

Übung partitioned view (partitionierte Sicht)

Ziel: Üben/Erstellen von partitionierter Sicht (Anlegen von Testtabellen; Einfügen von Daten über die partitionierte Sicht)

Aufgabenstellung:

1. Erstellen Sie in Ihrer Test-Datenbank drei Testtabellen. Die Testtabellen müssen vom Spaltenaufbau her identisch sein und der Primary Key muss die Spalte, über die wir später sortieren wollen, enthalten. Z.B. Bestellungen für die Jahre 2019, 2020 und aktuell.

```
CREATE TABLE Orders (  
  OrderId INT NOT NULL,  
  Country CHAR(3) NULL,  
  OrderDate DATETIME NULL,  
  OrderYear INT NOT NULL  
  CONSTRAINT PK_Order PRIMARY KEY (OrderId,OrderYear));  
GO
```

```
CREATE TABLE Orders_2020 (  
  OrderId INT NOT NULL,  
  Country CHAR(3) NULL,  
  OrderDate DATETIME NULL,  
  OrderYear INT NOT NULL  
  CONSTRAINT PK_Order_2020 PRIMARY KEY (OrderId,OrderYear)  
);  
GO
```

```
CREATE TABLE Orders_2019 (  
  OrderId INT NOT NULL,  
  Country CHAR(3) NULL,  
  OrderDate DATETIME NULL,  
  OrderYear INT NOT NULL  
  CONSTRAINT PK_Order_2019 PRIMARY KEY (OrderId,OrderYear)  
);
```

2. Fügen Sie ein paar Testdaten ein.

```
-- Testdaten für 2019
INSERT INTO [dbo].[Orders_2019] ([OrderId],[Country],[OrderDate]
,[OrderYear])
VALUES (201901, 'AUT', '2019-01-01', 2019 ),
(201902, 'AUT', '2019-04-01', 2019 )

-- Testdaten für 2020
INSERT INTO [dbo].[Orders_2020] ([OrderId],[Country],[OrderDate]
,[OrderYear])
VALUES (202001, 'AUT', '2020-01-01', 2020 ),
(202002, 'AUT', '2020-04-01', 2020 )

-- Testdaten für 2021
INSERT INTO [dbo].[Orders] ([OrderId],[Country],[OrderDate]
,[OrderYear])
VALUES (202101, 'AUT', '2021-01-01', 2021 ),
(202102, 'AUT', '2021-04-01', 2021 )
```

3. Kontrollieren Sie, ob das Einfügen funktioniert hat:

```
-- Daten ansehen
SELECT [OrderId],[Country],[OrderDate],[OrderYear] FROM
[dbo].[Orders]
SELECT [OrderId],[Country],[OrderDate],[OrderYear] FROM
[dbo].[Orders_2020]
SELECT [OrderId],[Country],[OrderDate],[OrderYear] FROM
[dbo].[Orders_2019]
```

4. Erstellen Sie eine View (Sicht), die alle Daten aus diesen drei Tabellen ausgibt.

```
CREATE VIEW v_Orders_Demo
AS
SELECT OrderId, Country, OrderDate, OrderYear FROM Orders
UNION ALL
SELECT OrderId, Country, OrderDate, OrderYear FROM Orders_2020
UNION ALL
SELECT OrderId, Country, OrderDate, OrderYear FROM Orders_2019
GO
```

5. Überprüfen Sie, welche Daten von der View (Sicht) jetzt ausgegeben werden.

```
SELECT *  
FROM v_Orders_Demo
```

6. Aktivieren Sie den Actual Execution Plan.

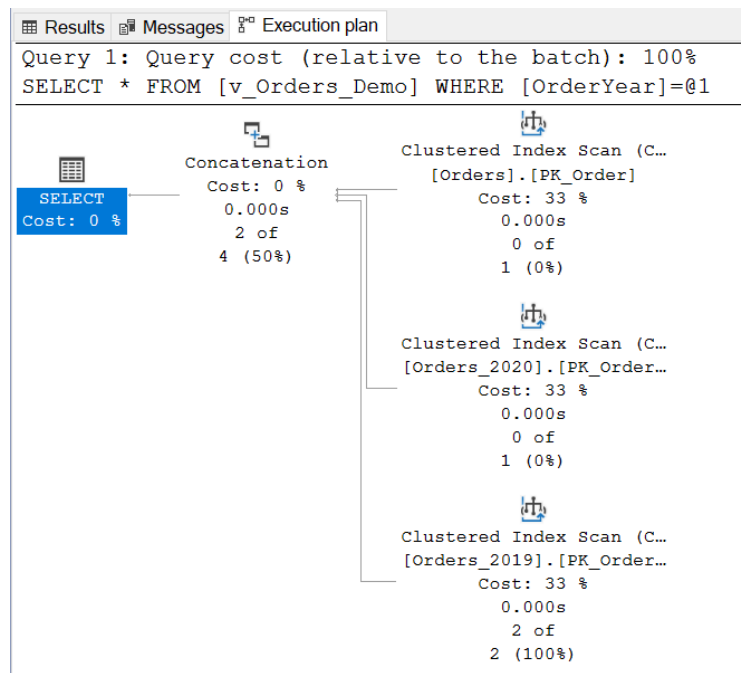


Sie können den Execution Plan über die Menüleiste aktivieren, oder indem Sie STRG+M drücken.

7. Führen Sie eine einfache Abfrage aus, die eine Jahresangabe enthält und betrachten Sie den Execution Plan.

```
SELECT *  
FROM v_Orders_Demo  
WHERE OrderYear = 2019
```

Was fällt Ihnen auf?



Im Execution Plan sehen wir, dass alle drei Tabellen abgefragt werden müssen, da unsere View (Sicht) ein UNION ALL mit diesen drei Tabellen verlangt. Wir können aber dafür sorgen, dass nur jeweils die Tabelle abgesucht wird, die die entsprechende Jahreszahl enthält.

8. Erstellen Sie CHECK-CONSTRAINTS für die drei Testtabellen: In den jeweiligen Tabellen sind nur noch Einträge mit den jeweiligen Jahren gestattet.

-- CHECK CONSTRAINT FÜR PARTITIONED VIEW (damit nur noch 1 Tabelle angesehen wird)

ALTER TABLE Orders ADD CONSTRAINT CK_Order CHECK (OrderYear >= 2021)

ALTER TABLE Orders_2020 ADD CONSTRAINT CK_Order_2020 CHECK (OrderYear = 2020)

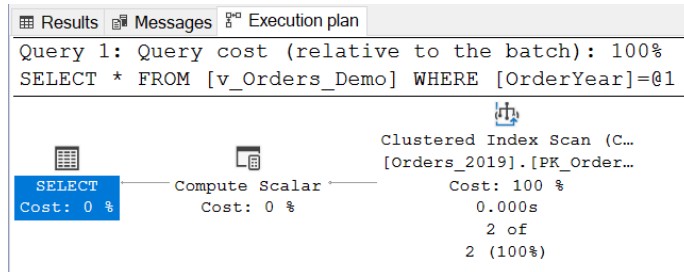
ALTER TABLE Orders_2019 ADD CONSTRAINT CK_Order_2019 CHECK (OrderYear = 2019)

9. Führen Sie die Abfrage noch einmal durch und betrachten Sie wieder den Execution Plan.

```

SELECT *
FROM v_Orders_Demo
WHERE OrderYear = 2019

```



Bei dieser Abfrage muss nur noch eine einzige Tabelle betrachtet werden, denn dank der CHECK-CONSTRAINTS ist jetzt bekannt, dass 2019 nur in dieser einen Tabelle vorkommen kann!

10. Fügen Sie jeweils einen neuen Wert in die entsprechende Tabelle ein; verwenden Sie dafür die partitioned view.

```

-- alle auf einmal einfügen:
INSERT INTO [dbo].[v_Orders_Demo] ([OrderId],[Country],[OrderDate]
,[OrderYear])
VALUES      (202103, 'AUT', '2021-03-01', 2021 ),
            (202003, 'AUT', '2020-03-02', 2020 ),
            (201903, 'AUT', '2019-03-03', 2019 )

```

Sie können die partitioned view (partitionierte Sicht) dafür verwenden, Werte einzufügen, die automatisch in die entsprechende Tabelle einsortiert werden (dank CHECK-Constraint).

11. Überprüfen Sie, ob es wie gewünscht funktioniert hat:

```

SELECT * FROM Orders
SELECT * FROM Orders_2019
SELECT * FROM Orders_2020

```

12. Überlegen Sie, welche Nachteile eine partitionierte Sicht haben könnte.