Software Craftsmanship Guild

526 S. Main St, Suite 609

Akron, OH 44311

Intro to Data Manipulation Language (DML)

Software Craftsmanship Guild

Lesson Goals

- Learn how to insert, update, and delete information from database tables

First, let's make a sandbox

```
Create Database MovieCatalogue
GO

CREATE TABLE Movie
(
    MovieID int Identity(1,1) NOT NULL Primary Key,
    Title varchar(50) NOT NULL,
    RunTime int NULL,
    Rating varchar(5) NULL,
    ReleaseDate Date NULL
)
```

Inserting Data

- Every table starts out empty (of course). The only way to populate a table is to *insert* data into the table
- When we insert data we typically provide a list of columns we want to populate followed by the list of values we would like to put into the columns
  - Do note that the order matters, if you don't list the values in the same order as the columns at best you will get an error and at worst the data will go into the wrong spot!

Inserting a Movie

- Notice we don't put the MovieID into the list. It is an identity field so the database will create the id key. If we tried to pass it a value it would fail.

```
INSERT INTO Movie (Title, RunTime, Rating, ReleaseDate)
VALUES ('Caddyshack', 98, 'R', '7-25-1980')

SELECT * FROM Movie
```

Null Columns

Columns marked as null are optional, and thus do not need to be specified in the insert statement. Both of these inserts are valid:

```
-- not sure what the runtime is right now
INSERT INTO Movie (Title, RunTime, Rating, ReleaseDate)
VALUES ('Ghostbusters', NULL, 'PG', '06/08/1984')

-- we can omit columns we want to be null
INSERT INTO Movie (Title, Rating)
VALUES ('Groundhog Day', 'PG')

SELECT * FROM Movie
```

Row Constructors

- This only works in SQL 2008 or higher

```
INSERT INTO Movie (Title, RunTime, Rating)
VALUES ('The Lion King', 89, 'G'),
('Beauty and the Beast', 84, 'G'),
('Aladdin', 90, 'G')

SELECT * FROM Movie
```

Lab - Create a table Customer

ColumnName	Type	Attribute
CustomerID	Int	Primary Key
CustomerType	Varchar(30)	Not null
FirstName	Varchar(20)	Null
LastName	Varchar(30)	Null
CompanyName	Varchar(30)	Null

Add this data

	CustomerID	CustomerType	FirstName	LastName	CompanyName
1	1	Consumer	Mark	Williams	NULL
2	2	Consumer	Lee	Young	NULL
3	3	Consumer	Patricia	Martin	NULL
4	4	Consumer	Mary	Lopez	NULL
5	5	Business	NULL	NULL	MoreTechnology.com

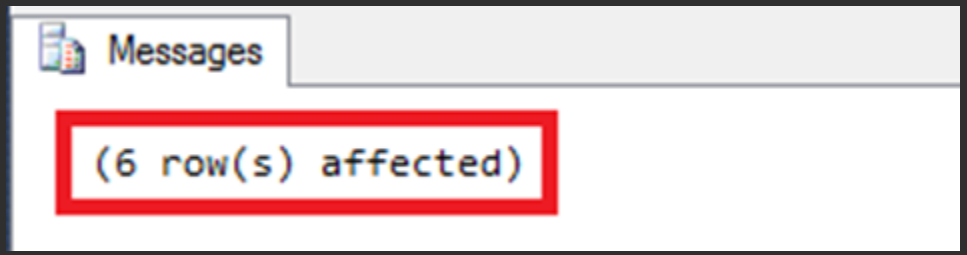
Updating Data

- When we want to change existing data in the database we use the UPDATE keyword.
- We must be EXTRA careful when writing update statements, as we will see in the following example

A bad update

Let's say we found the runtime of Ghostbusters to be 107 minutes…

```
UPDATE Movie
    SET RunTime = 107
```



Oh noes!

Hope we have a backup…

	MovieID	Title	RunTime	Rating	ReleaseDate
1	1	Caddyshack	107	R	1980-07-25
2	2	Ghostbusters	107	PG	1984-06-08
3	3	Groundhog Day	107	PG	NULL
4	4	The Lion King	107	G	NULL
5	5	Beauty and the Beast	107	G	NULL
6	6	Aladdin	107	G	NULL

Let's fix the data properly, specifying the key in a where clause

```
update movie
    set runtime = 89
where movieid = 4 -- lion king

update movie
    set runtime = 84
where movieid = 5 -- beauty

update movie
    set runtime = 90
where movieid = 6  -- aladdin
```

We can write complex where statements

- Just FYI, we don't have to update on just the primary key, we could update all customers with a certain zip code to a new zip code for example.
- Protip: Always build a select statement to select the data you want to update first and make sure it looks right before doing the update. The more rows that will be updated, the more careful you should be.

Giving Sally's Employees A Raise

- We can update data in one table based on the criteria in another table. It's a slightly more complicated query.
- As an example, let's give Sally's (EmployeeID = 11) employees a $1,000 raise.

```
SELECT *
FROM Employee as e
    INNER JOIN PayRates AS pr ON e.EmpID = pr.EmpID
WHERE ManagerID = 11

UPDATE pr
    SET YearlySalary = YearlySalary + 1000
FROM Employee as e
    INNER JOIN PayRates AS pr ON e.EmpID = pr.EmpID
WHERE ManagerID = 11
```

My SQL

```
UPDATE Payrates AS pr
INNER JOIN Employee AS e ON pr.EmpID = e.EmpID
SET YearlySalary = YearlySalary + 1000
WHERE ManagerID = 11;
```

Notice that 6 rows were affected…

- But only the two employees with a YearlySalary got the raise
- We cannot do math on a null
  - Anything compared to null is false
  - Any math on a null is null

A better way to write that query

```
UPDATE pr
    SET YearlySalary = YearlySalary + 1000
FROM Employee as e
    INNER JOIN PayRates AS pr ON e.EmpID = pr.EmpID
WHERE ManagerID = 11 AND YearlySalary IS NOT NULL
```

Lab Exercises

1. Sally (employeeID 11) is getting married, change her last name to Green
2. All the employees in the Spokane location are becoming contractors, update their status field to External
3. The location for Seattle has a typo, update the street field to read 111 1st Ave
4. A new policy requires that grants for employees in Boston be made for $20,000. There are two Boston records which aren't set to $20,000. Please fix them!

Deleting Data

- Deletes are similar to updates in that if you don't specify a where filter the entire table will be deleted
- SQL Server will not allow you to delete a row that is referenced by another table as its foreign key. You must delete all the relations first, then delete the primary row

```
DELETE FROM Movie
```

Now we need to put data back in…

```
INSERT INTO Movie (Title, Runtime, Rating)
VALUES
    ('A-List Explorers', 96,'PG-13'),
    ('Bonker Bonzo', 75,'G'),
    ('Chumps to Champs', 75,'PG-13'),
    ('Dare or Die', 110,'R'),
    ('EeeeGhads', 88,'G')
```

Delete only long movies

```
DELETE Movie
WHERE RunTime > 90
```

Multiple Table Deletes

- There are no employees in Chicago, so this is a safe statement to run

```
DELETE e
FROM Employee AS e
    INNER JOIN Location AS l ON e.LocationID = l.LocationID
WHERE City = 'Chicago'
```

Lab Exercises

1. We are moving our G movies to a kids website, delete all the G rated movies
2. SWCCorp is not teaching long classes anymore for management, delete all records where the duration is more than 20 hours

Fin

- Next up- Table Maintenance!