**School of Computing**
**ST1501 Data Engineering CA1**
**AY2022/2023 Semester 1**

_____

# A.    Instructions and Guidelines

1.    This is an **INDIVIDUAL** assignment which requires the student to design and setup a database to support a business scenario and demonstrate the competency in writing SQL statement.

2.    The deadline of this assignment is on 6/6/2022 9 am.

3.    Submissions should be made via the ST1501 CA1 Assignment Submission link by the stated deadline.

4.    **A compulsory demo/interview will be conducted**. During the session, you may need to give a short demonstration of the database that you have created and the SQL statements that you have submitted. Your module tutor will ask questions related to database design, SQL statements.

5.    This assignment will account for **35%** of the module grade.

6.    No marks will be awarded, if the work is copied or you have allowed others to copy your work.

7.    50% of the marks will be deducted for assignments that are received within ONE (1) calendar day after the submission deadline. No marks will be given thereafter. Exceptions to this policy will be given to students with valid LOA on medical or compassionate grounds. Students in such cases will need to inform the module tutor as soon as reasonably possible.  Students are not to assume on their own that their deadline has been extended.

8.    **Warning**: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person.
Plagiarism is a serious offence, and if you are found to have committed, aided, and/or abetted the offence of plagiarism, disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

## B.    The Business Scenario

**Background**



Jackson Chan and a few friends have pooled in resources to set up a video rental company (VHS Tapes and Equipment Rental) with intention to have several outlets in the States. Aside from the capital they have invested, the friends need a software application specialist to develop a system for them to manage the video rental, transactions and the daily operations of the business. The system will facilitate an information system that a video company can use to manage the rentals of videos and video equipment.

The specialist suggests to have a client server application used by multiple users at several locations.  Users will be able to interface with the system via Web Page while connected to the Internet.  The system will provide communication to a centralized RDBMS database system. The system is created by using any current market web application framework like ReactJS or Flask with Django or NodeJS as the middle tier.

**System General Requirements**

The system will provide necessary information via a web page to manage video, video camera and camcorder rentals, reservations and returns.  The system will have the ability to be scalable, which means it will be able to functions if it is changed in size or volume to meet a user needs.  The architecture of system will be opened which will allow the customer to add to the system and adapt it as necessary.
- Provide user authentication/access levels
- Store information about stores
- Store information about employees
- Store information about customers

- Store information about transactions
- Store information about movies
- Store information about video cameras.
- Store information about VCRs
- Store information about rentals
- Provide catalog of all movies
- Provide reports:
    - Total number of outgoing movies per store and per month
    - The number of rentals per customer
    - Ten Most frequently reserved titles for the last year
    - Average number of outgoing movies per store
    - Monthly revenue of the company for the 12 months

The store will have a catalogue. The catalogue of movies contains the title of the movie, the producer, the director, two lead actors, the category of movie, the number of VHS cassettes and DVDs, and charge per day. Items can be categorised under the types:

- movie
- vcr
- videocamera

In future, more items can be added to the list for rentals. The video store carries multiple copies of the same title, and a store could have been assigned any number of copies of each title. A store that has more copies of a given title than assigned to it will return these at the end of each week to head office once, which redistributes them to appropriate stores. Similarly, the store carries multiple pieces of equipment under its charge and customers may rent a particular piece of the equipment. Store has inventory of all items and the web page they will be using has management capabilities to:
- Query, add, update, and remove movie, and equipment
- Show status of movie, and equipment
- Show rentals of items
- Handle reservation, rental and returning of movies and equipment by members. Transaction can be either 'Reserve' or 'Rent'. Assume that a new record is created for each individual transaction. i.e a reserve transaction will not later be updated to rent if the customer decides to rent the items out.
- Manage payments for rental by members (including late charges)
  Late charges are computed based on the number of days overdue. For movies, $0.50 per day and for equipment $1.50 per day. There is a non-partial return policy enforced once the items are on loan for each particular transaction. It means that a customer must return all loaned items at the same time and not on each item basis for that transaction. The amount paid covers the rental charge as well as the penalty charge. It will be computed once all loaned items are returned including any late charges.

The application must provide a user interface for Video Store for renting videos, video cameras and camcorders. Some snapshots of the application are appended in the Appendix for a better appreciation of the whole system.

Various employees will use the system across various stores. The system needs to be protected from unauthorized users/access to prevent malicious activities. The application will need to provide the means to limit access and functions of the system by user. A manager may have the ability to create, modify, and delete information, but a sales associate can only rent or reserve movies and equipment or view information about customers. The system is flexible in allowing the customer define the level of authorization. Employees are paid under the following categories:

- manager
- assistantManager
- partTime

## C. The Submission Deliverables

Deliverable should be a zip file with the following file-naming convention: "ST1501-YourClass-YourStudentID-YourName.zip.

**Example ST1501-DITFT2B01-9999999-Benson Ang.zip**

Zip file should include the following items:
a) Use the submission template provided for the following:
   - The Entity Relationship Diagram to support all the requirements stated in section B.
b) Two SQL Scripts in SQL extension
   - Create Table SQL Script for all tables
     All tables must be implemenatable in MS Server taking into consideration the resolution of M:N relationships
   - Explain your database design (describe the purpose of each table, list the required fields (indicate PK), the relationship between tables (indicate FK) disregard referential actions for FK
   - Insert Records SQL Script for all tables
     Create sufficient variety of data to be inserted into the tables so that you are able to fully support the SQL SELECT statements for the queries and their corresponding query results.
c) Write SQL statement for each query and display their results.
   - Query 1 - View total number of outgoing movies per store and per month.
   - Query 2 - View the average number of rental days
     i) per customer
     ii) per title
   - Query 3 - View 10 most frequently reserved titles for last year
     The query must be able to retrieve records from any one year past from date of query
   - Query 4 - View 100 best customers in terms of money paid to the company
   - Query 5 - View monthly revenue of the company for the 12 months
     The query must be flexible to retrieve any previous 12 months from any date of query and exclude penalty charges

> ➢ Query 6- View the total amount of penalty incurred by each customer for movies and  equipment respectively

## D.    Submission Template

**Entity Relationship Diagram**

**Database Diagram from SQL Server after tables creation with PK and FK**

**Database Design Explanation**

List all the tables and their fields. Indicate the PK and FK and their referenced tables.

| Table | Purpose | Field | DataType | Primary Key | Foreign Key |
|---|---|---|---|---|---|
| *Table 1* | | *Field1* | | *Field1* | *Field4 references Table 2 Field1* |
| | | *Field2* | | | |
| | | *Field3* | | | |
| | | *Field4* | | | |
| *Table 2* | | *Field1* | | *Field1* | |
| | | *Field2* | | | |
| | | *Field3* | | | |
| *Table 3* | | *Field1* | | *Field1* | |
| | | *Field2* | | | |
| | | *Field3* | | | |

Table – Name the table
Purpose – State what the table is used for and contains what details
Field – List all the columns of the table
DataType – State varchar, integer, decimal etc
Primary Key – List the PK (Simple or Composite)
Foreign Key – List the FK (Simple or Composite) and its reference table

**SQL Queries and Results**

| Query 1 | Query 1 Results |
|---|---|
| | |
| Query 2(i) | Query 2(i) Results |
| | |
| Query 2(ii) | Query 2(ii) Results |
| | |
| Query 3 | Query 3 Results |
| | |
| Query 4 | Query 4 Results |
| | |

|  |  |
| --- | --- |
|  |  |
| Query 5 | Query 5 Results |
|  |  |
| Query 6 | Query 6 Results |
|  |  |

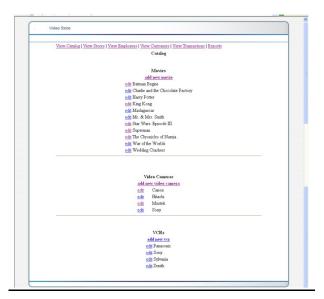## E.    The Assessment Criteria

| Components | | Weightage |
|---|---|---|
| **Database Design (Database Diagram)**<br>• The database supports the described business scenario.<br>• The chosen table names, field names and attributes are descriptive.<br>• The database is normalized. | | 35% |
| **Database Creation**<br>• The Create Table SQL script implements the database design, including the primary key and foreign key definition.<br>• The Insert Records SQL script creates enough data for testing. | | 25% |
| **The Query Statements**<br>• The SQL query statement implemented the query scenarios.<br>• The SQL query result shows that the design supports the query scenarios.<br><br>No mark will be given to the query if the database design cannot support the required query scenario. | Qry1 | 6% |
| | Qry2(i) | 3% |
| | Qry2(ii) | 4% |
| | Qry3 | 4% |
| | Qry4 | 3% |
| | Qry5 | 4% |
| | Qry6 | 6% |
| **CA1 Assignment Demo/Interview**<br>The CA1 Demo/Interview will be conducted where you have:<br>• To explain the database design and the submission<br>• To explain/answer questions given by the module tutor | | 10% |
| **Total** | | 100% |

**\*\*\* End of Assignment Specification \*\*\***

## F.    Appendix

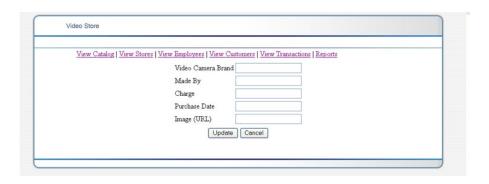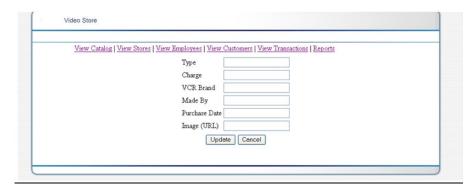### Catalog Information Screen



### Movie Information Screen



### Video Camera Information Screen

### VCR Information Screen



### Stores Information Screen



### Add/Edit  Store Information Screen



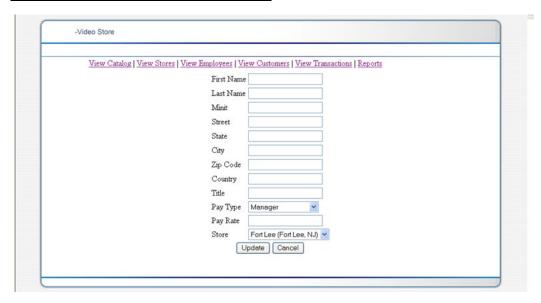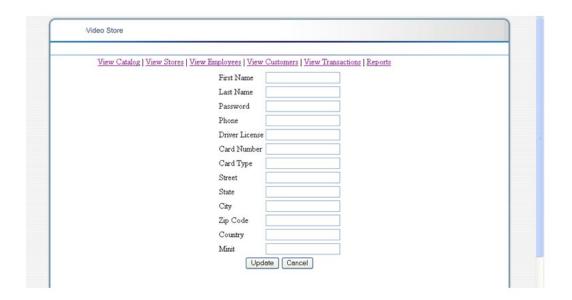### Employee Information Screen

## Add/Edit Employee Information Screen



## Customer Information Screen



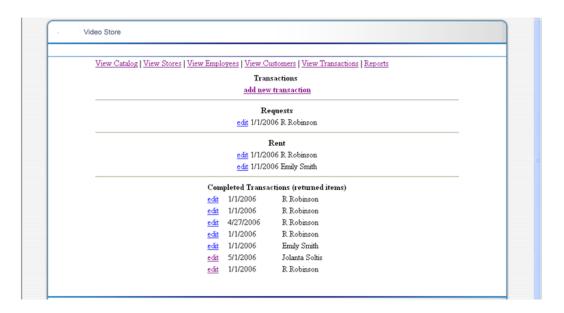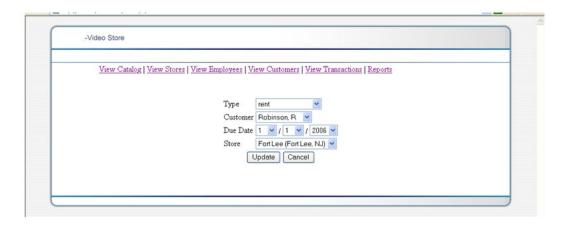## Add/Edit Customer Information Screen

## Transaction Information Screen



## Add/Edit Transaction Information Screen