1)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:

Costume rented

Name of person who rented the costume (formatted nicely)

Date rented (just the date without a time)

Sort the result with the latest rental first

A screenshot of a computer screen

Description automatically generated

2) Describe what NULL represents related to databases. Compare it to other

possible values that could be used in a database.

NULL illustrators vacant value in shows that specific data will not have any existance in the database.Null value is different than zero or any value that contains space.It is more liek we do not know the data at all for that field.Null value is field that appers to be blank in the table.We can use IS NULL and IS NOT NULL to check the null value.

Null shows that the value is not known. And, two null values can not be equal. To cite an illustation,

SELECT CustomerID,

CustomerFirstName,

CustomerLastName

FROM Customer.CustomerDetails

WHERE CustomerID IS NULL

In this example it will only show Null value.

SELECT CustomerID,

CustomerFirstName,

CustomerLastName

FROM Customer.CustomerDetails

WHERE CustomerID IS Not NULL

It will give output as all CustomerID which does not have any null value

3)What T-SQL data type would you use to store the following information and

why would you use it?

a) A value that can be either true or false.

=> A Boolean value use for either true or false. It shows value in 0 or 1 where 0 means false and 1 means true. It is used because of logical values.

b) A value that’s a very large number used for scientific purposes

=> Float can be perfect suitable for this one because scientific number is in point value.

c)The age of a person

=> We can use any int but small int is very suitable for this because it does not take us more space and only store data between 0 to 255.

d)The cost of items at a supermarket

=> cost will be like currency and we use decimal for this one. However, you can use numeric as well.

e) Values to represent when the next solar eclipse will occur

=> We will use date in this one to show on what date what time it will occur next time.

f) A list of ingredients for pumpkin pie

=> I will prefer text for this because it can store data till 255 also it stores both text and number.

g) A Word documents

=> We should use varvinary(MAX) because we need to convert data into binary so this is the best option.

4) A screenshot of a cell phone

Description automatically generated

a) What are the various candidate keys for this table?

1)Id, 2) Name, 3) Size 4) Colour

b) Are each of these keys’ natural keys or surrogate keys?

1)Id Surrogate 2) Name Natural 3) size Natural 4) colour Natural

5)Origin Natural

c)What would make the best choice for the primary key and why?

=> In my point of view, I feel like ID makes better primary key then others because primary key should be unique and, in this table, you can see ID is unique than others. Thus, it will be easy to get identify data as well.

5) Describe what a database foreign key is and give a real-life example of one.

=> A foreign key provides reference to another table key mostly it’s primary key, but it can also give reference to unique key.

It is useful for maintain referential integrity.

6) What will the following query retrieve?

A screenshot of a social media post

Description automatically generated

We will get this error if we run this query. It will not give us any data because we have written down wrong name of returndamged column name. Thus, it is unable to fetch data. We can change that error by changing the name of error.

After change the column name you will get output like this.

A screenshot of a computer

Description automatically generated

7) Re-write the following query using an IN statement:

SELECT costumeName

FROM dbo.Costume

WHERE colour = ‘Brown’

OR colour = ‘Black’

OR colour = ‘Green’

OR colour = ‘White’

A screenshot of a computer

Description automatically generated

USE Costume

GO

SELECT costumeName

FROM dbo.Costume WHERE colour IN( 'Brown','Black','Green','White')

8) 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 rentedOrder the single set of results by costume name. Note that there may costumes that have not been rented so take that into account.

A screenshot of a computer screen

Description automatically generated

USE Costume

GO

SELECT TOP 60 PERCENT c.costumeName, c.costumeDescription, c.colour, c.numPieces,

i.numberInStock,rent.rentedDate as LastRentedDate

FROM Costume AS c, CostumeInventory AS i, CostumeRental AS rent

WHERE c.costumeDescription like '%Mask%'

AND c.id = i.costumeId

AND i.costumeId = rent.costumeId

ORDER BY c.costumeName

9) Describe 5 traits of good SQL coding style.

1)Only use number, letters and underscore for field’s name.

2) Do not use reserved key in names.

3)Never give same table name as their column name.

4)Use uppercase for reserved keys.

5)Table has at least one meaningful and useful unique column.

10) 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.

A screenshot of a computer

Description automatically generated

USE Costume

GO

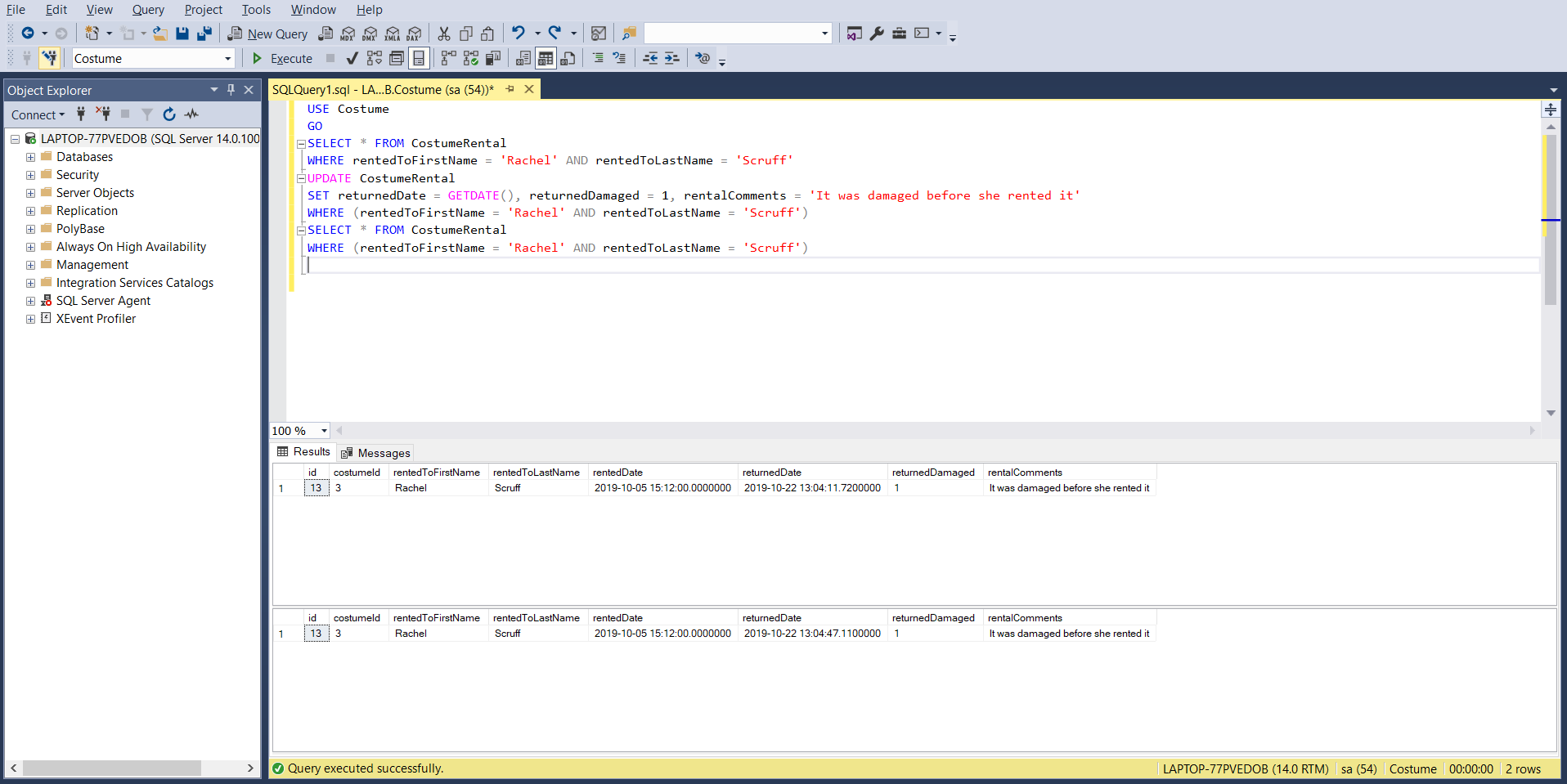
SELECT SUM(numberInStock) AS TotalNumberInStock,

SUM(numberOnOrder) AS TotalNumberOrder,

SUM(numberLost) AS TotalLostCostumeNumber

FROM dbo.CostumeInventory

**11)** 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.



USE Costume

GO

SELECT \* FROM CostumeRental

WHERE rentedToFirstName = 'Rachel' AND rentedToLastName = 'Scruff'

UPDATE CostumeRental

SET returnedDate = GETDATE(), returnedDamaged = 1, rentalComments = 'It was damaged before she rented it'

WHERE (rentedToFirstName = 'Rachel' AND rentedToLastName = 'Scruff')

SELECT \* FROM CostumeRental

WHERE (rentedToFirstName = 'Rachel' AND rentedToLastName = 'Scruff')

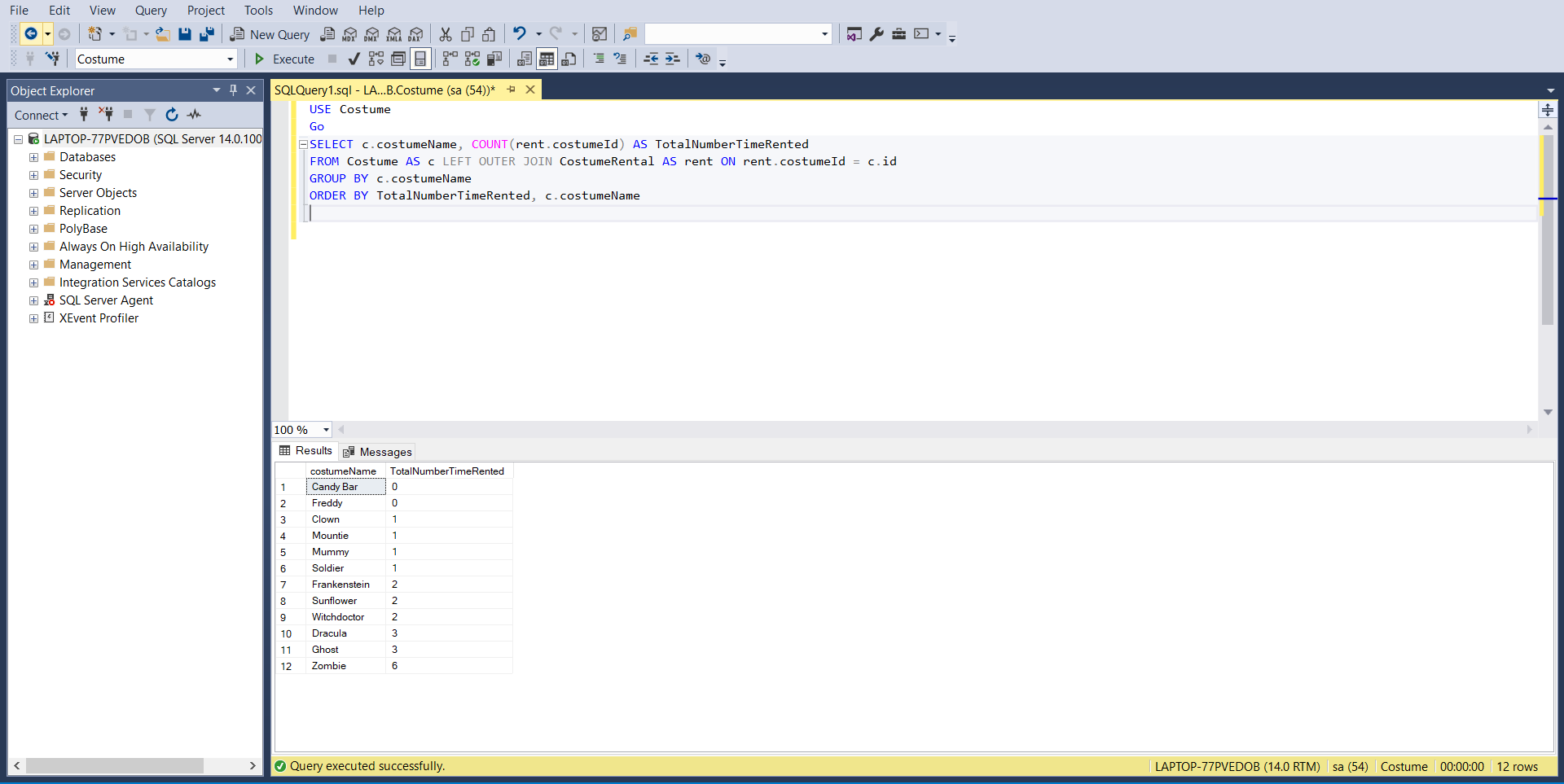
12) We need to remove all the data from a table in the database that has lots and lots of data. What’s the quickest way to clear it all out so that no rollback log is created?

We can remove data using delete and truncate. However, when you want to remove all the date from table in database. It is better to use Truncate in this you can directly remove whole table in one command then delete where you need to use where clause.

Syntax:

TRUNCATE TABLE Table\_Name;

13) 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.



USE Costume

Go

SELECT c.costumeName, COUNT(rent.costumeId) AS TotalNumberTimeRented

FROM Costume AS c LEFT OUTER JOIN CostumeRental AS rent ON rent.costumeId = c.id

GROUP BY c.costumeName

ORDER BY TotalNumberTimeRented, c.costumeName

14) 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:

Costume Name

Costume Description

UPC Code

A screenshot of a computer screen

Description automatically generated

USE Costume

Go

CREATE TABLE ScaryCostu(

costumeID int FOREIGN KEY REFERENCES dbo.Costume,

costumeDescription nvarchar(200),

upcCode varchar(15));

INSERT INTO ScaryCostu

SELECT DISTINCT costume.id, costume.costumeDescription, costume.upcCode

FROM dbo.Costume AS costume JOIN dbo.ScareRating AS rating ON rating.id = costume.scareRatingId

WHERE rating.id >= 60;

SELECT \* FROM ScaryCostu