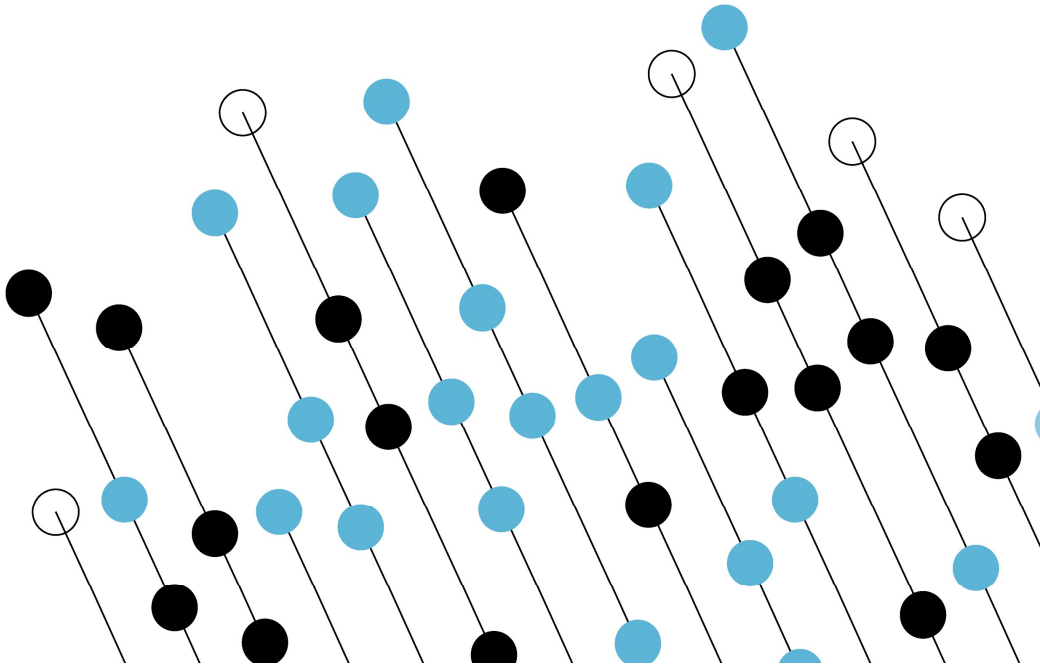
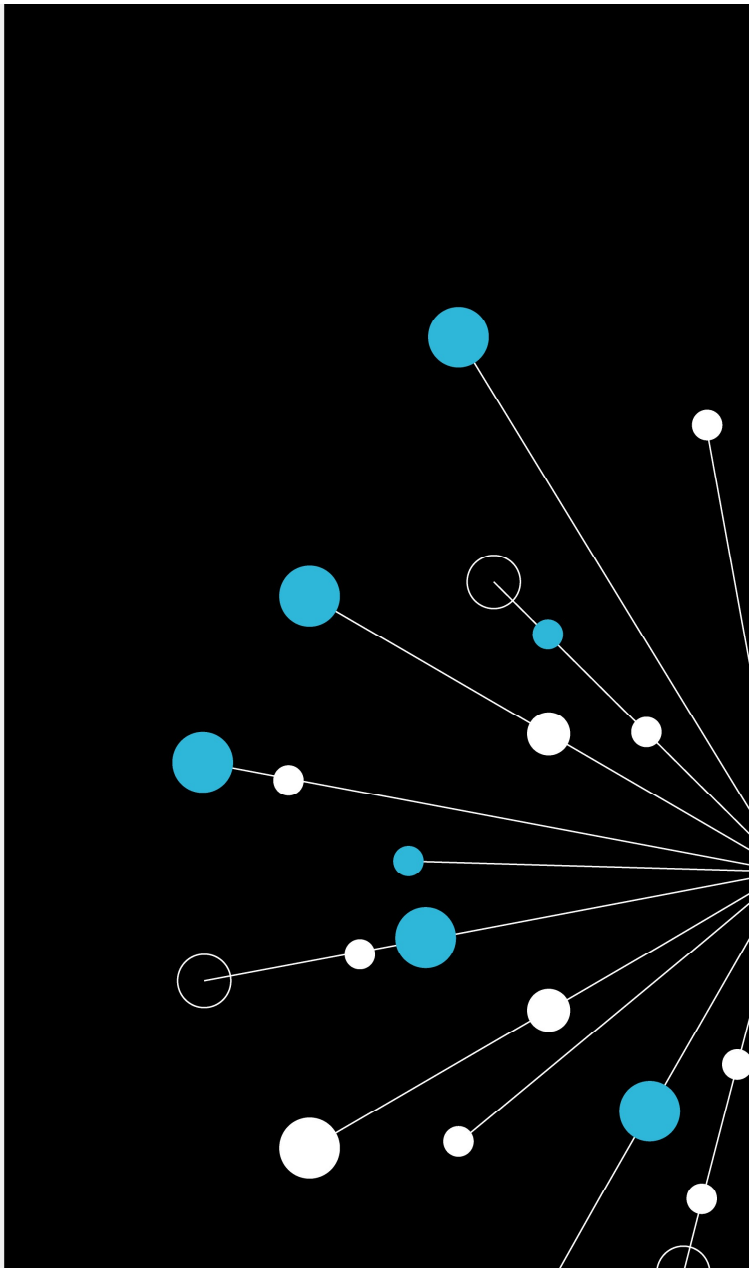
SQL: Create, Delete, Populate



SQL

Contents

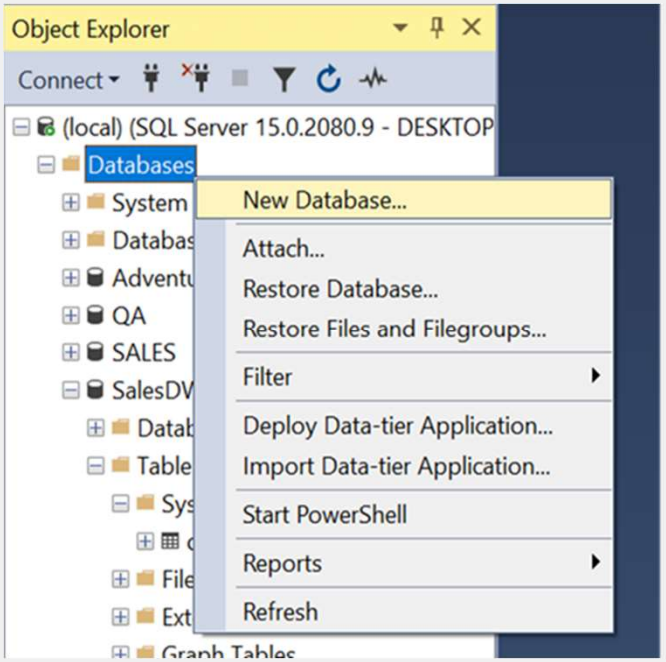
- Create and delete Database
- Create, populate and delete Table

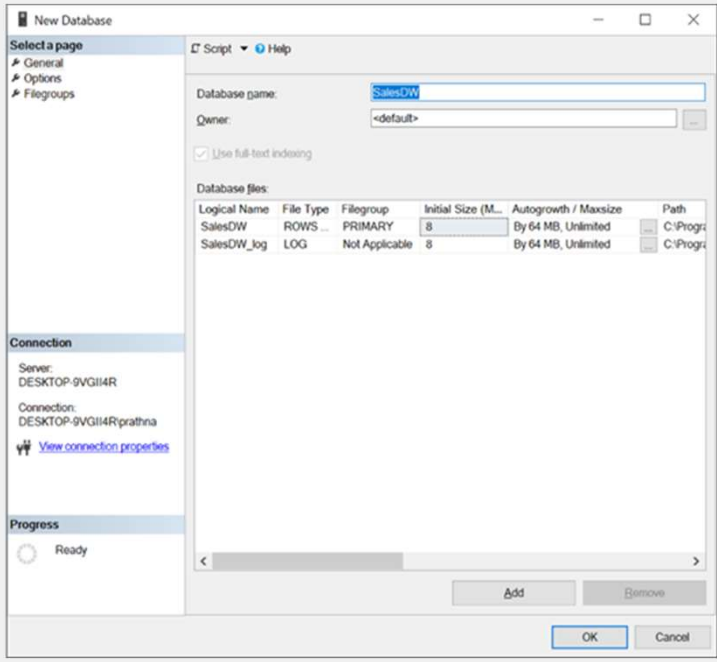


QA

Creating a database

```
-- Using a query
USE master
GO
DROP DATABASE IF EXISTS SalesDW
GO
CREATE DATABASE DW_Sales
GO
USE DW_Sales
GO
```

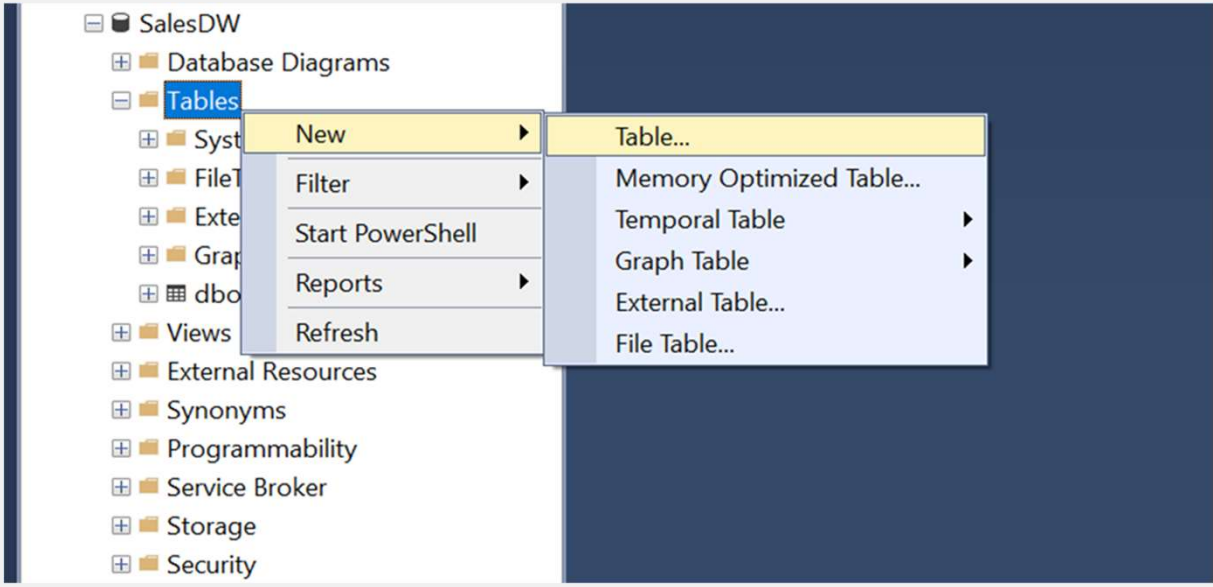




Creating a table

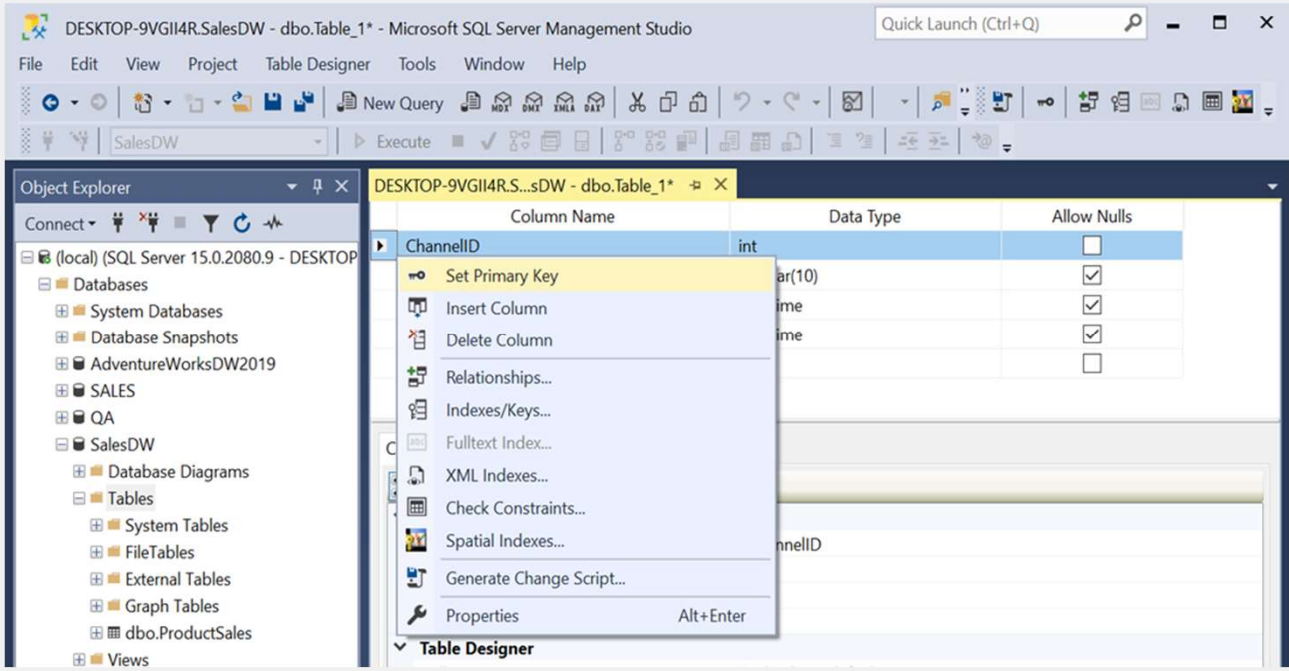
... and add columns with suitable data types.

```
-- Using a query
CREATE TABLE Product(
ProductID INT NOT NULL,
ProductDesc VARCHAR(30) NOT NULL,
CategoryID INT NOT NULL,
Price DECIMAL(5,2) NOT NULL,
DateCreated DATE NOT NULL
)
```



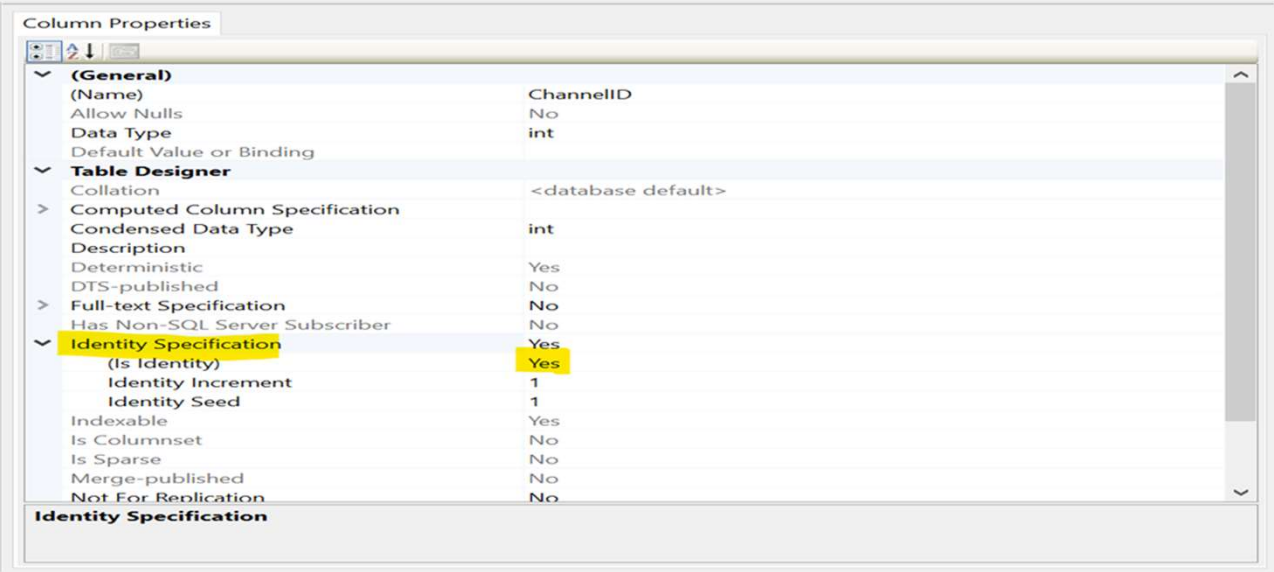
Setting a natural primary key

```
-- Using a query
CREATE TABLE Product(
ProductID INT NOT NULL PRIMARY KEY,
ProductDesc VARCHAR(30) NOT NULL,
CategoryID INT NOT NULL,
Price DECIMAL(5,2) NOT NULL,
DateCreated DATE NOT NULL
)
```



Setting a surrogate primary key

After setting the primary key, set the identity specification in the column properties:

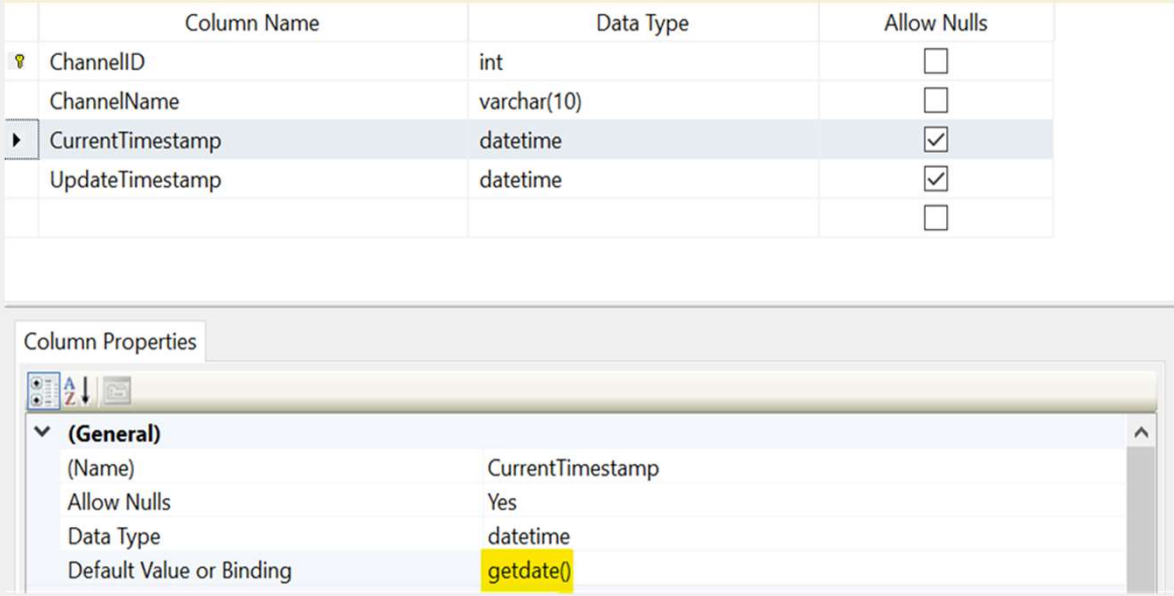


-- Using a query

```
CREATE TABLE Product(
ProductID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
ProductDesc VARCHAR(30) NOT NULL,
CategoryID INT NOT NULL,
Price DECIMAL(5,2) NOT NULL,
DateCreated DATE NOT NULL
)
```

Setting a default value for a column

```
-- Using a query
CREATE TABLE Product(
ProductID  INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
ProductDesc VARCHAR(30) NOT NULL,
CategoryID INT NOT NULL,
Price DECIMAL(5,2) NOT NULL,
DateCreated DATE DEFAULT getdate()
)
```



Populating a table – inserting a single row

Let's now store a record in the table we created:

```
INSERT INTO Product( ProductDesc, CategoryID, Price)
VALUES( "Hamlet", 1, 9.99)
```

We don't have to provide a value for ProductID – it is generated automatically, and each value is unique.

Unless we need to insert a different data value, the default date (today) will be assumed.

Populating a table – inserting multiple rows

Multiple rows to insert in a table can be derived via SELECT statement on another table.

SELECT replaces the VALUES clause.

```
INSERT INTO Product( ProductDesc, CategoryID, Price)
SELECT ProdDesc, CatID, CoverPrice
FROM BookStore
WHERE Agreed == 'Y'
```

SELECT could find 0, 1 or multiple rows.

For the above code to work, any other mandatory columns in the Product table must have a default value defined.

Delete

To delete records, use DELETE FROM.

This will empty the table:

DELETE FROM Product

To delete only certain records, it is necessary to define the conditions (which are those records) using WHERE.

This will delete only records where the price is less than 10:

DELETE FROM Product

WHERE PRICE < 10

ALWAYS make sure to run a test first using the equivalent SELECT statement. SELECT is read-only – DELETE is not!

Delete versus drop

To delete the entire table, use DROP:

DROP TABLE Product

To delete a database, use DROP too:

DROP DATABASE SalesDW

Updating values

```
UPDATE   salesperson
SET      sales_target = 400000
WHERE    dept_no = 3
```

** Note **
UPDATE 'tablename'

Not
UPDATE 'columnname'

```
UPDATE   salesperson
SET      sales_target = sales_target * 1.2,
         notes = 'Has had 20% increase'
WHERE    dept_no = 3
```

'SET' appears only once

```
UPDATE   salesperson
SET      sales_target = 400000
```