INF10025 Data Management and Analytics

Task 3 – Pass and Credit

Overview

- **See Canvas → Assignments for due date**
-
- In this task, you'll really get to the core of conventional database analysis and design: ERDs and SQL. To help you learn, we'll be making use of the iSQLJunior tool (a web interface for interacting with an Oracle DB). Details on how to use and how to connect are provided in the lectures and on Canvas.
- For submission, it's the same process as with the other tasks: complete tasks, document them (usually by creating screen shots) save as pdf, and submit online.
- To get started, download the files **T03P.DOCX** and **T03C.DOCX** from Canvas
- Paste the required screen captures from the tasks into these files
- When complete, save the files as T03P.PDF and T03C.PDF
- Finally submit both files on Canvas via Assignment .
- **When writing sql scripts you must follow CODES AND STANDARDS as shown in slide 29 in Lecture5 – Part 3**

Pass Level Tasks

Completion Criteria: For the Pass Task to be marked "Complete" seven (7) sub-tasks must be marked "Correct".

Pass 3a

- **Download** the file named **moviedb_sql- 1.txt** from **Canvas**.
- Open this file in a text editor, e.g. **in Notepad, Notepad++ or some other Text editor**
- Change all occurrences of **movieXXXX** to **movie9999** (where 9999 is the last 4 digits of your student ID). Similarly replace XXXX in all other tables with the last 4 digits of your student ID. To do that perform a **find and replace**. Enter XXXX in the Find box and last four digits of your student ID in the Replace box, click on Replace All.
- **Save the changes**.
- Go to https://feenix-isqljr.swin.edu.au.

  The username id **'s'** + your **student id**. (e.g. s123456789)
  The password is your **birthdate** (6 digits ddmmyy format). The database is **DAD**.
- **Copy and paste all text from your *modified* moviedb_sql-1.txt** (which now has your **student ID instead of XXXX**) into iSQLjr
- **Execute** the script**.** This will create the movie and other tables and populate them with data.

- Write the following SQL statement and execute it. Note, 9999 should be replaced with the last 4 digits of your student ID.

```
SELECT Title
FROM movie9999
```

SQL is not case sensitive but for readability it is a standard approach that SQL keywords are written in all upper-case letters.

Notice how only movie titles are listed. There is no need to screen capture these details.

- Write the following SQL statement and execute it.

  SELECT '123459999' as STUID, title

  FROM Movie9999

  (Note in the above statement 123459999 should be replaced with your Student ID)

  Notice how each row contains your student id and movie title. There **is no need to screen capture these details.**

  You are expected to *include your student ID* in all queries unless otherwise requested.

- Write an SQL statement to list only the **StudID**, **Movieno,TITLE,** and **RELYEAR** columns of every row in the movie table in **Descending** Movieno **sequence**
- **Save it in your Notepad++ file**
- **Copy and paste the script into iSQlj and execute it**
- **Screen Capture** your SQL text box (or your saved script from Notepad++)
- **Screen Capture** the **first 15 rows** of the result set
- **Paste both screen captures in the appropriate position in the document named T03P.DOCX**

Pass 3b

- Write and execute the SQL statement to list only the StuId, title, relyear, colourcode and ratingcode of every row in the movie table that has a M rating. The list must be in **Ascending** Relyear, title **sequence** (note two fields to be sorted).
- **Screen Capture** the SQL text box plus the **first 10 rows** of the result set (if there are more than 10 rows).
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**

Pass 3c

- Write and execute the SQL statement to list only the Stuid, movie no, title, relyear of every row in the movie table that has either title **Oceans Eleven** or **Hamlet.**
  (Note: Remember that oracle is **Case sensitive**)
- The list must be in **Ascending** title/ Relyear sequence
- **Screen Capture** the SQL text box plus **all rows** of the result set
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**

Pass 3d

- Write and execute the SQL statement to list all columns of every row in the movie table that has a TMDB_SCORE of **8.4** or greater and a release year of **2009** or less.
- The list must be in Ascending **Movieno** sequence
- Remember to **include your student ID**
- **Screen Capture** the SQL text box plus **all rows** of the result set
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**

  Note: You should be able to complete the above tasks by the end of Week 05

Reminder: You are expected to include your studentid in all queries unless otherwise requested.

Pass 3e

- Write a **single** SQL statement that lists your StuId plus all columns of movies that have one of these movieno values: 78, 164, 1726, 2501,57201
  DO NOT use the OR operator in your solution. Instead, use the **IN** operator.

- The list must be in **Ascending Movieno** sequence.
- **Screen Capture** the **SQL text box** plus **all rows** of the result set
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**

Pass 3f

- Write a **single** SQL statement that lists all columns of the Actor table where the Actor's password contains this string of 4 letters **ABCD** regardless of case.

  This means that **all of** these password would match AbCdbb ABCDbb abcdbb ddaBcd dABCda
- The list must be in Descending ActorNo sequence.
- **Screen Capture** the **SQL text box** plus **all the rows** of the result set (Hint: at least 5 rows should match. If you have less than 5 rows – then you are doing something wrong)
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**

Pass 3g

- Write a **single** SQL statement that lists all columns of movies plus StuId that match EITHER of the following criteria:
- Rating 'PG' and the tmdb_score is greater than to 7.6

  or

  Rating is 'M' and the tmdb_score is less than or equal to 4.1

- In addition, the query **must** only list movies that have a **runtime** Less than 134 minutes
- The list must be in **Ascending movie no** sequence.

  Do **not** write multiple SQL statements. **This must be a single SQL statement**
- CHECK YOUR RESULTS and ensure that all movies in the result set meet the criteria above (Hint: If you have more than 8 or 9 rows – then you are doing something wrong)
- **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 T03P.DOCX**

Pass 3h

Write a single SQL statement that does the following:

- For each row in the MOVIE table display the stuid, Movieno, title, release year, tmdb_score, rating code and the matching Long description from the RATING table.
- The list must be in **Ascending MovieNo** sequence.
- **This will require you to Join two tables.**
- **Screen Capture** the **SQL text box** plus **the first 5 rows** of the result set.
- **Paste the screen captures in the appropriate position in the document named T03P.DOCX**

Pass 3i

- Write a **single** SQL statement that does the following:
- For each row in the MOVIE table, display the title, MovieNo, release year, runtime, rating code and the **matching short description** from the RATING table.
- Only do this if the release year is **2012 or greater and runs for 94 minutes or less**.
- 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 T03P.DOCX**

Credit Level Tasks

Completion Criteria: **For the Credit Task to be marked "Complete" Eight (8) sub-tasks must be marked "Correct".**

Note: for Credit 3a to 3e you may find it beneficial to review the content from both Week 5 and Week 6 to be able to complete the work.

Credit 3a

- For each row in the MOVIE table, display the stuid, Movieno,title, release year, rating code,runtime, the matching **short description** from the RATING table and the **colour name** from the COLOURTYPE table.
- Only display rows that have runtime of greater than 170 minutes
- Only display rows that have a rating code of M or MA.
- 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 T03C.DOCX**

Credit 3b

- Write a SQL statement which does the following:
- Display the StuId, MovieNo, Title, Runtime, Rating code, Rating Short Description, tmdb score for movies that meet **any** of these criteria:
- Rating code of MA plus a runtime between 100-105 (inclusive)
- Rating code of G plus a runtime less than 90
- Rating code of PG plus a runtime between 120-125 (inclusive)
- Rating code of M plus a runtime 170 minutes or more
- The query must **only i**nclude movies that have a tmdb_score between 6.8 and 7.4 inclusive
- The list must be in **Ascending Movie no** sequence.
- CHECK YOUR RESULTS and ensure that all movies in the result meet the criteria above. (Hint Between 5 and 10 rows should match)
- **Paste the SQL from this script plus the results into the appropriate position in the document named T03C.DOCX**

Credit 3c

- 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 descending release year 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 T03C.DOCX**

Credit 3d

- You are now going to do some Updates
- Write a single statement to **Update t**he tmdb_votes value to 0 for all rows.
- Paste the statement into ISQlj and execute.
- **Screen capture** the ***Update*** statement paste in appropriate position document named **T03C.docx**
- Write a **single** SQL statement that displays all the rows in the movie table in Descending Movieno
- **Screen capture** the first 6 rows (make sure that capture shows tmdb_votes = 0)
- Now write another single statement to **Update** the tmdb_votes to **1** where rows meet **either** of the following criteria:

  The run time is 140 or less and the rating code is G

  The run time is 170 or greater and the rating code is M

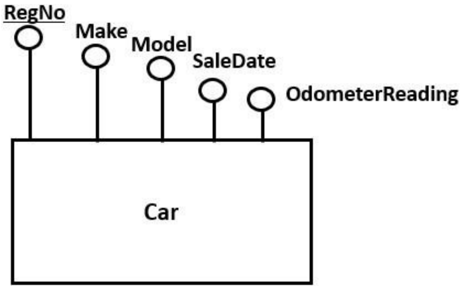
- **Paste this update into iSql junior and execute**
- **Screen Capture** the SQL text box showing the update
- Write a single SQL statement that displays the stuid, movieno, title, release year, runtime, tmdb_votes and rating code for all movies that have tmdb_votes equal to the value 1
- The list must be in **Ascending title** sequence
- **Screen Capture** the SQL text box of the **Select** statement plus **7 rows** of the result set
- **Paste the screen captures in the appropriate position in the document named T03C.DOCX**
- Note if you make a mistake with the script involving Tmdb_votes = 1 then go back and Update tmdb_votes = 0 and start again

Credit 3e

- Write a SQL statement that **displays** stuid plus **all columns** from movies in the movie table.
- Only display movies where the title does **not** contain any of these letters: **A B I O** (any case)
- The list must be in **Descending 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 T03C.DOCX**

Credit 3f

- You're now going to create your own Relational Schema and Table and your own sample data.
- Create the Relational schema for a relation named CAR9999 (where 9999 is the last 4 digits of your student ID) based on the following ERD.
- **Paste the relational schema code in the appropriate position in the document named T03C.DOCX**



Create a SQL script named create_car_sql.txt which contains SQL including the following:

- Has a **Drop table** statement that drops a table named CAR9999 (replace 9999 with the last **4 digits** of **Stuid**)
- Has a **Create Table** statement that creates the table named CAR9999 based on the relational schema that you have created. Make sure **Primary Key** is clearly shown
  The RegNo, Make, and Model columns are varchar datatypes (specify any appropriate length)
- The OdometerReading column is numeric (you may specify the length)
- The Sale Date is a Date datatype.
- **Paste the SQL from this script into** iSQLj and execute
- **Screen capture the SQL from this script into the appropriate position in the document named T03C.DOCX**

Credit 3g

- Create a SQL script named **Insert_car_sql.txt** which contains SQL that includes insert statements that add the following data to the table:

RegNo	Make	Model	SaleDate	OdometerReading
1AMT7U	Aston Martin	Vantage	March 12 2022	5300

You need to make up another 3 records of Data of your own. Your name (or part of your name) is to be included in one of the pieces of Data

- **Paste the Insert statements into iSQLj and execute**

- **Paste the Insert SQL statements from this script into the appropriate position in the document named T03C.DOCX**

Help on inserting dates and formats can be found at:

https://www.techonthenet.com/oracle/questions/insert_date.php

https://www.techonthenet.com/oracle/functions/to_date.php

http://www.dba-oracle.com/f_to_date.htm

- You should now have created and populated your Car Table in iSQLj

Credit 3h

- Write and execute the SQL statement to list all rows of the Car table in Ascending Primary Key sequence.
- **Screen Capture** the SQL text box plus **all rows** of the result set.
- **Paste the screen captures in the appropriate position in the document named T03C.DOCX**

Credit 3i

- Write and execute the SQL statement to list all rows of the Car table for cars that have **travelled less than** some distance (**you decide what value to use**).
- **Screen Capture** the SQL text box plus **all rows** of the result set.
- **Paste the screen captures in the appropriate position in the document named T03C.DOCX**

Credit 3j

- Write and execute the SQL statement to list all rows of the Car table that have a purchase date after Jan 01 2019. (**or use your own date**)
  For help: https://www.techonthenet.com/oracle/functions/to_date.php
- Screen Capture the SQL text box plus all rows of the result set.
- **Paste the screen captures in the appropriate position in the document named T03C.DOCX**

References

- https://feenix-isqljr.swin.edu.au/
- When using iSQLJr, if 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 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

SQL Help and Tips

- http://proquest.safaribooksonline.com/book/databases/sql/9780321584069 via Swinburne library

- http://www.w3schools.com/sql/
- https://www.techonthenet.com/sql/
- Lecture 5 and 6 of this unit