INF10025 Data Management and Analytics

Task 4 – Pass and Credit

Overview

- In this task, you are going to continue with more advanced ERD and SQL work.
- For submission, it's same process as the other the other tasks: complete tasks, document them (usually by asking for screen grabs) and submit online.
- To get started, download the files **T04P.DOCX** and **T04C.DOCX** from Canvas.
- When complete, generate the files **T04P.PDF** and **T04C.PDF**
- Finally log into Canvasand submit both files into the appropriate weekly tasks.

Pass Level Tasks

Completion Criteria: For the Pass Task to be marked "Complete" nine (9) sub-tasks must be marked "Correct".

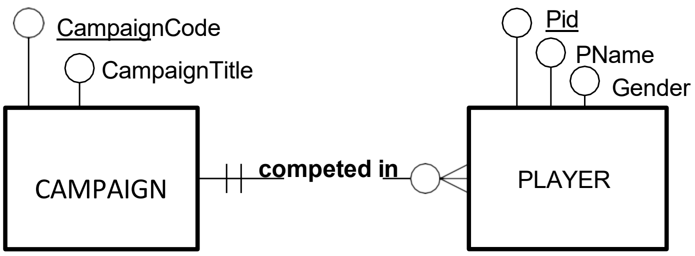
Pass 4a

- Consider the **Relational Schema** shown below. It has been created from an ERD involving a relationship between two Entities EMPLOYEE and BRANCH

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

BRANCH (BranchId, BranchName)
Primary Key (BranchId)

- Consider the **ERD diagram shown below**
- Create a Relational Schema based on this diagram.
- Indicate all **Primary and Foreign Keys** (see the above example to help you)
- **Paste the text of your relational schema into the document named T04P.DOCX**



Pass 4b

- Create and execute **two 'Create Table' statements** based on your relational schema above. Make sure each table has last 4 digits of your Stuid in its name (e.g. Campaign9999)
- Tables must include appropriate primary and foreign keys. Do **not** introduce **surrogate** keys.
- **Paste the text of your SQL statements into the document named T04P.DOCX**

Pass 4c

- Use **insert** statements to add the following data to your **Campaign Table**

1	Campaign One
2	Campaign Two
3	Campaign Three

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

21	Ali Aien	M	2
22	Peter Ymer	M	1
23	Sarah Rogers	F	1
24	Sheelan Aien	F	2

- Add volunteer #25 and #26. Use **your** name and gender and those of a friend, movie star etc
- Allocate both of yourselves to Campaign 3
- **Paste the text of your SQL statements into the document named T04P.DOCX**

Pass 4d

- Execute these statements in iSQL Jr (be careful of strange characters when you cut/paste!)
- Describe Campaign; select *from Campaign;
- **Screen Capture** the results of these statements
- Execute these statements in iSQL Jr Describe Player; Select *from Player;
- **Screen Capture** the results of these statements
- **Paste both captures in the appropriate position in the document named T04P.DOCX**

Pass 4e

In this task you are going to test that your tables have been created correctly by testing Primary Keys and Foreign Keys

- Write an **Insert** statement to add the following data to your **CampaignTable and execute**

  1 , Campaign four

  This statement must fail (If it doesn't fail, you will need to investigate / correct the primary key clause in your ***create table*** statement).
- Screen capture your **insert statement and the result - paste into the document named T04P.DOCX**
- Write a short explanation why it failed **in the document T04P.DOCX**
- Use an **insert** statement to add the following data to your **Player Table**

  27, Marcus Pontembelli, M, 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 iSQL Jr and **Screen Capture the SQL statement and the results of the statement**.
- Write a short explanation why it failed **in the document T04P.DOCX**

- Write a single **Delete** statement to Delete **Campaign Two** from the **Campaign 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 iSQL Jr and **Screen Capture the statement and the result of the statement.**
- Write a short explanation why it failed **in the document T04P.DOCX**
- **Paste all screen captures in the appropriate position in the document T04P.DOCX**

Pass 4f

- Write a single SQL statement lists the **Player name** and **Campaign title** for every row in the Player table.
- **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 T04P.DOCX**

Pass 4g

- Write a single SQL statement that **counts the number of rows** in the Player 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 Campaigncode** in the Player table.
- **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 T04P.DOCX**

Pass 4h

- You are now going to write SQL scripts in Access using **moviedatabaseDAD.accdb**
- Ensure that the tables are named **movie9999**, **actor9999**, **casting9999, etc** (where 9999 is the last 4 digits of your student id). All table names must end with 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 **BirthCountry** in the actor table e.g. USA might have 50; Germany might have 10 etc.
- **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 T04P.DOCX**

Pass 4i

- 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 runtimes (your choice) and for a limited list 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 in the document T04P.DOCX**

You should be able to get to this point by the end of Week 07

Pass 4j

- Read the following narrative:

> ***'SwinEats'*** *has food delivery jobs. Each job requires one driver. Each job has a job ID, a title, a duration (in hours), and a description. Drivers all have a name, an address and a mobile phone number and some do more than one job. When a driver does a job, the date is recorded, as well as a fixed fee. E.g. Jim did Job #47 on April 15th 2018. The fee that Jim has set for Job #47 is $10.*

- Create an ERD diagram based on the above narrative. **Make sure that every entity name ends with 4 digits of your student ID**.
- The ERD may be neatly hand-drawn or can be drawn in a package like **draw.io**.
- The **symbols** and style of the **ERD must match** the ERDs seen in lectures of **this unit**. If there are weak entities, make sure you use proper notation to indicate those.
- Scan or take a screenshot of the **ERD** and place it in the document named **T04P.DOCX**
- Convert the ERD to a **Relational schema** (DO NOT use surrogate keys)
- **Make up 2 pieces of Test data** for each entity.
- **Copy and paste the relational schema and place it in the document T04P.DOCX**
- **Copy and paste the Test data into the document T04P.DOCX**

Pass 4k

- Read the following narrative:

Hawthorn Comedy Inc. *is planning to hold a festival made up of various comedy shows.*
Every show will have an ID, a title and a description. Each show is delivered by a comedian. Comedians have a name, an address, contact number, and email. One comedian may deliver more than one show at the festival. Members of the public will come and visit the festival as attendees. As each attendee enters the festival, their name, address and phone number will be recorded. Each attendee will be given a unique barcode id. During their visit, attendees will rate the show with a score of 1 to 10. This will be recorded by staff at Hawthorn Comedy Inc. Attendees will be able to vote for more than one show.

- Create an ERD diagram based on the above narrative. **Make sure that every entity name ends with 4 digits of your student ID**.
- The ERD may be neatly hand-drawn or can be drawn in a package like **draw.io**.
- The **symbols** and style of the **ERD must match** the ERDs seen in lectures of **this unit**. If there are weak entities, make sure you use proper notation to indicate those.
- Scan or take a screenshot of the **ERD** and place it in the document named **T04P.DOCX**
- Convert the ERD to a **Relational schema** (DO NOT use surrogate keys)
- **Make up 2 pieces of Test data** for each entity.
- **Copy and paste the relational schema and place it in the document T04P.DOCX**
- **Copy and paste the Test data into the document T04P.DOCX**

Credit Level Tasks

Completion Criteria: For the Credit Task to be marked "Complete" Six (6) sub-tasks must be marked "Correct".

For these tasks you're going to be using iSQLJr (Oracle) and the tables from **moviedatabaseDAD.accdb**

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

Credit 4a

- List the stuid, MovieNo, movie title, colour name and the short rating description of every movie (Hint: This will require 2 inner joins)
- The list must be in Descending movie title sequence
- **Screen Capture** the **SQL text** plus the **first 10 rows** of the result set
- **Paste the screen captures in the appropriate position in the document T04C.DOCX**

Credit 4b

- List the stuid, actor fullname, movie title and release year for each casting row in the Casting table. (Hint: This will require 2 inner joins)
- The list must be in descending actor fullname 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 T04C.DOCX**

Credit 4c

- List the stuid, rating code and count of movies that belong to that movie code based on the movie table. Hint: This will require a group by clause based one non aggregate expression). Example (values will differ):

  M 145
  MA 91
  PG 72
  G 26

  - The list be should be in Descending Count 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 T04C.DOCX**

Credit 4d

- Same as above, but this time only include results where the count is greater than 21 (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 in the document named T04C.DOCX**

Credit 4e

- Same as above, but only count movies that were made between 1996 & 2010 (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 in the document named T04C.DOCX**

Credit 4f

- List the stuid, actor fullname, gender, and the number of movies that he/she has appeared in
- Change the **heading** from Count to **Total Movies**
- The list must be in Descending actor fullname sequence. Examples (values will differ)

Tom Hanks	M 9
Jamie Fox	M 4
Lucy Liu	F 5

- **Screen Capture** the **SQL text box** plus the **first 15 rows** of the result set
- **Paste the screen captures in the appropriate position in the document named T04C.DOCX**

Credit 4g

- List the stuid, actor fullname, the rating code and the number of movies appeared in for that code
- The list must be in Ascending actor fullname / rating code sequence. (Hint: This will require a group by clause based on two non-aggregate expressions)

Tom Hanks	PG	2
Tom Hanks	M	6
Tom Hanks	MA	1
Jamie Fox	M	4
Lucy Liu	PG	2
Lucy Liu	M	3

- **Screen Capture** the **SQL text box** plus the **first 15 rows** of the result set
- **Paste the screen captures in the appropriate position in the document named T04C.DOCX**

Credit 4h

- **Download** the SQL script named **task04_customers.txt**
- Edit the script and use **Search and Replace** to rename all occurrences of **customerXXXX to customer9999** (where 9999 is the last 4 digits of your student id).
- List the sum of all Sales this year by State / Gender
- The list must be in ascending state / gender sequence.
- Only include those customers who have a rating code of 1, 2, or 3.

State	Gender	Total Sales Last Year
New South Wales	F	598
New South Wales	M	3386
South Australia	F	2074
Tasmania	F	846
Tasmania	M	229
Victoria	F	6695
Victoria	M	4372
Western Australia	F	11247
Western Australia	M	3161

- **Screen Capture** the **SQL text box** plus the **all rows** of the result set
- Examine your results **and write a short comment** on what the Total Sales show with respect **to Gender and State**
- **Paste the screen captures and your comments in the appropriate position in the document named T04C.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