CP5503 Enterprise Database Systems Oracle

Assignment – Part 1 (15%)

Due: End of Week 5 Prac

**Assignment – Part 1 Objectives & Structures:**

This part provides experience in designing a small database including both logical and physical designs. Beside the main tool **SQL Developer**, you also practice on **Data Modeler**. You are going to create a database to manage screenings and ticketing of Springfield Cinema Multiplex (SCM).

The database should record the details of screenings and sold tickets. SCM consists of 4 cinema theatres. Each theatre can be planned to screen a number of movies each day.

Theatre information should consist of a description and a number of available rows (min 1, max 26). Each row should be identified by a capital letter from ‘A’ to ‘Z’ and the total number of seats (min 1). Each seat can be identified by a row and a seat number, for example, seat ‘A10’: row ‘A’, seat number: 10.

Each movie record should contain its title, rating, description, duration and released date. Each movie should be scheduled to be screened in a number of plans (default 1). Each plan should contain the information of the starting date, the ending date of the plan and the number of screenings per day of the movie. In addition, each plan should also contain the minimum and maximum time to screen the movie a day, for example, 09:00 (default value) and 22:00 (default value) correspondingly. Normally, SCM will plan the coming screening at least one week in advance.

Each screening should contain references to a screening plan and an available theatre. It should also contain the date (DD/MM/YYYY) and time (HH:MM) of the screening. There is at least a break of 30 minutes between 2 consecutive screenings in the same theatre.

Ticket information should consist of a reference to a screening, an issued date, a seat number and a ticket type. At this moment, there are 2 types of tickets: Adult and Concession. Each type will be associated with a price and an expired date (the last date that the price is valid).

There are four tasks of the assignment – Part 1:

**Task1 (5 pts):** You need to create a database schema called **scm\_jcxxxxxx** to store all the database objects for this assignment. Firstly, it consists of the creation of a tablespace named **scmts\_jcxxxxxx** where jcxxxxxx is your jc username. The tablespace should have the initial size of 100 MB and can be extended if required. Secondly, you need to create a user account named **scm\_jcxxxxxx** and grant appropriate privileges to the user for creating the database objects. The default tablespace of **scm\_jcxxxxxx** must be **scmts\_jcxxxxxx**. In addition, the user **scm\_jcxxxxxx** can also have rights to create users and allow them to connect to the database. You need to follow the **principle of least privilege** when granting privileges.

**SUBMISSION FILE(S):**

Firstly, create a folder named **jcxxxxxx** (your jc number, for example, jc165984) to store all of your files in this assignment-Part 1.

For this task, you have to produce a script called **A1Task1.sql** which contains SQL statement(s) for

- creating the **scmts\_jcxxxxxx** tablespace

- creating the **scm\_jcxxxxxx** user

- granting appropriate privileges to the **scm\_jcxxxxxx** user

And put this file into the folder.

**Task 2 (20 pts):** You need to use **Data Modeler** to model the SCM database. It should consist of a logical model, a relational model and a generated DDL script. Here is a suggestion of SCM entities.

Entity: **Theatre**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| theatre\_id | NUMBERIC, Precision=1 | Primary UID |
| theatre\_description | VARCHAR, Size=100 |  |
| theatre\_total\_rows | NUMBERIC, Precision=3 | Mandatory |

Entity: **Theatre\_Row**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| row\_id | NUMBERIC, Precision=4 | Primary UID |
| theatre\_id | NUMBERIC, Precision=1 | Mandatory; Foreign key to the entity Theatre |
| row\_name | VARCHAR, Size=2 | Mandatory |
| row\_total\_seats | NUMBERIC, Precision=3 | Mandatory |

Entity: **Seat**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| seat\_id | NUMBERIC, Precision=5 | Primary UID |
| row\_id | NUMBERIC, Precision=4, | Mandatory; Foreign key to the entity Row |
| seat\_number | NUMBERIC, Precision=3 | Mandatory |

Entity: **Movie**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| movie\_id | NUMBERIC, Precision=4 | Primary UID |
| movie \_title | VARCHAR, Size=100 | Mandatory |
| movie\_rating | VARCHAR, Size=10 | Mandatory |
| movie\_released\_date | DATE | Mandatory |
| movie\_description | VARCHAR, Size=400 |  |
| movie\_length | NUMBERIC, Precision=3 | Mandatory |

Entity: **Screening\_Plan**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| plan\_id | NUMBERIC, Precision=4 | Primary UID |
| movie\_id | NUMBERIC, Precision=4 | Mandatory; Foreign key to the entity Movie |
| plan\_start\_date | DATE | Mandatory |
| plan\_end\_date | DATE | Mandatory |
| plan\_min\_start\_hh24 | NUMBERIC, Precision=2 | Mandatory; The earliest time to start screening |
| plan\_max\_start\_hh24 | NUMBERIC, Precision=2 | Mandatory; The latest time to start screening |
| plan\_no\_of\_screenings | NUMBERIC, Precision=2 | Mandatory; The number of screenings per day of the movie |

Entity: **Screening**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| screening\_id | NUMBERIC, Precision=6 | Primary UID |
| plan\_id | NUMBERIC, Precision=4 | Mandatory; Foreign key to the entity Screening\_Plan |
| theatre\_id | NUMBERIC, Precision=1 | Mandatory; Foreign key to the entity Theatre |
| screening\_date | DATE | Mandatory |
| screening\_start\_hh24 | NUMBERIC, Precision=2 | Mandatory |
| screening\_start\_mm60 | NUMBERIC, Precision=2 | Mandatory |

Entity: **Ticket\_Type**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| t\_type\_id | NUMBERIC, Precision=1 | Primary UID |
| t\_type | VARCHAR, Size=20 | Mandatory |
| t\_type\_price | NUMBERIC, Precision=4, Scale=2 | Mandatory |
| t\_type\_start\_date | DATE | Mandatory; The first date that the price is valid |
| t\_type\_end\_date | DATE | The last date that the price is valid |

Entity: **Ticket**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| ticket\_id | NUMBERIC, Precision=8 | Primary UID |
| t\_type\_id | NUMBERIC, Precision=1 | Mandatory; Foreign key to the entity Ticket\_Type |
| screening\_id | NUMBERIC, Precision=6 | Mandatory; Foreign key to the entity Screening |
| seat\_id | NUMBERIC, Precision=5 | Mandatory; Foreign key to the entity Seat |
| ticket\_date | DATE | Mandatory |

Entity: **Employee**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Datatype** | **Other Information** |
| e\_id | NUMBERIC, Precision=3 | Primary UID |
| e\_fname | VARCHAR, Size=30 | Mandatory |
| e\_lname | VARCHAR, Size=30 | Mandatory |
| e\_email | VARCHAR, Size=50 | Mandatory |
| e\_phone | VARCHAR, Size=20 | Mandatory |
| e\_street | VARCHAR, Size=50 | Mandatory |
| e\_suburb | VARCHAR, Size=30 | Mandatory |
| e\_postcode | VARCHAR, Size=4 | Mandatory |
| e\_gender | VARCHAR, Size=1 | Mandatory |
| date\_employed | DATE | Mandatory |

**SUBMISSION FILE(S):**

Produce a model named **scm\_jcxxxxxx** and store it in the folder **jcxxxxxx**. Data Modeler will automatically create a folder named **scm\_jcxxxxxx** for you.

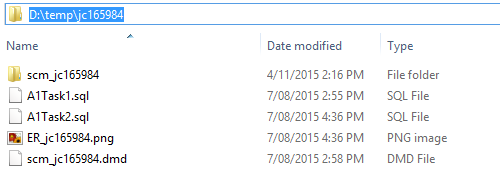
The model should consist of a logical model and a relational model.

In addition, you should also produce:

- A generated script file named **A2Task2.sql**; store it in the same folder **jcxxxxxx**

- An E-R diagram named **ER\_jcxxxxxx.png**; store it in the same folder **jcxxxxxx**

The content of your folder **jcxxxxxx** should be similar to:



**Task 3 (20pts):** You need to modify the script file **A2Task2.sql** to complete additional constraints specified below. In SQL Developer, you need to connect the Oracle DB server as **scm\_jcxxxxxx** user and then execute the modified script file to create the required tables.

Table 1: **Theatre**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| theatre\_id | Value between 1 and 5 |
| theatre\_description | No additional constraints |
| theatre\_total\_rows | Value between 1 and 26 |

Table 2: **Theatre\_Row**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| row\_id | Value must be positive |
| theatre\_id | No additional constraints |
| row\_name | Value between ‘A’ and ‘Z’. |
| row\_total\_seats | Value is equal or greater than 1 |
| The combination of (theatre\_id, theatre\_row\_name) is unique | |

Table 3: **Seat**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| seat\_id | Value must be positive |
| row\_id | No additional constraints |
| seat\_number | Value is equal or greater than 1 |
| The combination of (theatre\_row\_id, seat\_number) is unique | |

Table 4: **Movie**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| movie\_id | Value must be positive |
| movie \_title | No additional constraints |
| movie\_rating | Value must be one of the following items: G, PG, M, MA15+, R18+ |
| movie\_released\_date | Default should be the current date |
| movie\_description | No additional constraints |
| movie\_length | No additional constraints |

Table 5: **Screening\_Plan**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| plan\_id | Value must be positive |
| movie\_id | No additional constraints |
| plan\_start\_date | Default is next Monday (Note: use NEXT\_DAY(SYSDATE, 'MON')). |
| plan\_end\_date | Default value is next Monday + 6 (days) |
| plan\_min\_start\_hh24 | Value between 9 (9:00 am) and 22 (10:00 pm); Default value is 9 |
| plan\_max\_start\_hh24 | Value between 9 and 22 (10:00 pm); Default value is 22; Greater than or equal to plan\_min\_start\_hh24 |
| plan\_no\_of\_screenings | Value is equal or greater than 1 |
| The combination of (movie\_id, plan\_start\_date) is unique | |

Table 6: **Screening**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| screening\_id | Value must be positive |
| plan\_id | No additional constraints |
| theatre\_id | No additional constraints |
| screening\_date | Default should be the current date |
| screening\_start\_hh24 | Value between 9 (9:00 am) and 22 (10:00 pm) |
| screening\_start\_mm60 | Value between 0 (minutes) and 59 (minutes) |
| The combination of (theatre\_id, screening\_date, screening\_start\_hh24, screening\_start\_mm60) is unique | |

Table 7: **Ticket\_Type**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| t\_type\_id | Value must be positive |
| t\_type | No additional constraints |
| t\_type\_price | Value must be positive |
| t\_type\_start\_date | No additional constraints |
| t\_type\_end\_date | Either null or ≥ t\_type\_start\_date |
| The combination of (t\_type, t\_type\_start\_date) is unique | |

Table 8: **Ticket**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| ticket\_id | Value must be positive |
| t\_type\_id | No additional constraints |
| screening\_id | No additional constraints |
| seat\_id | No additional constraints |
| ticket\_date | Default should be the current date and time |
| The combination of (screening\_id, seat\_id) is unique | |

Table 9: **Employee**

|  |  |
| --- | --- |
| **Column** | **Constraints & Comments** |
| e\_id | Value must be positive |
| e\_fname | No additional constraints |
| e\_lname | No additional constraints |
| e\_email | Unique and contains an **@** (Bonus: if being able to set x@x.x) |
| e\_phone | No additional constraints |
| e\_street | No additional constraints |
| e\_suburb | No additional constraints |
| e\_postcode | Four numerical digits; from 0200 to 9999 |
| e\_gender | 1 character: ***M***(Male) or ***F***(Female) |
| date\_employed | Default should be the current date and time |
| The combination of (e\_fname, e\_lname) is unique | |

**SUBMISSION FILE(S):**

Produce a script called **A1Task3.sql** (a modified version of **A1Task2.sql**) and put it in the folder **jcxxxxxx**.

**Task 4 (10pts):** **Loading Data, Creating Sequences**

You should use the file (Ass1\_data.zip) provided to construct INSERT statements to load the data into the tables. In SQL Developer, you need to connect the database server as **scm\_jcxxxxxx** user and then create sequence generators for tables to facilitate the auto generation of primary key values. One sequence is for one table. You will then write INSERT statements to load the data. The sequence generators must be used in the INSERT statements.

**SUBMISSION FILE(S):**

Produce a script called **A1Task4.sql** which contains SQL statement(s) for

- creating the required sequence generators

- inserting data into the tables

And put it in the folder **jcxxxxxx**.

**ASSIGNMENT-PART 1 SUBMISSION:**

Compress (zip) your folder **jcxxxxxx** to a single zip file **jcxxxxxx.zip** and upload it on LearnJCU by 5 pm Friday 18/12/2015.

The final content of your folder **jcxxxxxx** should be similar to:

