**Database Development (COMP08002)**

**Coursework for *Stream2U* Case Study (Extended)**

|  |  |  |
| --- | --- | --- |
| **Student Name(s)** | **BannerID(s)** | **\* Student Signature** |
| Robbie Gow | B00315699 |  |
| Stephen Gault |  |  |
| Chris Allan |  |  |

\* Each student must sign to indicate that he/she contributed an appropriate amount of work to justify receiving a group mark for this coursework.

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

|  |  |
| --- | --- |
| **Lecture Name** | **Your Campus?**  **(Ayr, Dumfries, Hamilton or Paisley)** |
| Database Development | Hamilton |

**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) |
| **Relational Schema** | *Enter Name(s)* | 20 |  |  |
| *Feedback (Summary)* | RG |  |  |  |
| **Data Dictionary** | *Enter Name(s)* | 20 |  |  |
| *Feedback (Summary)* | RG |  |  |  |
| **Create Database** | *Enter Name(s)* | 30 |  |  |
| *Feedback (Summary)* | CA |  |  |  |
| **Insert Records** | *Enter Name(s)* | 10 |  |  |
| *Feedback (Summary)* |  |  |  |  |
| **SQL Statements** | *Enter Name(s)* | 20 |  |  |
| *Feedback (Summary)* |  |  |  |  |
| **Presentation** | *Enter Name(s)* |  |  |  |
| *Feedback (Summary)* |  |  |  |  |
| **Final Total & Grade** |  | **100** |  |  |

1. **ER model for *Stream2U***

Actor

actorID {PK}

firstName

lastName

gender

nationality

/fullName

1..1

*Performs*

MembershipType

mTypeID {PK}

mTypeName

minFilmsStream

mBenefit

0..\*

ActorFilm

actorFilmID {PK}

character

voiceOnly

0..\*

1..1

*Describes*

*Includes*

1..1

1..\*

Member

memberID {PK}

firstName

lastName

dob

gender

dateJoined

eMail

passwordHash

passwordSalt

/daysSinceJoin

/fullName

Stream

streamID {PK} memberRating

requestView

startView freeStream

memberReview

reviewDate

/requestToStart

Film

filmID {PK}

title

genre [1..\*]

classification

rentalCharge

*Delivers*

*Requestses*

1..1

1..\*

1..1

1..\*

1..\*

*Directs*

0..\*

deviceAddDate

1..\*

*Has*

Director

directorID {PK}

firstName

lastName

gender

nationality

/fullName

1..\*

*Produceses*

Device

deviceTypeID {PK}

deviceType

deviceModel

Make

makeID {PK}

makeName

1..\*

1..1

[Assumption – An actor can play only one character in a given film.]

**2. Relational Schema for *Stream2U* Database**

**tblMembershipType** (mTypeID, mTypeName, minFilmsStream, mBenefit)

**Primary Key** mTypeID

**tblMember** (memberID, firstName, lastName, dob, gender, dateJoined, eMail, passwordHash, passwordSalt, daysSinceJoin, fullName, mTypeID)

**Primary Key** memberID

**Foreign Key** mTypeID references tblMembershipType(mTypeID)

**Calculated** fullName, daysSinceJoin

**tblDevice** (deviceTypeID, deviceType, deviceModel)

**Primary Key** deviceTypeID

**tblMake** (makeID, makeName)

**Primary Key** makeID

**tblFilm** (filmID, title, genre, classification, rentalCharge)

**Primary Key** filmID

**tblActorFilm** (actorFilmID, actorID, filmID, character, voiceOnly)

**Primary Key** actorFilmID

**Foreign Key** actorID references tblActor(actorID), filmID references tblFilm(filmID)

**tblActor** (actorID. firstName, lastName, gender, nationality, fullName)

**Primary Key** actorID

**Calculated** fullName

**tblDirects** (filmID, directorID)

**Foreign Key** filmID references tblFilm(filmID), actorID references tblActor(actorID)

**tblDirector** (directorID, firstName, lastName, gender, nationality, fullName)

**Primary Key** directorID

**Calculated** fullName

**tblStream** (streamID, memberID, filmID, memberRating, requestView, startView, freestream, memberReview, reviewDate, requestToStart)

**Primary Key** streamID

**Calculated** requestToStart

**Foreign Key** memberID references tblMember(memberID)

**Foreign Key** filmID references tblFilm(filmID)

**3. Data Dictionary for *Stream2U* Database**

The **MembershipType** Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property (including data classification)** |
| mTypeID | Unique identifier of type | tinyint | No | PK |  | (Public) |
| mTypeName | Name of membership type | nvarchar(8) | No |  |  | Only values ‘Bronze’, ‘Silver’, ‘Gold’, ‘Platinum’  (Public) |
| minFilmsStream | Minimal number of films streamed to qualify | smallint | No |  |  | (Public) |
| mBenefit | Describes the benefit achieved on streaming more than a minimum number of films | nvarchar(100) | No |  |  | (Public) |

The **tblMember** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Column Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| memberID | Uniquely identifies each member | int | No | PK |  | Set identity specification seed as 1 and increment as 1.  (Restricted) |
| firstName | First name of member | nvarchar (30) | No |  |  | (Restricted) |
| lastname | Last name of member | nvarchar (30) | No |  |  | (Restricted) |
| dob | Date of birth of member | date | No |  |  | (Restricted) |
| gender | Gender of member | nchar(1) | Yes |  | ‘F’ | Only hold ‘F’, ‘M’ or U (Unassigned)  (Restricted) |
| dateJoined | Date member joined | date | No |  | getDate() | (Restricted) |
| eMail | eMail address of member | varchar(40) | No |  |  |  |
| passwordHash |  |  |  |  |  |  |
| passwordSalt |  |  |  |  |  |  |
| /daysSinceJoin | Days since member has joined |  | No |  |  |  |
| /fullName | Displays first and last name of member together. |  |  |  |  | Use RTRIM (Calculated) and persistent storage.  (Restricted) |
| mType | Identifies the member’s current membership type | tinyint | No | FK | ‘Bronze’ | (Restricted) |

The **tblDevice** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| deviceTypeID | Uniquely identifies each device | int | No | PK |  |  |
| deviceType | Type of device that is being used | nvarchar (25) | No |  |  |  |
| deviceModel | Model of device that is being used | nvarchar (25) | No |  |  |  |

The **tblMake** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| makeID | Uniquely identifies each Make | Int | No | PK |  |  |
| makeName | Name of the Make | nvarchar (30) | No |  |  | (Restricted) |

The **tblFilm** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| filmID | Uniquely identifies each film | int | No | PK |  | Set identity specification seed as 101 and increment as 1.  (Public) |
| title | Title of film | nvarchar  (50) | No |  |  | (Public) |
| genre | Genre of film | nchar(20) | No |  |  | Only hold: Only hold: ‘Action’, ‘Adult’, ‘Children’, ‘Comedy’, ‘Horror’, ‘Romance’ or ‘Sci-Fi’  (Public) |
| classification | British Board classification of film | nvarchar (3) | No |  | 12A | Only hold: ‘U’, ‘PG’, ‘12A’, ‘15’ or ‘18’  (Public) |
| rentalCharge | Nominal charge to stream a film. Used by company to calculate monetary value of films streamed. | money | No |  | 3.00 | Only hold between 1.00 and 10.00  (Restricted) |

The **tblActorFilm** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| actorFlimID | Uniquely identifies each Actor in a film | int | No | PK |  |  |
| actorID | Uniquely identifies each Actor | Int | No | FK |  |  |
| filmID | Uniquely identifies each film | Int | No | FK |  |  |
| character | Role of actor in a film | nvarchar (30) | No |  |  | (Restricted) |
| voiceOnly | Identify if actors role is voice only | bit | No |  | ‘0’ | Only hold ‘0’ or ‘1’  (Restricted) |

The **tblActor** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| actorID | Uniquely identifies each Actor | int | No | PK |  |  |
| firstName | First name of Actor | nvarchar (30) | No |  |  | (Restricted) |
| lastName | Last name of Actor | nvarchar (30) | No |  |  | (Restricted) |
| gender | Gender of Actor | nchar(1) | Yes |  | ‘F’ | Only hold ‘F’, ‘M’ or U (Unassigned)  (Restricted) |
| nationality | Nationality of Actor | nvarchar (30) | No |  |  | (Restricted) |
| /fullName | Displays first and last name of Actor together. |  |  |  |  | Use RTRIM (Calculated) and persistent storage.  (Restricted) |

The **tblDirects** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| filmID | Uniquely identifies each film | Int | No | FK |  | Set identity specification seed as 101 and increment as 1.  (Public) |
| directorID | Uniquely identifies each director | Int | No | FK |  | Set identity specification seed as 101 and increment as 1.  (Public) |

The **tblDirector** table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default**  **Value** | **Constraint /Column Property** |
| directorID | Uniquely identifies each director | int | no | PK |  | Set identity specification seed as 101 and increment as 1.  (Public) |
| firstName | First name of director | nvarchar (30) | no |  |  | (Restricted) |
| lastName | Last name of director | nvarchar(30) | no |  |  | (Restricted) |
| gender | Gender of director | nchar(1) | yes |  | ‘F’ | Only hold ‘F’, ‘M’ or U (Unassigned)  (Restricted) |
| nationality | Nationality of director | nvarchar (30) | no |  |  | (Restricted) |
| /fullName | Displays first and last name of member together. |  |  |  |  | Use RTRIM (Calculated) and persistent storage.  (Restricted) |

The **tblStream** Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Meaning** | **Data Type** | **Nulls** | **PK / FK** | **Default Value** | **Add Constraint/Column Property** |
| streamID | Uniquely identifies each request to stream a film. | int | No | PK |  | Set identity specification seed as 1001 and increment as 1.  (Restricted) |
| memberID | Identifies member requesting film. | int | No | FK |  | (Restricted) |
| filmID | Identifies film to be streamed. | int | No | FK |  | (Restricted) |
| memberRating | Member’s rating of film from 0.5 star (very poor) to 10 stars (brilliant). | numeric(3,1) | Yes |  | 5 | Only values 0.5 to 10.0 (Constraint).  (Public) |
| requestView | Date and time that member made request. | datetime | No |  | getDate() | (Restricted) |
| startView | Date and time that member started to view film. | datetime | No |  |  | (Restricted) |
| freeStream | Identify if stream is free or not | bit | No |  | 0 | Only hold ‘0’ or ‘1’  (Restricted) |
| memberReview | Member’s review of film | nvarchar(MAX) | No |  |  | (Public) |
| reviewDate | Date member reviewed | Date | No |  | getDate() | (Restricted) |
| /requestToStart | Time (in seconds) between member’s request to rent (stream) film and member starting to watch film. |  |  |  |  | Use DATEDIFF() function and persistent storage.  (Restricted) |

**4. Build the *Stream2U*database**

The Stream2U Database (Diagram shown using Standard Mode)

**The Stream2U Database (shown in Table View – Standard) – Replace diagram with your Extended Version**



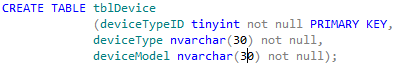
**The Stream2U Database (shown in Table View – Column Names) – Replace diagram with your Extended Version**



**5. Add Records into the *Stream2U*database**

**Ensure** that you have between 5 and 10 records in each of your tables with the exception of the tblStream table which requires to have between 20 – 30 records.

**6. Querying your *Stream2U* database**

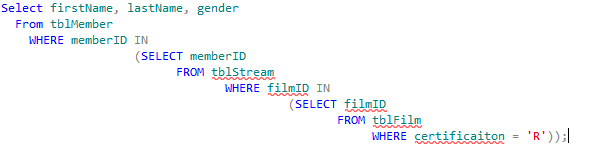
****

****

****



****

****

