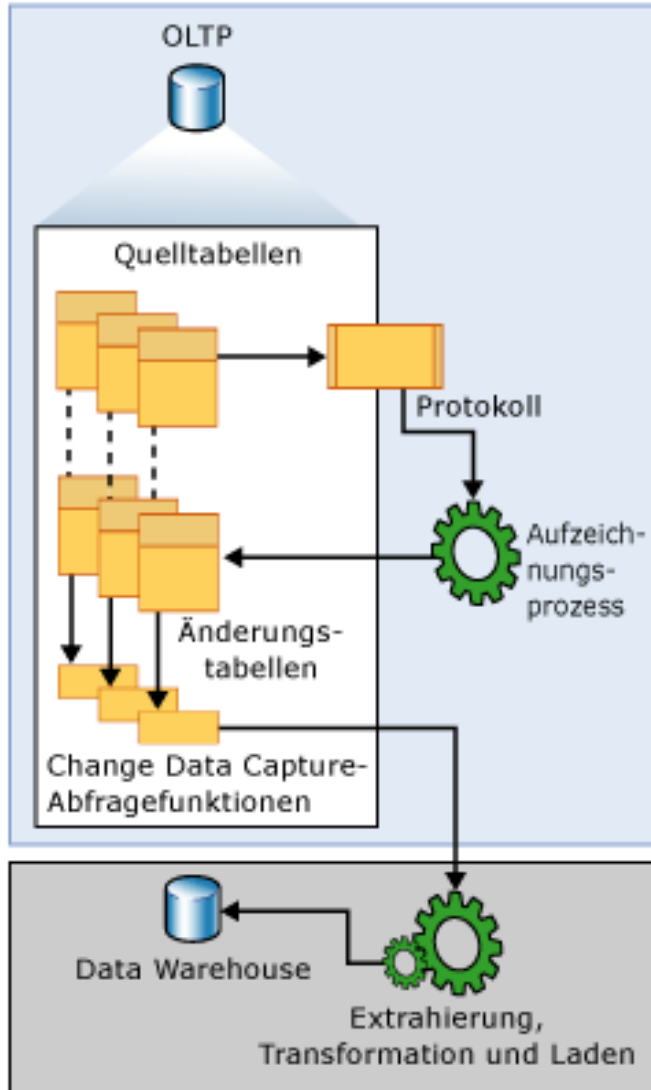


Change Data Capture  
Output  
Merge

# Change Data Capture



- Details zu Änderungen (Einfüge- , Aktualisierungs- und Löschartivitäten auf die Tabellen) werden in einem leicht verwendbaren relationalen Format bereitgestellt
- Wurde in SQL Server 2008 eingeführt, ist aber nur in dem Enterprise Edition verfügbar

# Change Data Capture (CDC)

- `sys.sp_cdc_enable_db`
  - Change Data Capture wird explizit für die Datenbank aktiviert (damit Änderungen an einzelnen Tabellen nachverfolgt werden können)
- `sys.sp_cdc_enable_table`
  - Wenn die Datenbank für Change Data Capture aktiviert ist, dann können Quelltabellen als nachverfolgte Tabellen identifiziert werden
- Für mehrere Informationen: <https://msdn.microsoft.com/de-de/library/cc645937.aspx>

# OUTPUT Klausel

- Gibt Informationen aus bzw. Ausdrücke basierend auf den einzelnen Zeilen zurück, auf die eine INSERT-, UPDATE-, DELETE- oder MERGE-Anweisung Auswirkungen hat
- Bietet Zugriff zu den `inserted` oder `deleted` Tabellen

# OUTPUT Klausel

```
UPDATE Categories
SET CategoryName = 'Dried Produce'
OUTPUT inserted.CategoryId,
        deleted.CategoryName,
        inserted.CategoryName,
        SUSER_SNAME () ← gibt den Username aus
INTO CategoryChanges
WHERE CategoryId = 7
```

# MERGE Klausel

- Führt Einfüge-, Aktualisierungs- oder Löschvorgänge in einer Zieltabelle anhand der Ergebnisse eines Joins mit einer Quelltablelle
- Kann z.B. zwei Tabellen synchronisieren, indem Zeilen in einer Tabelle anhand von Unterschieden, die in der anderen Tabelle gefunden wurden, eingefügt, aktualisiert oder gelöscht werden

# MERGE KLAUSEL

```
MERGE Table_definition AS Target
USING (Table Source) AS Source
(
    Column Keys
)
ON (
    Search Terms
)
WHEN MATCHED THEN
    UPDATE SET
        or
    DELETE
WHEN NOT MATCHED BY TARGET/SOURCE THEN
    INSERT
```

# MERGE Klausel

```
MERGE Books
USING
( select MAX(BookId) BookId, Title, MAX(Author)
Author, MAX(ISBN) ISBN, MAX(Pages) Pages
  from Books
  group by Title )MergeData
ON Books.BookId = MergeData.BookId
WHEN MATCHED THEN
    UPDATE SET Books.Title = MergeData.Title,
    Books.Author = MergeData.Author,
    Books.ISBN = MergeData.ISBN,
    Books.Pages = MergeData.Pages
WHEN NOT MATCHED BY SOURCE THEN DELETE;
```



# MERGE Klausel

- Wie würde folgende Tabelle nach dem Merge aussehen?

BookId	Title	Author	ISBN	Pages
1	Microsoft SQL Server 2005 For Dummies	Andrew Watt	NULL	NULL
2	Microsoft SQL Server 2005 For Dummies	NULL	NULL	432
3	Microsoft SQL Server 2005 For Dummies	NULL	978-0-7645-7755-0	NULL

- Nach dem Merge:

BookId	Title	Author	ISBN	Pages
3	Microsoft SQL Server 2005 For Dummies	Andrew Watt	978-0-7645-7755-0	432

# MERGE Klausel

```
MERGE Target AS T
USING Source AS S
ON (T.EmployeeID = S.EmployeeID)
WHEN NOT MATCHED BY TARGET AND S.EmployeeName LIKE 'S%'
    THEN INSERT (EmployeeID, EmployeeName)
        VALUES (S.EmployeeID, S.EmployeeName)
```

Angestellten aus der Tabelle S, deren Name mit S anfängt und die in der Tabelle T nicht existieren (nach Id), werden jetzt auch in T eingefügt mit Id und Name.

# Aufgabe

- Gegeben seien folgende Tabellen:
  - Kunden (id , nachname, vorname)
  - KundenNew (id , nachname, vorname)
- In der Tabelle KundeNew hat man die aktuelle Daten über die Kunden einer Firma.
- Benutze die Merge Klausel, um die Tabelle Kunden zu aktualisieren, sodass man auch hier die aktuellen Informationen über Kunden hat.
- Welche sind die möglichen Änderungen die vorkommen können?
  - Man nimmt an, dass die Ids der Kunden nicht geändert wurden
  - Ein Kunde kann seinen Namen ändern
  - Ein Kunde kann gelöscht werden
  - Ein neuer Kunde kann eingefügt werden