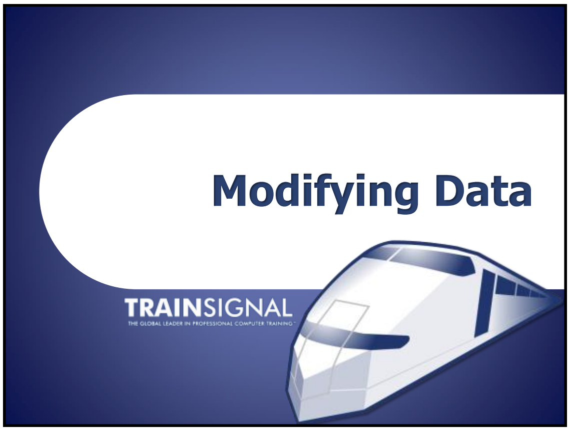
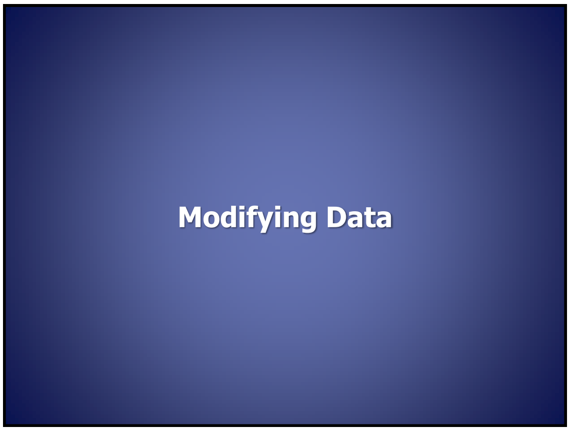
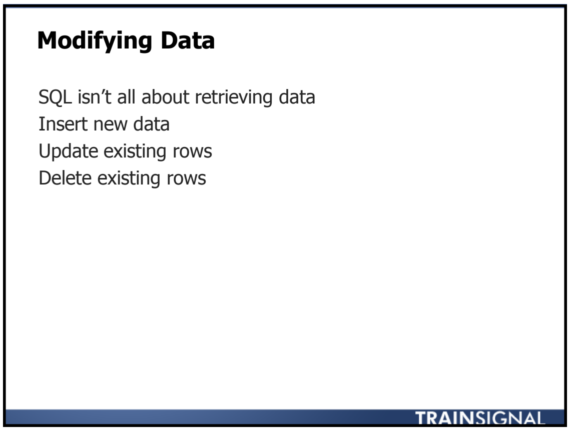


Modifying Data

TRAINSIGNAL
THE GLOBAL LEADER IN PROFESSIONAL COMPUTER TRAINING™



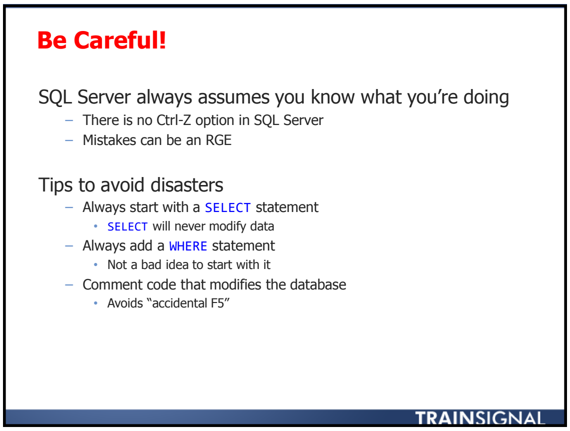
Modifying Data



Modifying Data

SQL isn't all about retrieving data
Insert new data
Update existing rows
Delete existing rows

TRAINSIGNAL



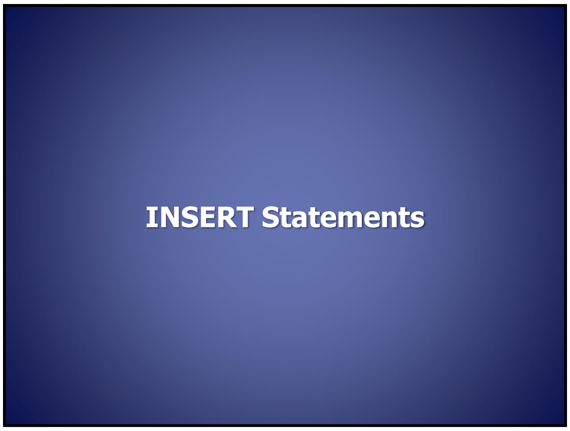
Be Careful!

SQL Server always assumes you know what you're doing

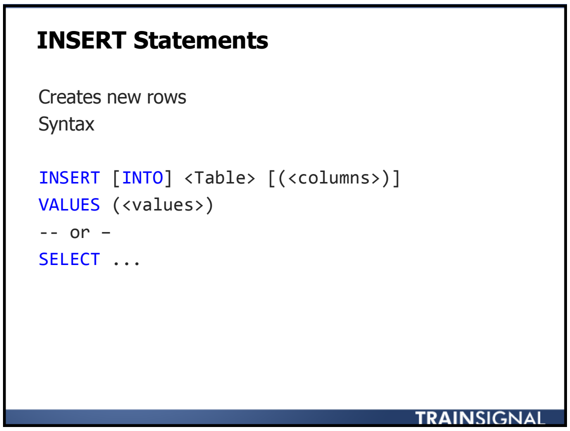
- There is no Ctrl-Z option in SQL Server
- Mistakes can be an RGE

Tips to avoid disasters

- Always start with a `SELECT` statement
  - `SELECT` will never modify data
- Always add a `WHERE` statement
  - Not a bad idea to start with it
- Comment code that modifies the database
  - Avoids "accidental F5"



INSERT Statements



INSERT Statements

Creates new rows

Syntax

```
INSERT [INTO] <Table> [(<columns>)]
VALUES (<values>)
-- or –
SELECT ...
```

Providing Data

Column list is optional

- If provided, you must provide data in that order
- If not provided, you must provide data in order of table
  - Skip any column marked as "IDENTITY" (auto-count)
  - Generally not a best practice

Special values

- DEFAULT keyword to indicate you wish to use default value
- NULL keyword for null values

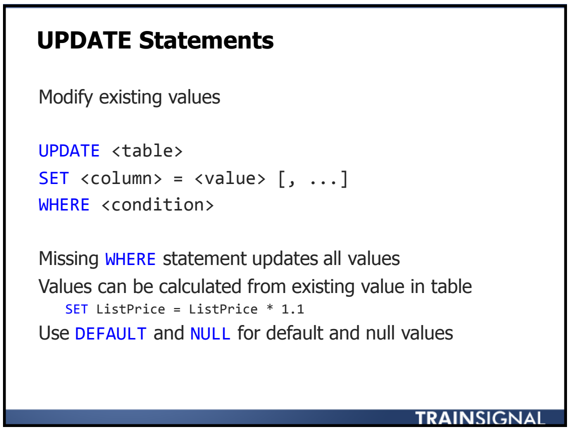
Providing Data

Row constructors

- Provide multiple rows in one INSERT statement
- Separate value lists with commas

```
INSERT ...
VALUES (<values>)
     , (<values>)
```

UPDATE Statements



UPDATE Statements

Modify existing values

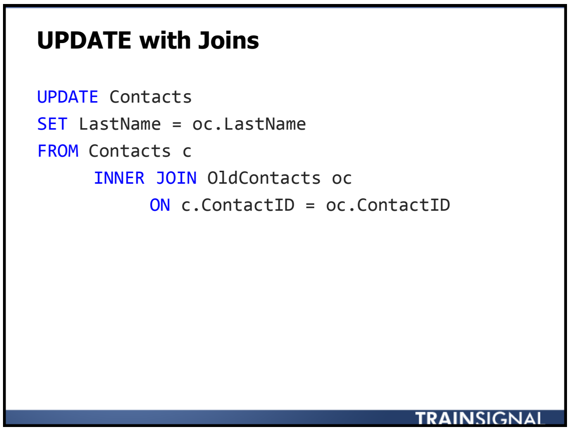
```
UPDATE <table>
SET <column> = <value> [, ...]
WHERE <condition>
```

Missing `WHERE` statement updates all values

Values can be calculated from existing value in table

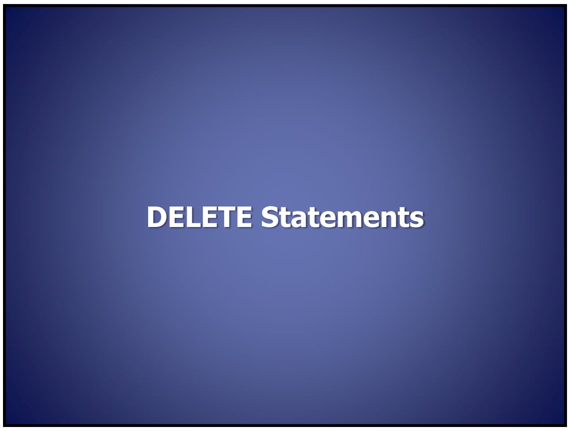
```
SET ListPrice = ListPrice * 1.1
```

Use `DEFAULT` and `NULL` for default and null values

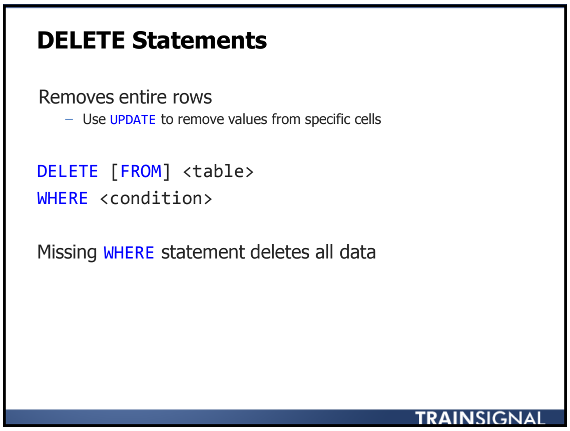


UPDATE with Joins

```
UPDATE Contacts
SET LastName = oc.LastName
FROM Contacts c
     INNER JOIN OldContacts oc
          ON c.ContactID = oc.ContactID
```



DELETE Statements



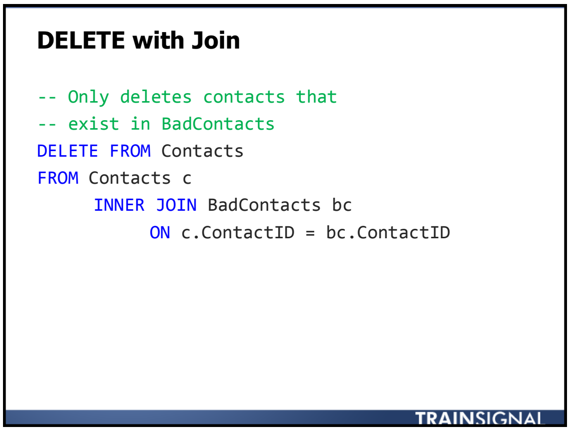
DELETE Statements

Removes entire rows

- Use UPDATE to remove values from specific cells

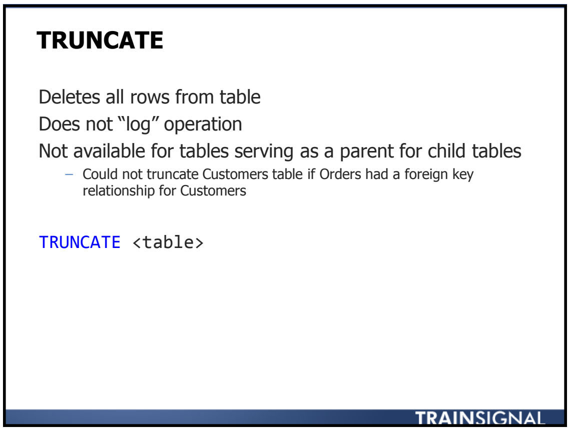
```
DELETE [FROM] <table>
WHERE <condition>
```

Missing WHERE statement deletes all data



DELETE with Join

```
-- Only deletes contacts that
-- exist in BadContacts
DELETE FROM Contacts
FROM Contacts c
     INNER JOIN BadContacts bc
          ON c.ContactID = bc.ContactID
```



TRUNCATE

Deletes all rows from table

Does not “log” operation

Not available for tables serving as a parent for child tables

- Could not truncate Customers table if Orders had a foreign key relationship for Customers

```
TRUNCATE <table>
```