

DML STATEMENTS

Learning Goals

**By the end of this lecture
students should be able to:**

Describe each data manipulation language
(DML) statement

Sql Insert into Statement

```
INSERT INTO agents VALUES ("A001","Jodi","London",12,"075-1248798");
```

agent_code	agent_name	working_area	commission	phone_no
A001	Jodi	London	12	075-1248798

Table : agents

Insert rows into a table

Update rows in a table

Delete rows from a table



Table of contents

- **Insert Statement**
- **Update Statement**
- **Delete Statement**
- **Select Statement**

Section1


INSERT STATEMENT

INSERT Statements (1/3)

- The **INSERT INTO** statement is used to add one or more rows to a table or a view

Sql Insert into Statement

```
INSERT INTO agents VALUES ("A001","Jodi","London",.12,"075-1248798");
```



agent_code	agent_name	working_area	commission	phone_no
A001	Jodi	London	.12	075-1248798

Table : agents

INSERT Statements (2/3)

- **Syntax:**

(1) Inserting data to all columns

```
INSERT INTO table_name  
VALUES (value1,value2,value3,...);
```

Ex1: `USE Fsoft_Training`
`INSERT INTO` `dbo.Persons`
`VALUES (1,'Tom', 'B. Erichsen','Skagen 21','Stavanger')`

(2) Inserting data to selected columns

```
INSERT INTO table_name(column1,column2,column3,...)  
VALUES (value1,value2,value3,...);
```

Ex2: `USE Fsoft_Training`
`INSERT INTO` `dbo.Customer (CustomerName, City, Country)`
`VALUES ('Cardinal', 'Stavanger', 'Norway');`

■ Demo

- ✓ Inserting data to selected columns
- ✓ Inserting data to all columns with identity column
- ✓ Insert many rows at one time

- **Always check data in various cases:**

- ✓ Normal/Abnormal
- ✓ Invalid data type
- ✓ Special characters: ~!@#\$%^&*()_
- ✓ Special string characters: ' char(10) char(13) tab space
- ✓ Max length, Max/Min value
- ✓ Duplicated value in UNIQUE constraints

Section2

UPDATE STATEMENT

UPDATE Statement (1/2)

- The **UPDATE** statement is used to changes existing data in a table or view

SQLQuery1.sql

```
1 SELECT TOP 5 * FROM Sales.CurrencyRate
2 GO
3
4 UPDATE Sales.CurrencyRate
5 SET AverageRate = AverageRate + 0.01,
6     EndOfDayRate = EndOfDayRate + 0.01
7 GO
8
9 SELECT TOP 5 * FROM Sales.CurrencyRate
10
```

Results Messages Before Update

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate
1	1	2001-07-01 00:00:00.000	USD	ARS	1.00	1.0002
2	2	2001-07-01 00:00:00.000	USD	AUD	1.5491	1.55
3	3	2001-07-01 00:00:00.000	USD	BRL	1.9379	1.9419
4	4	2001-07-01 00:00:00.000	USD	CAD	1.4641	1.4683
5	5	2001-07-01 00:00:00.000	USD	CNY	8.2781	8.2784

After Update

	CurrencyRateID	CurrencyRateDate	FromCurrencyCode	ToCurrencyCode	AverageRate	EndOfDayRate
1	1	2001-07-01 00:00:00.000	USD	ARS	1.01	1.0102
2	2	2001-07-01 00:00:00.000	USD	AUD	1.5591	1.56
3	3	2001-07-01 00:00:00.000	USD	BRL	1.9479	1.9519
4	4	2001-07-01 00:00:00.000	USD	CAD	1.4741	1.4783
5	5	2001-07-01 00:00:00.000	USD	CNY	8.2881	8.2884

- Best Practice**

- ✓ Use the **@@ROWCOUNT** function to return the number of inserted rows to the client application.

UPDATE Statement (2/2)

- **Syntax:**

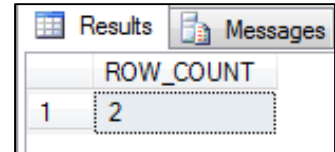
```
UPDATE table_name  
SET column1=value1,column2=value2,...  
WHERE some_column=some_value;
```

Notice the WHERE clause in the SQL UPDATE statement!

The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated!

- **Ex:**

```
USE Fsoft_Training  
UPDATE dbo.Customer  
SET PostalCode = '4006'  
WHERE Country = 'Norway'  
SELECT @@ROWCOUNT AS ROW_COUNT
```



Results		Messages
ROW_COUNT		
1	2	

Section3

DELETE STATEMENT

DELETE Statement (1/2)

- Removes one or more rows from a table or view

CustomerId	CustomerName	ContactName
1	Alfreds Futterkiste	Maria Anders
2	Around the Horn	Thomas Hardy
3	Berglunds snabbköp	Christina Berglund
4	Antonio Moreno	Antonio Moreno
5	Ana Trujillo	Ana Trujillo

- Best Practice:**

To delete all the rows in a table, use TRUNCATE TABLE. TRUNCATE TABLE is faster than DELETE and uses fewer system and transaction log resources.

TRUNCATE TABLE has restrictions, for example, the table cannot participate in replication



DELETE Statement (2/2)

- **Syntax:**

```
DELETE FROM table_name  
WHERE some_column=some_value;
```

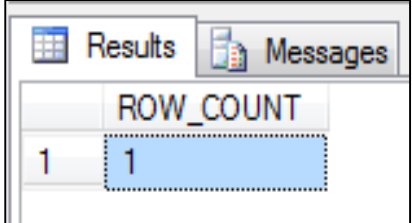
- **Notice the WHERE clause in the SQL DELETE statement!**

The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted!

Please note that the DELETE FROM command cannot delete any rows of data that would violate FOREIGN KEY or other constraints.

- **Ex:**

```
USE Fsoft_Training  
DELETE dbo.Customer  
WHERE Country = 'Germany'  
SELECT @@ROWCOUNT AS ROW_COUNT
```



Results		Messages	
	ROW_COUNT		
1	1		

Section4

SELECT STATEMENT

SELECT Statement (1/4)

- Retrieves rows from the database and enables the selection of one or many rows or columns from one or many tables



SELECT Statement (2/4)

■ Syntax:

SELECT [ALL/DISTINCT/TOP [WITH TIES]] <Column name1>, <Column name2>

FROM <Table name>

[**WHERE** <Search condition>]

[**GROUP BY** grouping columns]

[**HAVING** search condition]

[**ORDER BY** sort specification]

Ex1: **USE** AdventureWorks

GO

SELECT ProductID, Name

FROM Production.Product

ORDER BY Name **ASC**;

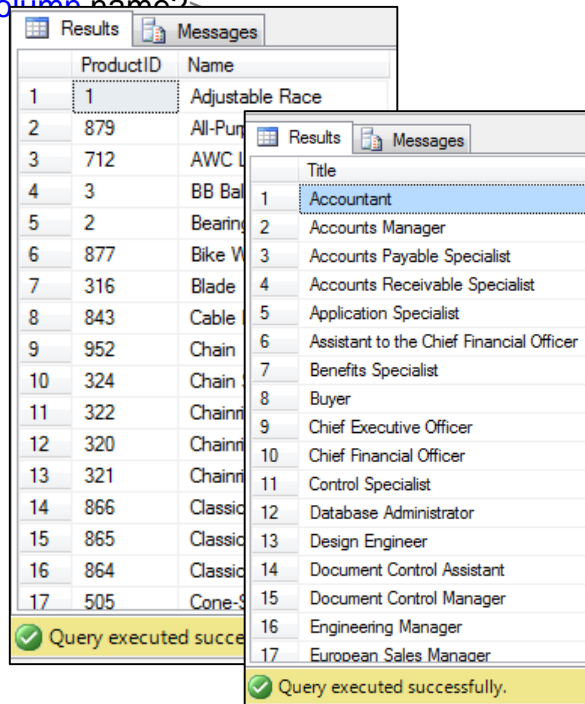
(504 rows)

Ex2: **SELECT DISTINCT** E.Title

FROM HumanResources.Employee E

ORDER BY E.Title;

(67 rows)



ProductID	Name
1	Adjustable Race
2	All-Purpose Bearing
3	AWC Ball Bearings
4	BB Ball Bearings
5	Bearing
6	Bike Wheel
7	Blade
8	Cable
9	Chain
10	Chainring
11	Chainring
12	Chainring
13	Chainring
14	Classic
15	Classic
16	Classic
17	Cone-Sprocket

Title
Accountant
Accounts Manager
Accounts Payable Specialist
Accounts Receivable Specialist
Application Specialist
Assistant to the Chief Financial Officer
Benefits Specialist
Buyer
Chief Executive Officer
Chief Financial Officer
Control Specialist
Database Administrator
Design Engineer
Document Control Assistant
Document Control Manager
Engineering Manager
European Sales Manager

SELECT Statement (3/4)

- The **SELECT INTO** statement selects data from one table and inserts it into a different table.

- **Syntax:**

```
SELECT *  
INTO new_table_name  
FROM old_tablename
```

- **Tip:**

- ✓ The SELECT INTO statement can also be used to create a new, empty table using the schema of another. Just add a WHERE clause that causes the query to return no data:

```
SELECT *  
INTO newtable  
FROM table1  
WHERE 1=0;
```

SELECT Statement (4/4)

- **SQL Alias syntax:**

- ✓ *For table*

- ```
SELECT column_name(s)
FROM table_name AS alias_name
```

- ✓ *For Column(s)*

- ```
SELECT column_name AS alias_name  
FROM table_name
```

- **Ex:**

- ```
USE AdventureWorks
GO
SELECT c.CustomerID, s.Name
FROM Sales.Customer AS c
JOIN Sales.Store AS s
ON c.CustomerID = s.SalesPersonID
```

# Summary

- Insert Statement
- Delete Statement
- Select Statement