Week 7 Submission Tasks

References

http://www.w3schools.com/sql/

https://www.techonthenet.com/sql/

Submission Process

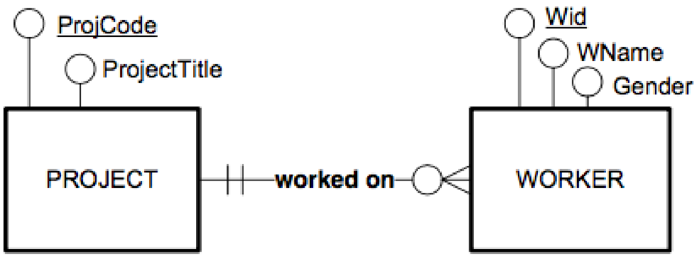
Download **DAD_task_submission_template.docx** from Canvas..

Paste the required screen captures from the tasks below into this file.

Submit the (.docx) file into the appropriate weekly task on **Canvas**.

Task 1.

Consider this ERD:



Create a Relational Schema based on this diagram. Indicate all Primary & Foreign Keys. E.g.:

(note this relational schema is an example, it does NOT match the above ERD intentionally)

BRANCH (BranchId, BranchName)
Primary Key (BranchId)

EMPLOYEE (EmpName, PhoneNo, BranchId)
Primary Key (EmpName)
Foreign Key (BranchId) references Branch

Paste the text of your relational schema in the **DAD_task_submission_template.docx**

Task 2.

Create and execute two Create Table statements based on your relational schema above.

Tables must include appropriate primary and foreign keys.

Do not introduce any surrogate keys

Paste the text of your SQL statements into the appropriate position.

Task 3.

Use **insert** statements to add the following data to your **Project Table**

1	Project One
2	Project Two
3	Project Three

Use **insert** statements to add the following 4 data rows into your **Worker Table**

21	Dave Jones	M	2
22	Emma Quilt	F	2
23	Fred Gingers	M	1
24	Pat Smith	F	2

Add employee # 25 and #26. Use **your** name and gender and those of a friend (or movie star etc)
Allocate both of yourselves to Project 3.

Paste the text of your SQL statements into the appropriate position.

Task 4.

Execute these statements in Azure

```
select table_name from information_schema.tables;
exec sp_columns project;
select * from project;
exec sp_columns worker;
select * from worker;
```

Screen Capture the results of each of these statements Paste in the appropriate position.

Task 5.

Testing Primary Keys and Foreign Keys

Use **insert** statements to add the following data to your **Project Table**

1 Project Four

This statement **must fail** as it will cause a duplicate primary key. (If it doesn't fail, you will need to **investigate / correct** the primary key clause in your create table statement).

Execute the statement and Screen Capture the results of the statement.

Use **insert** statements to add the following data to your **Worker Table**

25 Helen Nogood F 4

This statement **must fail** as it will cause a foreign key problem. (If it doesn't fail, you will need to investigate / correct the foreign key clause in your create table statement).

Execute the statement in and Screen Capture the results of the statement.

Write a single delete statement to delete Project 2 from the Project Table

This statement must fail as it will cause a parent / child constraint problem. (If it doesn't fail, you will need to **investigate / correct** the foreign key clause in your create table statement).

Execute the statement in and Screen Capture the results of the statement.

Paste all screen captures in the appropriate position.

Task 6.

Write a single SQL statement lists the worker name and project title for every row in the worker table.

Screen Capture the SQL text box plus the all rows of the result set
Paste the screen capture in the appropriate position.

Task 7.

Write a single SQL statement that counts the number of rows in the Worker table.

Screen Capture the SQL text box plus the all rows of the result set

Write a single SQL statement that counts the number of rows by projectcode in the Worker table.

Screen Capture the SQL text box plus the all rows of the result set

Paste the screen captures in the appropriate position.

Task 8.

Use the **Access Database** named movie_actor_casting_only.accdb

Ensure that that the tables are named movie9999, actor9999 and casting9999 (where 9999 is the last
4
digits of your student id).

Create a new Query Design and click on the **SQL icon at the bottom right of the screen**.
Write a single **SQL statement** that counts the number of rows by ratingcode in the Movie table. E.g.

G	12
PG	8
M	115

Screen Capture the SQL text box plus the all rows of the result set
Paste the screen captures in the appropriate position.

Task 9.

Create a new Query Design and click on the SQL icon at the bottom right of the screen.

Using the movie table, write an SQL statement that list various columns (your choice) from the movie table for a limited range of years (your choice) and for a limited range of rating codes (your choice).

The displayed result must be in movie title sequence.

Screen Capture the SQL window that contains your code.
Screen Capture the resulting Query Design grid generated by your SQL statement
Screen Capture the first 10 rows of the datasheet view.
Paste the screen captures in the appropriate position.

Task 10.

List the Movie number, movie title, colour name and the long rating description of every movie
The list must be in ascending movie title sequence (Hint: This will require 2 inner joins)
Screen Capture the SQL text box plus the first 10 rows of the result set
Paste the screen captures in the appropriate position

Setup

- Download the file named **MOVIE_RATING_COLOUR_ACTOR_CASTING.sql** from Canvas.
- Edit the script.
- Rename all tables to end with XXXX where XXXX is the last 4 digits of your student number.
- **Save the changes**.
- Run the script in AZURE Query Editor to create the database.

Task 11.

- List the actor fullname, movie title and release year for each casting row in the casting table. The list must be in ascending actor fullname sequence
(Hint: This will require 2 inner joins)
- Screen Capture the SQL text box plus the first 10 rows of the result set
Paste the screen captures in the appropriate position

Task 12.

List the rating code and the count of movies that belong to that movie code based on the movie table

The list be in descending count sequence.

(Hint: This will require a group by clause based one non aggregate expression) e.g.

M	145
MA	91
PG	72
G	96

(these values are not accurate)

Screen Capture the SQL text box plus the all rows of the result set

Paste the screen captures in the appropriate position.

Task 13.

Same as task 12, but only include results where the count is greater than 50.
(Hint: This will require the use of the Having clause)

Screen Capture the SQL text box plus the all rows of the result set
Paste the screen captures in the appropriate position.

Task 14.

Same as task 13, but only count movies that were made between 1998 & 2014 (inclusive)..
(Hint: This will require the use of the Where clause)

Screen Capture the SQL text box plus the all rows of the result set

Paste the screen captures in the appropriate position.

Task 15.

List the actor fullname and the number of movies that he/she has appeared in

The list must be in ascending actor fullname sequence. e.g.

Bill Murray	9
Bill Nighy	4
Christine Taylor	5

(these values are not accurate)

Screen Capture the SQL text box plus the first 10 rows of the result set

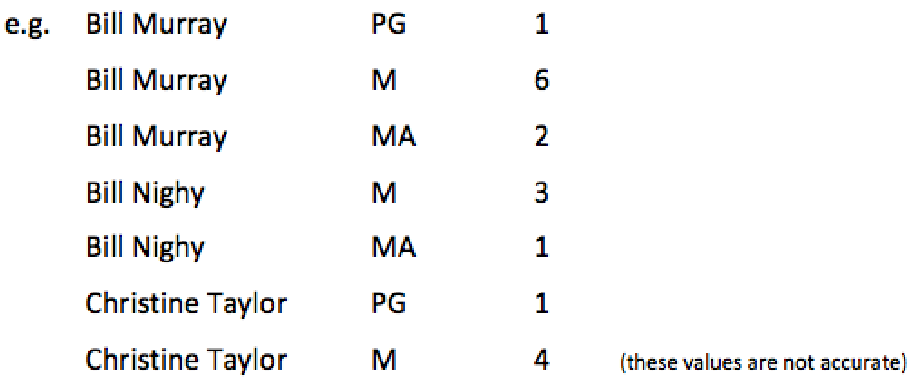
Paste the screen captures in the appropriate position.

Task 16.

List the actor fullname, the rating code and the number of movies that he/she has appeared in

The list must be in ascending actor fullname / rating code sequence.

(Hint: This will require a group by clause based two non-aggregate expressions)



e.g.	Bill Murray	PG	1	
	Bill Murray	M	6	
	Bill Murray	MA	2	
	Bill Nighy	M	3	
	Bill Nighy	MA	1	
	Christine Taylor	PG	1	
	Christine Taylor	M	4	(these values are not accurate)

Screen Capture the SQL text box plus the first 10 rows of the result set

Paste the screen captures in the appropriate position.