Update - Text lecture

In previous lectures we learned about select like below:

```
SELECT title
     ,release_year
FROM movies
WHERE release_year IS NULL;
```

If we want to update or change the data, we use UPDATE instead of select:

Use a select to find what to update

```
SELECT *
FROM movies
WHERE Title = 'Fight Club';
```

Then run an update on the same selection criteria:

```
UPDATE movies
SET ReleaseYear = 1999
WHERE Title = 'Fight Club';
```

We can update multiple rows and multiple columns at the same time:

```
SELECT *
FROM movies
WHERE title IN ('DodgeBall: A True Underdog Story'
              ,'Along Came Polly'
              ,'Anchorman:The Legend of Ron Burgundy');
```

Rows matched:3 Changed:3

If you run this again, it will say matched: 3, Changed: 0

Use SELECT * to verify your WHERE before updating a table.

Update based on comparison - Text lecture

Let's create some data to work with:

```
CREATE TABLE CustBal (CustID int NOT NULL AUTO_INCREMENT
          ,CheckingBalance DECIMAL(8,2)
          ,SavingsBalance DECIMAL(8,2)
          ,CreditCardBalance DECIMAL(8,2)
          ,CreditCardTotalCredit DECIMAL(8,2)
          ,CreditCardAvailableCredit DECIMAL(8,2)
          ,TotalBalance DECIMAL(8,2)
          ,ActiveStatus boolean
          ,primary key (CustID));
```

```
INSERT INTO CustBal ( CheckingBalance
                    ,SavingsBalance
                    ,CreditCardBalance
                    ,CreditCardTotalCredit
                    ,CreditCardAvailableCredit
                    ,TotalBalance
                    ,ActiveStatus)
VALUES (835.27 , 2223.01 ,   123.09 ,  1000.00 , NULL , NULL , 1)
     ,(165.27 , 1223.01 ,     0.00 ,  1000.00 , NULL , NULL , 1)
     ,(899.27 , 3223.01 ,  9125.09 , 20000.00 , NULL , NULL , 1)
     ,(  0.00 ,    0.00 ,    00.00 ,     0.00 , NULL , NULL , 0)
     ,(123.27 , 4223.01 ,   123.00 ,   800.00 , NULL , NULL , 1)
     ,( 65.00 , 6223.01 , 21123.09 , 20000.00 , NULL , NULL , 1)
     ,(315.90 , 7223.01 ,     0.00 , 21000.00 , NULL , NULL , 1)
     ,(992.22 , 1923.01 , 20000.00 , 20000.00 , NULL , NULL , 1)
     ,( 75.21 , 4223.01 ,  1123.51 , 15000.00 , NULL , NULL , 1)
     ,(165.22 ,    0.01 , 15123.08 , 16000.00 , NULL , NULL , 0)
     ,(455.23 ,    0.00 ,  4123.00 ,  5000.00 , NULL , NULL , 1);
Let's say we wanted a process to update the customer accounts
1) Use Select Statements to plan out what you want to do
SELECT CustID
     ,CreditCardTotalCredit
     ,CreditCardBalance
     ,CreditCardAvailableCredit              AS OldCreditCardAvailableCredit
     ,CreditCardTotalCredit - CreditCardBalance AS
NewCreditCardAvailableCredit
FROM CustBal
WHERE ActiveStatus = 1;

SELECT CustID
     ,CheckingBalance
     ,SavingsBalance
     ,CreditCardBalance
     ,TotalBalance                                 AS OldTotalBalance
     ,CheckingBalance + SavingsBalance - CreditCardBalance  AS
NewTotalBalance
FROM CustBal
WHERE ActiveStatus = 1;
```

We could have combined them into one larger select, but breaking it apart makes it easier to read in the terminal window.

2) Put it together into an update statement

```
UPDATE CustBal
SET CreditCardAvailableCredit = CreditCardTotalCredit - CreditCardBalance
   ,TotalBalance              = CheckingBalance + SavingsBalance -
CreditCardBalance
WHERE ActiveStatus = 1;
```

Delete - Text lecture

Delete is just like Select but instead of returning rows with information, it removes them from the table. As a rule, always use a select statement to verify what you are about to remove.

Example, remove all of the inactive accounts from our CustBal table:

First, identify the inactive accounts:

```
SELECT *
FROM CustBal
WHERE ActiveStatus = 0;
```

To delete, replace SELECT * with DELETE and execute;

```
DELETE
FROM CustBal
WHERE ActiveStatus = 0;
```