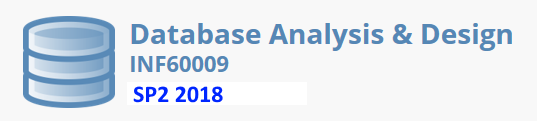
****

**Task 7 - Pass and Credit Requirements**

**Overview**

* This week we’re going to start linking our ERD work with our work in iSQLJr - a little more concretely. We’ll also make use of some the aggregate expressions and related clauses covered in the lecture.
* 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 **W07P.DOCX** and **W07C.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 W07P.PDF and W07C.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
* Consider the ERD below.
* Create a Relational Schema based on this diagram.
* Indicate all Primary & Foreign Keys (see example below – yours )
  + **EMPLOYEE (EmpName, PhoneNo, BranchId)**

**BRANCH (BranchId, BranchName)**

**Primary Key (BranchId)**

* + **Primary Key (EmpName)**
  + **Foreign Key (BranchId) references Branch**
* **Paste the text of you relational schema into the document named W07P.DOCX**

EVENT

EventTitle

EventCode

**worked on**

VOLUNTEER

VName

Gender

Vid

* Create and execute **two ‘Create Table’ statements** based on your relational schema above.[[1]](#footnote-1)
* Tables must include appropriate primary and foreign keys. Do not introduce any **surrogate** keys
* **Paste the text of your SQL statements into the document named W07P.DOCX**
* Use **insert** statements to add the following data to your **Event Table**

1 Event One

2 Event Two

3 Event Three

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

21 Dean Jones M 2

22 Eliza Quads F 1

23 Freek Gys M 2

24 Ali 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 Event 3
* **Paste the text of your SQL statements into the document named W07P.DOCX**
* Execute these statements in iSQL Jr

describe event;

select \*from event;

* **Screen Capture** the results of these statements
* Execute these statements in iSQL Jr

describe volunteer;

select \*from volunteer;

* **Screen Capture** the results of these statements
* **Paste both captures in the appropriate position in the document named W07P.DOCX**
* Now you’re going to test Primary Keys and Foreign Keys
* Use **insert** statements to add the following data to your **Event Table**

1 Event 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 in iSQL Jr and **Screen Capture** the results of the statement.
* Use **insert** statements to add the following data to your **Volunteer Table**

27 Mustin Dartin F 4 (Update: original version indicated ‘25’)

* 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 results of the statement.
* Write a single **delete** statement to delete **Event 2** from the **Event 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 results of the statement.
* **Paste all screen captures in the appropriate position in the document named W07P.DOCX**
* Write a single SQL statement lists the **volunteer name** and **event title** for every row in the volunteer 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 W07P.DOCX**
* Write a single SQL statement that **counts the number of rows** in the volunteer 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 eventcode** in the volunteer 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 W07P.DOCX**
* Use the Access Database from week 4 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).
* 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 W07P.DOCX
* 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 in the document named W07P.DOCX

## **Credit level Tasks**

**ALL** tasks in this section MUST be completed for you to successfully complete the Credit Level Tasks. For these tasks you’re going to be using iSQLJr (Oracle) and the same tables you were using in Pass tasks 8 and 9.

* List the Movie number, 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 box** plus the **first 12 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W07C.DOCX**
* List the 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 12 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W07C.DOCX**
* List the 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 W07C.DOCX**
* Same as above, but this time only include results where the count is greater than 20 (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 W07C.DOCX**
* Same as above, but only count movies that were made between 1995 & 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 W07C.DOCX**
* List the actor fullname, gender, and the number of movies that he/she has appeared in
* 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 10 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W07C.DOCX**
* List the 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 10 rows** of the result set
* **Paste the screen captures in the appropriate position in the document named W07C.DOCX**
* **Download** the SQL script named **week7\_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 last 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
* **Paste the screen captures in the appropriate position in the document named W07C.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 7 of this unit

1. Remember: 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: 1) Listing all of the tables in your account: SELECT TABLE\_NAME FROM TABS; 2) Then drop a table. DROP TABLE <table-name>. Note: You must drop child tables before dropping the parent table [↑](#footnote-ref-1)