IT-234 – database concepts

UNIT 5 – ENTERING DATA AND USING SQL COMMANDS TO QUERY EXISTING DATA

overview

You now have an established database, and tables have been implemented.

This unit will explore how to populate your tables with data.

The first method you will learn is simply to type the data in using the Microsoft SSMS Designer tools.

While typing is tedious and error prone, it is quick and will suffice for small tasks.

overview

Another technique is to import data from a file.

You will leverage a provided flat data file and data migration script to populate the normalized tables.

You will also analyze the data migration script to understand how it functions.

overview

▶ Once you have data in the tables, what can you do with it?

▶ This is the fundamental purpose of the database: To allow you to retrieve data.

▶ You will examine different SQL syntax for selecting data into a result set. There are many ways to limit and format the result set into exactly what you require.

overview

The quandary you must overcome is to determine if the result set has returned correct results or not.

A well-formed SQL query should return something.

But you will need to analyze the results to determine if what was returned is what you asked for and if what you asked for is really what you wanted.

Sounds confusing at first, but do not worry; a little practice is all you will need.

overview

After completing this unit, you should be able to:

- ➢ Use a variety of methods to populate a database table with data.

- ➢ Examine the data placed into the table using SQL queries.

Importing data

▶ Importing data into a new database can be done one of two ways:

1. You manually type information in the new database and all its various tables

2. You import data using queries.

Importing data

For this week's assignment, you have a CSV file that contains all the movies for your new database.

So, to experience both methods, you will start by entering one record from the Movies flat file dataset contained in the file named **Movies_Import_Temp.csv**.

Importing data

The record you will be entering is the "Arsenic and Old Lace" movie record since it is the first one in the Movies CSV file.

| Title	Director_FirstName	Director_LastName	Genre	Star_FirstName	Star_LastName	Rating	Producer_FirstName	Producer_LastName
Arsenic and Old Lace	Frank	Capra	Comedy	Cary	Grant	8.1		Warner Bros.

Importing data

▶ Since the Star, Director, Genre and Producer tables contain the Primary Keys (PKs) and the Movie table contains the Foreign Keys (FKs) for these 4 tables, you should enter the data into these four tables (Star, Director, Genre and Producer) BEFORE you enter information into the Movies table.

1. Open the *Microsoft SQL Server Management Suite (SSMS)* application.

2. Expand the **Databases** item in the **Object Explorer** window pane. Navigate to the Tables folder located under the **Movies_DB** database. Expand the **Tables** folder by clicking on . The tables contained in the **Movies_DB** database should appear.

SSMS MANUAL DATA ENTRY
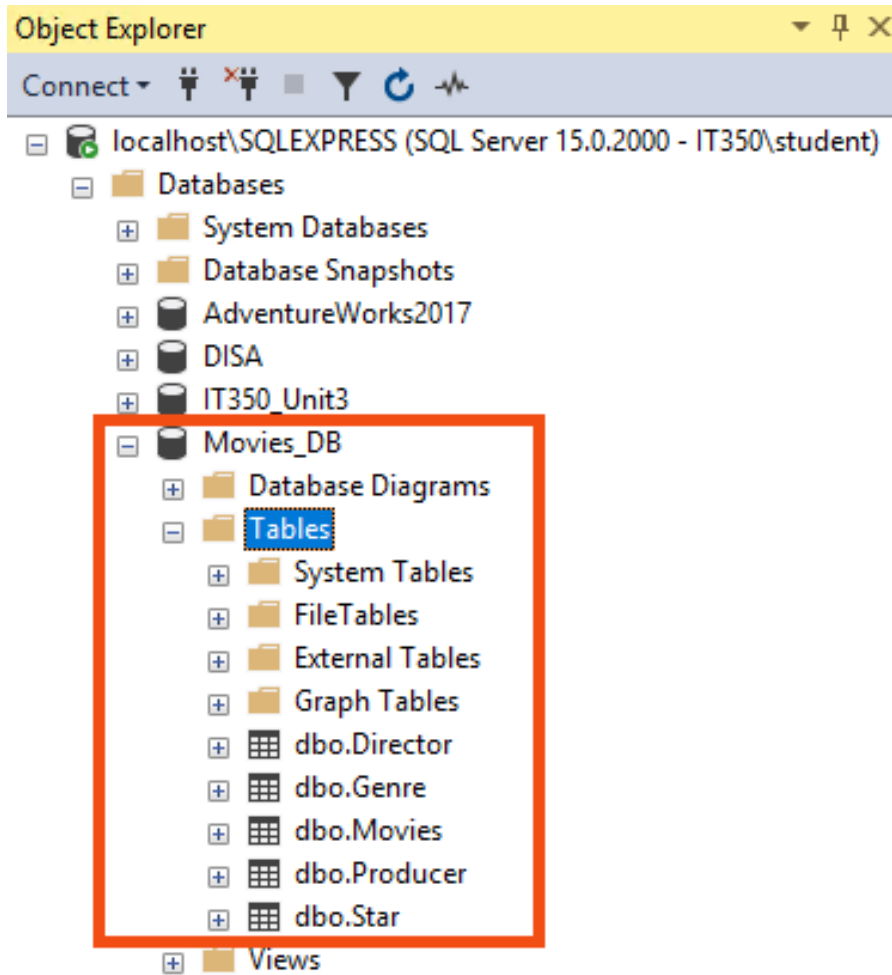
SSMS MANUAL DATA ENTRY
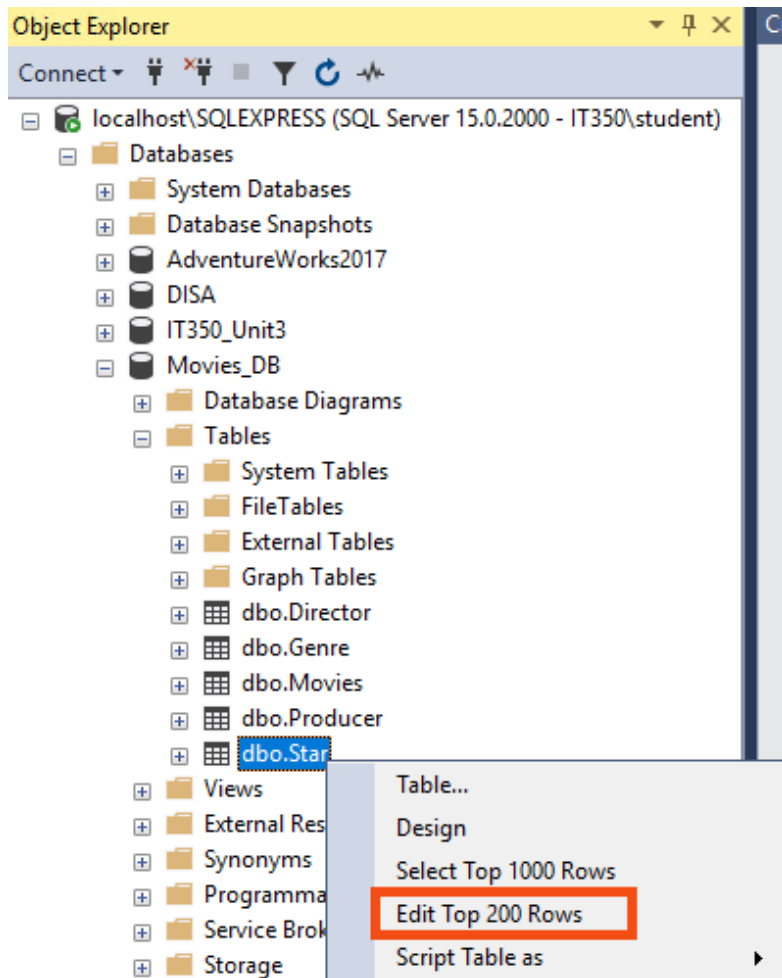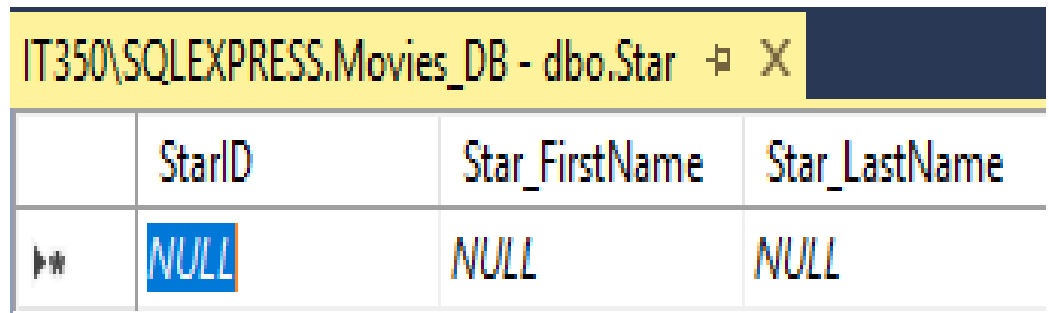
SSMS MANUAL DATA ENTRY

RIGHT-CLICK ON THE DBO.STAR TABLE. SELECT THE EDIT TOP 200 ROWS ITEM FROM THE RIGHT-CLICK MENU.

SSMS MANUAL DATA ENTRY
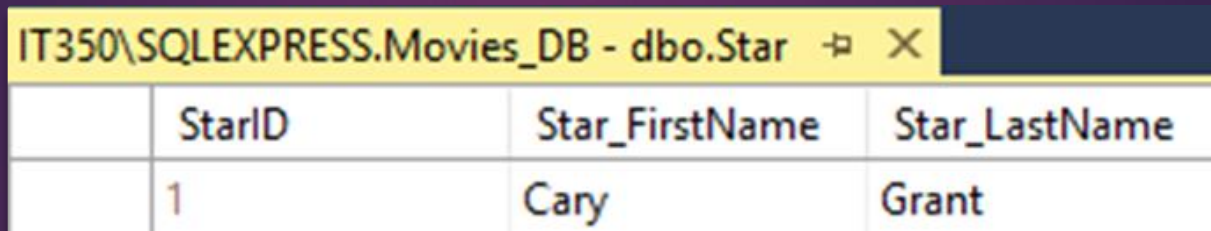
4. The contents of the **Star** table will appear in the right window pane of the Microsoft SSMS application. The table currently contains no data, so what you will see will look similar to the illustration below.

	StarID	Star_FirstName	Star_LastName
▶*	NULL	NULL	NULL

IT350\SQLEXPRESS.Movies_DB - dbo.Star

SSMS
MANUAL
DATA ENTRY

5. You will now manually enter the applicable "**Arsenic and Old Lace**" movie data into the **Star** table.

   ➢ Click in the cell directly below the **Star_FirstName** header. Enter **Cary** into that cell.

   ➢ Then press the **TAB** key on your keyboard to move into the cell below the **Star_LastName** header. Enter **Grant** into that cell.

   ➢ Press the **TAB** key again. The cell below the **StarID** header should now show a value of 1. The entire record should look like what is in the illustration on the next slide.

SSMS MANUAL DATA ENTRY

SSMS MANUAL DATA ENTRY

6. Repeat Steps 3-5 for the **Director**, **Genre**, and **Producer** tables. You will be entering the relevant data from the "**Arsenic and Old Lace**" record contained in the flat file dataset into these tables.

SSMS MANUAL DATA ENTRY

7.   Repeat Steps 3-5 for the **Movies** table. However, you will only be entering flat file data into the **Title** and **Rating** fields. The foreign key attributes (**StarID**, **DirectorID**, **ProducerID**, and **GenreID**) contained in the **Movies** table reference the associated primary keys in the other tables.

SSMS MANUAL DATA ENTRY

7. [continued] If Steps 3-6 were accomplished correctly, the StarID, DirectorID, ProducerID, and GenreID values in the "Arsenic and Old Lace" record entry for the Movies table should all be set to a value of 1.

	MovieID	Title	DirectorID	StarID	GenreID	ProducerID	Rating
▶	1	Arsenic and Ol...	1	1	1	1	8.1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SSMS MANUAL DATA ENTRY

8. Next you decide that this was a rather painful method.

   ➢ Very tedious

   ➢ In addition, manual of entry can be error prone (e.g., "fat fingering").

   ➢ An automated approach using scripts containing queries and data manipulation language can make life easier

INSERT STATEMENTS

The SQL INSERT INTO Statement is used to add new rows of data to a table in the database.

There are two basic syntaxes of the INSERT INTO statement

INSERT STATEMENTS

Insert Syntax Type 1

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

Here, column1, column2, column3, ...columnN are the names of the columns in the table into which you want to insert the data.

INSERT STATEMENTS

Insert Syntax Type 2

```
INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...valueN);
```

➢ You may not need to specify the column(s) name in the SQL query if you are adding values for all the columns of the table.

➢ But make sure the order of the values is in the same order as the columns in the table.

Basic select statement

Database tables are objects that stores all the data in a database.

In a table, data is logically organized in a row-and-column format which is similar to a spreadsheet.

In a table, each row represents a unique record, and each column represents a field in the record.

Basic select statement

▶ For example, a *customers* table contains customer data such as customer identification number, first name, last name, phone, email, and address information as shown below:

customer_id	first_name	last_name	phone	email	street	city	state	zip_code
1	Debra	Burks	NULL	debra.burks@yahoo.com	9273 Thorne Ave.	Orchard Park	NY	14127
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street	Campbell	CA	95008
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.	Redondo Beach	CA	90278
4	Daryl	Spence	NULL	daryl.spence@aol.com	988 Pearl Lane	Uniondale	NY	11553
5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.	Sacramento	CA	95820
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road	Fairport	NY	14450
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.	Buffalo	NY	14215
8	Jacquline	Duncan	NULL	jacquline.duncan@yahoo.com	15 Brown St.	Jackson Heights	NY	11372
9	Genoveva	Baldwin	NULL	genoveva.baldwin@msn.com	8550 Spruce Drive	Port Washington	NY	11050
10	Pamelia	Newman	NULL	pamelia.newman@gmail.com	476 Chestnut Ave.	Monroe	NY	10950

Basic select statement

▶ To query data from a table, you use the **SELECT** statement.

▶ The following illustrates the most basic form of the **SELECT** statement:

```
SELECT
    select_list

FROM
    schema_name.table_name;
```
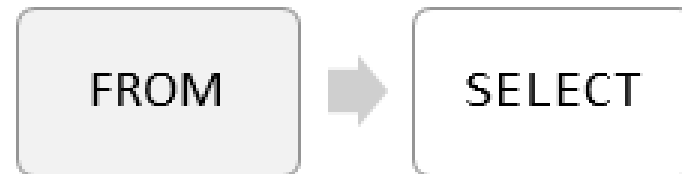
Basic select statement

▶ **SELECT** syntax:

  ➢ First, specify a list of comma-separated columns from which you want to query data in the **SELECT** clause.

  ➢ Second, specify the source table and its schema name on the **FROM** clause.

Basic select statement

When processing the **SELECT** statement, SQL Server processes the **FROM** clause first and then the **SELECT** clause even though the **SELECT** clause appears first in the query.

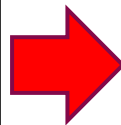| FROM	➡	SELECT

Basic select statement

To get data from all columns of a table, you can specify all the columns in the select list.

You can also use **SELECT *** as a shorthand to save some typing:

```
SELECT
    *
FROM
    sales.customers;
```

customer_id	first_name	last_name	phone	email	street	city	state	zip_code
1	Debra	Burks	NULL	debra.burks@yahoo.com	9273 Thorne Ave.	Orchard Park	NY	14127
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street	Campbell	CA	95008
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.	Redondo Beach	CA	90278
4	Daryl	Spence	NULL	daryl.spence@aol.com	988 Pearl Lane	Uniondale	NY	11553
5	Charlotte	Rice	(916) 381-6003	charlotte.rice@msn.com	107 River Dr.	Sacramento	CA	95820
6	Lyndsey	Bean	NULL	lyndsey.bean@hotmail.com	769 West Road	Fairport	NY	14450
7	Latasha	Hays	(716) 986-3359	latasha.hays@hotmail.com	7014 Manor Station Rd.	Buffalo	NY	14215
8	Jacquline	Duncan	NULL	jacquline.duncan@yahoo.com	15 Brown St.	Jackson Heights	NY	11372
9	Genoveva	Baldwin	NULL	genoveva.baldwin@msn.com	8550 Spruce Drive	Port Washington	NY	11050
10	Pamelia	Newman	NULL	pamelia.newman@gmail.com	476 Chestnut Ave.	Monroe	NY	10950
11	Deshawn	Mendoza	NULL	deshawn.mendoza@yahoo.com	8790 Cobblestone Street	Monsey	NY	10952
12	Robby	Sykes	(516) 583-7761	robby.sykes@hotmail.com	486 Rock Maple Street	Hempstead	NY	11550

Basic select statement

The **WHERE** clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=
BETWEEN	Between a certain range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

Basic select statement

THE FOLLOWING OPERATORS CAN BE USED IN THE WHERE CLAUSE:

To filter rows based on one or more conditions, you use a **WHERE** clause as shown in the following example:

```
SELECT
    *
FROM
    sales.customers
WHERE
    state = 'CA';
```

customer_id	first_name	last_name	phone	email	street	city	state	zip_code
2	Kasha	Todd	NULL	kasha.todd@yahoo.com	910 Vine Street	Campbell	CA	95008
3	Tameka	Fisher	NULL	tameka.fisher@aol.com	769C Honey Creek St.	Redondo Beach	CA	90278
5	Charolette	Rice	(916) 381-6003	charolette.rice@msn.com	107 River Dr.	Sacramento	CA	95820
24	Corene	Wall	NULL	corene.wall@msn.com	9601 Ocean Rd.	Atwater	CA	95301
30	Jamaal	Albert	NULL	jamaal.albert@gmail.com	853 Stonybrook Street	Torrance	CA	90505
31	Williemae	Holloway	(510) 246-8375	williemae.holloway@msn.com	69 Cypress St.	Oakland	CA	94603
32	Araceli	Golden	NULL	araceli.golden@msn.com	12 Ridgeview Ave.	Fullerton	CA	92831
33	Deloris	Burke	NULL	deloris.burke@hotmail.com	895 Edgemont Drive	Palos Verdes Peninsula	CA	90274
40	Ronna	Butler	NULL	ronna.butler@gmail.com	9438 Plymouth Court	Encino	CA	91316
46	Monika	Berg	NULL	monika.berg@gmail.com	369 Vernon Dr.	Encino	CA	91316
47	Bridgette	Guerra	NULL	bridgette.guerra@hotmail.com	9982 Manor Drive	San Lorenzo	CA	94580

In this example, the query returns the customers who locate in California.

Basic select statement

Basic select statement

When the **WHERE** clause is available, SQL Server processes the clauses of the query in the following sequence: **FROM**, **WHERE**, and **SELECT**.

FROM	➡	WHERE	➡	SELECT

Basic select statement

A field with a NULL value is a field with no value.

If a field in a table is optional, it is possible to insert a new record or update a record without adding a value to this field.

Then, the field will be saved with a NULL value.

Basic select statement

A NULL value is different from a zero value or a field that contains spaces.

A field with a NULL value is one that has been left blank during record creation!

- NULL is the absence of a value

Basic select statement

It is not possible to test for NULL values with comparison operators, such as =, <, or <>.

We have to use the **IS NULL** and **IS NOT NULL** operators instead.

Basic select statement

IS NULL Syntax

```sql
SELECT column_names
FROM table_name
WHERE column_name IS NULL;
```

IS NOT NULL Syntax

```sql
SELECT column_names
FROM table_name
WHERE column_name IS NOT NULL;
```

Basic select statement

The **LIKE** operator is used in a **WHERE** clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the **LIKE** operator:

| The percent sign (%) represents zero, one, or multiple characters | The underscore sign (_) represents one, single character. |

Basic select statement

The following SQL statement selects all customers with a CustomerName starting with "c

```
SELECT * FROM Customers
    WHERE CustomerName LIKE 'a%';
```

The following SQL statement selects all customers with a CustomerName ending with "c"

```
SELECT * FROM Customers
    WHERE CustomerName LIKE '%a';
```

Populate one table using another table

You can populate the data into a table through the select statement over another table; provided the other table has a set of fields, which are required

```
INSERT INTO first_table_name [(column1, column2, ... columnN)]
    SELECT column1, column2, ...columnN
    FROM second_table_name
    [WHERE condition];
```

Unit 5 To Do!

Reading & Videos

Discussion Forum

Seminar (or Alternative)

Assignment

UNIT 5 ASSIGNMENT

UNIT 5 ASSIGNMENT

▶ Purpose:

▶ There are varieties of ways to get data inserted into a table.

▶ One way entails manual data entry through use of the Designer tools contained in Microsoft SQL Server Management Studio (SSMS).

▶ Another way involves importing the data from an Excel file.

UNIT 5 ASSIGNMENT

Purpose:

You will import the Movies flat file dataset into a temporary table in the Movies database.

The data from the temporary table will then be migrated to the normalized tables using a provided data migration script.

Use the import instructions document to accomplish this task.

UNIT 5 ASSIGNMENT

Purpose:

After you have entered new data, you will query the table, based on specific data requested.

You can leverage the revised database design diagram as a resource for this assignment.

Download and use this diagram for your analysis.

UNIT 5 ASSIGNMENT

Assignment Instructions:

Part 1: Establish a New Movies Database Instance

Create a new version of the Movies database called "**Movies_DB**" using the SQL script and instructions provided in the assignment posting

UNIT 5 ASSIGNMENT

Assignment Instructions:

Part 2: Manually Populate the Normalized Tables

- Incorporate the first record from the Movies flat file dataset into the normalized tables in the Movies database.

UNIT 5 ASSIGNMENT

Assignment Instructions:

Use the Designer tools from the Microsoft SQL Server Management Studio (SSMS) to accomplish this task.

Detailed instructions for the manual data entry are provided in the assignment posting

UNIT 5 ASSIGNMENT

Assignment Instructions:

Part 3: Import the Flat File Dataset into the Database

- Establish a temporary table called **Movies_Import_Temp** in the **Movies_DB** database.
- The instruction document and flat file dataset, which is a comma-separated values (CSV) file, are provided in the assignment posting

UNIT 5 ASSIGNMENT

▶ Assignment Instructions:

▶ Part 4: Migrate Data to the Normalized Tables Using a Migration Script

▶ After the manual insertions are completed, migrate the remaining data contained in the Movies_Import_Temp table using the provided data migration script

UNIT 5 ASSIGNMENT

▶ Assignment Instructions:

▶ The data migration script along with Instructions for executing it in a Microsoft SQL Server Management Studio (SSMS) query window are provided in the assignment posting.

▶ Briefly describe the purpose and function of the individual INSERT statements in the provided data migration script.

UNIT 5 ASSIGNMENT

▶ Assignment Instructions:

▶ Part 5: Contrast Data Migration Techniques

▶ Write one or more paragraphs contrasting the techniques (i.e., manual versus scripted) used to populate the normalized tables.

▶ Provide pros and cons for each technique.

UNIT 5 ASSIGNMENT

Assignment Instructions:

Part 6: Use basic SELECT statement to retrieve data from tables in the Movies database.

- Create SQL statements to retrieve data from the Movies Database for the problems specified in the assignment posting.
- Create screenshots to show each SQL query and the results of each query execution.

UNIT 5 ASSIGNMENT

	DirectorID	Director_FirstName	Director_LastName
1	5	Francis Ford	Coppola
2	6	Sofia	Coppola

▶Assignment Instructions:

▶ Problem 1 - List all of the directors with the last name of "Coppola."

	Producer_LastName
1	Metro-Goldwyn-Mayer
2	Mirisch Company
3	Mirisch-Rich Productions

▶ Problem 2 - Show the last names of movie producers whose last names start with the letter M.

UNIT 5 ASSIGNMENT

▶ Assignment Instructions:

▶ Problem 3 - List all movie titles and ratings for movies with a rating greater than or equal to 8. Show the results in alphabetical order by movie title.

	Title	Rating
1	Arsenic and Old Lace	8.1
2	Casablanca	8.6
3	Kill Bill Vol 1	8.1
4	Mr. Smith Goes To Washington	8.3
5	Ratatouille	8.0
6	Reservior Dogs	8.3
7	Saving Private Ryan	8.6
8	Sunset Blvd.	8.5
9	The Incredibles	8.0
10	The Philadephia Story	8.1

UNIT 5 ASSIGNMENT

▶Assignment Instructions:

▶ Problem 4 - List all movie titles and ratings for movies with a rating between 5 and 6. Show the results in alphabetical order by movie title.

	Title	Rating
1	Bling Ring	5.6
2	Critics Choice	5.7
3	Massacre Harbor	6.0

UNIT 5 ASSIGNMENT

▶Assignment Instructions:

▶ Problem 5 - Show the producer records that do not have first name values. In other words, the first name value is NULL for these producer records. Present the results in alphabetical order.

	ProducerID	Producer_FirstName	Producer_LastName	Producer_Oscars
1	5	NULL	Columbia Pictures	NULL
2	10	NULL	Dreamworks SKG	NULL
3	15	NULL	Metro-Goldwyn-Mayer	NULL
4	16	NULL	Mirisch Company	NULL
5	17	NULL	Mirisch-Rich Productions	NULL
6	18	NULL	Paramount Pictures	NULL
7	20	NULL	RKO Pictures	NULL
8	21	NULL	Romulus Films	NULL
9	22	NULL	Twentieth Century Fox	NULL
10	24	NULL	Warner Bros.	NULL

UNIT 5 ASSIGNMENT

Microsoft SQL Server Express and SQL Server Management Studio (SSMS) MUST be installed to complete this Assignment.

Compose your Assignment in a Word document.

UNIT 5 ASSIGNMENT

Assignment Requirements:

- Embed the screenshots of your SQL statements and confirmatory output (e.g., table structure definitions) into the Word document.

- The assignment is due by the final day of the Unit 5 week.

UNIT 5 ASSIGNMENT

Name your assignment document according to this convention: **IT234_<*YourName*>_Unit5.docx** (replace **<*YourName*>** with your full name).

Submit your completed assignment to the Unit 5 Assignment Dropbox by the final day of the Unit 5 week.

Review the Unit 5 Assignment Rubric before beginning this activity.

Any Questions?