



# WEEK #6

# ASSIGNMENT

Part of 10% Individual Assignment

Half-Way There Assignment

Mark Morell

Database Management – Fall 2019

Submitted By: Aparna Tomar(7802150)

## Assignment Questions

**NOTE:** Use the Costume database to answer *all* of these questions

Question #	Question	Score																									
1	<p>Write a single query to show a list of costumes that have been rented within the last 2 weeks. In your query include the following information:</p> <ul style="list-style-type: none"><li>• Costume rented</li><li>• Name of person who rented the costume (formatted nicely)</li><li>• Date rented (just the date without a time)</li></ul> <p>Sort the result with the latest rental first</p>	4																									
2	<p>Describe what NULL represents related to databases. Compare it to other possible values that could be used in a database.</p>	4																									
3	<p>What T-SQL data type would you use to store the following information and why would you use it?</p> <ul style="list-style-type: none"><li>a) A value that can be either true or false</li><li>b) A value that's a very large number used for scientific purposes</li><li>c) The age of a person</li><li>d) The cost of items at a supermarket</li><li>e) Values to represent when the next solar eclipse will occur</li><li>f) A list of ingredients for pumpkin pie</li><li>g) A Word document</li></ul>	7																									
4	<p>Answer these questions using the following partial table of dog breeds:</p> <table><tr><th>id</th><th>name</th><th>size</th><th>colour</th><th>origin</th></tr><tr><td>1</td><td>Labrador Retriever</td><td>Large</td><td>Black, Brown, Golden</td><td>Canada, UK</td></tr><tr><td>2</td><td>Bulldog</td><td>Medium</td><td>Brown, White</td><td>UK</td></tr><tr><td>3</td><td>German Shepherd</td><td>Large</td><td>Brown, Black</td><td>Germany</td></tr><tr><td>4</td><td>Rajapalayam</td><td>Large</td><td>White</td><td>India</td></tr></table> <ul style="list-style-type: none"><li>a) What are the various candidate keys for this table?</li><li>b) Are each of these keys natural keys or surrogate keys?</li><li>c) What would make the best choice for the primary key and why?</li></ul>	id	name	size	colour	origin	1	Labrador Retriever	Large	Black, Brown, Golden	Canada, UK	2	Bulldog	Medium	Brown, White	UK	3	German Shepherd	Large	Brown, Black	Germany	4	Rajapalayam	Large	White	India	6
id	name	size	colour	origin																							
1	Labrador Retriever	Large	Black, Brown, Golden	Canada, UK																							
2	Bulldog	Medium	Brown, White	UK																							
3	German Shepherd	Large	Brown, Black	Germany																							
4	Rajapalayam	Large	White	India																							
5	<p>Describe what a database foreign key is and give a real-life example of one.</p>	4																									
6	<p>What will the following query retrieve?</p> <pre>SELECT costumeName FROM dbo.Costume WHERE id IN ( SELECT costumeId               FROM dbo.CostumeRental               WHERE returnedDate IS NOT NULL                 AND returnDamaged = 1 )</pre>	4																									
7	<p>Re-write the following query using an IN statement:</p> <pre>SELECT costumeName FROM dbo.Costume WHERE colour = 'Brown'       OR colour = 'Black'       OR colour = 'Green'       OR colour = 'white'</pre>	2																									

<b>8</b>	<p>Write a query to retrieve the <u>top 60%</u> of all costumes that have a mask in their costume description. Provide the following information:</p> <ul style="list-style-type: none"> <li>• Costume Name</li> <li>• Costume Description</li> <li>• Costume Colour</li> <li>• Number of pieces</li> <li>• Number currently in stock</li> <li>• The date it was last rented</li> </ul> <p>Order the single set of results by costume name. Note that there may be costumes that have not been rented so take that into account</p>	6
<b>9</b>	Describe 5 traits of good SQL coding style	5
<b>10</b>	Write a query that simply gives us (from a single query) the total number of costumes that are in-stock, the total number that are on-order and the total number that have been lost. Be sure to include appropriate column heading names in your query.	3
<b>11</b>	Write a query to update the costume rental information. Rachel Scruff just now returned the costume that she had rented and it is damaged. Include a comment that she claims it was damaged before she rented it. In your results show your update statement and then write a SELECT statement to show that the data has been updated in the table.	4
<b>12</b>	We need to remove all of the data from a table in the database that has lots and lots of data. What's the quickest way to clear it all out so that no rollback log is created?	2
<b>13</b>	Write a single query that retrieves the costumes names and how many times that they've been rented as shown in the CostumeRental table. Sort the results by the number of times that they've been rented and then secondarily within the same query by costume name. Note that this is a bit tricky so a) Start building your SQL in steps to help get you what you need and b) don't forget to include costumes that have never been rented. This should result in 12 rows being returned.	6
<b>14</b>	<p>Write a query that would create a brand new table of costumes named "ScaryCostumes" that include only those that are rated at "Scary" or above on the scare-factor. Include the following information in the new table:</p> <ul style="list-style-type: none"> <li>• Costume Name</li> <li>• Costume Description</li> <li>• UPC Code</li> </ul>	3
	<b>Total</b>	<b>60</b>
	<b>Total From In-Class Component</b>	<b>20</b>
	<b>Grand Total</b>	<b>80</b>

Answer1:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'AdventureWorks2017', including tables like 'dbo.Costume', 'dbo.CostumeInventory', 'dbo.CostumeRental', 'dbo.Country', 'dbo.Province', and 'dbo.ScareRating'. The main window shows a query titled 'Q1\_ASSIGNMENTS.s...Costume (sa (53))'. The query is as follows:

```
1 /*
2 Write a single query to show a list of costumes that have been rented within the last 2 weeks.
3 In your query include the following information:
4 • Costume rented
5 • Name of person who rented the costume (formatted nicely)
6 • Date rented (just the date without a time)
7 Sort the result with the latest rental first
8 */
9
10 --Review of participating tables
11
12 --Query
13 Use Costume
14 Go
15 Select
16 c.costumeName,
17 cr.rentedToFirstName + ' ' + cr.rentedToLastName as customerName,
18 (Select Cast(cr.rentedDate as date)) as 'rentedOn'
19 From CostumeRental cr
20 INNER JOIN Costume c ON c.id = cr.costumeId
21 Where (cr.rentedDate) >= GETDATE() - 14
22 Order By 'rentedOn' DESC;
23
24
25
```

Below the query, the 'Results' tab shows the output of the query:

	costumeName	customerName	rentedOn
1	Frankenstein	Billy Orwell	2019-10-09
2	Sunflower	Andrea Yan	2019-10-09
3	Dracula	Ivory Fang	2019-10-07

Answer2:

**Missing Value:** In the world of database, **NULL** handles the concept of missing values in a record. It is used to signify missing or unknown values.

In day to day life, there are many situations when data for a field in a record is not available may be because user did not provide the information, or the information just does not exist.

NULL is tricky as no two NULLs are equal. It makes sense as if SQL start making two NULL equal then it will produce unwanted results in which all NULLs are same which would have been disastrous.

Because Two NULLs are never equal, therefore in order to test if field's value is equal to NULL then it should be done using IS NULL and IS NOT NULL operators provided in SQL instead of '=' or '!= '.

IS NULL and IS NOT NULL operators knows how to deal with missing values a.k.a. NULL value.

Answer3:

- Bit:** As only true or false value need to be stored. Therefore, BIT data type is recommended.
- Float:** Scientific numbers generally are real values. That is why Float data type is recommended.
- Small Int:** As age of Sapiens is expected to be around 120 yrs at most.

- d. **Small Money:** In RDBMS, any amount which represent price represented in local currency can be handled nicely by using Money data types.
- e. **Datetime:** In order to provide complete information about timing of solar eclipse, you will need date as well as time. Therefore, datetime.
- f. **NVarchar:** List of ingredients usually are limited and could also contain characters from Non-ASCII set of numbers such as special characters. That is why NVarchar is recommended.
- g. **NText:** A word document could be very long and can contain special characters and accents. Therefore, T-SQL data type NText is best suitable for this job.

#### Answer4:

- a. **Candidate Keys:** Except ID and Name columns, all other columns have low-cardinality levels therefore unfit for candidate key.
  - a. **Id: Simple, System generated**
  - b. **Name + Origin:** Breed name generally are Unique but in order to remove ambiguity, therefore **Name + Origin** is being recommended are Composite Candidate Key.
- b. **Candidate keys have both surrogate keys and natural keys.**
  - a. **ID is a surrogate key**
  - b. **Name + Origin: Composite Primary Key**
- c. **ID** because it is numeric and simple.

#### Answer5:

**Foreign Key:** Concept of foreign keys comes when we need to retrieve information by joining more than one tables. In those scenarios, **Primary Key of a table A** becomes **foreign key of another table B**, if table B would like to use records of table A.

OR

When one or more columns in a table refer to the primary key in another table. Foreign Key a.k.a. **Referencing Key**. The table containing the foreign key is called the child table.

#### Example:

**Let's take two tables for example, one representing Student while other represent course taken by the student.**

The "StudentID" columns in the "Student" table is the PRIMARY KEY in the "Student" table.

The "StudentID" column in the "CourseEnrolled" table is a FOREIGN KEY in the "CourseEnrolled" table.

The FOREIGN KEY constraint makes sure that LINK between 'Student' table and 'CourseEnrolled' table is healthy and data 'CourseEnrolled' table is not stale or corrupted.

The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it must be one of the values contained in the table it points to.

StudentID	LastName	FirstName	Age
780554	Zhang	Larry	30
785468	Hansen	Tom	24
568797	Malhotra	Rohan	26

**Table: Student**

OrderNumber	StudentID	CourseTaken
1	780554	Mobile Application Development
2	780554	Web Technologies
3	785468	Programming Database

**Table: CourseEnrolled**

**Answer6:**

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'AdventureWorks2017', including tables like 'Costume', 'CostumeInventory', 'CostumeRental', 'Country', 'Province', and 'ScareRating'. The main window displays a SQL query in the 'Query Editor' tab, titled 'Q6\_ASSIGNMENT5.s...Costume (sa (54))'. The query is as follows:

```

1  /*
2  1. Fixed coloum name: returnDamaged --> returnedDamaged
3  2. Ran the given Query
4  3. This query provides names of thoses Costumes Which:
5     - Was rented
6     - Was returned
7     - Is Damaged
8  4. This query helps management to do things:
9     - Fix the damaged dresses
10    - Make dress unavailable for rental
11  5. Result: Zombie & Ghost costumes are damaged.
12  */
13
14  Use Costume
15  SELECT costumeName as 'DamagedCostumes'
16  FROM dbo.Costume
17  WHERE id IN ( SELECT costumeId
18                FROM dbo.CostumeRental
19                WHERE returnedDate IS NOT NULL
20                  AND returnedDamaged = 1 )
21

```

Below the query editor, the 'Results' tab is active, showing the output of the query. The results are displayed in a table with two columns: 'DamagedCostumes' and 'id'. The results are as follows:

DamagedCostumes	id
Zombie	1
Ghost	2

## Answer7:

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'AdventureWorks2017', with the 'Costume' table expanded under 'dbo'. The central pane shows a SQL query in the 'Q7\_ASSIGNMENTS5.s...Costume (sa (52))' window. The query is as follows:

```
1 USE Costume
2 GO
3 SELECT costumeName
4 FROM dbo.Costume
5 WHERE colour IN ( 'Brown', 'Black', 'Green', 'White');
6
7
```

At the bottom, the 'Results' tab is active, showing a table with 8 rows of data. The first row is highlighted.

	costumeName
1	Dracula
2	Frankenstein
3	Zombie
4	Mummy
5	Soldier
6	Candy Bar
7	Witchdoctor
8	Ghost

#### Answer8:

```
/*
Write a query to retrieve the top 60% of all costumes that have a mask in their costume description.
Provide the following information:
• Costume Name
• Costume Description
• Costume Colour
• Number of pieces
• Number currently in stock
• The date it was last rented
Order the single set of results by costume name.
Note that there may costumes that have not been rented so take that into account
*/
Use
Costume
SELECT DISTINCT TOP 60 PERCENT
c.costumeName,
c.costumeDescription,
c.colour,
c.numPieces,
(select CostumeInventory.numberInStock From CostumeInventory
where c.id = CostumeInventory.costumeId) as numInStock,
(Cast(cr.rentedDate as date)) as 'dateLastRented'
From Costume c
LEFT OUTER JOIN CostumeRental cr ON c.id = cr.costumeId
Where c.costumeDescription like '%[m-M]ask%'
ORDER BY c.costumeName ASC;
```

100 %

Results Messages

	costumeName	costumeDescription	colour	numPieces	numInStock	dateLastRented
1	Freddy	Mask, hat and fingers	Red	3	3	NULL
2	Witchdoctor	Mask, staff, wig	Black	3	3	2018-01-04
3	Witchdoctor	Mask, staff, wig	Black	3	3	2019-10-05
4	Zombie	Tattered clothing, mask	Brown	2	17	2018-11-23
5	Zombie	Tattered clothing, mask	Brown	2	17	2019-08-31

**Answer 9:** Software companies now a days spends huge time and energy to templatize coding styles with in the company. Recent research done by Basalaj et. al. (Basalaj, W., & van den Beuken, F. (2006)). proved both qualitative and quantitatively the direct relationship between Coding Standards Compliance and Quality of the software produced.

Principle of coding standards extends to the world of T-SQL as well and few recommended coding standards of SQL are:

1. When writing query, always write DB name in the beginning of query for example:

**Use dbo.Student**

**Go**



2. Always comment you SQL where it is appropriate. Brief multiline description of what the query does is expected.
3. Do not use Alias names for the columns which has space in its name. For instance:
  - a. "FirstName" → Recommended
  - b. "First Name" → Not Recommended
4. Use "camelCase" identifiers.
5. Use uppercase SQL keywords and functions.
6. Use meaningful names and identifiers (singular nouns).
7. Breakdown Query into multiple lines.

Answer10:

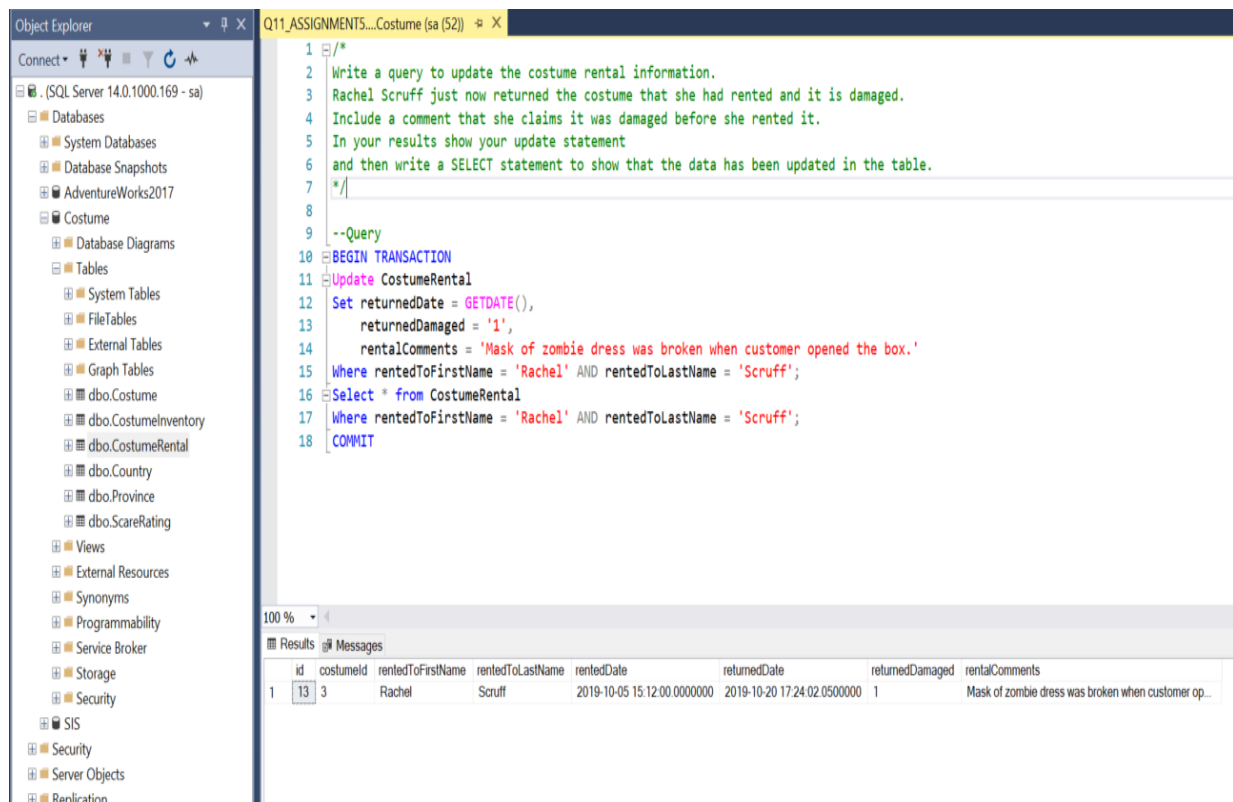
The screenshot displays the SQL Server Enterprise Manager interface. On the left, the Object Explorer shows the database structure for 'AdventureWorks2017', including tables like 'CostumeInventory' and 'CostumeRental'. The main window shows a SQL query in the 'Query Editor' tab, titled 'Q10\_ASSIGNMENT5...Costume (sa (54))'. The query is a SELECT statement that calculates the total number of costumes in stock, on order, and lost. The query is as follows:

```
1  /*
2  Write a query that simply gives us (from a single query)
3  the total number of costumes that are in-stock,
4  the total number that are on-order
5  and the total number that have been lost.
6  Be sure to include appropriate column heading names in your query.
7  */
8
9  Use
10 Costume
11 SELECT SUM(ci.numberInStock) AS TotalCostumesInStock,
12 SUM(ci.numberOnOrder) AS TotalCostumesOnOrder,
13 SUM(ci.numberLost) AS TotalCostumesOnOrder
14 FROM CostumeInventory ci;
15
```

Below the query editor, the 'Results' tab shows the output of the query. The results are displayed in a table with three columns: 'TotalCostumesInStock', 'TotalCostumesOnOrder', and 'TotalCostumesOnOrder'. The first row shows the values 122, 77, and 30 respectively.

	TotalCostumesInStock	TotalCostumesOnOrder	TotalCostumesOnOrder
1	122	77	30

### Answer11:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'AdventureWorks2017', specifically the 'Costume' database and its 'CostumeRental' table. The main pane shows a SQL query being executed in the 'Q11\_ASSIGNMENTS...Costume (sa (52))' window. The query is as follows:

```
1 /*
2 Write a query to update the costume rental information.
3 Rachel Scruff just now returned the costume that she had rented and it is damaged.
4 Include a comment that she claims it was damaged before she rented it.
5 In your results show your update statement
6 and then write a SELECT statement to show that the data has been updated in the table.
7 */
8
9 --Query
10 BEGIN TRANSACTION
11 Update CostumeRental
12 Set returnedDate = GETDATE(),
13    returnedDamaged = '1',
14    rentalComments = 'Mask of zombie dress was broken when customer opened the box.'
15 Where rentedToFirstName = 'Rachel' AND rentedToLastName = 'Scruff';
16 Select * from CostumeRental
17 Where rentedToFirstName = 'Rachel' AND rentedToLastName = 'Scruff';
18 COMMIT
```

The Results pane at the bottom shows the output of the query, displaying the updated record in the 'CostumeRental' table:

	id	costumeId	rentedToFirstName	rentedToLastName	rentedDate	returnedDate	returnedDamaged	rentalComments
1	13	3	Rachel	Scruff	2019-10-05 15:12:00.0000000	2019-10-20 17:24:02.0500000	1	Mask of zombie dress was broken when customer op...

### Answer12:

Quickest way to clear a table with no rollback log is TRUNCATE.

TRUNCATE is faster than DELETE.

TRUNCATE is not Logged.

TRUNCATE perform deletion on each row of the table. No condition clause allowed.

### Answer13:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the server structure for 'CI070000000681 (SQL Server 14.0.1000.16)'. The 'Databases' folder is expanded, showing 'AdventureWorks2017' and 'Costume'. The 'Costume' database is selected, and its 'Tables' folder is expanded, showing 'CostumeRental'. The main pane displays a SQL query in a text editor. The query is as follows:

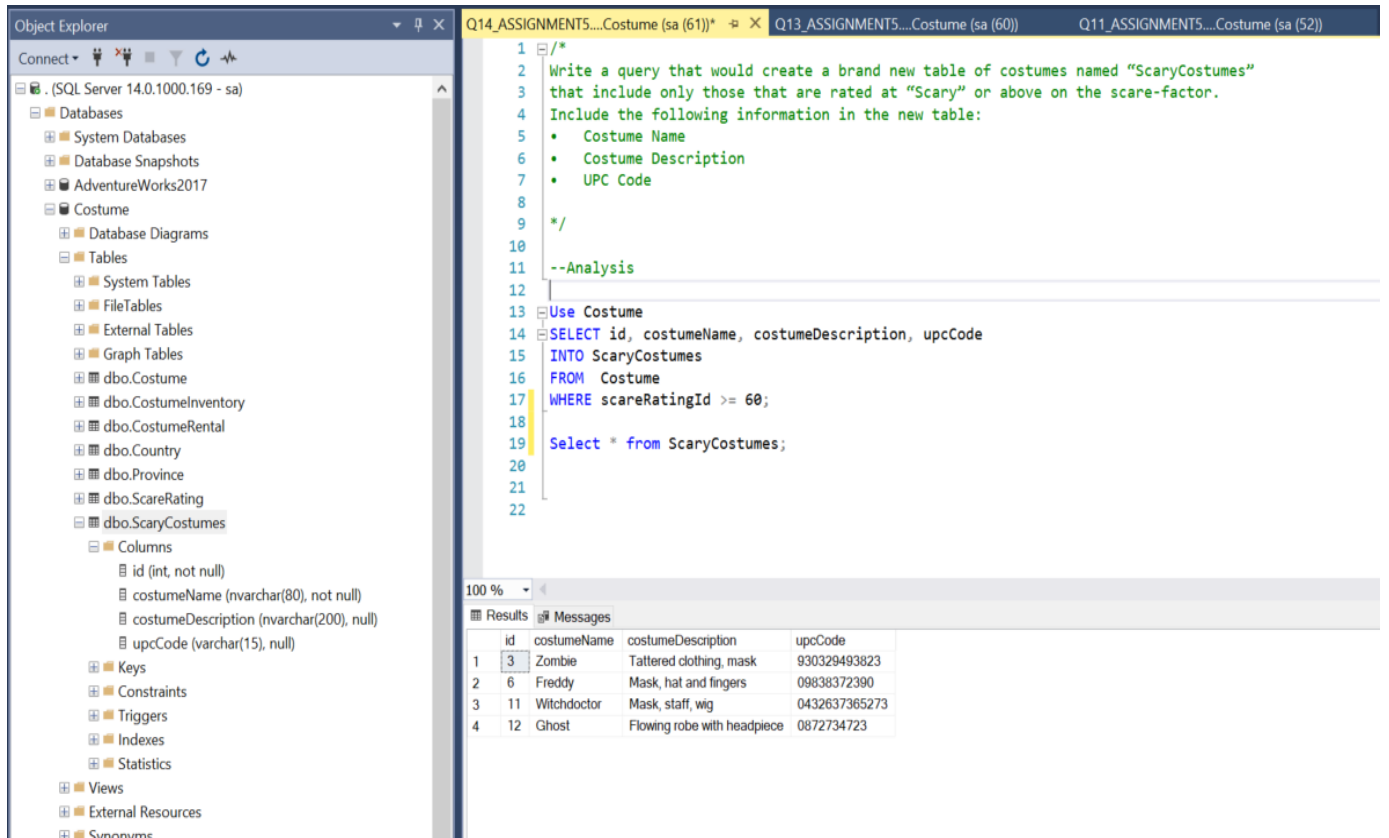
```
/*
Write a single query that retrieves the costumes names and how many times
that they've been rented as shown in the CostumeRental table.
Sort the results by the number of times that they've been rented and
then secondarily within the same query by costume name.
Note that this is a bit tricky so
a)
Start building your SQL in steps to help get you what you need and
b) don't forget to include costumes that have never been rented.
This should result in 12 rows being returned.
*/

Select c.costumeName,
COUNT(cr.costumeId) as numberOfTimesRented
From Costume c
LEFT OUTER JOIN CostumeRental cr ON c.id = cr.costumeId
GROUP BY c.costumeName
ORDER BY numberOfTimesRented DESC, c.costumeName
```

Below the query editor, the 'Results' tab is active, showing the query results in a table. The table has two columns: 'costumeName' and 'numberOfTimesRented'. The results are sorted by 'numberOfTimesRented' in descending order, and then by 'costumeName' in ascending order. The results are as follows:

	costumeName	numberOfTimesRented
1	Zombie	6
2	Dracula	3
3	Ghost	3
4	Frankenstein	2
5	Sunflower	2
6	Witchdoctor	2
7	Clown	1
8	Mountie	1
9	Mummy	1
10	Soldier	1
11	Candy Bar	0
12	Freddy	0

## Answer14:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'AdventureWorks2017', including tables like 'Costume', 'CostumeInventory', 'CostumeRental', 'Country', 'Province', 'ScareRating', and 'ScaryCostumes'. The 'ScaryCostumes' table is highlighted, showing its columns: 'id' (int, not null), 'costumeName' (nvarchar(80), not null), 'costumeDescription' (nvarchar(200), null), and 'upcCode' (varchar(15), null).

The main window displays a SQL query in the 'Query Editor' pane. The query is as follows:

```
1  /*
2  Write a query that would create a brand new table of costumes named "ScaryCostumes"
3  that include only those that are rated at "Scary" or above on the scare-factor.
4  Include the following information in the new table:
5  • Costume Name
6  • Costume Description
7  • UPC Code
8
9  */
10
11 --Analysis
12
13 Use Costume
14 SELECT id, costumeName, costumeDescription, upcCode
15 INTO ScaryCostumes
16 FROM Costume
17 WHERE scareRatingId >= 60;
18
19 Select * from ScaryCostumes;
20
21
22
```

The query is executed, and the results are displayed in the 'Results' pane. The results show a table with 4 rows and 4 columns: 'id', 'costumeName', 'costumeDescription', and 'upcCode'.

	id	costumeName	costumeDescription	upcCode
1	3	Zombie	Tattered clothing, mask	930329493823
2	6	Freddy	Mask, hat and fingers	09838372390
3	11	Witchdoctor	Mask, staff, wig	0432637365273
4	12	Ghost	Flowing robe with headpiece	0872734723

## References:

- Basalaj, W., & van den Beuken, F. (2006). Correlation between coding standards compliance and software quality. *White paper, Programming Research Ltd.*