**Database Application (COMP09050)**

**Coursework**

This document is a template for the coursework report part of your Database Application coursework.

On page 2 of this document you should add the requested details.

Read the Database Applications Coursework Specification document on Moodle and use this document to capture the requested work.

Before submitting this document - delete this page so that Page 2 becomes the front page of your coursework report.

Also delete the Task text and boxes and any examples supplied by me from your final coursework document.

**Important Note**

This template describes a coursework for 2-3 students working together.

For students working solo or as part of a larger group – please also use this template HOWEVER please refer to your DA Coursework Specification for the size and scope of your work.

**Database Applications (COMP09050)**

**Coursework**

Enter details of group members

|  |  |
| --- | --- |
| **Student Name(s)** | **BannerID(s)** |
| James Agbotta | B00310481 |

Enter details of case study

|  |  |
| --- | --- |
| **Company/Organisation Name** | **URL Address** |
| BBC Good Food | https://www.bbcgoodfood.com/ |

Enter details for group database

|  |  |
| --- | --- |
| **Database Name** | **Location?**  **(UWS Server/Student Laptop)** |
|  | Student Computer |

Enter details of your lecture and campus

|  |  |
| --- | --- |
| **Lecture Name** | **Your Campus?**  **(Dumfries, Lanarkshire or Paisley)** |
| Carolyn Begg (Database Applications) | PAISLEY |

**Summary of your Coursework Marks & Feedback**

**(Please include this page (just as you see it below) but with student names/initials against each section).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Who**  **Responsible for the work?** | **Section**  **Marks** | **Your Marks** | **Your Grade**  (E, D, C, B2, B1, A1-3) |
| **Database Planning** | *Enter Name(s)* | 10 |  |  |
| *Feedback (Summary)* | JA |  |  |  |
| **Systems Definition** | *Enter Name(s)* | 5 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Requirement Collection** | *Enter Name(s)* | 10 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Conceptual Design (EER Model)** | *Enter Name(s)* | 15 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Logical Design (Relations)** | *Enter Name(s)* | 15 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Physical Design (D. Dictionary)** | *Enter Name(s)* | 15 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Implement Database** | *Enter Name(s)* | 15 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Implement Useful Objects** | *Enter Name(s)* | 15 |  |  |
| *Feedback (Summary)* | *JA* |  |  |  |
| **Final Total & Grade** |  | **100** |  |  |

1. **Database Planning (10%)**

**Comixology**

**Tasks**

*Include an introduction to your company.*

*Present screen dumps of the home page and four other important web pages that show the website and identify the business of your chosen company (or organisation). Choose webpages that are best for illustrating the aspect of the business (e.g. business process(es) that you plan to focus on. Where appropriate (i.e. helpful) – annotated the screen dumps. Present reasons for your selection of the company (or organisation) you are using for this coursework.*

*Please do not choose any real film or TV streaming service for your coursework as this will be too similar to the Stream2U case study used in the labs.*

Comixology is a digital marketplace for the sale of digital versions of physical comic books.

**2. Systems Definition Stage (5%)**

**Tasks**

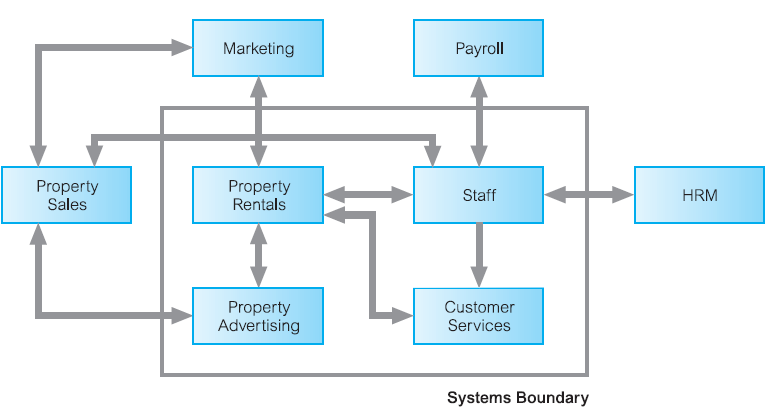
*Using English statements to describe the boundaries of your case study to identify what business process(es) of the company are included in your coursework and which are not. Support your description with a simple diagram.*

*Identify two user views that interact with the business processes identified above.*

**Simple English language descriptions of the business process(es) and the user views are sufficient for this stage. Illustrate the relationship between the business processes forming part of the study.**

**The purpose of this section is to define the scope of your investigation – in other words – what aspects of the business that use data will be used to help you construct an EER model.**

**An example diagram to show the boundary of a study.**

****

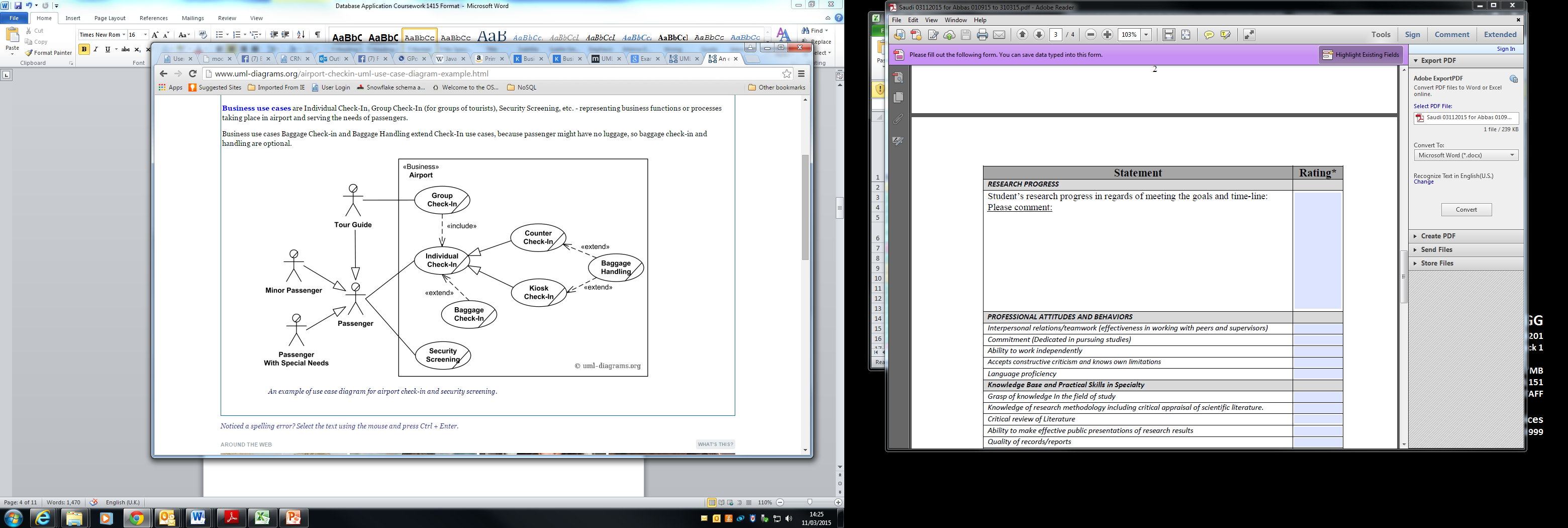
**3. Requirements Collection and Analysis (10%)**

**Tasks**

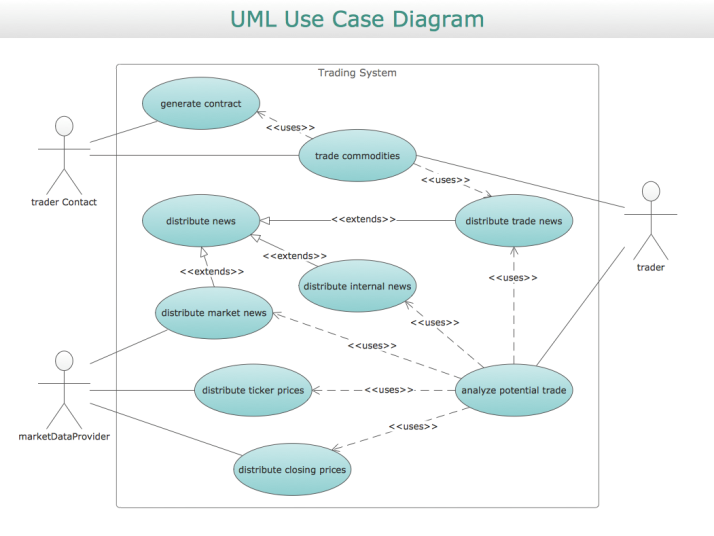
*Present a diagrammatic representation of your two user views interacting with your business process(es). Case diagrams and (high-level) descriptions are appropriate for this section. There is no need to provide detailed step-by-step descriptions. The Use Case descriptions are included here to help explain in general what the diagram is showing.*

**Here are examples of Use Case diagrams - Replace with your diagram(s).**

**Support your Use Case diagram(s) with Use Case descriptions.**



Source <http://www.uml-diagrams.org/airport-checkin-uml-use-case-diagram-example.html>



Source : http://conceptdraw.com/samples/business-process-diagrams-unified-modeling-language

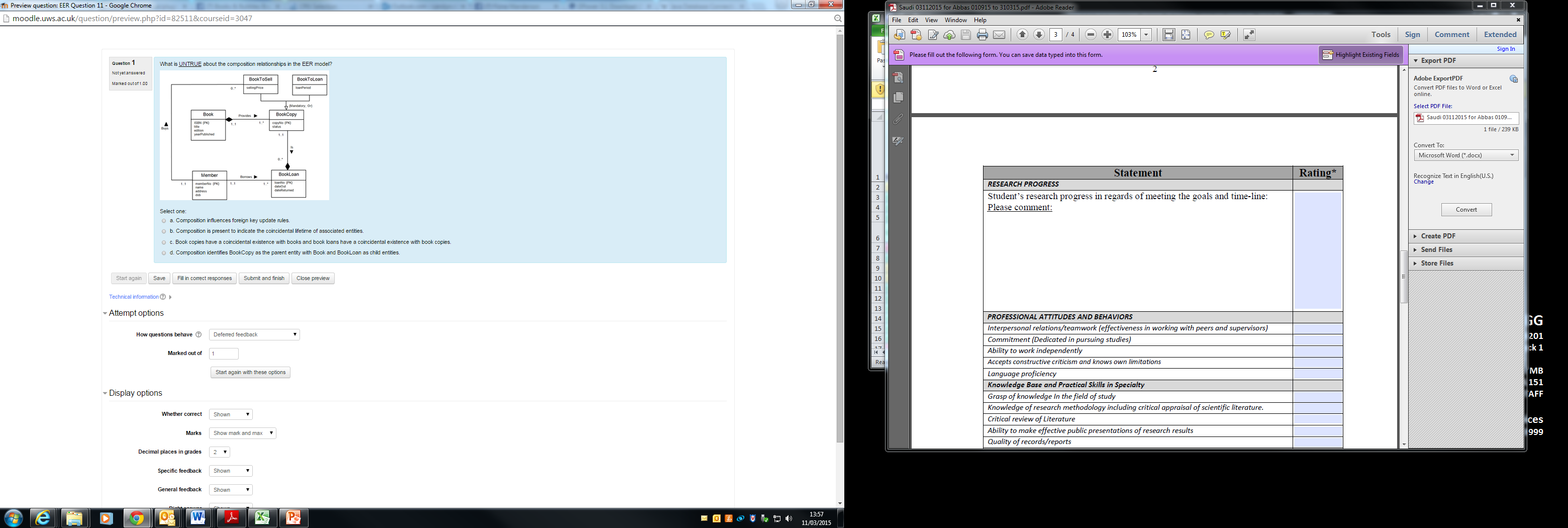
**4. Conceptual Database Design - EER Model (15%)**

**Tasks**

*Present an enhanced entity relationship (EER) diagram to represent the conceptual design for your case study database.*

*Your EER model should be drawn using Unified Modelling Language (UML) notation and model all of the data requirements for your business process(es). The EER model should be easily readable and fit one side of an A4 page. All attributes should be identified with the primary key attributes tagged {PK} and all relationships appropriately named and described using multiplicity. You should use a drawing tool or better still a CASE tool that can create UML class diagrams. Although the concepts of EER models do not include foreign keys – many of the CASE tools do create diagrams that include foreign keys. Therefore, please present a diagram that either shows all FK or none at all – both representations will be considered equally correct. Your EER model should include between 10 - 12 entities (not including subclasses), each described using between 5 and 10 attributes, one example of superclass/subclasses (with at least 2 subclasses), one example of aggregation or composition.*

**Replace with your EER (Note that aggregation is not shown in this example and your model should contain more entities and attributes than shown here)**



**5. Logical Database Design - Relational Schema (15%)**

**Tasks**

*Map your EER model to a relational schema* *to represent the logical design for your case study database. Use the following format for each relation and include Foreign Key and Derived Attribute where appropriate.*

*In particular describe how you have chosen to represent the superclass/subclasses, aggregation or composition concepts in your relational schema.*

**Replace with your relational schema**

|  |  |
| --- | --- |
| **Relation** | **Comments** |
| *List each relation separately.* | *Add any information that you wish to about your table. For example, if you want to explain your choice of Derived Attribute, Update/Delete rule.*  *In particular, comment on the guidelines for the mapping of your enhanced ER concepts. If you veer away from the guideline – say why here*. |
| **Member** (memberNo, firstName, lastName, dob, gender, dateJoined, eMail, passwordHash, passwordSalt, dateSinceJoined, fullName, membershipTypeID)  **Primary Key** memberNo  **Foreign Key** membershipTypeID **references** Membership (membershipTypeID) **on Delete** No Action **on Update** Cascade  **Derived Attribute** dateSinceJoined  **Derived Attribute** fullName | *The dateSinceJoined attribute is frequently accessed to ensure that members receive appropriate communications (e.g. eMails).*  *The fullName attribute is displayed on the member’s login page and is derived and stored for fast retrieval and display.*  *(Note - Derived is aka Calculated or Computed)* |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**6. Physical Database Design - Data Dictionary**

**Tasks**

*Present a data dictionary to represent the physical design for your case study database.*

*Your data dictionary should describe how the relational schema will be represented as physical tables.*

*Each relation name now described as table - should have the prefix* ***tbl****. For each column in your tables, describe the meaning of the column; identify an appropriate data type; specify whether nulls are allowed; identify any special role played (i.e. PK, PPK (partial primary key) or FK); any default value; any appropriate constraint/property (e.g. data range, example values, formula for calculation, business rules) and finally the data classification (i.e. Restricted, Private or Public).*

**Replace with your data dictionary for all tables**

**tblMember** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property (including Data Classification)** |
| memberID | Uniquely identifies each member | int | No | PK |  | (Private)  Set identity specification seed as 1 and increment as 1. |
| firstName | First name of member | nvarchar (30) | No |  |  | (Private) |
| lastname | Last name of member | nvarchar (30) | No |  |  | (Private) |
| dob | Date of birth of member | date | No |  |  | (Private) |
| gender | Gender of member | nchar(1) | Yes |  | ‘F’ | (Private)  Only hold ‘F’ or ‘M’ or ‘U’ (Constraint ) |
| dateJoined | Date member joined | date | No |  | getDate() | (Private) |
| eMail | Member’s e-mail |  |  |  |  | (Restricted) |
| password | User’s password hashed. | binary(64) | No |  |  | (Restricted)  Password must be hashed using e.g. Hashbytes( ) using algorithm e.g. SHA2\_512 |
| salt | Used to strengthen password protection | uniqueIdentifier | No |  |  | (Restricted)  Random unique string combined with password – then hashed. |
| daysSinceJoined | The number of days that have passed since becoming a member |  |  |  |  | (Private)  Use DATEDIFF function (Calculated). |
| fullName | Displays first and last name of member together. |  | Yes |  |  | (Private)  Use RTRIM (Calculated) and persistent storage. |
| membershipTypeID | Identifies the member’s current type of membership. |  |  | FK |  | (Private)  References Membership (membershipTypeID) on Delete No Action on Update Cascade |

**7. Prototype - Database Implementation (15%)**

**Tasks**

*Present evidence of your database by displaying two database diagrams and screen dumps showing the rows in each of your tables.*

*Present two database diagrams.*

*Diagram 1 – Display using* ***Table View – Column.*** *Ensure that all entities, attributes and named relationships are visible and that the database diagram is easily readable and can fit on one side of a A4 page. (Purpose of the diagram is to view entities, column and relationship names.)*

*Diagram 2 – Display using* ***Table View - Standard*** *but in this case ensure that the descriptions of each column are clear. (Purpose of the diagram is to view details of the columns of each table).*

**Replace with your diagram**

**Example - The *Stream2U* Database shown in Table View – Column**



**Replace with your diagram**

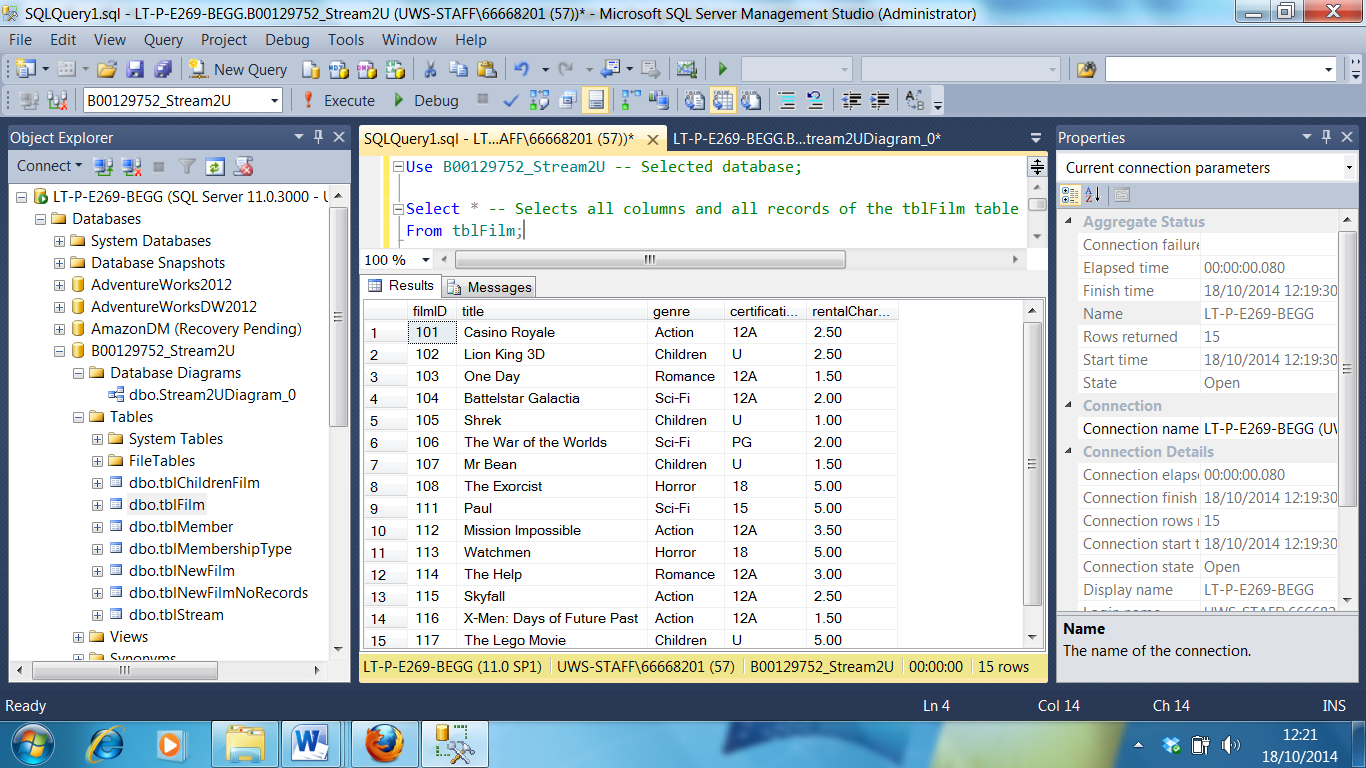
**Example - The *Stream2U* Database shown in Table View – Standard**



**Ensure that all of your parent tables have at least 5 rows, child tables with at least 15 rows and that you can demonstrate your useful database objects working in Section 8.**

**Replace with screen dumps of rows in each of your tables.**

**Example - The *tblFilm* table showing all rows**



**8. Prototype - Useful Database Objects (15%)**

**Tasks**

*Describe each object and including the purpose of each object in relation to satisfying a user view requirement. Use screen dumps of your database to present evidence that your useful database objects have been created and when possible show evidence of your objects in use.*

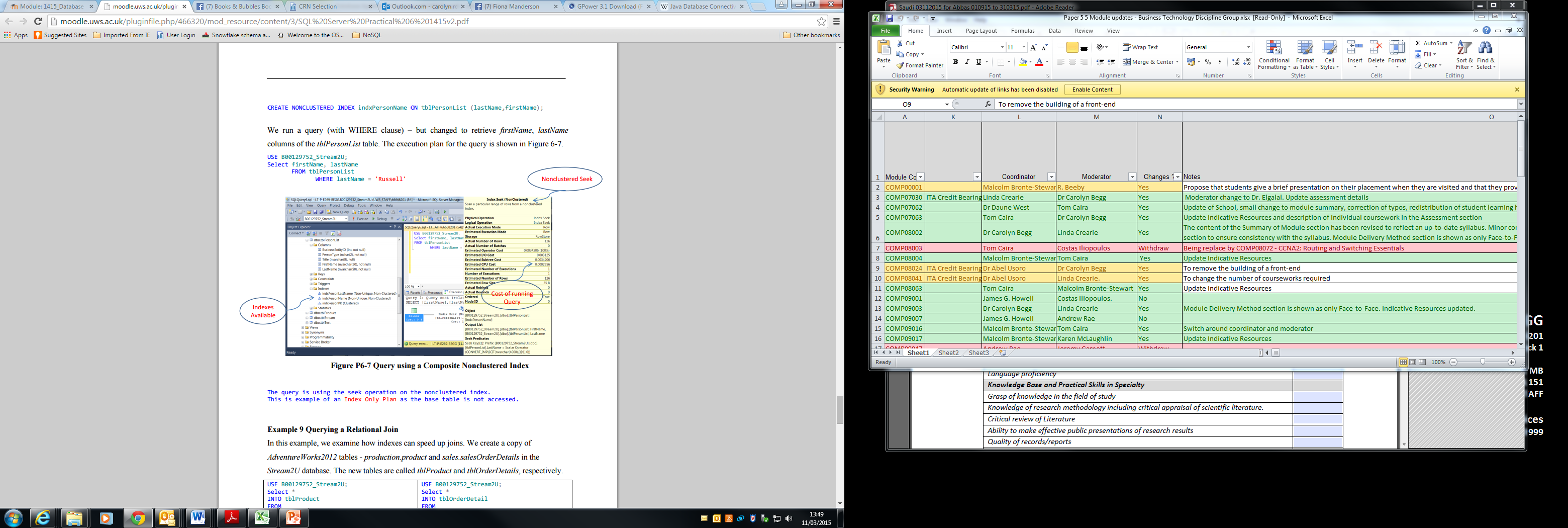
*You should create nonclustered indexes, 5 views, 5 SQL scripts and 5 stored procedures (or triggers). You should use these examples to demonstrate the breadth and depth of your knowledge. Identify your objects by prefixing the object names with* ***indx*** *for indexes,* ***vw*** *for views,* ***q*** *for the scripts and* ***sp*** *for stored procedures (or triggers). Use SQL code and screen dumps to show the creation and where appropriate the results of executing your database object. For each database object, identify the user view user and requirement being supported and where appropriate - the guideline that you are following to justify each index, the type of view for each view and the purpose of each SQL script and stored procedure.*

**Replace entries with your database objects**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Object Type** | **Purpose**  **(Identify user view(s))** | **Code to Create** |
| *Use appropriate prefix to name object* | *Specify if object is an index, view, query or stored procedure (trigger)* | *Identify the user view(s) and why the object has been created. Usually only one purpose will be sufficient. For indexes identify the guideline (GL) that supports the creation of object.* | *Cut and paste code to create object to here* |
| indxPersonName | Composite Non-clustered index | For use by Member User View  1. To speed up searches of the *tblPersonList* table using member’s name*.* (GL 4)  2. To support an index only plan*.* (GL 7) | CREATE NONCLUSTERED INDEX indxPersonName ON tblPersonList (LastName, FirstName); |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Present screen dumps to show that the objects described above physically exist in the database and when possible and appropriate show evidence (e.g. results set) of the objects executing.**

**indxPersonName**



**Submit Your Coursework**

**Tasks**

*Before submitting the report, enter the name or names of the students responsible for the work submitted for each section of this coursework in the Summary of your Coursework Marks & Feedback form (found near the beginning of this template). This will be used as a rough guide as to the workload spread across the group and will not use the percentage of work done unless the majority of the group wish to highlight that a member’s contribution was disproportionately low compared with the work of others.*

*Using a link in the Formal Assessment Coursework (50%) section on MyUWS (Moodle) - upload your coursework report (as a pdf document) which includes all the sections detailed above by the set deadline (see Announcements section).*