POWERING
POTENTIAL

# SQL:

## Create, Delete, Populate

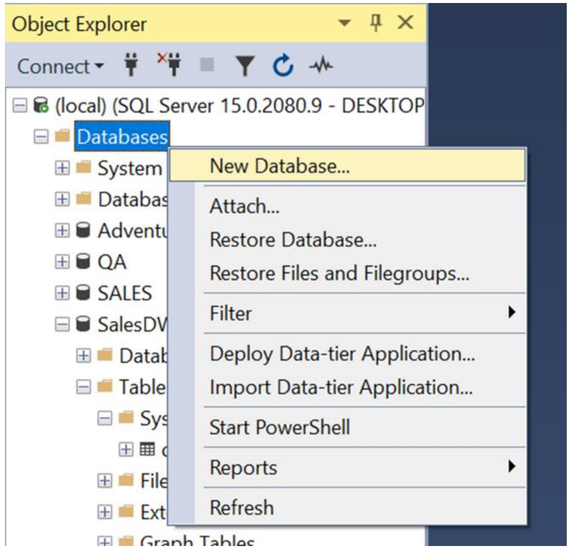
# SQL

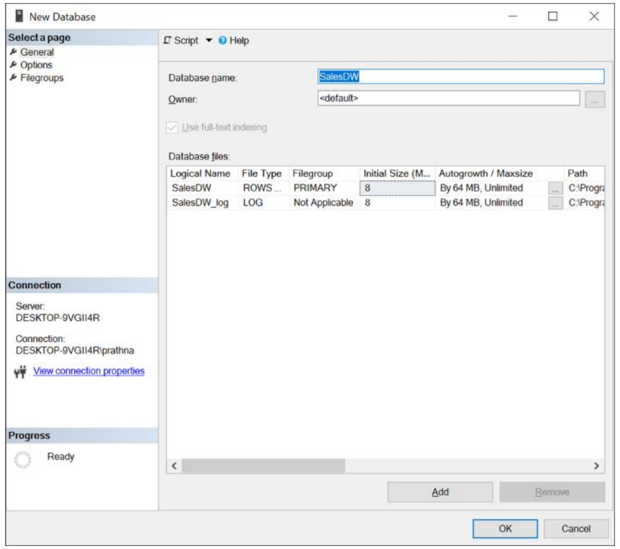
## Lesson Objectives and Contents

→Create and delete Database

→Create, populate and delete Table

# CREATING A DATABASE





Database name: SalesDW

Owner: <default>

☑ Use full-text indexing

Database files:

| Logical Name | File Type | Filegroup | Initial Size (M... | Autogrowth / Maxsize | Path |
|---|---|---|---|---|---|
| SalesDW | ROWS ... | PRIMARY | 8 | By 64 MB, Unlimited | C:\Progr |
| SalesDW_log | LOG | Not Applicable | 8 | By 64 MB, Unlimited | C:\Progr |

Connection

Server:
DESKTOP-9VGII4R

Connection:
DESKTOP-9VGII4R\prathna

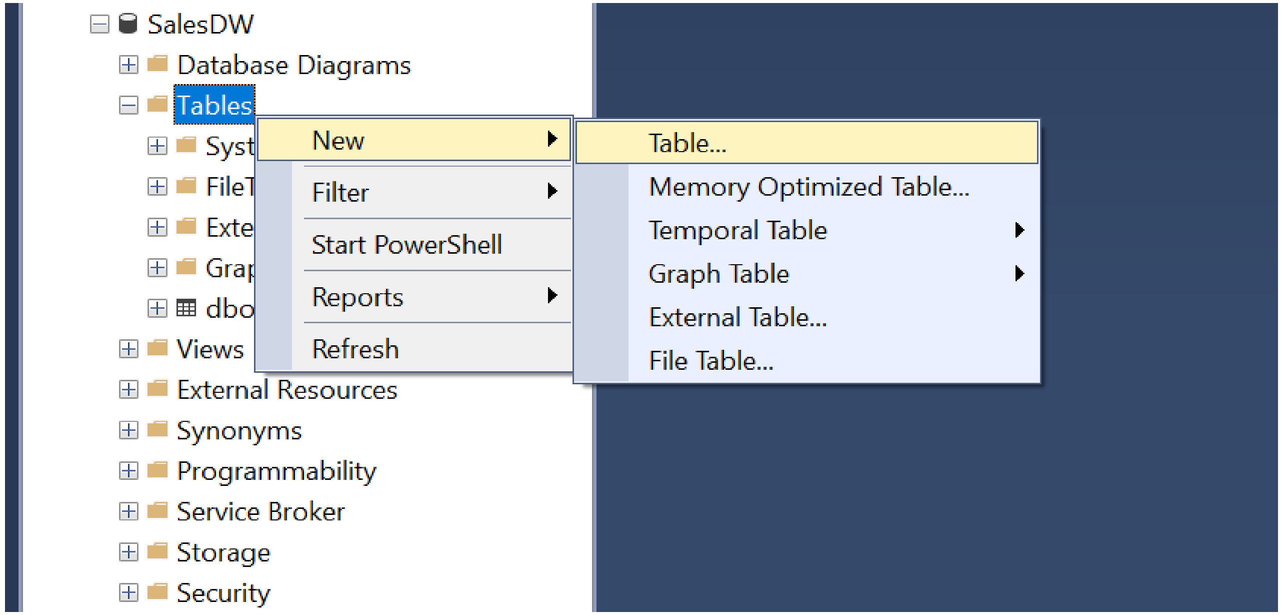
View connection properties

Progress

Ready

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

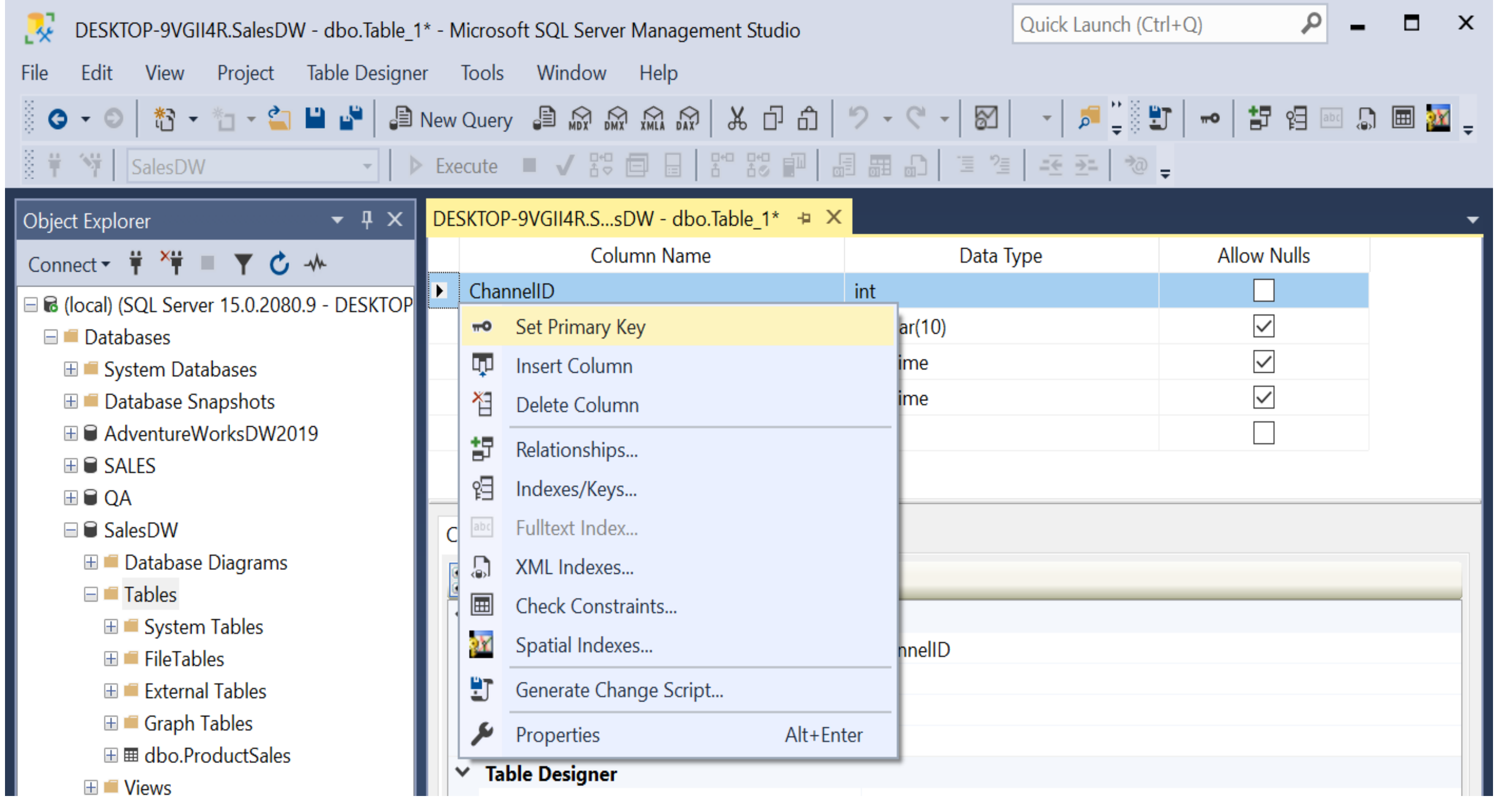
# CREATING A TABLE



... 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
)
```

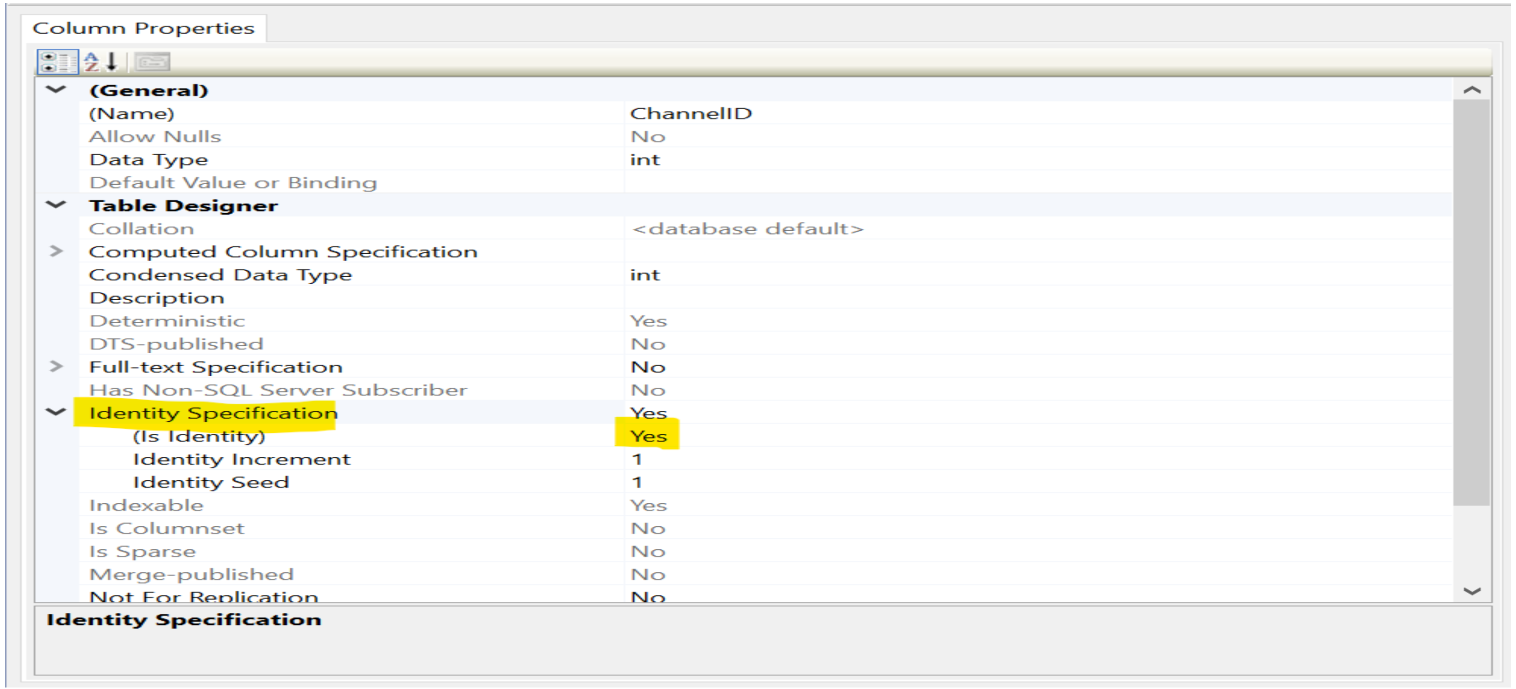
# SETTING A NATURAL PRIMARY KEY



```
-- 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:



| | |
|---|---|
| **(General)** | |
| (Name) | ChannelID |
| Allow Nulls | No |
| Data Type | int |
| Default Value or Binding | |
| **Table Designer** | |
| Collation | <database default> |
| Computed Column Specification | |
| Condensed Data Type | int |
| Description | |
| Deterministic | Yes |
| DTS-published | No |
| Full-text Specification | No |
| Has Non-SQL Server Subscriber | No |
| Identity Specification | Yes |
| (Is Identity) | Yes |
| Identity Increment | 1 |
| Identity Seed | 1 |
| Indexable | Yes |
| Is Columnset | No |
| Is Sparse | No |
| Merge-published | No |
| Not For Replication | No |

**Identity Specification**

**-- Using a query**

CREATE TABLE Product(

ProductID INT **IDENTITY(1,1)** NOT NULL **PRIMARY KEY**,

ProductDesc VARCHAR(30) NOT NULL,

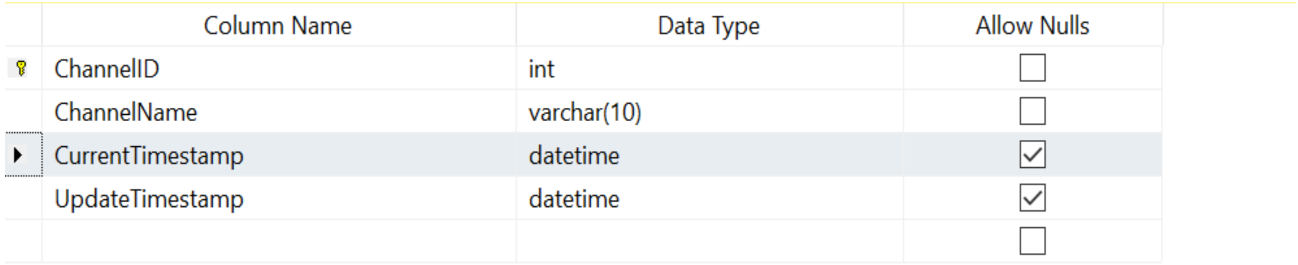
CategoryID INT NOT NULL,

Price DECIMAL(5,2) NOT NULL,

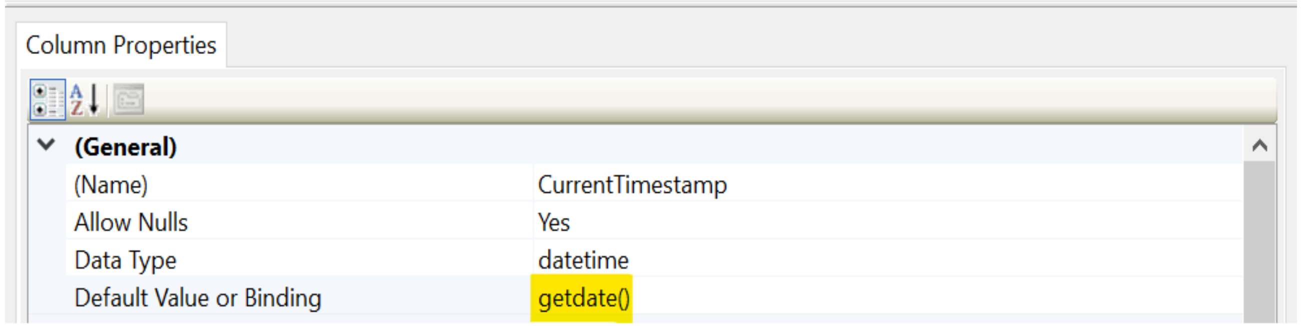
DateCreated DATE NOT NULL

)

# SETTING A DEFAULT VALUE FOR A COLUMN



| Column Name | Data Type | Allow Nulls |
|---|---|---|
| ChannelID | int | ☐ |
| ChannelName | varchar(10) | ☐ |
| CurrentTimestamp | datetime | ☑ |
| UpdateTimestamp | datetime | ☑ |
| | | ☐ |



Column Properties

(General)

| | |
|---|---|
| (Name) | CurrentTimestamp |
| Allow Nulls | Yes |
| Data Type | datetime |
| Default Value or Binding | getdate() |

**-- 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

**And 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 VERSUS DROP

**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
```