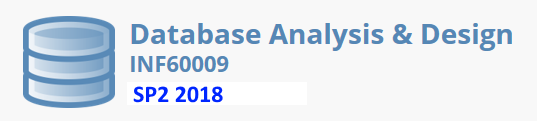
****

**Task 6 - Pass and Credit Requirements**

## **Overview**

* This week we’re going to focus on applying some of the more advanced SQL concepts and statements covered in the lectures. Again, we’re going to make use of iSQLJunior – so make sure your account is working.
* Also, you find that you get an error message **ORA-01536: space quota exceeded for tablespace 'USERS',** it means that you have **too many tables** in your account (probably from last week) and that you need to drop some of them. This can be done by:
  + Listing all of the tables in your account: SELECT TABLE\_NAME FROM TABS;
  + Then drop a table. DROP TABLE <table-name>
  + Note: You must drop child tables **before** dropping the parent table
* For submission, it’s same process as the other weeks: complete tasks, document them (usually by asking for screen grabs), and submit online.

## **Getting Started & Submitting**

* Download the files **W06P.DOCX** and **W06C.DOCX** from blackboard
* Paste the required screen captures from the tasks below into these files
* When complete, use the File / Export menu option to generate the files W06P.PDF and W06C.PDF
* Finally log into Doubtfire and submit both files into the appropriate weekly tasks.

## **Pass level Tasks**

* **ALL tasks** in this section MUST be completed for you to successfully complete the Pass Level Tasks
* **Download** the file named **movie\_rating\_colour.txt** from **Blackboard**.
* **Edit the script.**
* Perform a **find and replace**. Change all occurrences of
  + **movieXXXX** to **movie9999**
  + **ratingXXXX to rating9999**
  + **colourtypeXXXX to colourtype9999**
* where 9999 is the last 4 digits of your student ID.
* **Save the changes.**
* Go to <https://feenix-isqljr.swin.edu.au/> and login to your Oracle account.
* Write a **single** SQL statement that lists all columns of movies that have one of these movie no values: 70, 137, 1858, 286217. Use OR operators in your solution.
* The list must be in Ascending movie no sequence.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that lists all columns of movies that have one of these movie no values: 871, 10764, 65754, 211672. Do NOT use the OR operators in your solution. Use the IN operator.
* The list must be in Ascending movie no sequence.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that lists all columns of movies that have a title that begins with the word **'The'**.
* Make the query work for **any combination** of upper or lowercase characters.
* The list must be in Ascending Title sequence.
* **Screen Capture** the **SQL text box** plus the **first 8 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that lists all columns of movies that match all the following criteria:
  + Rating is "M"
  + Length of the film is less than or equal to 110 minutes
* The list must be in Ascending movie no sequence.
* **Screen Capture** the **SQL text box** plus the **first 10 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that lists all columns of movies that match either of the following criteria:
  + Rating 'PG' and the tmdb score is greater than 7.6
  + Rating is 'MA' and the tmdb score is greater than 7.8
* The list must be in Ascending movie no sequence.
* CHECK YOUR RESULTS and ensure that all movies in the result meet the criteria above.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Same as the query above, except ensure that movies listed have a runtime greater than 130 minutes.
* CHECK YOUR RESULTS and ensure that all movies in the result meet the criteria above.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that lists all columns of movies that match **all** of the following criteria:
  + Release Year is one of the following: 2013 or 2014
  + Run Time > 150
* The list must be in Ascending movie no sequence.
* CHECK YOUR RESULTS and ensure that all movies in the result meet the criteria above.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that does the following:
  + For each row in the MOVIE table, display the title, release year, runtime, rating code and the matching short description from the RATING table
  + The list must be in Ascending movie no sequence.
* **This will require you to Join two tables**
* **Screen Capture** the **SQL text box** plus the **first 10 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**
* Write a **single** SQL statement that does the following:
  + For each row in the MOVIE table, display the title, release year, runtime, rating code and the **matching long description** from the RATING table. Only do this if the release year is **2016** **or** **greater and runs for 100 minutes or more**.
  + The list must be in Ascending movie no sequence.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06P.DOCX**

## **Credit level Tasks**

* **ALL** tasks in this section MUST be completed for you to successfully complete the Credit Level Tasks
* Write a **single** SQL statement that does the following:
  + For each row in the MOVIE table, display the title, release year, rating code, the matching **short description** from the RATING table and the **colour name** from the COLOURTYPE table.
  + Only display rows that have runtime of less than 85 minutes
* The list must be in Ascending movie no sequence.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06c.DOCX**
* Write a **single** SQL statement which does the following:
* Display the Movie No, Title, Run time, Rating Short Description, tmdb score for movies that meet **any** of these criteria:
  + Rating code of MA plus a runtime between 100-102 (inclusive)
  + Rating code of G plus a runtime 90 minutes or less
  + Rating code of PG plus a runtime between 120-125 (inclusive)
  + Rating code of M plus a runtime 175 minutes or more
* The SQL statement must also only include movies that between 6.5 to 7.5 (inclusive)
* The list must be in Ascending movie no sequence.
* **Paste the SQL from this script plus the results of the first 10 rows into the appropriate position in the document named W06C.DOCX**
* Write a **single** SQL statement which does the following:
* Display the Movie No, Title, tmdb score, Rating Short Description for movies that meet **any** of these criteria:
  + The Rating Long Description contains the word 'recommended' (any case)
  + The tmdb id contains the text '44'
  + The title of the movie contains the letter 't' (any case)
* The list must be in Ascending movie no sequence.
* **Paste the SQL from this script plus the first 6 rows of the result set into the appropriate position in the document named W06C.DOCX**
* Write a **single** SQL statement that displays all the release years in the movie table.
* The list must not repeat any value.
* The list must be in acscending release year sequence.
* **Screen Capture** the **SQL text box** plus the **first 20 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06C.DOCX**
* Write a **single** SQL statement that displays all columns from movies in the movie table. Only display movies where the title does **not** contain any of these letters: **A E I U** (any case)
* The list must be in Ascending title sequence.
* **Screen Capture** the **SQL text box** plus the **all rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06C.DOCX**
* **Update** the tmdb\_votes value to 0 for all rows.
* **Screen capture** statement plus sample result (e.g. 6 rows) and paste in appropriate position document named W06C
* **Now, update** the tmdb\_votes to 1 where rows meet **either** of the following criteria:
* The run time is 185 or greater and the rating code is M
* The run time is 160 or greater and the rating code is MA
* **Screen Capture** the SQL text box of the Select statement plus the first 6 rows of the result set
* Write a single SQL statement that displays the movie no, title, release year, run time and rating code for all movies that have tmdb-votes equal to the value 1
* The list must be in Ascending title sequence
* **Screen Capture** the SQL text box of the **Select** statement plus the **first 8 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W06C.DOCX**

## **References**

* Chapter 4 <http://proquest.safaribooksonline.com/book/databases/sql/9780321584069> via Swinburne library
* <http://www.w3schools.com/sql/>
* <https://www.techonthenet.com/sql/>
* Lecture 6 of this unit