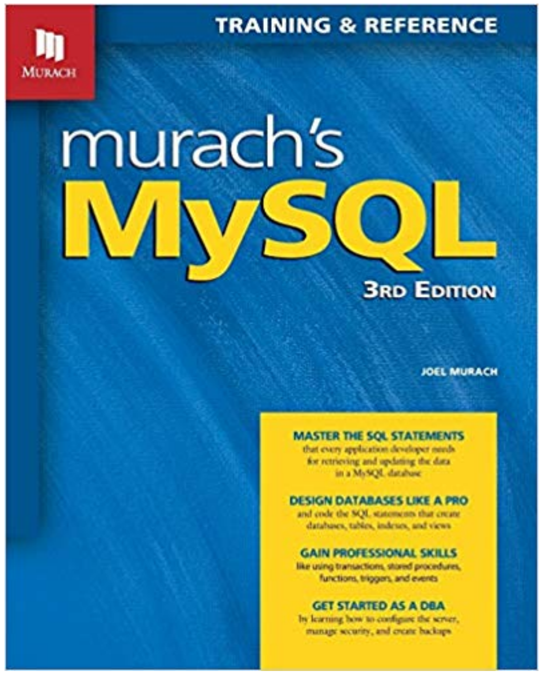
INSERTING, UPDATING and DELETING tuples
The CUD operations

Topic 3

Lesson 9 – changing data tuples in the database

Chapter 5 Murach's MySQL


TRAINING & REFERENCE
MURACH
murach's
MySQL
3RD EDITION
JOEL MURACH
MASTER THE SQL STATEMENTS that every application developer needs for retrieving and updating the data in a MySQL database
DESIGN DATABASES LIKE A PRO and code the SQL statements that create databases, tables, indexes, and views
GAIN PROFESSIONAL SKILLS like using transactions, stored procedures, functions, triggers, and events
GET STARTED AS A DBA by learning how to configure the server, manage security, and create backups

Database CRUD operations

- A database management system must provide a user with the ability to CREATE data, READ data, UPDATE data and DELETE delete.
- The INSERT command allows users to **C**REATE tuples in the database. In a relational database, we CREATE tuples in the database.
- The SELECT commands allows a user to **R**EAD data from the database. We have looked at this command extensively.
- The UPDATE command allows users to **U**PDATE tuples in the database. A user can update multiple tuples with one command.
- The DELETE command allows users to **D**ELETE tuples in the database

INSERT Operation in SQL

INSERT allows you to INSERT tuples into an already existing table.

The structure of the table and the data types for the fields listed in the column_list must align with the tuples being inserted into the table. Column_list can be a subset of the fields in the table

You can INSERT multiple tuples with one INSERT command by providing a comma separated list of tuples.

```
INSERT [INTO] table_name [(column_list)]
VALUES (expression_1[, expression_2]...)[,
       (expression_1[, expression_2]...)]...
```

EXAMPLE: INSERT a tuple

```
 INSERT INTO student (name, school,
     credits_req, yr_grad)
     VALUES (‘Smith’, ’Khoury’, 128, 2025);
```

Column definitions for the Student table:

```
id                INT PRIMARY KEY AUTO_INCREMENT,
name              VARCHAR(50)     NOT NULL,
school            VARCHAR(50) DEFAULT ’Undefined’,
credits_earned  INT                 DEFAULT 0,
credits_req      INT            ,
yr_grad                INT NOT NULL
```

EXAMPLE 2: INSERT multiple tuples

```
INSERT INTO student (name, school,
    credits_earned, credits_req, yr_grad)
    VALUES (‘Daria’, ’Khoury’, 4, 128, 2025),
           (‘Smith’, ’Khoury’, 12, 128, 2025);
```

Column definitions for the Student table:

```
id             INT PRIMARY KEY AUTO_INCREMENT,
name           VARCHAR(50)    NOT NULL,
school         VARCHAR(50) DEFAULT ’Undefined’,
credits_earned  INT                DEFAULT 0,
credits_req     INT          ,
yr_grad            INT NOT NULL
```

INSERT data using a SELECT statement

- You can also retrieve the data you wish to insert using a SELECT statement.
- The syntax is:

```
   INSERT [INTO] table_name [(column_list)]
select_statement
```

- EXAMPLE: INSERT graduated students into the alumni table
- INSERT INTO alumni (id, name, yr_grad)

```
            SELECT id, name, yr_grad
            FROM student WHERE id = 2019;
```

UPDATE Command

- Like the CREATE command, the UPDATE command can be used to UPDATE many different types of database objects.
- To change the values of data in the database, use the “UPDATE table_name” command. Since it works at the table level, you will typically use a WHERE command to limit the tuples that should be updated.
- You can change multiple columns with one UPDATE command.
- The result from the UPDATE command is the number of tuples or rows changed. Example: **`(1 row affected)`**
- Always, create a SELECT command that retrieves the tuples you want to update first, to make sure you are retrieving the tuples you want to change.

UPDATE syntax

```
UPDATE table_name
SET column_name_1 = expression_1
 [, column_name_2 = expression_2]...
[WHERE search_condition]
```

ID	Name	School	Credits_Earned	Credits_Req	yo_grad
7	Haines	Khoury	32	120	2021
8	Lee	D’Amore McKim	64	128	2020
9	Frred	D’Amore McKim	50	120	2020

EXAMPLE: UPDATE command

```
UPDATE student SET name = ‘Smythe’
WHERE name=‘Smith’;
```

All Smith values become Smythe.

ID	Name	School	Credits_Earned	Credits_Req	yr_o_grad
1	Smith	Khoury	32	120	2019
2	Shah	D’Amore McKim	64	128	2019
3	Li	Khoury	50	120	2020

ID	Name	School	Credits_Earned	Credits_Req	yr_o_grad
1	Smythe	Khoury	32	120	2019
2	Shah	D’Amore McKim	64	128	2019
3	Li	Khoury	50	120	2020

EXAMPLE: UPDATE multiple fields

```
UPDATE student SET name = ‘Smythe’,
                        credits_earned = 40,
                          WHERE name=‘Smith’;
```

ID	Name	School	Credits_Earned	Credits_Req	yr_o_grad
1	Smith	Khoury	32	120	2019
2	Shah	D’Amore McKim	64	128	2019
3	Li	Khoury	50	120	2020

ID	Name	School	Credits_Earned	Credits_Req	yr_o_grad
1	Smythe	Khoury	40	120	2019
2	Shah	D’Amore McKim	64	128	2019
3	Li	Khoury	50	120	2020

SAFE UPDATE mode in MySQL workbench

- By default, MySQL Workbench runs in safe update mode.
- Safe update mode prevents you from updating rows if the WHERE clause is omitted or doesn't refer to a primary key or foreign key column.
- You can turn safe update mode off by selecting the Edit→Preferences command, selecting the SQL Editor node, changing the "Safe Updates" option, and restarting MySQL Workbench.
- If you turn off safe update mode and omit the WHERE clause, **all rows in the table will be updated.**

DELETE command

- The delete command removes tuples from a table.
- You typically provide a WHERE clause to limit the tuples you are deleting from the table.
- Its behavior is also affected by SAFE UPDATE mode.
- Its syntax is:

```
DELETE FROM table_name
[WHERE search_condition]
```

Example: DELETE a tuple

```
DELETE FROM student where id = 3;
```

ID	Name	School	Credits_Earned	Credits_Req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

ID	Name	School	Credits_Earned	Credits_Req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128

Example: DELETE multiple tuples

```
DELETE FROM student where id IN (2,3);
```

ID	Name	School	Credits_Earned	Credits_Req
1	Smith	Khoury	32	120
2	Shah	D’Amore McKim	64	128
3	Li	Khoury	50	120

ID	Name	School	Credits_Earned	Credits_Req
1	Smith	Khoury	32	120

Summary

In this module you learned:

- DELETE operation
- INSERT operation
- UPDATE operation