Departement Informatika
Department of Informatics

MS SQL 2008

Practical 2 | Data Manipulation
Chapter 10 | Hotek, 2008

Learning objectives

- Insert, update, delete and merge data in tables

Inserting data manually

Common syntax for inserting data into a table:

```
INSERT INTO <table>
(<column list>)
VALUES (<value1>, <value2>, <value3>, …, n)
```

For example:

```
INSERT INTO Location.Address
(HouseNumberName, StreetName, City, Province, Country)
VALUES ('21 Montana House', 'Lynnwood Road', 'Pretoria', 'Gauteng',
'South Africa')
GO
```

Inserting data manually

Before SQL2008, only one row can be inserted at a time manually, for example

```
INSERT INTO Location.RegionLookupTables VALUES ('Gauteng', '01')
INSERT INTO Location.RegionLookupTables VALUES ('Mpumalanga', '02')
INSERT INTO Location.RegionLookupTables VALUES ('Western Cape', '03')
INSERT INTO Location.RegionLookupTables VALUES ('KwaZulu Natal', '04')
INSERT INTO Location.RegionLookupTables VALUES ('Northern Cape', '05')
INSERT INTO Location.RegionLookupTables VALUES ('Limpopo', '07')
INSERT INTO Location.RegionLookupTables VALUES ('Eastern Cape', '08')
INSERT INTO Location.RegionLookupTables VALUES ('North West', '09')
go
```

Inserting data manually…

But now can be done in blocks, for example…

```
INSERT INTO Location.RegionLookupTables VALUES
('Gauteng', '01'), ('Mpumalanga', '02'), ('Western Cape', '03'),
('KwaZulu Natal', '04'), ('Northern Cape', '05'), ('Free State', '06'),
('Limpopo', '07'), ('Eastern Cape', '08'), ('North West', '09')
go
```

Bulk Loading Data from other tables

```
INSERT INTO <target_table>
SELECT <columns>
FROM <source_table> (For example)
      use SQL2008SBS;

      INSERT INTO LookupTables.Country
      (CountryName)
      SELECT Name
      FROM AdventureWorks.Person.CountryRegion
      GO
```

NB: Ensure the datatypes in the source and target are similar and that the null value issues are taken care of.

SELECT INTO

Combines CREATE TABLE, INSERT, and SELECT into one to create a new table from values in another table. For example:

```
use AdventureWorks;

SELECT LastName, FirstName, Phone
INTO HomeAffairs.People.PhoneContact
FROM Person.Contact
```

UPDATING existing data in a table

- Changes existing data in a table or view.
- Most simple form utilises UPDATE and SET clauses.

For example:

```
UPDATE People.PhoneContact
SET Phone = '012 420 5422'
WHERE FirstName LIKE 'Gu%'
```

Deleting data

- The DELETE statement removes rows from a table or view. Simplest form is;

- DELETE FROM <table_name>

The statement can also be confined by using the WHERE clause in addition

For example:

```
DELETE FROM People.PhoneContact
WHERE Phone LIKE '012%'
```

Delete data

- The TRUNCATE TABLE statement is a fast, efficient method of deleting all rows in a table. TRUNCATE TABLE is similar to the DELETE statement without a WHERE clause. However, TRUNCATE TABLE is faster and uses fewer system and transaction log resources

Simplest form is:

```
TRUNCATE TABLE  [ { database_name.[ schema_name ].
| schema_name . } ]    table_name  [ ; ]
```

Exercise

- Create a database – VideoShop
- Create accompanying tables
- Insert data into the tables
- Delete data from table
- Update data in table