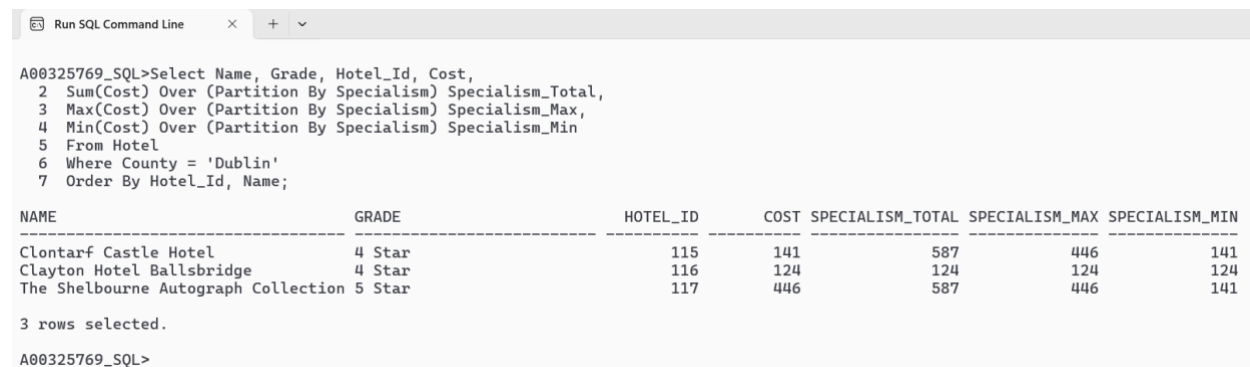
Max(Cost) Over (Partition By Specialism) Specialism_Max,

Min(Cost) Over (Partition By Specialism) Specialism_Min

From Hotel

Where County = 'Dublin'

Order By Hotel_Id, Name;



The screenshot shows a SQL command line window with the following content:

```
A00325769_SQL>Select Name, Grade, Hotel_Id, Cost,  
2 Sum(Cost) Over (Partition By Specialism) Specialism_Total,  
3 Max(Cost) Over (Partition By Specialism) Specialism_Max,  
4 Min(Cost) Over (Partition By Specialism) Specialism_Min  
5 From Hotel  
6 Where County = 'Dublin'  
7 Order By Hotel_Id, Name;
```

NAME	GRADE	HOTEL_ID	COST	SPECIALISM_TOTAL	SPECIALISM_MAX	SPECIALISM_MIN
Clontarf Castle Hotel	4 Star	115	141	587	446	141
Clayton Hotel Ballsbridge	4 Star	116	124	124	124	124
The Shelbourne Autograph Collection	5 Star	117	446	587	446	141

3 rows selected.
A00325769_SQL>

12. List the name, Grade, hotel id, and cost for each individual hotel located in galway as well as a cumulative running cost total and average cost for hotels broke down by grade. Sort the results by Grade and then by hotel name

Cl scr

Select Name, Grade, Hotel_Id, Cost,

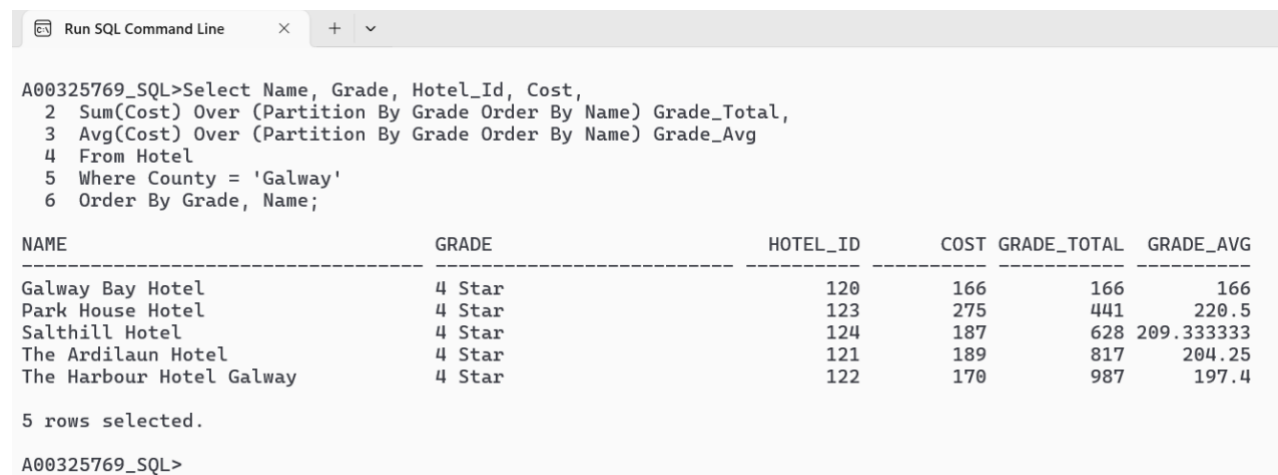
Sum(Cost) Over (Partition By Grade Order By Name) Grade_Total,

Avg(Cost) Over (Partition By Grade Order By Name) Grade_Avg

From Hotel

Where County = 'Galway'

Order By Grade, Name;



The screenshot shows a SQL command line window titled "Run SQL Command Line". The query entered is:

```
A00325769_SQL>Select Name, Grade, Hotel_Id, Cost,  
2 Sum(Cost) Over (Partition By Grade Order By Name) Grade_Total,  
3 Avg(Cost) Over (Partition By Grade Order By Name) Grade_Avg  
4 From Hotel  
5 Where County = 'Galway'  
6 Order By Grade, Name;
```

The results are displayed in a table with the following columns: NAME, GRADE, HOTEL_ID, COST, GRADE_TOTAL, and GRADE_AVG. The data is sorted by grade and then by hotel name.

NAME	GRADE	HOTEL_ID	COST	GRADE_TOTAL	GRADE_AVG
Galway Bay Hotel	4 Star	120	166	166	166
Park House Hotel	4 Star	123	275	441	220.5
Salthill Hotel	4 Star	124	187	628	209.333333
The Ardilaun Hotel	4 Star	121	189	817	204.25
The Harbour Hotel Galway	4 Star	122	170	987	197.4

5 rows selected.

A00325769_SQL>

13. List the Name, cost, county, hotel id, cost for each hotel as well as a cumulative running cost for the current hotels plus the two earlier/ previous hotels processed at the name (i.e there should be no overlap between names) where specialism in concerts. Sort the output by county and then hotel name.

Cl scr

Select Name, Cost, County, Hotel_Id,

Sum(Cost) Over (Partition By County Order By Name Rows 2 Preceding) Cost_Total

From Hotel

Where Specialism = 'Concerts'

Order By County, Name;

```

Run SQL Command Line
A00325769_SQL>Select Name, Cost, County, Hotel_Id,
  2 Sum(Cost) Over (Partition By County Order By Name Rows 2 Preceding) Cost_Total
  3 From Hotel
  4 Where Specialism = 'Concerts'
  5 Order By County, Name;

NAME                                COST COUNTY      HOTEL_ID COST_TOTAL
-----
Borrow View Bread and Breakfast      86 Carlow         102        86
Lord Bengal Inn                     129 Carlow         104       215
Talbot                               92 Carlow         103       307
River Island Hotel                   69 Cork           106        69
Harveys Point                       221 Donegal        113       221
Tara Hotel                           198 Donegal        111       419
Clayton Hotel Ballsbridge            124 Dublin          116       124
Lutterllstown Castle Resort          220 Fingal          119       220
Park House Hotel                     275 Galway          123       275
Glasson Lakehouse                   272 Westmeath       126       272

10 rows selected.

A00325769_SQL>

```

14. List the Atmosphere, visit date, storage, and fine imposed, and what is the cumulative total of fines imposed over the past 90 days for each atmosphere category, for inspections where the storage condition was Poor.

Cl scr

Select Atmosphere, Visit_Date, Storage, Fine_Imposed,

Sum(Fine_Imposed) Over (Partition By Atmosphere Order By Visit_Date Range 90 Preceding)

Fine_Imposed_Total

From Inspection

Where Storage = 'Poor'

Order By Atmosphere, Visit_Date;

```
Run SQL Command Line
A00325769_SQL>Select Atmosphere, Visit_Date, Storage, Fine_Imposed,
2 Sum(Fine_Imposed) Over (Partition By Atmosphere Order By Visit_Date Range 90 Preceding) Fine_Imposed_Total
3 From Inspection
4 Where Storage = 'Poor'
5 Order By Atmosphere, Visit_Date;
```

ATMOSPHER	VISIT_DAT	STORAGE	FINE_IMPOSED	FINE_IMPOSED_TOTAL
Excellent	12-DEC-23	Poor	200	200
Excellent	26-MAR-24	Poor		
Excellent	15-APR-24	Poor	100	100
Excellent	23-APR-24	Poor		100
Excellent	17-JUL-24	Poor	100	100
Excellent	09-AUG-24	Poor	100	200
Excellent	10-AUG-24	Poor		200
Excellent	14-NOV-24	Poor	200	200
Good	05-DEC-23	Poor		
Good	19-DEC-23	Poor		
Good	19-JAN-24	Poor	500	500
Good	22-JAN-24	Poor		500
Good	11-FEB-24	Poor	500	1000
Good	10-MAR-24	Poor	200	1200
Good	30-MAR-24	Poor		1200
Good	17-JUN-24	Poor	100	100
Good	16-AUG-24	Poor	300	400
Good	15-SEP-24	Poor	200	600
Good	30-SEP-24	Poor	300	800
Good	11-OCT-24	Poor	500	1300
Good	05-NOV-24	Poor	500	1800
Poor	13-MAR-24	Poor	100	100
Poor	09-MAY-24	Poor		100
Poor	05-JUN-24	Poor	300	400
Poor	11-SEP-24	Poor	100	100
Poor	17-OCT-24	Poor	300	400

```
26 rows selected.
A00325769_SQL>
```


15. List the Atmosphere, visit date, storage, and fine imposed, and what is the cumulative total of fines imposed at any time during the year before and the year after the individual under consideration (i.e. +365 days or -365 days) for each atmosphere category, for inspections where the storage condition was Poor.

Cl scr

Select Atmosphere, Visit_Date, Storage, Fine_Imposed,

Sum(Fine_Imposed) Over (Partition By Atmosphere Order By Visit_Date Range Between 365
Preceding And 365 Following) Fine_Imposed_Total

From Inspection

Where Storage = 'Poor'

Order By Atmosphere, Visit_Date;

```
Run SQL Command Line
A00325769_SQL>Select Atmosphere, Visit_Date, Storage, Fine_Imposed,
2 Sum(Fine_Imposed) Over (Partition By Atmosphere Order By Visit_Date Range Between 365 Preceding And 365 Following) Fine_Imposed_Total
3 From Inspection
4 Where Storage = 'Poor'
5 Order By Atmosphere, Visit_Date;
```

ATMOSPHER	VISIT_DAT	STORAGE	FINE_IMPOSED	FINE_IMPOSED_TOTAL
Excellent	12-DEC-23	Poor	200	700
Excellent	26-MAR-24	Poor		700
Excellent	15-APR-24	Poor	100	700
Excellent	23-APR-24	Poor		700
Excellent	17-JUL-24	Poor	100	700
Excellent	09-AUG-24	Poor	100	700
Excellent	10-AUG-24	Poor		700
Excellent	14-NOV-24	Poor	200	700
Good	05-DEC-23	Poor		3100
Good	19-DEC-23	Poor		3100
Good	19-JAN-24	Poor	500	3100
Good	22-JAN-24	Poor		3100
Good	11-FEB-24	Poor	500	3100
Good	10-MAR-24	Poor	200	3100
Good	30-MAR-24	Poor		3100
Good	17-JUN-24	Poor	100	3100
Good	16-AUG-24	Poor	300	3100
Good	15-SEP-24	Poor	200	3100
Good	30-SEP-24	Poor	300	3100
Good	11-OCT-24	Poor	500	3100
Good	05-NOV-24	Poor	500	3100
Poor	13-MAR-24	Poor	100	800
Poor	09-MAY-24	Poor		800
Poor	05-JUN-24	Poor	300	800
Poor	11-SEP-24	Poor	100	800
Poor	17-OCT-24	Poor	300	800

```
26 rows selected.
A00325769_SQL>
```

16. List the storage, outcome visit date recorded within a 100-day range for inspections that failed between January 1, 2024, and May 30, 2024, Arrange the results by visit date.

Cl scr

Select Storage, Visit_Date, Outcome, Visit_Date-100 Wind100_Dys,

First_Value(Storage) Over (Order By Visit_Date Asc Range 100 Preceding) Storage_1st,

First_Value(Visit_Date) Over (Order By Visit_Date Asc Range 100 Preceding) Vdate_1st

From Inspection

Where Outcome = 'Fail'

And Visit_Date Between '01-Jan-2024' And '30-May-2024'

Order By Visit_Date Asc;

```

Run SQL Command Line
A00325769_SQL>Select Storage, Visit_Date, Outcome, Visit_Date-100 Wind100_Dys,
 2 First_Value(Storage) Over (Order By Visit_Date Asc Range 100 Preceding) Storage_1st,
 3 First_Value(Visit_Date) Over (Order By Visit_Date Asc Range 100 Preceding) Vdate_1st
 4 From Inspection
 5 Where Outcome = 'Fail'
 6 And Visit_Date Between '01-Jan-2024' And '30-May-2024'
 7 Order By Visit_Date Asc;

STORAGE  VISIT_DAT  OUTC  WIND100_D  STORAGE_  VDATE_1ST
-----
Poor      19-JAN-24  Fail  11-OCT-23  Poor      19-JAN-24
Poor      22-JAN-24  Fail  14-OCT-23  Poor      19-JAN-24
Fair      25-JAN-24  Fail  17-OCT-23  Poor      19-JAN-24
Fair      04-FEB-24  Fail  27-OCT-23  Poor      19-JAN-24
Fair      06-FEB-24  Fail  29-OCT-23  Poor      19-JAN-24
Adequate  07-FEB-24  Fail  30-OCT-23  Poor      19-JAN-24
Fair      08-FEB-24  Fail  31-OCT-23  Poor      19-JAN-24
Poor      11-FEB-24  Fail  03-NOV-23  Poor      19-JAN-24
Fair      14-FEB-24  Fail  06-NOV-23  Poor      19-JAN-24
Poor      13-MAR-24  Fail  04-DEC-23  Poor      19-JAN-24
Adequate  15-MAR-24  Fail  06-DEC-23  Poor      19-JAN-24
Poor      30-MAR-24  Fail  21-DEC-23  Poor      19-JAN-24
Adequate  03-APR-24  Fail  25-DEC-23  Poor      19-JAN-24
Fair      08-MAY-24  Fail  29-JAN-24  Fair      04-FEB-24
Adequate  09-MAY-24  Fail  30-JAN-24  Fair      04-FEB-24
Poor      09-MAY-24  Fail  30-JAN-24  Fair      04-FEB-24
Adequate  28-MAY-24  Fail  18-FEB-24  Poor      13-MAR-24
Fair      28-MAY-24  Fail  18-FEB-24  Poor      13-MAR-24

18 rows selected.

A00325769_SQL>

```

17. List the Storage, visit date, outcome, and fines imposed 30 days before and 30 days after each failed inspection between February 1, 2024, and March 30, 2024, and outcomes is fail, arrange the results by visit date.

Cl scr

Select Storage, Visit_Date, Outcome, Visit_Date-30 "30DyEar", Visit_Date + 30 "30DyLst",

Avg(Fine_Imposed) Over (Order By Visit_Date Asc Range 30 Preceding) Av_Fine_Imposed_30Dy_Before,

Avg_Fine_Imposed_30Dy_Before,

Avg(Fine_Imposed) Over (Order By Visit_Date Desc Range 30 Preceding)

Avg_Fine_Imposed_30Dy_After From Inspection

Where Outcome = 'Fail'

And Visit_Date Between '01-Feb-2024' And '30-Mar-2024'

Order By Visit_Date;

```
Run SQL Command Line
A00325769_SQL>Select Storage, Visit_Date, Outcome, Visit_Date-30 "30DyEar", Visit_Date + 30 "30DyLst",
2 Avg(Fine_Imposed) Over (Order By Visit_Date Asc Range 30 Preceding) Av_Fine_Imposed_30Dy_Before,
3 Avg(Fine_Imposed) Over (Order By Visit_Date Desc Range 30 Preceding) Av_Fine_Imposed_30Dy_After From Inspection
4 Where Outcome = 'Fail'
5 And Visit_Date Between '01-Feb-2024' And '30-Mar-2024'
6 Order By Visit_Date;

STORAGE VISIT_DAT OUTC 30DyEar 30DyLst AV_FINE_IMPOSED_30DY_BEFORE AV_FINE_IMPOSED_30DY_AFTER
-----
Fair 04-FEB-24 Fail 05-JAN-24 05-MAR-24 500 340
Fair 06-FEB-24 Fail 07-JAN-24 07-MAR-24 300 300
Adequate 07-FEB-24 Fail 08-JAN-24 08-MAR-24 366.666667 366.666667
Fair 08-FEB-24 Fail 09-JAN-24 09-MAR-24 366.666667 300
Poor 11-FEB-24 Fail 12-JAN-24 12-MAR-24 400 300
Fair 14-FEB-24 Fail 15-JAN-24 15-MAR-24 340 233.333333
Poor 13-MAR-24 Fail 12-FEB-24 12-APR-24 100 300
Adequate 15-MAR-24 Fail 14-FEB-24 14-APR-24 233.333333 500
Poor 30-MAR-24 Fail 29-FEB-24 29-APR-24 300

9 rows selected.

A00325769_SQL>
A00325769_SQL>
```

18. List the name, town, hotel id, county and cost from hotel and the total and average costs for hotels specializing in weddings, grouped by hotel, town, and county, arrange the results in hotel id.

Cl scr

Select Name, Town, Hotel_Id, County, Cost,

Sum (Cost) Over (Partition By Hotel_Id) Hotel_Id_Total,

Avg (Cost) Over (Partition By Town) Town_Total,

Sum (Cost) Over (Partition By County) County_Total

From Hotel

Where Specialism = 'Weddings'

Order By Hotel_Id;

Run SQL Command Line							
<pre>A00325769_SQL>Select Name, Town, Hotel_Id, County, Cost, 2 Sum (Cost) Over (Partition By Hotel_Id) Hotel_Id_Total, 3 Avg (Cost) Over (Partition By Town) Town_Total, 4 Sum (Cost) Over (Partition By County) County_Total 5 From Hotel 6 Where Specialism = 'Weddings' 7 Order By Hotel_Id;</pre>							
NAME	TOWN	HOTEL_ID	COUNTY	COST	HOTEL_ID_TOTAL	TOWN_TOTAL	COUNTY_TOTAL
Step House Hotel	Graiguenamanagh	100	Carlow	171	171	171	252
Woodford Dolmen Hotel	Clonegal	101	Carlow	81	81	81	252
Castlemartyr Resort	Castlemartyr	108	Cork	221	221	221	221
Red Castel Hotel	Castlefin	110	Donegal	105	105	105	323
Lough Eske Castel	Greencastle	112	Donegal	218	218	218	323
Clontarf Castle Hotel	Dublin City	115	Dublin	141	141	141	587
The Shelbourne Autograph Collection	Dublin	117	Dublin	446	446	446	587
Roganstown Hotel	Garristown	118	Fingal	130	130	130	130
Galway Bay Hotel	Glenamaddy	120	Galway	166	166	166	523
The Harbour Hotel Galway	Cloonboo	122	Galway	170	170	170	523
Salthill Hotel	Athenry	124	Galway	187	187	187	523
Sheraton Athlone Hotel	Castlepollard	125	Westmeath	181	181	181	484
Athlone Springs Hotel	Athlone	127	Westmeath	141	141	151.5	484
Radisson Blu Hotel	Athlone	129	Westmeath	162	162	151.5	484
14 rows selected.							
A00325769_SQL>							

19. list the hotel name, town, cost, grade and number of for all the hotels with a Grade of 4 Star and at least 20 bedrooms, along with a running total of cost across all hotels, the total cost by town, and a sequential number for each hotel within the same town, sorted by town and name.

Cl scr

Break On Town Skip 1

Select Name, Town, Cost, Grade, Num_Bedrooms,

Sum(Cost) Over

(Order By Town, Name) Run_Total_All_Hotels,

Sum(Cost) Over

(Partition By Town) Total_By_Town,

Row_Number() Over

(Partition By Town Order By Name) Seq

From Hotel

Where Grade = '4 Star'

And Num_Bedrooms >= 20

Order By Town, Name;

Run SQL Command Line							
A00325769_SQL>Break On Town Skip 1							
A00325769_SQL>							
A00325769_SQL>Select							
2	Name, Town, Cost, Grade, Num_Bedrooms,						
3	Sum(Cost) Over						
4	(Order By Town, Name) Run_Total_All_Hotels,						
5	Sum(Cost) Over						
6	(Partition By Town) Total_By_Town,						
7	Row_Number() Over						
8	(Partition By Town Order By Name) Seq						
9	From Hotel						
10	Where						
11	Grade = '4 Star'						
12	And Num_Bedrooms >= 20						
13	Order By Town, Name;						
NAME	TOWN	COST	GRADE	NUM_BEDROOMS	RUN_TOTAL_ALL_HOTELS	TOTAL_BY_TOWN	SEQ
Salthill Hotel	Athenry	187	4 Star	20	187	187	1
Athlone Springs Hotel	Athlone	141	4 Star	20	328	303	1
Radisson Blu Hotel		162	4 Star	30	490	303	2
Lord Bengal Inn	Ballon	129	4 Star	39	619	129	1
Clayton Hotel Ballsbridge	Brittas	124	4 Star	22	743	124	1
Woodford Dolmen Hotel	Clonegal	81	4 Star	81	824	81	1
Hodson Bay Hotel	Clonmellon	136	4 Star	25	960	136	1
The Harbour Hotel Galway	Cloonboo	170	4 Star	20	1130	170	1
Glasson Lakehouse	Collinstown	272	4 Star	50	1402	272	1
Roganstown Hotel	Garristown	130	4 Star	20	1532	130	1
Galway Bay Hotel	Glenamaddy	166	4 Star	25	1698	166	1
Step House Hotel	Graiguenamanagh	171	4 Star	20	1869	171	1
12 rows selected.							
A00325769_SQL>							

20. list the Inspection id, hotel id and visit date for the inspections where the Overall Score is greater than 80 and imposed a Fine Imposed. Additionally, display the first Outcome and Visit Date from the previous 5 rows ordered by Visit Date, and sort the results by Visit Date in ascending order.

Cl scr

Select Inspection_Id, Hotel_Id, Visit_Date,

FIRST_VALUE(Outcome)

OVER (ORDER BY Visit_Date ASC ROWS 5 PRECEDING) AS Outcome_Prec,

FIRST_VALUE(Visit_Date)

OVER (ORDER BY Visit_Date ASC ROWS 5 PRECEDING) AS VisitDate_Prec

FROM

Inspection

WHERE

Overall_Score > 80

AND Fine_Imposed > 0

ORDER BY

Visit_Date ASC;

```

Run SQL Command Line
A00325769_SQL>Select Inspection_Id, Hotel_Id, Visit_Date,
2     FIRST_VALUE(Outcome)
3     OVER (ORDER BY Visit_Date ASC ROWS 5 PRECEDING) AS Outcome_Prec,
4     FIRST_VALUE(Visit_Date)
5     OVER (ORDER BY Visit_Date ASC ROWS 5 PRECEDING) AS VisitDate_Prec
6 FROM
7     Inspection
8 WHERE
9     Overall_Score > 80
10    AND Fine_Imposed > 0
11 ORDER BY
12    Visit_Date ASC;

INSPECTION_ID  HOTEL_ID  VISIT_DAT  OUTC  VISITDATE
-----
7145           118  19-JAN-24  Fail  19-JAN-24
9781           123  04-FEB-24  Fail  19-JAN-24
6994           115  06-FEB-24  Fail  19-JAN-24
7373           114  07-FEB-24  Fail  19-JAN-24
5059           123  11-FEB-24  Fail  19-JAN-24
5029           104  10-MAR-24  Fail  19-JAN-24
5199           125  15-MAR-24  Fail  04-FEB-24
6830           103  23-MAR-24  Fail  06-FEB-24
5274           109  17-JUN-24  Fail  07-FEB-24
3935           100  17-JUL-24  Fail  11-FEB-24
6865           122  16-AUG-24  Pass  10-MAR-24
6115           127  11-OCT-24  Fail  15-MAR-24
7465           112  27-OCT-24  Pass  23-MAR-24
3532           112  05-NOV-24  Pass  17-JUN-24
6261           106  14-NOV-24  Fail  17-JUL-24

15 rows selected.
A00325769_SQL>

```

21. List the name, county and the year open of the hotels. Calculate the cumulative total of year opened for all hotels in the location of cork, sort the results by county and name.

Cl scr

```
Select Name, County, Year_Opened,  
        Sum(Year_Opened) Over  
          (Partition By County  
           Order By Name)County_Total
```

From Hotel

Where County = 'Cork'

Order By County, Name;

```
A00325769_SQL>Select Name, County, Year_Opened,  
2          Sum(Year_Opened) Over  
3            (Partition By County  
4              Order By Name)County_Total  
5 From Hotel  
6 Where County = 'Cork'  
7 Order By County, Name;
```

NAME	COUNTY	YEAR_OPENED	COUNTY_TOTAL
Castlemartyr Resort	Cork	2008	2008
Midleton Park Hotel	Cork	2023	4031
River Island Hotel	Cork	1995	6026
The Waterfront Hotel	Cork	1960	7986
The West Cork Hotel	Cork	1902	9888

5 rows selected.

A00325769_SQL>

22. List the name, county, cost and the year open of each hotel. Insert the new column called running total that include cost of all hotels that located in the same or earlier county sort it by hotel name and county. And insert the new column called county hotel count where the hotels located in same county sort this by using hotel name, filter the hotels that located in the county called cork and Dublin sort the results with county and hotel name.

Cl scr

Select Name, County, Cost, Year_Opened,

(Select Sum(Cost)

From Hotel H2

Where H2.County < Hotel.County

Or(H2.County = Hotel.County AND H2.Name <= Hotel.Name))

Running_Total,

(Select Sum(Cost)

From Hotel H3

Where H3.County = Hotel.County

And H3.Name <= Hotel.Name)

County_Hotel_Count

From Hotel

Where County In ('Cork', 'Dublin')

ORDER BY County, Name;

```
Run SQL Command Line
A00325769_SQL>Select Name, County, Cost, Year_Opened,
2      (Select Sum(Cost)
3      From Hotel H2
4      Where H2.County < Hotel.County
5      Or(H2.County = Hotel.County AND H2.Name <= Hotel.Name))
6      Running_Total,
7      (Select Sum(Cost)
8      From Hotel H3
9      Where H3.County = Hotel.County
10     And H3.Name <= Hotel.Name)
11     County_Hotel_Count
12 From Hotel
13 Where County In ('Cork', 'Dublin')
14 ORDER BY County, Name;
```

NAME	COUNTY	COST	YEAR_OPENED	RUNNING_TOTAL	COUNTY_HOTEL_COUNT
Castlemartyr Resort	Cork	221	2008	780	221
Midleton Park Hotel	Cork	116	2023	896	337
River Island Hotel	Cork	69	1995	965	406
The Waterfront Hotel	Cork	124	1960	1089	530
The West Cork Hotel	Cork	35	1902	1124	565
Clayton Hotel Ballsbridge	Dublin	124	2004	2120	124
Clontarf Castle Hotel	Dublin	141	1998	2261	265
The Shelbourne Autograph Collection	Dublin	446	1998	2707	711

```
8 rows selected.
A00325769_SQL>
```


23. list the name and average cost of the hotels with multiple hotel names

Cl scr

Select Avg(Cost)

From Hotel

Where Name = 'Step House Hotel';

```
Run SQL Command Line
A00325769_SQL>Select Avg(Cost)
  2 From Hotel
  3 Where Name = 'Step House Hotel';

AVG(COST)
-----
      171

1 row selected.
A00325769_SQL>
```

Cl scr

Select Avg(Cost)

From Hotel

Where Name In ('Step House Hotel', 'Woodford Dolmen Hotel');

```
Run SQL Command Line
A00325769_SQL>Select Avg(Cost)
  2 From Hotel
  3 Where Name In ('Step House Hotel', 'Woodford Dolmen Hotel');

AVG(COST)
-----
      126

1 row selected.
A00325769_SQL>
```

Cl scr

Select Avg(Cost)

From Hotel

Where Name In ('Step House Hotel', 'Woodford Dolmen Hotel', 'Borrow View Bread and Breakfast', 'Talbot');

```
Run SQL Command Line
A00325769_SQL>Select Avg(Cost)
  2 From Hotel
  3 Where Name In ('Step House Hotel', 'Woodford Dolmen Hotel', 'Borrow View Bread and Breakfast', 'Talbot');

AVG(COST)
-----
     107.5

1 row selected.
A00325769_SQL>
```

You tube Links

Video 1: <https://youtu.be/O-3PR-J7ZMA?si=766Kgf26OYpwF7Lg>

Video 2 : https://youtu.be/nfUINh1_9mc?si=jkrtA_Y40TEs_4xC