

LESSON 3.3

98-364 Database Administration Fundamentals

# Update Data

## Lesson Overview

In this lesson, you will learn about:

- UPDATE
- CASE

## UPDATE

- The `UPDATE` statement is used to update existing records in a table:

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

- Notice the `WHERE` clause in the `UPDATE` syntax. This clause specifies which record or records should be updated.
- If you omit the `WHERE` clause, all records will be updated.

## Simple UPDATE

- All rows can be affected if a `WHERE` clause is eliminated from an `UPDATE` statement
- If all the schools in a **Colleges** table move their main campus to Phoenix, AZ, this example shows how the **Colleges** table could be updated:

```
UPDATE Colleges SET city = ' Phoenix', state = 'AZ'
```

- This example changes the Fees of all the **Colleges** to `NULL`:

```
UPDATE Colleges SET Fees = NULL
```

- You can compute values in an update. This example doubles all fees in the **Colleges** table:

```
UPDATE Colleges SET Fees = Fees * 2
```

## UPDATE statement with a WHERE clause

- The `WHERE` clause specifies the rows to update.
- This example shows how to update a **Colleges** table for all Mesa, AZ colleges when a ZIP code changes:

```
UPDATE Colleges
```

```
SET ZIP = '86689-1234'
```

```
WHERE state = 'AZ' AND city = 'Mesa'
```

## UPDATE statement with data from another table

- This example modifies the `ytd_enrollment` column in a **Colleges** table to reflect the most recent enrollment in an **Enrollment** table:

```
UPDATE Colleges SET ytd_enrollment =  
    Colleges.ytd_Enrollment + Enrollment.qty  
FROM Colleges, Enrollment  
WHERE Colleges.subject_id = Enrollment.subject_id  
AND Enrollment.date = (SELECT AX(Enrollment.date)  
FROM Enrollment)
```

- This assumes that only one enrollment total is recorded for a subject on a given date. If more than one total is recorded on the same day, only one will be recorded. This is because a single `UPDATE` statement never updates the same row twice.

## CASE

- CASE can be used in Structured Query Language (SQL) anywhere an expression can be used. It is a conditional statement providing if/then/else logic for SELECT, WHERE, HAVING, IN, DELETE, UPDATE, etc.
- It creates when-then-else functionality (WHEN this condition is met, THEN do this).

### CASE

```
WHEN the condition THEN result_1  
ELSE result_2  
END
```

## The CASE Clause: Two Forms

```
CASE  
WHEN condition THEN result_1  
ELSE result_2  
END
```

```
CASE the expression  
WHEN the condition THEN result_1  
ELSE result_2  
END
```

- The **WHEN** clause can be repeated any number of times.
- The **ELSE** clause can be omitted.
- The value of the **CASE** expression is the value of the first **WHEN** clause that is true. If none are true, the result is the **ELSE**.
- If there is no **ELSE** clause, then the result is **NULL**.
- All expressions must be compatible data types.



## UPDATE Using CASE

```
UPDATE class_info
SET fee = CASE
WHEN (fee < 5 AND grade > "D") THEN fee * 1.25
WHEN (fee < 5 AND grade < "B") THEN fee * 1.15
ELSE fee= 7 END
```

- This CASE expression combines multiple WHEN clauses and Boolean logic (AND) to set the current fee.
- This example also shows that a calculate function (\*) can be placed within the UPDATE CASE structure.

## Lesson Review

1. What is the purpose of the UPDATE statement?
2. Describe the CASE statement.
3. Give an example of the CASE statement that fits the description you just gave.