Electronic Stores Business Model

## Initial Proposal:

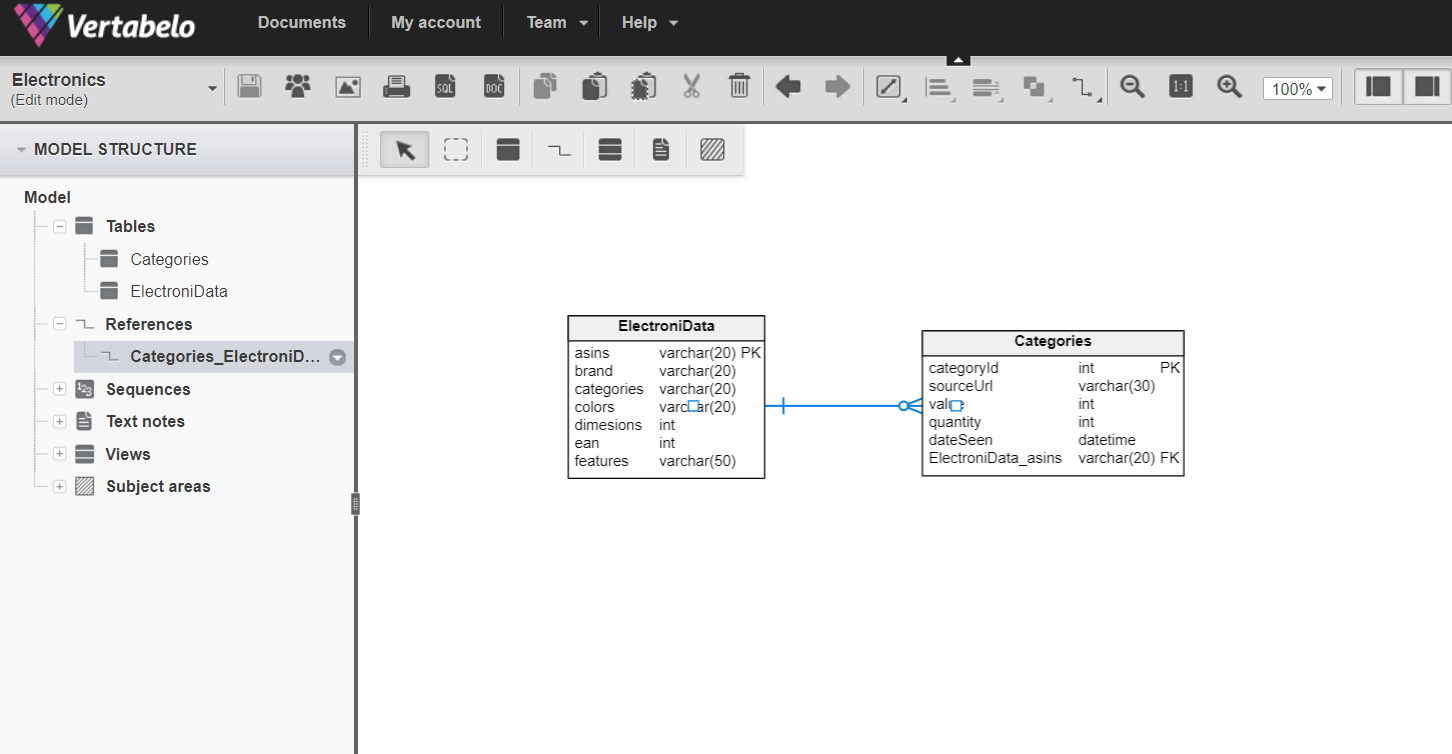
All your efforts this semester will focus on this project to be done BY YOURSELF for which you will develop, document, implement, secure, and program with a database. You will post the artifacts of work (XML files, model diagrams, code) as a GitHub repository, and you will add to it steadily throughout the semester. You will document all your efforts in a single Word document that you'll write throughout the semester, sharing it with me periodically to review and provide comment.

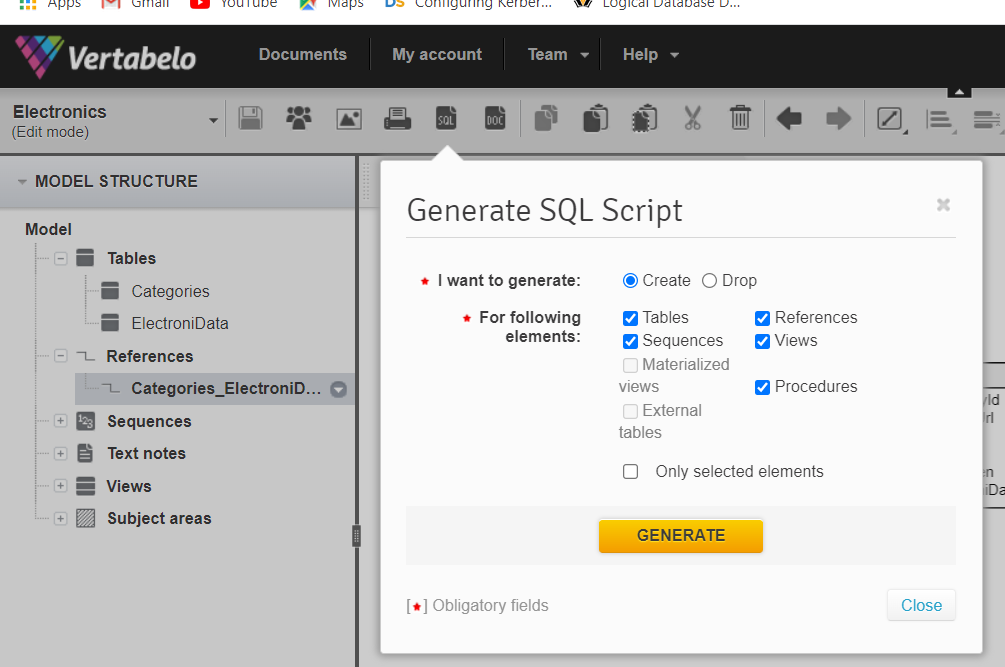
The Idea of proposing the Business model is, now-a-days the usage of the electronics is becoming the most in each city and each country. Due to this increase of the usage we are trying to build a business case model which will help us building our career. Each module or metrics will try to portray that either said metrics is being followed in the code or not with the graphical representation or tabular formats. After completing the analysis of the given code, the behavior of the code is above average which is confirmed by the analysis.

## Relational Database Design Process:

This will look a lot like the description component of the Aquarium assignment. Then illustrate that you understand normalization and how to use it to reduce uncontrolled redundancy in your database design by creating a single ERD diagram using either UML or Crow’s Foot notation and upload that drawing to **GitHub**. The ERD must be a **physical** model (includes data types and key information).

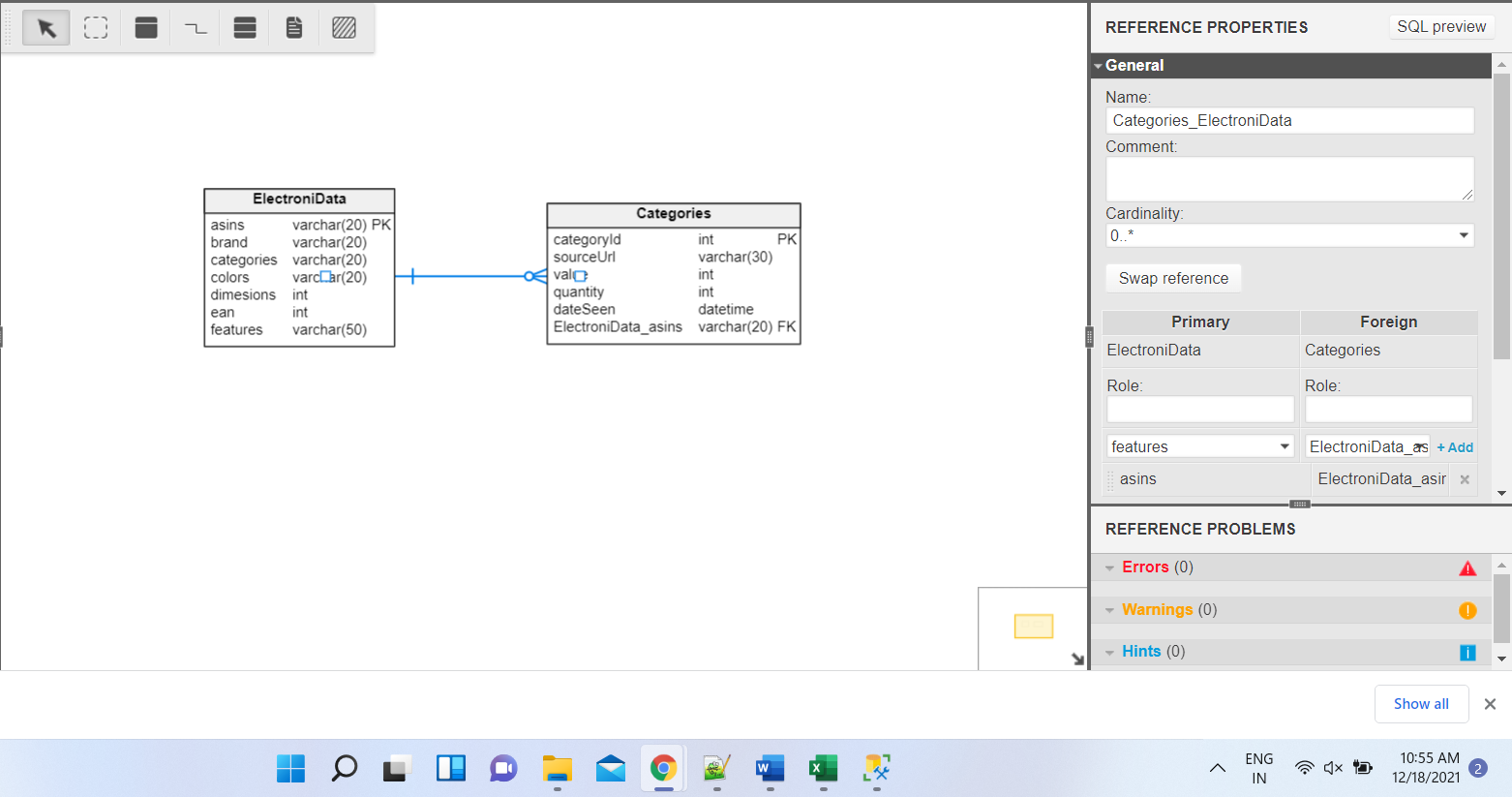
We are building a electronics store which contains any number of models and categories such as mobiles, laptops, keyboards, mouse, hard disk, batteries, pen drives, and so on. We are going to build a database design process for this new requirement which helps us to identify and drill down the process that we are going to build.





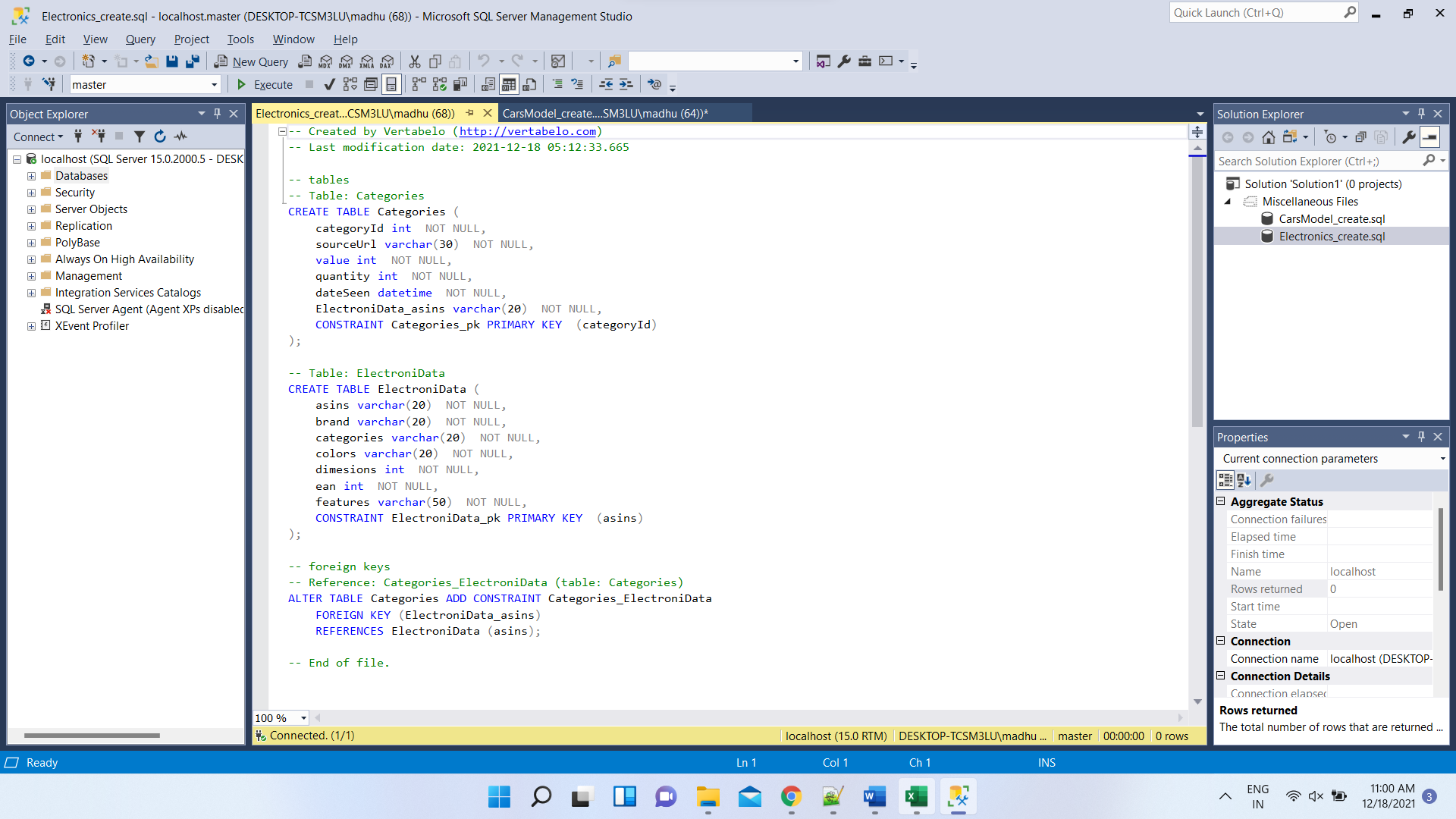
## Data Sources:

Here we will get the supply from all over the companies that provides the electronic goods which contains almost all the types of electronics that are required from the young age to all the software companies that purchases the required goods for their employees.



## Data Definition Language Scripts:

In data definition language scripts we can create the tables and for the same we can alter which means we can change the definition of the table which is already created and also we can drop the table that was created with the drop command that we can use, we can rename the table using the rename table.



-- Created by Vertabelo (http://vertabelo.com)

-- Last modification date: 2021-12-18 05:12:33.665

-- tables

-- Table: Categories

CREATE TABLE Categories (

categoryId int NOT NULL,

sourceUrl varchar(30) NOT NULL,

value int NOT NULL,

quantity int NOT NULL,

dateSeen datetime NOT NULL,

ElectroniData\_asins varchar(20) NOT NULL,

CONSTRAINT Categories\_pk PRIMARY KEY (categoryId)

);

-- Table: ElectroniData

CREATE TABLE ElectroniData (

asins varchar(20) NOT NULL,

brand varchar(20) NOT NULL,

categories varchar(20) NOT NULL,

colors varchar(20) NOT NULL,

dimesions int NOT NULL,

ean int NOT NULL,

features varchar(50) NOT NULL,

CONSTRAINT ElectroniData\_pk PRIMARY KEY (asins)

);

-- foreign keys

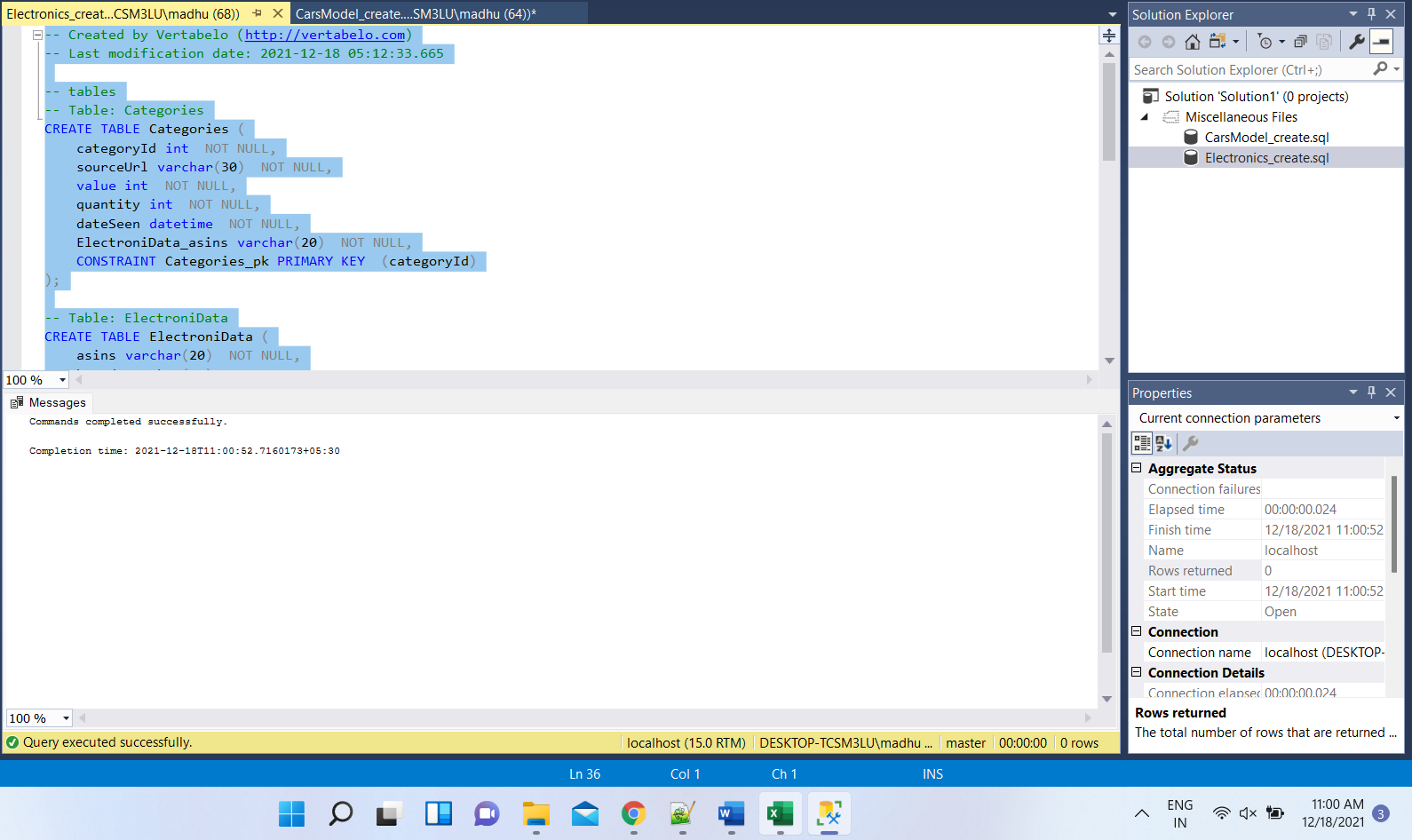
-- Reference: Categories\_ElectroniData (table: Categories)

ALTER TABLE Categories ADD CONSTRAINT Categories\_ElectroniData

FOREIGN KEY (ElectroniData\_asins)

REFERENCES ElectroniData (asins);

-- End of file.



## Data Manipulation Language Scripts:

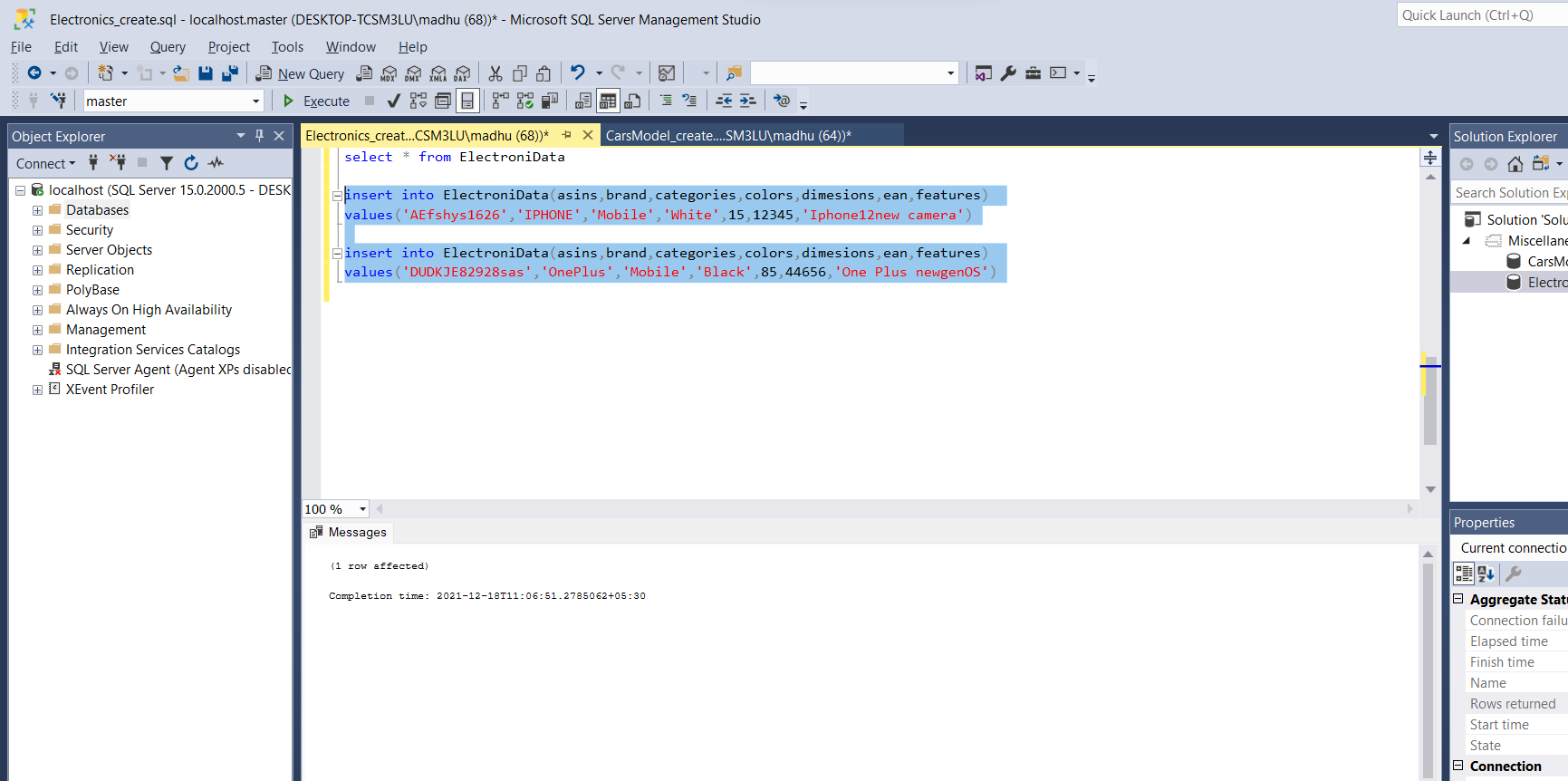
Here in the data manipulation language scripts we have different types of queries that we have to implement such are Insert , update, delete, merge etc. With these commands we can insert the data into the required tables and we can update the data that is required into the particular record and delete the record which is not required.

insert into ElectroniData(asins,brand,categories,colors,dimesions,ean,features)

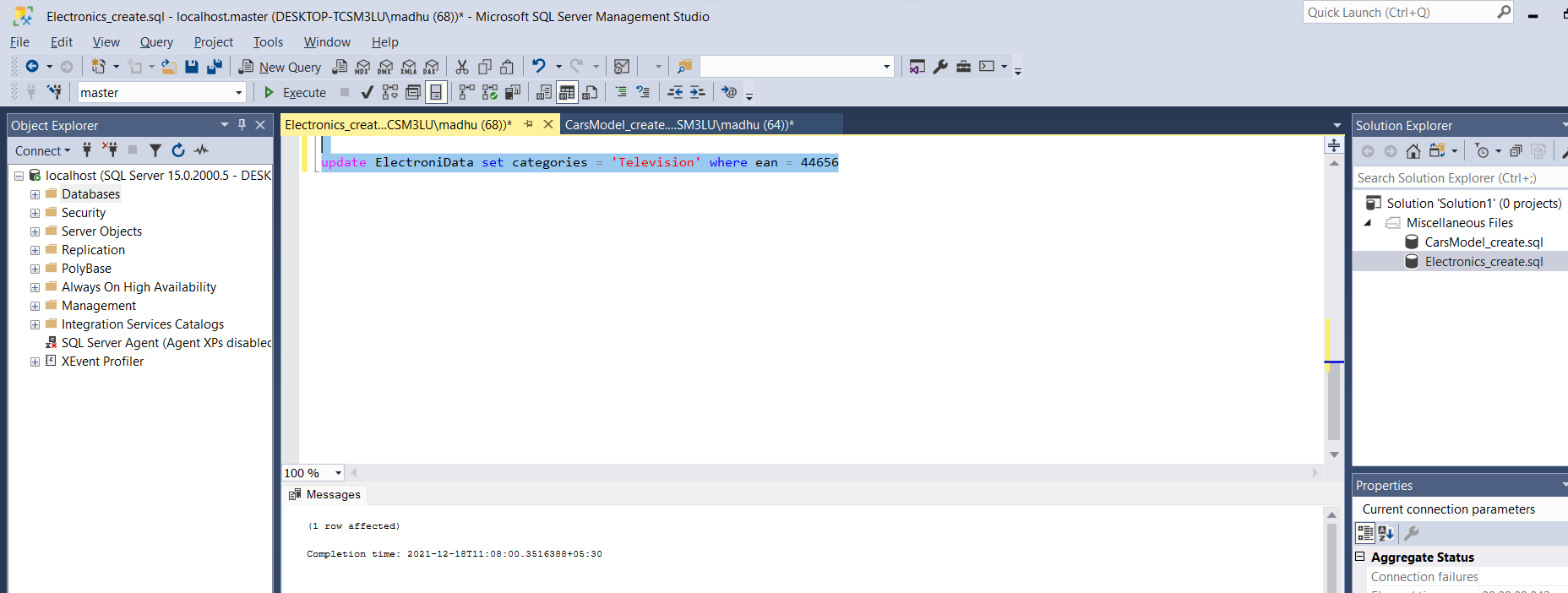
values('AEfshys1626','IPHONE','Mobile','White',15,12345,'Iphone12new camera')

insert into ElectroniData(asins,brand,categories,colors,dimesions,ean,features)

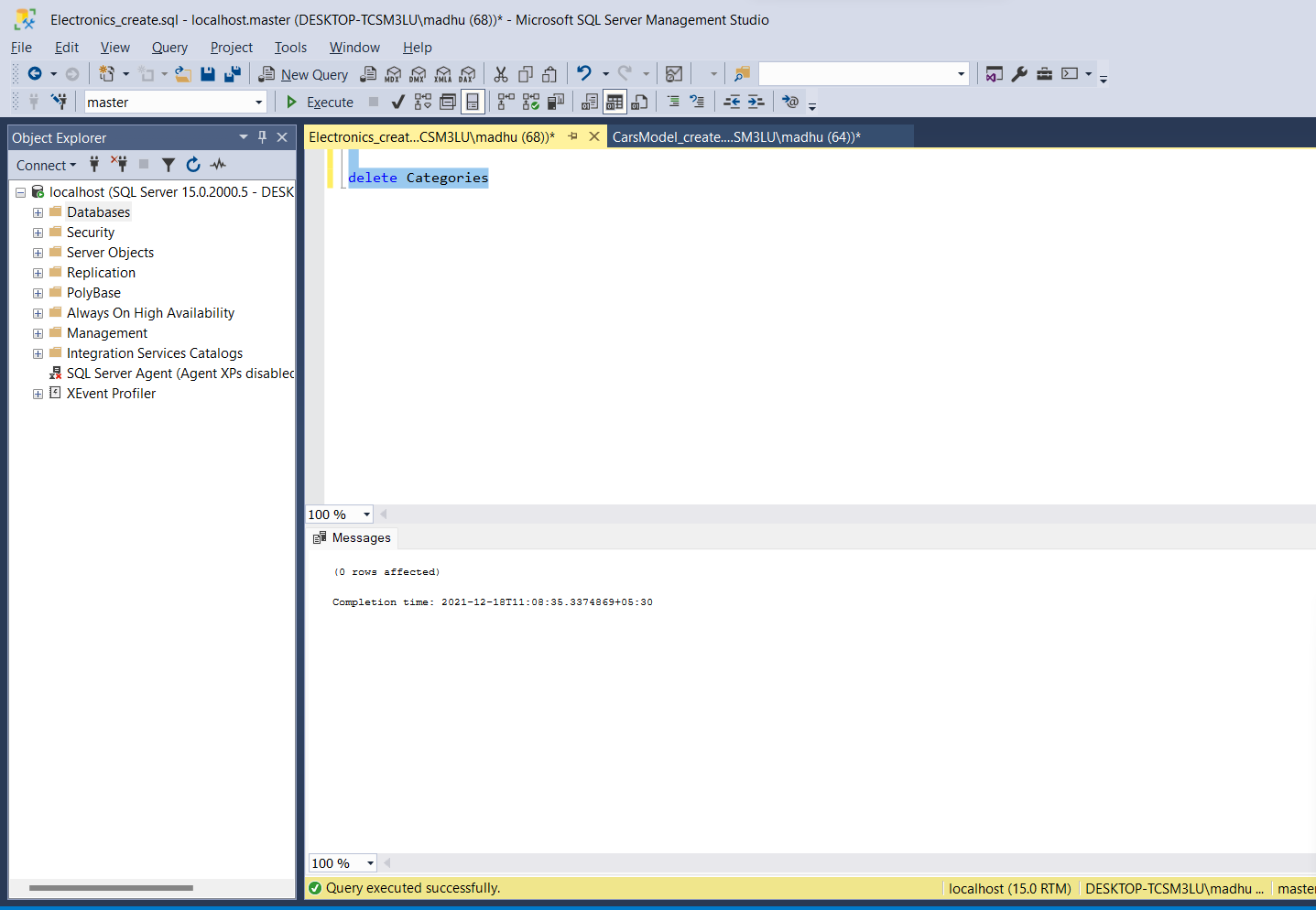
values('DUDKJE82928sas','OnePlus','Mobile','Black',85,44656,'One Plus newgenOS')



update ElectroniData set categories = 'Television' where ean = 44656

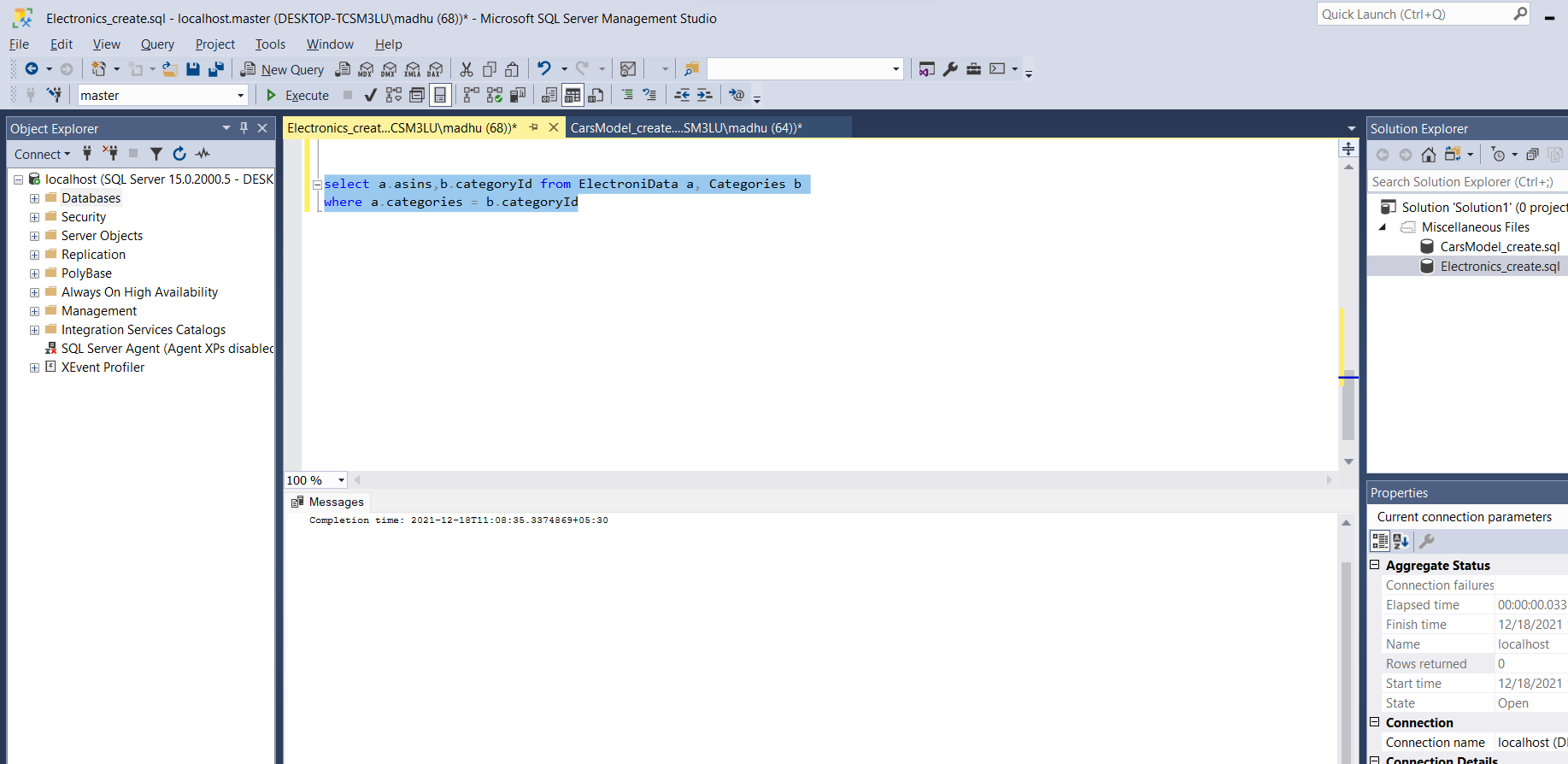


delete Categories



select a.asins,b.categoryId from ElectroniData a, Categories b

where a.categories = b.categoryId

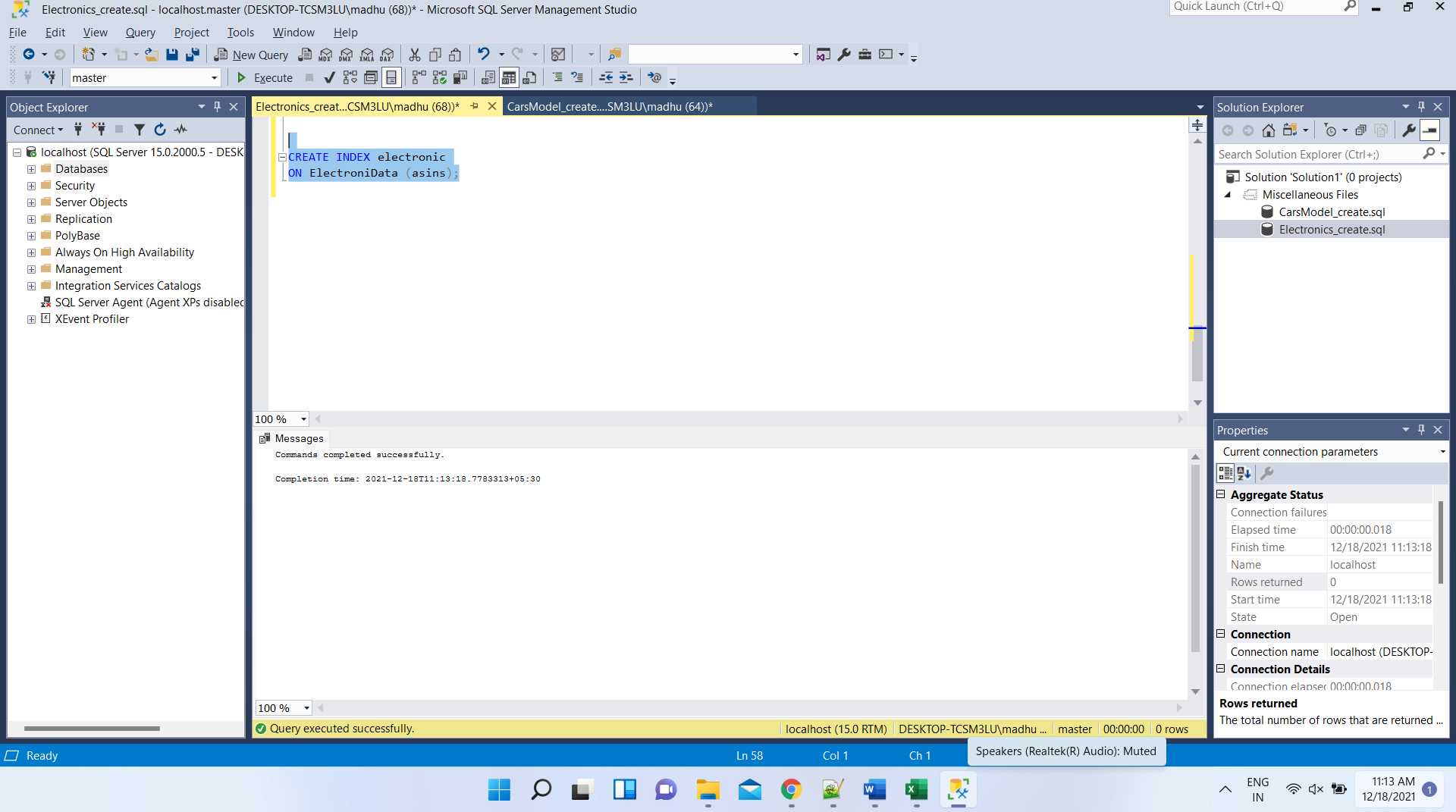


## Indexes:

Indexes is created in the database for a tables to have the performance issue controlled by the application.

CREATE INDEX electronic

ON ElectroniData (asins);



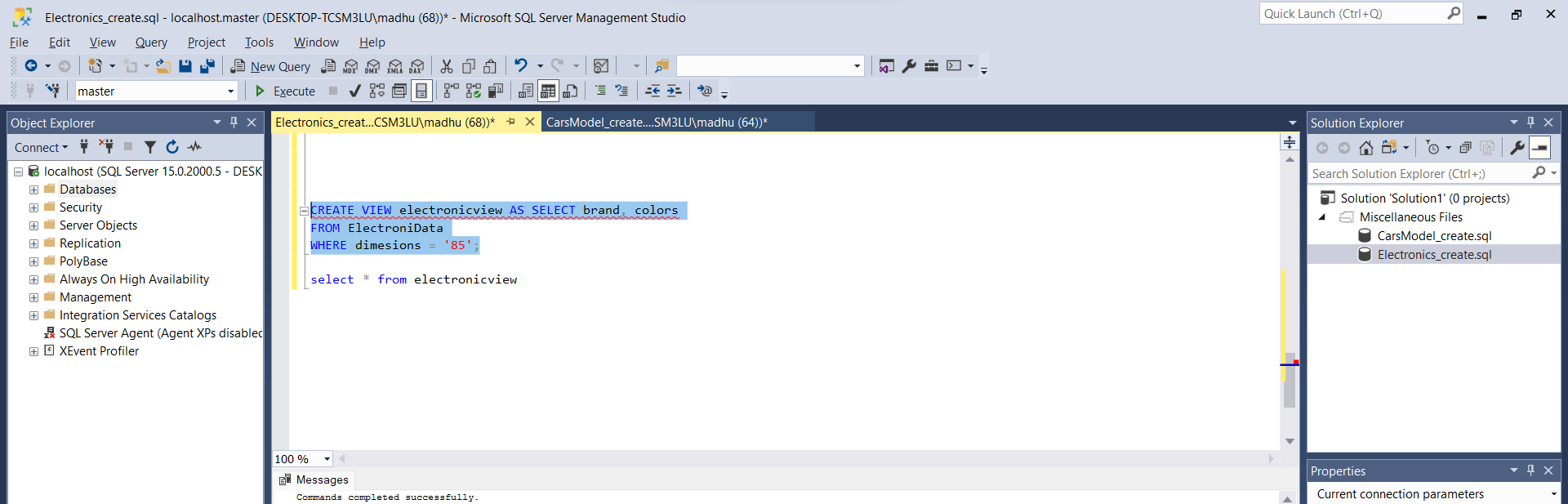
## Views:

Views are created in the database which have a same structure as table in the sql server that helps us build some logics in the application with ease.

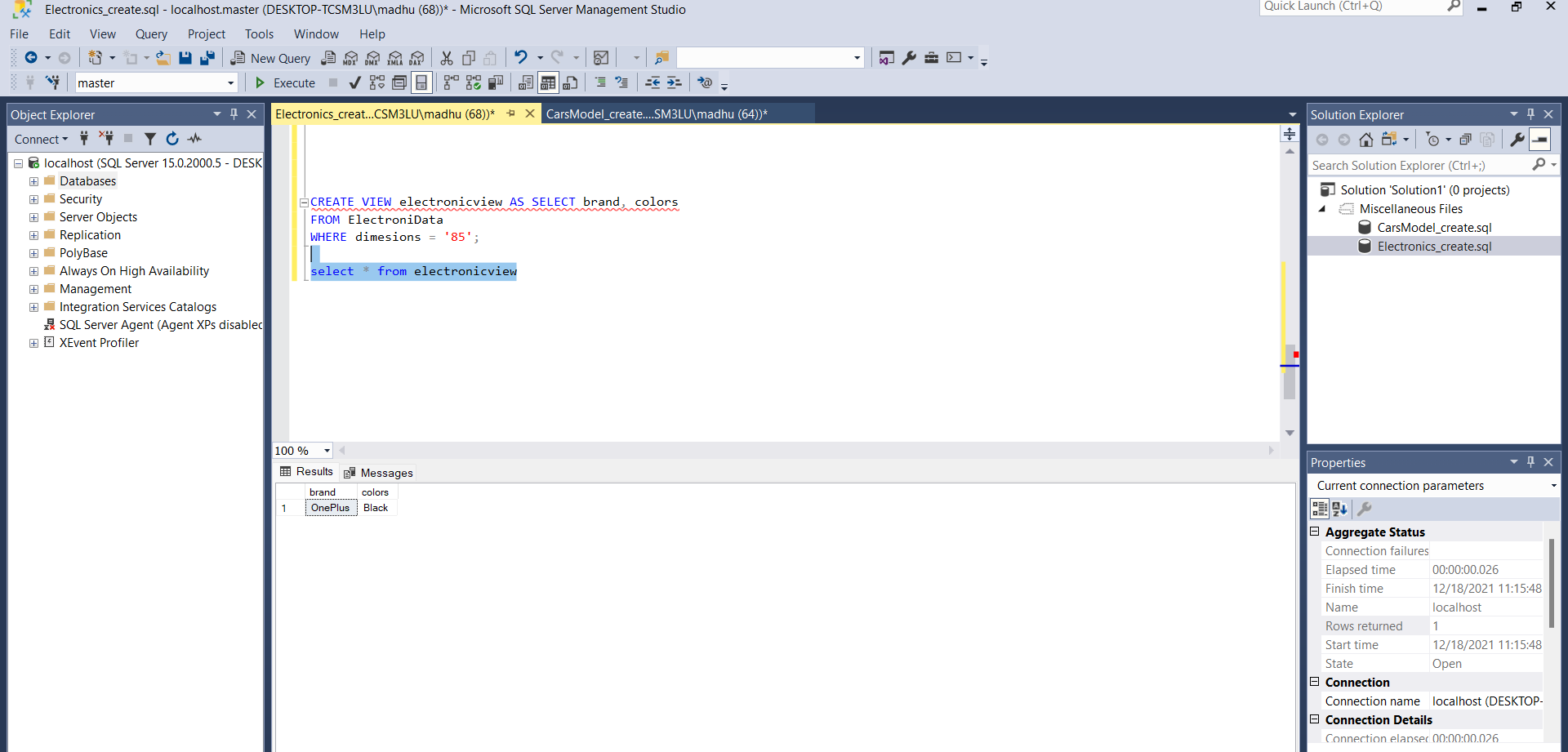
CREATE VIEW electronicview AS SELECT brand, colors

FROM ElectroniData

WHERE dimesions = '85';



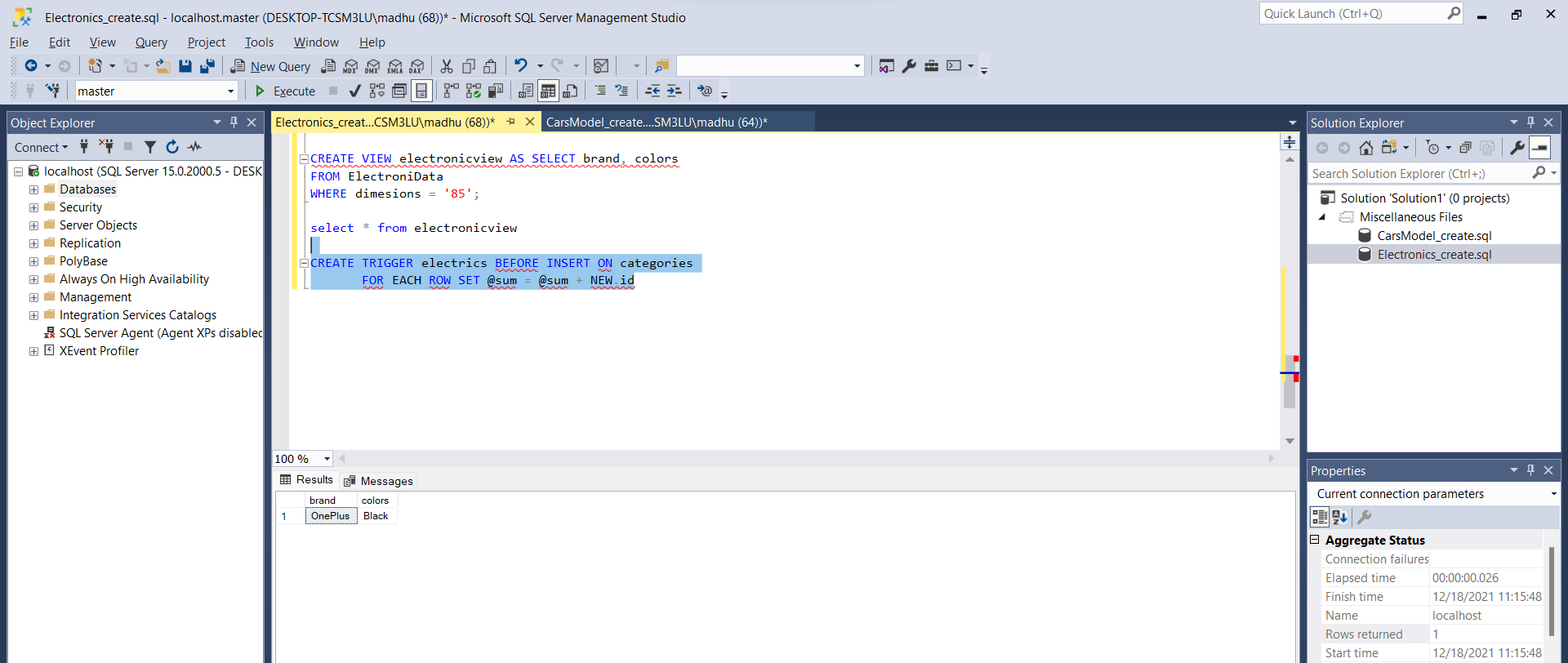
select \* from electronicview



## Triggers:

CREATE TRIGGER electrics BEFORE INSERT ON categories

FOR EACH ROW SET @sum = @sum + NEW.id



## Transactions:

Transaction in structured query language is a sequence of operations that are executed in the query window. We can consider the select statement or update statement or insert statement or delete statement in the sequence of operations. Transaction will ensure the integrity and consistency of the execution of queries. The changes that are done to the database in one or more fashion which can be done manually or through an automatic process through server is called transaction.

The Transaction properties are Atomicity, Consistency, Isolation and Durability which we call them as ACID properties. Transactions are important in ensuring acid behavior because while executing a query the transaction must ensure to follow the process of atomicity, consistency, isolation and durability by which we can achieve accuracy, data integrity and completeness.

Example on Implicit Transactions:

SET IMPLICIT\_TRANSACTIONS ON

UPDATE

ElectroniData

SET

brand = 'IPHONE12'

WHERE

ean = 12345

SELECT

IIF(@@OPTIONS & 2 = 2,

'Implicit Transaction Mode ON',

'Implicit Transaction Mode OFF'

) AS 'Transaction Mode'

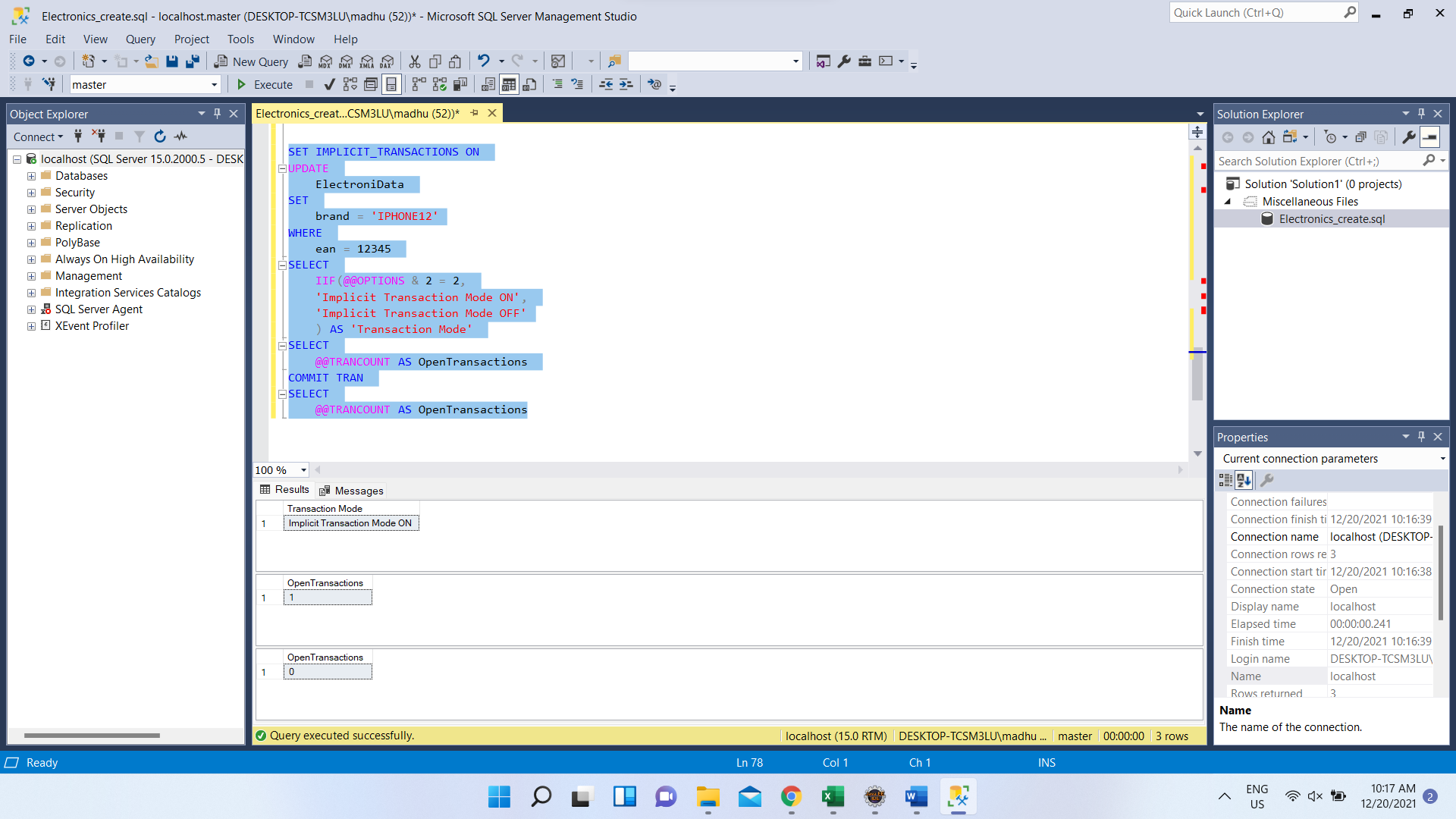
SELECT

@@TRANCOUNT AS OpenTransactions

COMMIT TRAN

SELECT

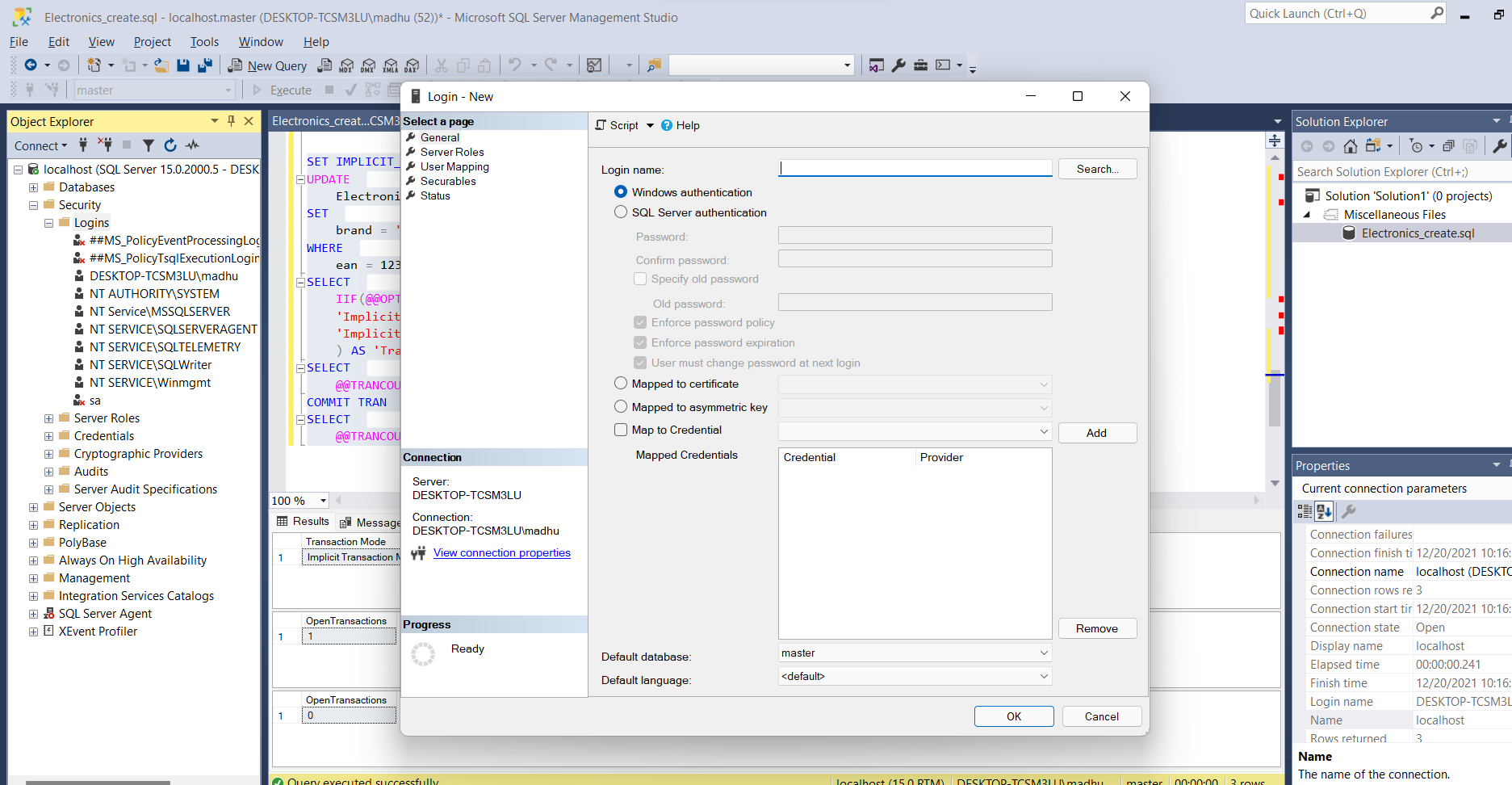
@@TRANCOUNT AS OpenTransactions



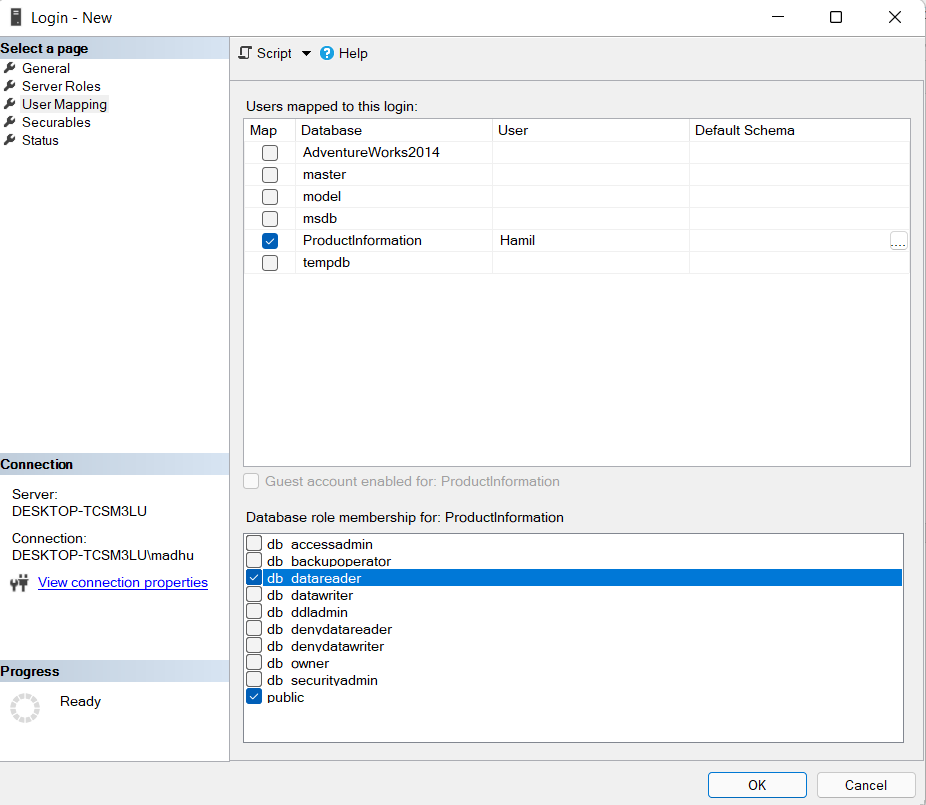
Transactions have different modes of transaction in which they are Autocommit transaction, Implicit transaction, and explicit transaction.

## Security:

We can create new users who are going to access my database with the privileges and roles that are assigned by me as an administrator. Below is the screen for creating new users.

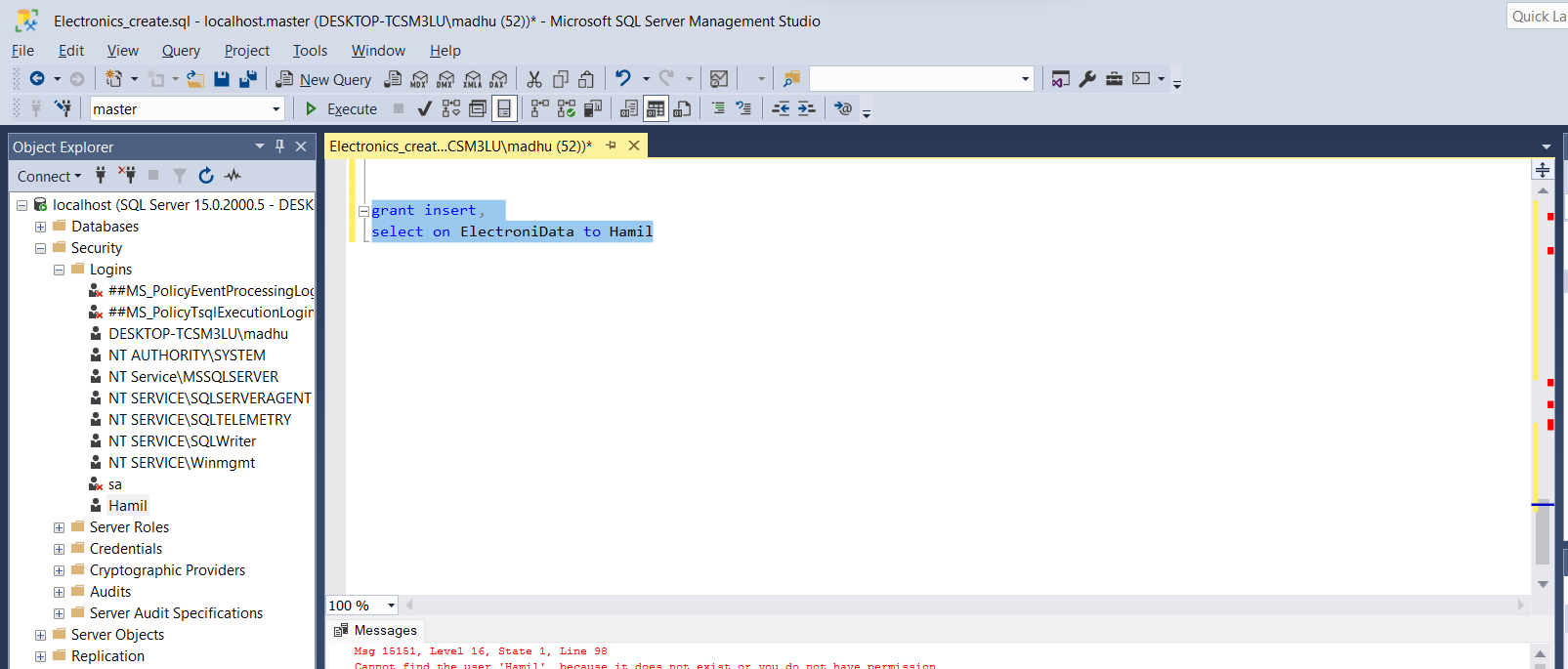


Here we are assigning new roles and actionable to be taken care by the user. We are providing the access to ProductInformation database with datareader role access.

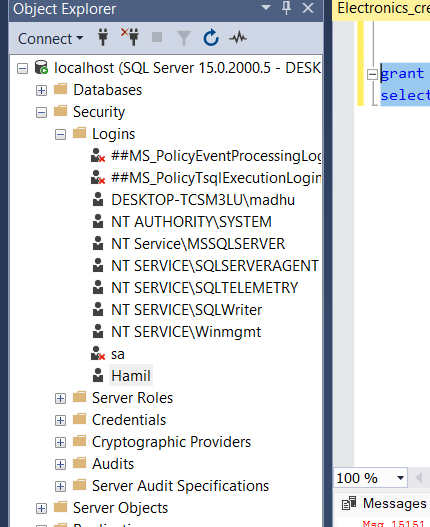


grant insert,

select on ElectroniData to Hamil



Below screenshot refers to the no of users currently present in the database that have different types of access privileges.



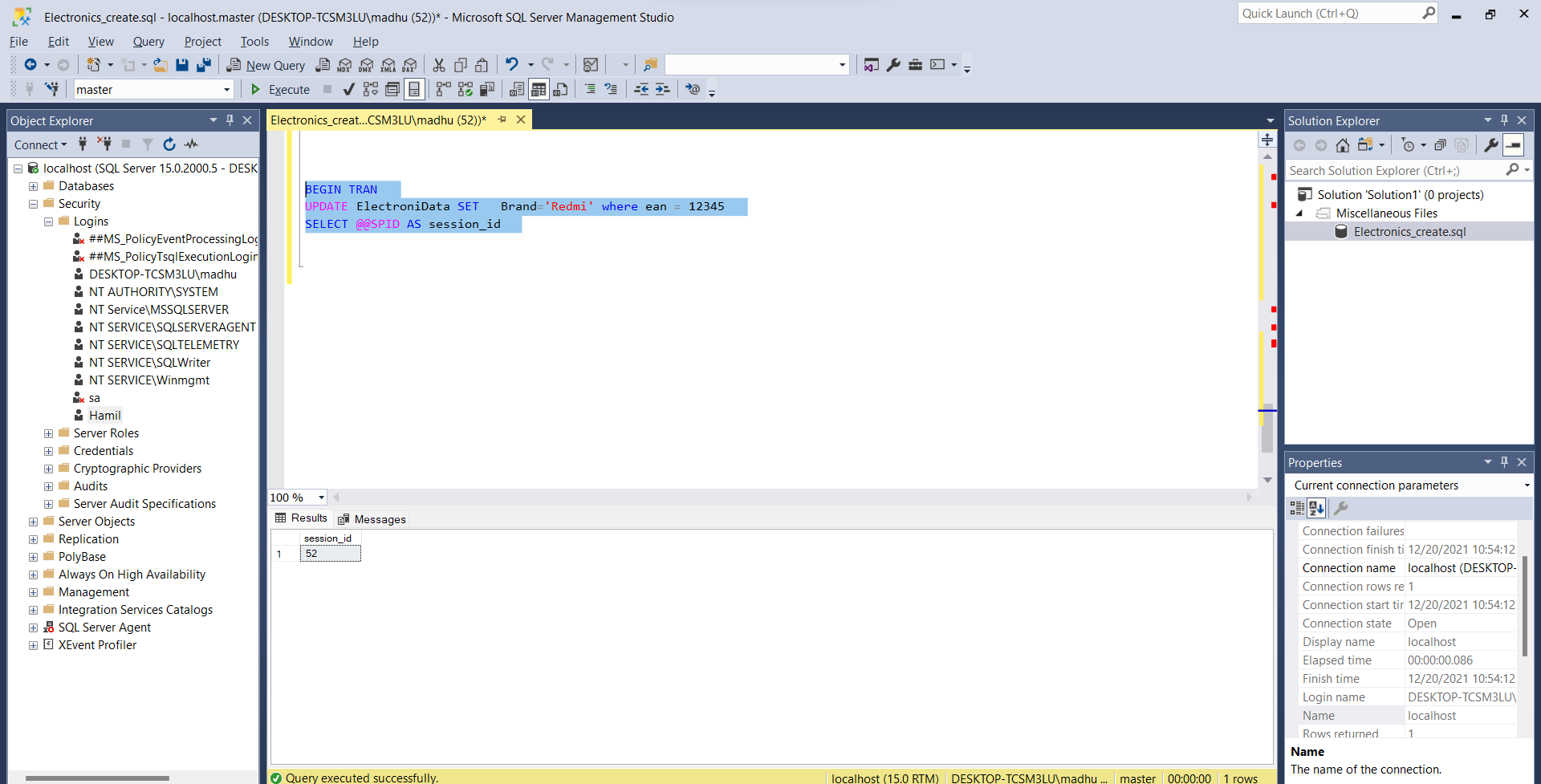
## Locking:

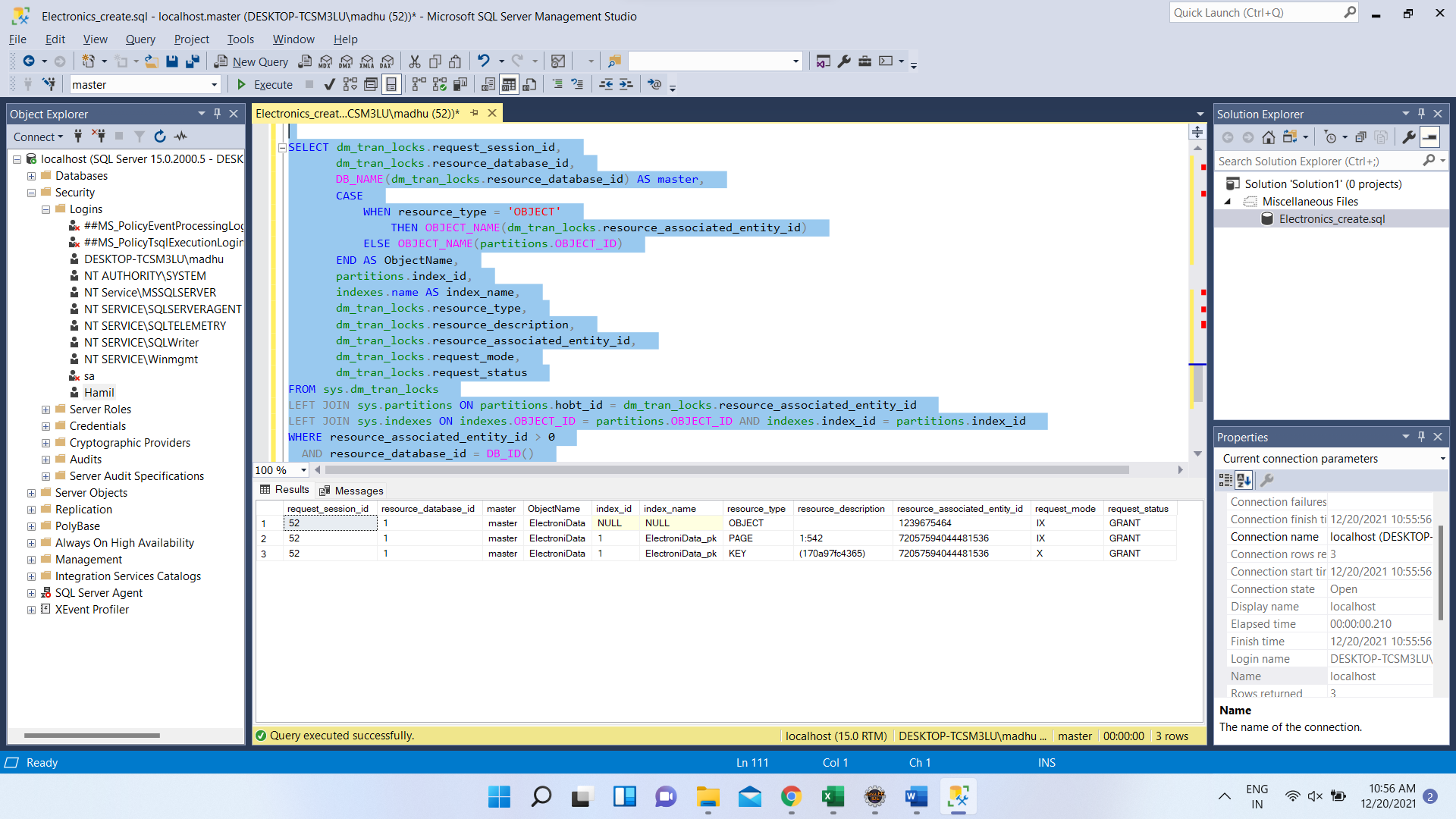
Lock is an mechanism that is present in the sql server to maintain restriction on a particular table or database or row or a record or on page level to avoid concurrency while multiple users attempt to use same data. This also helps in avoiding the in protection of data from being stolen and invalidated when everyone started using the data.

BEGIN TRAN

UPDATE ElectroniData SET Brand='Redmi' where ean = 12345

SELECT @@SPID AS session\_id





SELECT dm\_tran\_locks.request\_session\_id,

dm\_tran\_locks.resource\_database\_id,

DB\_NAME(dm\_tran\_locks.resource\_database\_id) AS master,

CASE

WHEN resource\_type = 'OBJECT'

THEN OBJECT\_NAME(dm\_tran\_locks.resource\_associated\_entity\_id)

ELSE OBJECT\_NAME(partitions.OBJECT\_ID)

END AS ObjectName,

partitions.index\_id,

indexes.name AS index\_name,

dm\_tran\_locks.resource\_type,

dm\_tran\_locks.resource\_description,

dm\_tran\_locks.resource\_associated\_entity\_id,

dm\_tran\_locks.request\_mode,

dm\_tran\_locks.request\_status

FROM sys.dm\_tran\_locks

LEFT JOIN sys.partitions ON partitions.hobt\_id = dm\_tran\_locks.resource\_associated\_entity\_id

LEFT JOIN sys.indexes ON indexes.OBJECT\_ID = partitions.OBJECT\_ID AND indexes.index\_id = partitions.index\_id

WHERE resource\_associated\_entity\_id > 0

AND resource\_database\_id = DB\_ID()

and request\_session\_id=52

ORDER BY request\_session\_id, resource\_associated\_entity\_id

## Backup:

Backup is important in sql where if any failure occurs in the execution of transaction which may crash the database and all the data may lost. In such cases it is important to have backup of the database which can be done periodically. Using backup we can restore the data that has lost and the operations may go as it is. Below is the example of how to create backups with the sql queries.

