Module Code & Module Title

CC5051NA – Databases

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2018-19 Autumn / 2018-19 Spring

Student Name: Summit Shakya

London Met ID: 17030948

College ID: NP01CP4A170019

Assignment Due Date: Week 5

Assignment Submission Date: 28th January, 2019

Table of Contents

Table of Figures

1. Introduction

Suyu Theatre is a location where you can watch various acts and plays throughout the year. The theatre was established in 2006 A.D. Suyu is located at Lazimpat, Kathmandu. Mr. Chris Silva is the man behind the establishment of the theatre. The theatre is associated with Moonlight Play School from where the actors are brought to perform. The company aims to show as many acts as possible to develop the theatre scenario in Nepal. The theatre does not focus on a specific theme. It supports play and acts of all genres. The theatre opens at 10am to 6pm.

2. Current business activities and operation

1. People Record

   The theatre keeps record of all its people involved whether they are customer or an employee. It records the data of its old and new customers. Theatre keeps most if their personal information such as address, age, DOB.

2. Ticket Record

   All the tickets are recorded in the database of the theatre. The ticket contains information such as Ticket ID, Play ID, People ID, type of booking made and available meal.

3. Play Record

   This is the record of the plays that are played in the theatre. One play has a unique ID along with its name, category, duration of the show.

3. Business Rules

1. There must be at least 50 people in the theatre for a play to be performed.
2. The usher is provided with free facility of accommodation and meal.
3. The children below 5 years age does not need ticket to enter the theatre.
4. Two types of booking can be made by customer which are ‘Early Booking’ and ‘Current Booking’.

4. Creating Entities

- **Play**

i. Play_ID
ii. Play_Name
iii. Play_Category
iv. Play_Length

- **Ticket**

i. Ticket_ID
ii. Play_ID
iii. Ticket_Price
iv. StartAt

- **People**

i. People_ID
ii. People_Name
iii. People_Type
iv. People_Country
v. People_Province
vi. People_City,
vii. PeopleStreet,
viii. People_StreetNo,
ix. People_PhoneNo,
x. People_FaxNo,
xi. People_CellNo,
xii. People_Email,
xiii. People_Age,
xiv. People_Sex,
xv. People_DOB,
xvi. People_CustomerType,
xvii. People_StaffType,
xviii. People_StaffSalary

5. Initial ERD

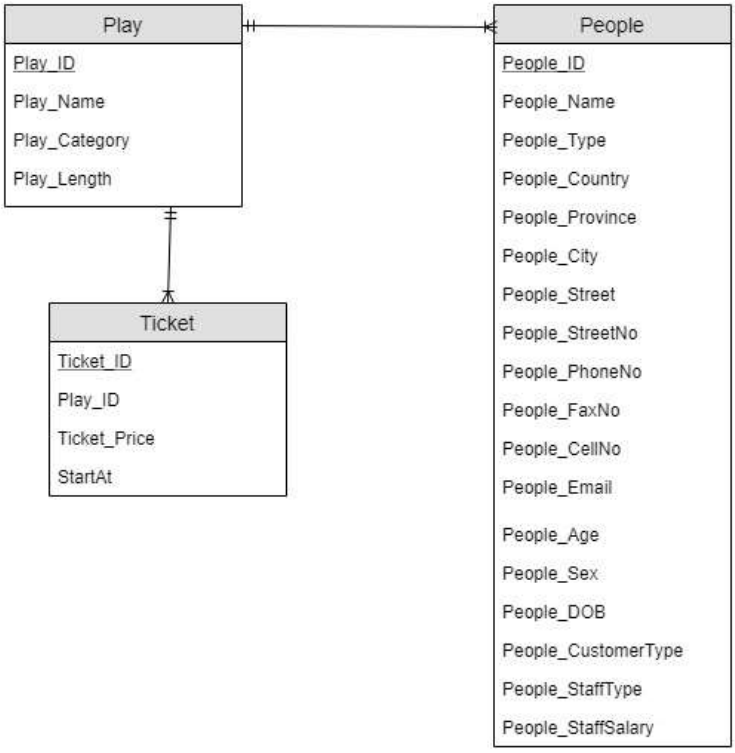

Figure 1Initial ERD

6. Normalisation

Scenario

In the theatre, there is a single play being acted. The particular show can be viewed by many people at a time. One audience will have multiple tickets.

UNF

People{Play{Ticket}}

People (People_ID, People_Name, People_Type, People_Country, People_Province, People_City, PeopleStreet, People_StreetNo, People_PhoneNo, People_FaxNo, People_CellNo, People_Email, People_Age, People_Sex, People_DOB, People_CustomerType, People_StaffType, People_StaffSalary {Play_ID, Play_Name, Play_Category, Play_Length {Ticket_ID, Ticket_Price, StartAt, Meal_ID, Meal_Type, Meal_Price, Meal_Free, Accomodation_Type, Booking_ID, Booking_Type, Booking_Date})

1NF

People1 (People_ID, People_Name, People_Type, People_Country, People_Province, People_City, People_Street, People_StreetNO, People_PhoneNo, People_FaxNo, People_CellNo, People_Age, People_Sex, People_DOB, People_CustomerType, People_StaffType, People_StaffSalary)

Play1 (People_ID, Play_ID, Play_Name, PlayCategory, PlayLength)

Ticket1 (Play_ID, People_ID, Ticket_ID, Ticket_Price, StartAt, Meal_ID, Meal_Type, Meal_Price, Meal_Free, Accomodation_Type, Booking_ID, Booking_Type, Booking_Date)

2NF

Table: Play

People_ID , Play_ID →

Play_ID →

People_ID →

Table: Ticket

Play_ID →

Ticket → StartAt, Price

PlayID, TicketID → Booking_ID, Booking_Type, Booking_Date, Meal_No, Meal_Price, Meal_Free, Meal_Type, Accomodation

People2 (People_ID, People_Name, People_Type, People_Country, People_Province, People_City, People_Street, People_StreetNO, People_PhoneNo, People_FaxNo, People_CellNo, People_Email, People_Age, People_Sex, People_DOB, People_CustomerType, People_StaffType, People_StaffSalary)

Play2(People_ID , Play_ID**)**

PlayInfo (Play_ID, Play_Name, Play_Category, Play_Length)

Ticket2 (Ticket_ID, StartAt, Ticket_Price)

TicketInfo2 (Ticket_ID, Play_ID, People_ID, Booking_ID, Booking_Type, Booking_Date , Meal_ID, Meal_Type, Meal_Price, Meal_Free, Accomodation_Type)

3NF

Ticket_ID → Booking_ID → Booking_Type

Ticket_ID → Meal_ID → Meal_Type, Meal_Price, Meal_Accomodation_Type

People3 (People_ID, People_Name, People_Type, , People_Country, People_Province, People_City, People_Street, People_StreetNo, People_PhoneNo, People_FaxNo, People_CellNo, People_Age, People_Sex, People_DOB)

Customer3 (People_ID, People_CustomerType)

Staff3 (People_ID, People_StaffType, People_StaffSalary)

Play3 (Play_ID, People_ID)

PlayInfo3 (Play_ID, Play_Name, Play_Category, Play_Length)

Ticket3 (Ticket_ID, Ticket_Price, StartAt)

TicketInfo3 (People_ID, Play_ID, Ticket_ID, Booking_ID*, Meal_ID*)

Booking3 (Booking_ID, Booking_Type, Booking_Date)

Meal3 (Meal_ID, Meal_Type, Meal_Price, Meal_Free, Accomodation_Type)

Here, the attribute People_CustomerType has been separated into Customer table and the attributes People_StaffType and People_StaffSalary have been separated into Staff table in 3NF even if there are no transitive dependencies on it. It has been separated to identify the supertype and subtype relation between the entities. The People is supertype and under this supertype lies two entities Customer and Staff. So, these attributes have been separated from the People table into Customer table and Staff table.

7. Final Entity Relationship Diagram (ERD)

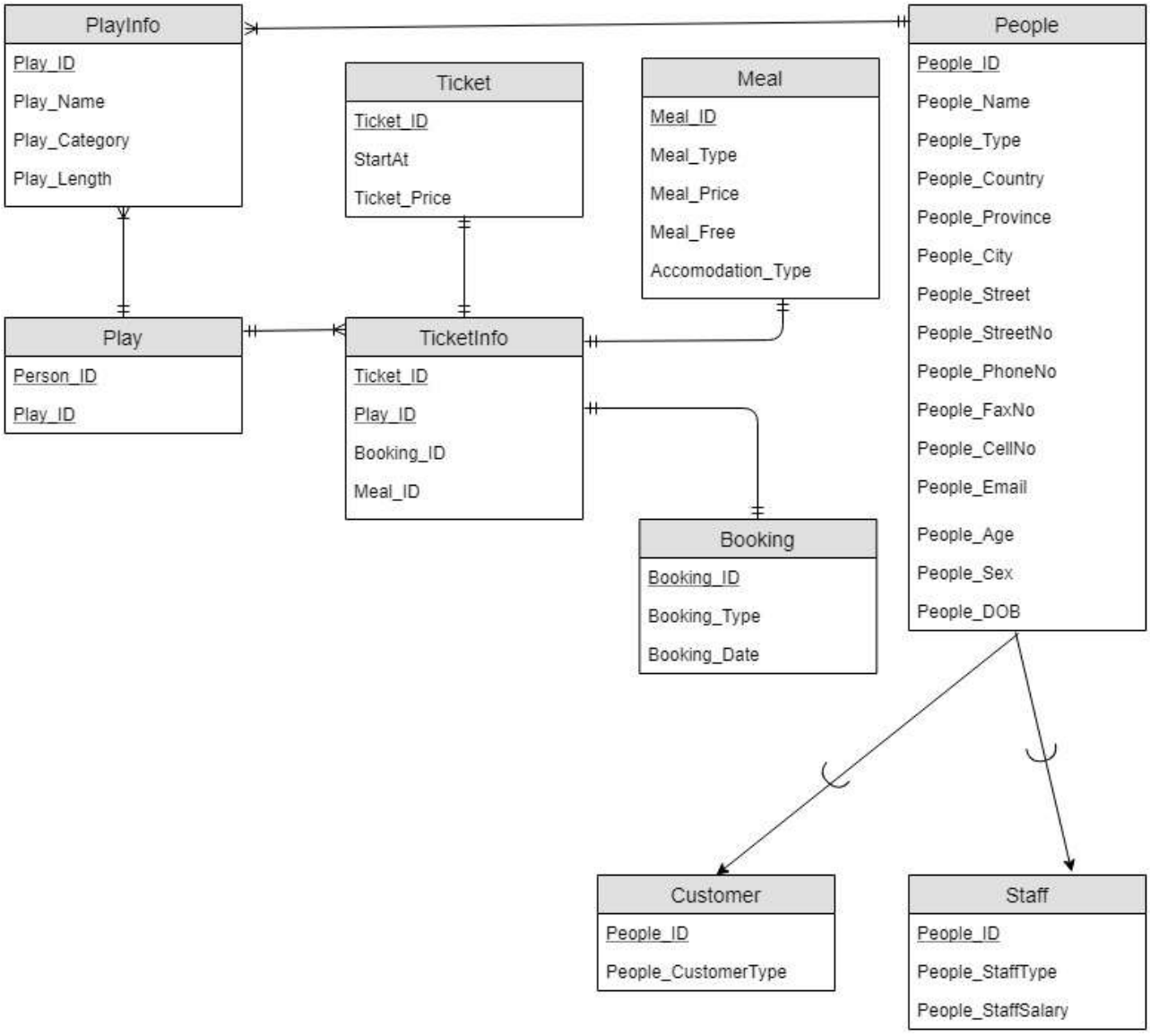

Figure 2 Final ERD

8. Creating Table

```
SQL> connect system
Enter password:
Connected.
SQL> CREATE USER Suyu_Theatre IDENTIFIED BY suyutheatre;

User created.

SQL> GRANT CONNECT, RESOURCE TO Suyu_Theatre;

Grant succeeded.

SQL> CONNECT Suyu_Theatre
Enter password:
Connected.
SQL>
```

Figure 3 Creating user and granting permission

```
SQL> CREATE TABLE PlayInfo(
  2  Play_ID number(5),
  3  Play_Name varchar2(25),
  4  Play_Category varchar(10),
  5  Play_Length varchar(10),
  6  CONSTRAINT plyinf PRIMARY KEY (Play_ID));

Table created.

SQL>
```

Figure 4Creating PlayInfo table

```
SQL> CREATE TABLE Ticket(
  2  Ticket_ID number(10),
  3  StartAt varchar2(10),
  4  Ticket_Price number(5),
  5  CONSTRAINT tick PRIMARY KEY (Ticket_ID));

Table created.

SQL>
```

Figure 5Creating Ticket table

```
SQL> CREATE TABLE Meal(
  2  Meal_ID number(5),
  3  Meal_Type varchar2(10),
  4  Meal_Price number(5),
  5  Meal_Free varchar2(5),
  6  Accomodation_Type varchar(15),
  7  CONSTRAINT meal PRIMARY KEY (Meal_ID));

Table created.

SQL>
```

Figure 6Creating Meal Table

```
SQL> CREATE TABLE Booking(
  2  Booking_ID number(10),
  3  Booking_Type varchar2(10),
  4  Booking_Date varchar2(11),
  5  CONSTRAINT booking PRIMARY KEY (Booking_ID));

Table created.
```

Figure 7Creating Booking Table

```
SQL> CREATE TABLE People(
  2  People_ID number(10),
  3  People_Name varchar2(25),
  4  People_Type varchar2(20),
  5  People_Country varchar2(15),
  6  People_Province varchar2(10),
  7  People_City varchar2(15),
  8  People_Street varchar2(15),
  9  People_StreetNo number(5),
 10  People_PhoneNo number(11),
 11  People_FaxNo number(11),
 12  People_CellNo number(11),
 13  People_Email varchar2(50),
 14  People_Age number(5),
 15  People_Sex varchar2(8),
 16  People_DOB varchar(11),
 17  CONSTRAINT people PRIMARY KEY (People_ID));

Table created.
```

Figure 8Creating People table

```
SQL> CREATE TABLE Customer(
  2  People_ID number(10),
  3  People_CustomerType varchar2(10),
  4  CONSTRAINT peop FOREIGN KEY (People_ID) REFERENCES People(People_ID));

Table created.
```

Figure 9Creating Customer Table

```
SQL> CREATE TABLE Staff(
  2  People_ID number(10),
  3  People_StaffType varchar2(10),
  4  People_StaffSalary number(10),
  5  CONSTRAINT staff FOREIGN KEY (People_ID) REFERENCES People(People_ID));

Table created.
```

Figure 10Creating Staff Table

```
SQL> CREATE TABLE Play(
  2  People_ID number(10),
  3  Play_ID number(10),
  4  CONSTRAINT play PRIMARY KEY (People_ID, Play_ID),
  5  CONSTRAINT play_peo FOREIGN KEY (People_ID) REFERENCES People(People_ID),
  6  CONSTRAINT play_pl FOREIGN KEY (Play_ID) REFERENCES PlayInfo(Play_ID));

Table created.
```

Figure 11Creating Play table

```
SQL> CREATE TABLE TicketInfo(
  2  Ticket_ID number(10),
  3  Play_ID number(10),
  4  People_ID number(10),
  5  Booking_ID number(10),
  6  Meal_ID number(10),
  7  CONSTRAINT tickinfo PRIMARY KEY (People_ID, Play_ID, Ticket_ID),
  8  CONSTRAINT tickinfo_per FOREIGN KEY (People_ID) REFERENCES People(People_ID),
  9  CONSTRAINT tickinfo_tick FOREIGN KEY (Ticket_ID) REFERENCES Ticket(Ticket_ID),
 10  CONSTRAINT tickinfo_play FOREIGN KEY (Play_ID) REFERENCES PlayInfo(Play_ID),
 11  CONSTRAINT book_fk FOREIGN KEY (Booking_ID) REFERENCES Booking(Booking_ID),
 12  CONSTRAINT meal_fk FOREIGN KEY (Meal_ID) REFERENCES Meal(Meal_ID));

Table created.

SQL>
```

Figure 12Creating TicketInfo table

9. Describing the tables

```
SQL> DESCRIBE play
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PEOPLE_ID                                 NOT NULL NUMBER(10)
 PLAY_ID                                   NOT NULL NUMBER(10)
```

Figure 13Describing Play table

```
SQL> Describe meal
 Name                                       Null?    Type
 ----------------------------------------- -------- ----------------------------
 MEAL_ID                                   NOT NULL NUMBER(5)
 MEAL_TYPE                                          VARCHAR2(10)
 MEAL_PRICE                                         NUMBER(5)
 MEAL_FREE                                          VARCHAR2(5)
 ACCOMODATION_TYPE                                  VARCHAR2(15)
```

Figure 14Describing Meal table

```
SQL> DESCRIBE PlayInfo
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PLAY_ID                                   NOT NULL NUMBER(5)
 PLAY_NAME                                          VARCHAR2(25)
 PLAY_CATEGORY                                      VARCHAR2(10)
 PLAY_LENGTH                                        VARCHAR2(10)
```

Figure 15Describing PlayInfo table

```
SQL> DESCRIBE Ticket
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TICKET_ID                                 NOT NULL NUMBER(10)
 STARTAT                                            VARCHAR2(10)
 TICKET_PRICE                                       NUMBER(5)
```

Figure 16Describing Ticket table

```
SQL> DESCRIBE Booking
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKING_ID                                NOT NULL NUMBER(10)
 BOOKING_TYPE                                       VARCHAR2(10)
 BOOKING_DATE                                       VARCHAR2(11)
```

Figure 17Describing Booking table

```
SQL> DESCRIBE People
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 PEOPLE_ID                                 NOT NULL NUMBER(10)
 PEOPLE_NAME                                        VARCHAR2(25)
 PEOPLE_TYPE                                        VARCHAR2(20)
 PEOPLE_COUNTRY                                     VARCHAR2(15)
 PEOPLE_PROVINCE                                    VARCHAR2(10)
 PEOPLE_CITY                                        VARCHAR2(15)
 PEOPLE_STREET                                      VARCHAR2(15)
 PEOPLE_STREETNO                                    NUMBER(5)
 PEOPLE_PHONENO                                     NUMBER(11)
 PEOPLE_FAXNO                                       NUMBER(11)
 PEOPLE_CELLNO                                      NUMBER(11)
 PEOPLE_EMAIL                                       VARCHAR2(50)
 PEOPLE_AGE                                         NUMBER(5)
 PEOPLE_SEX                                         VARCHAR2(8)
 PEOPLE_DOB                                         VARCHAR2(11)
```

Figure 18Describing People table

```
SQL> DESCRIBE Customer
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 PEOPLE_ID                                          NUMBER(10)
 PEOPLE_CUSTOMERTYPE                                VARCHAR2(10)
```

Figure 19Describing Customer table

```
SQL> DESCRIBE Staff
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 PEOPLE_ID                                          NUMBER(10)
 PEOPLE_STAFFTYPE                                   VARCHAR2(10)
 PEOPLE_STAFFSALARY                                 NUMBER(10)
```

Figure 20Describing Staff table

```
SQL> DESCRIBE TicketInfo
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 TICKET_ID                                 NOT NULL NUMBER(10)
 PLAY_ID                                   NOT NULL NUMBER(10)
 PEOPLE_ID                                 NOT NULL NUMBER(10)
 BOOKING_ID                                         NUMBER(10)
 MEAL_ID                                            NUMBER(10)
```

Figure 21Describing TicketInfo table

10. Inserting data into table

```
SQL> INSERT ALL INTO PlayInfo
  2  VALUES (1,'Black Rose','Romance','2hours')
  3  INTO PlayInfo
  4  VALUES (2,'Athens','War','1.5hours')
  5  INTO PlayInfo
  6  VALUES (3,'The Glass Menagerie','Memory','2.5hours')
  7  INTO PlayInfo
  8  VALUES (4,'A Dolls House','History','2:15')
  9  INTO PlayInfo
 10  VALUES (5,'Henry','Life Story','2:45')
 11  SELECT * FROM DUAL;

5 rows created.

SQL>
```

Figure 22 Inserting into PlayInfo table

```
SQL> INSERT ALL INTO Meal
  2  VALUES (1, 'Breakfast','100','Yes','Gold')
  3  INTO Meal
  4  VALUES (2, 'Breakfast','200','No','Premium')
  5  INTO Meal
  6  VALUES (3, 'Lunch','250','Yes','Gold')
  7  INTO Meal
  8  VALUES (4, 'Lunch','350','No','Premium')
  9  INTO Meal
 10  VALUES (5, 'Dinner','450','No','Premium')
 11  SELECT * FROM DUAL;

5 rows created.

SQL>
```

Figure 23Inserting into Meal table

```
SQL> INSERT ALL INTO BOOKING
  2  VALUES (1, 'Early','2018-11-12')
  3  INTO Booking
  4  VALUES (2, 'Current','2018-12-11')
  5  INTO Booking
  6  VALUES (3, 'Current','2018-11-11')
  7  INTO Booking
  8  VALUES (4, 'Early','2018-10-21')
  9  INTO Booking
 10  VALUES (5, 'Early','2018-09-02')
 11  SELECT * FROM DUAL;

5 rows created.
```

Figure 24Inserting into Booking table

```
SQL> INSERT ALL INTO People
  2  VALUES (1,'Manish Nepali', 'Customer','Nepal','StateNo3','Lalitpur','Jawalakhel',3,015555555,5555555,9876543210,'abc@xyz.com',21,'Male','1998-01-03')
  3  INTO People
  4  VALUES (2,'Ayusha Rai','Customer','Nepal','StateNo2','Nawalaprasi','Sashwatdham',9,014444444,4444444,9876543852,'xyz@abc.com',19,'Female','2000-08-02')
  5  INTO People
  6  VALUES (3,'Prabal Shrestha','Employee','Nepal','StateNo3','Kathmandu','Durbarmarg',4,012222222,2222222,9876575395,'xxx@yyy.com',20,'Male','1999-05-22')
  7  INTO People
  8  VALUES (4,'Neha Kunwar','Employee','Nepal','StateNo1','Bhojpur','Bokhim',5,011111112,1111112,9834917395,'aaa@bbb.com',25,'Male','1994-07-22')
  9  INTO People
 10  VALUES (5,'Chloe Martinez','Customer','Nepal','StateNo2','Bhaktapur','Thimi',1,019999999,9999999,9837164285,'yyy@bbb.com',25,'Female','2001-11-02')
 11  SELECT * FROM DUAL;

5 rows created.

SQL> INSERT ALL INTO People
  2  VALUES (6, 'Ashok Sharma', 'Employee', 'Nepal', 'StateNo1','Biratnagar','Amarmarg', 7, 8529634, 8529634, 789456135, 'qwe78@asd.com', 29, 'Male', '1990-04-21')
  3  INTO People
  4  VALUES (7, 'Ashina Nepal', 'Employee', 'Nepal', 'StateNo2','Birtamod','Mardi', 4, 7946315, 7946315, 7894598437, 'kjh78@asd.com', 24, 'Female', '1995-04-11')
  5  INTO People
  6  VALUES (8, 'Oshin Shrestha', 'Employee', 'Nepal', 'StateNo5','Khaptad','MohanMarg', 2, 1122334, 1122334, 789456248, 'kkkh78@add.com', 19, 'Female', '2000-07-14')
  7  INTO People
  8  VALUES (9, 'Nijan Shakya', 'Customer', 'Nepal', 'StateNo4','Simara','Arunmarg', 5, 7941973, 7941973, 9819752437, 'manu@asd.com', 22, 'Male', '1997-10-21')
  9  INTO People
 10  VALUES (10, 'Samir Thapa', 'Customer', 'Nepal', 'StateNo2','Bhaktapur','Sano Thimi', 4, 6644229, 6644229, 9811111111, 'manci@asd.com', 23, 'Male', '1996-01-21')
 11  SELECT * FROM DUAL;

5 rows created.

SQL>
```

Figure 25Inserting into People table

```
SQL> INSERT ALL INTO Customer
  2  VALUES (1, 'Old')
  3  INTO Customer
  4  VALUES (2, 'New')
  5  INTO Customer
  6  VALUES (5, 'Old')
  7  INTO Customer
  8  VALUES (9, 'Old')
  9  INTO Customer
 10  VALUES (10, 'New')
 11  SELECT * FROM DUAL;

5 rows created.
```

Figure 26Inserting into Customer table

```
SQL> INSERT ALL INTO Staff
  2  VALUES (3,'Usher',6000)
  3  INTO Staff
  4  VALUES (4,'Actor',10000)
  5  INTO Staff
  6  VALUES (6,'Actor',9000)
  7  INTO Staff
  8  VALUES (7,'Usher',7000)
  9  INTO Staff
 10  VALUES (8,'Actor',10500)
 11  SELECT * FROM DUAL;

5 rows created.

SQL>
```

Figure 27Inserting into Staff table

```
SQL> INSERT ALL INTO Ticket
  2  VALUES (1, '3:15pm',350)
  3  INTO Ticket
  4  VALUES (2, '12:15pm',450)
  5  INTO Ticket
  6  VALUES (3, '10:00am',350)
  7  INTO Ticket
  8  VALUES (4, '3:15pm',450)
  9  INTO Ticket
 10  VALUES (5, '10:00am',350)
 11  SELECT * FROM DUAL;

5 rows created.
```

Figure 28Inserting into Ticket table

```
SQL> INSERT ALL INTO TicketInfo
  2  VALUES (1,3,1,2,2)
  3  INTO TicketInfo
  4  VALUES (2,1,2,5,1)
  5  INTO TicketInfo
  6  VALUES (3,5,3,1,1)
  7  INTO TicketInfo
  8  VALUES (4,3,4,2,2)
  9  INTO TicketInfo
 10  VALUES (5,2,5,4,4)
 11  INTO TicketInfo
 12  VALUES (1,5,5,1,3)
 13  INTO TicketInfo
 14  VALUES (2,2,6,3,2)
 15  INTO TicketInfo
 16  VALUES (3,1,7,2,5)
 17  INTO TicketInfo
 18  VALUES (4,4,8,4,3)
 19  INTO TicketInfo
 20  VALUES (5,4,9,3,4)
 21  SELECT * FROM DUAL;

10 rows created.
```

Figure 29Inserting into TicketInfo table

```
SQL> INSERT ALL INTO Play
  2  VALUES (2,1)
  3  INTO Play
  4  VALUES (1,3)
  5  INTO Play
  6  VALUES (5,5)
  7  INTO Play
  8  VALUES (3,5)
  9  INTO Play
 10  VALUES (4,3)
 11  INTO Play
 12  VALUES (5,2)
 13  INTO Play
 14  VALUES (6,2)
 15  INTO Play
 16  VALUES (7,1)
 17  INTO Play
 18  VALUES (8,4)
 19  INTO Play
 20  VALUES (9,4)
 21  INTO Play
 22  VALUES (10,3)
 23  SELECT * FROM DUAL;

11 rows created.
```

Figure 30Inserting into Play table

11. Displaying Table

```
SQL> SELECt * FROM People;

PEOPLE_ID PEOPLE_NAME            PEOPLE_TYPE        PEOPLE_COUNTRY  PEOPLE_PRO PEOPLE_CITY    PEOPLE_STREET   PEOPLE_STREETNO PEOPLE_PHONENO PEOPLE_FAXNO PEOPLE_CELLNO PEOPLE_EMAIL
 PEOPLE_AGE PEOPLE_S PEOPLE_DOB
---------- -------------------- ---------- --------------- ---------- ------------ --------------- --------------- -------------- ------------ -------------- --------------------
         1 Manish Nepali        Customer   Nepal           StateNo3   Lalitpur     Jawalakhel                    3       15555555      5555555   9876543210 abc@xyz.com
        21 Male     1998-01-03
         2 Ayusha Rai           Customer   Nepal           StateNo2   Nawalaprasi  Sashwatdham                   9       14444444      4444444   9876543852 xyz@abc.com
        19 Female   2000-08-02
         3 Prabal Shrestha      Employee   Nepal           StateNo3   Kathmandu    Durbarmarg                    4       12222222      2222222   9876575395 xxx@yyy.com
        20 Male     1999-05-22
         4 Neha Kunwar          Employee   Nepal           StateNo1   Bhojpur      Bokhim                        5       11111112      1111112   9834917395 aaa@bbb.com
        25 Male     1994-07-22
         5 Chloe Martinez       Customer   Nepal           StateNo2   Bhaktapur    Thimi                         1       19999999      9999999   9837164285 yyy@bbb.com
        25 Female   2001-11-02
         6 Ashok Sharma         Employee   Nepal           StateNo1   Biratnagar   Amarmarg                      7        8529634      8529634    789456135 qwe78@asd.com
        29 Male     1990-04-21
         7 Ashina Nepal         Employee   Nepal           StateNo2   Birtamod     Mardi                         4        7946315      7946315   7894598437 kjh78@asd.com
        24 Female   1995-04-11
         8 Oshin Shrestha       Employee   Nepal           StateNo5   Khaptad      MohanMarg                     2        1122334      1122334    789456248 kkkh78@add.com
        19 Female   2000-07-14
         9 Nijan Shakya         Customer   Nepal           StateNo4   Simara       Arunmarg                      5        7941973      7941973   9819752417 mana@asd.com
        22 Male     1997-10-21
        10 Samir Thapa          Customer   Nepal           StateNo2   Bhaktapur    Sano Thimi                    4        6644229      6644229   9811111111 manci@asd.com
        23 Male     1996-01-21

10 rows selected.
```

Figure 31 Displaying People table

```
SQL> SELECT * FROM Customer;

 PEOPLE_ID PEOPLE_CUS
---------- ----------
         1 Old
         2 New
         5 Old
         9 Old
        10 New
```

Figure 32Displaying Customer table

```
 PEOPLE_ID PEOPLE_STA PEOPLE_STAFFSALARY
---------- ---------- ------------------
         3 Usher                    6000
         4 Actor                   10000
         6 Actor                    9000
         7 Usher                    7000
         8 Actor                   10500
```

Figure 33Displaying staff table

```
BOOKING_ID BOOKING_TY BOOKING_DAT
---------- ---------- -----------
         1 Early      2018-11-12
         2 Current    2018-12-11
         3 Current    2018-11-11
         4 Early      2018-10-21
         5 Early      2018-09-02

SQL>
```

Figure 34Displaying booking table

```
SQL> SELECT * FROM Meal;

   MEAL_ID MEAL_TYPE  MEAL_PRICE MEAL_ ACCOMODATION_TY
---------- ---------- ---------- ----- --------------
         1 Breakfast         100 Yes   Gold
         2 Breakfast         200 No    Premium
         3 Lunch             250 Yes   Gold
         4 Lunch             350 No    Premium
         5 Dinner            450 No    Premium

SQL>
```

Figure 35Displaying Meal table

```
SQL> SELECT * FROM PlayInfo;

   PLAY_ID PLAY_NAME                 PLAY_CATEG PLAY_LENGT
---------- ------------------------- ---------- ----------
         1 Black Rose                Romance    2hours
         2 Athens                    War        1.5hours
         3 The Glass Menagerie       Memory     2.5hours
         4 A Dolls House             History    2:15
         5 Henry                     Life Story 2:45

SQL>
```

Figure 36Displaying PlayInfo table

```
SQL> SELECT * FROM Play;

 PEOPLE_ID    PLAY_ID
---------- ----------
         1          3
         2          1
         3          5
         4          3
         5          2
         5          5
         6          2
         7          1
         8          4
         9          4
        10          3

11 rows selected.
```

Figure 37Displaying Play table

```
SQL> SELECT * FROM Ticket;

 TICKET_ID STARTAT    TICKET_PRICE
---------- ---------- ------------
         1 3:15pm              350
         2 12:15pm             450
         3 10:00am             350
         4 3:15pm              450
         5 10:00am             350

SQL>
```

Figure 38Displaying Ticket table

```
SQL> SELECT * FROM TicketInfo;

 TICKET_ID    PLAY_ID  PEOPLE_ID BOOKING_ID    MEAL_ID
---------- ---------- ---------- ---------- ----------
         1          3          1          2          2
         2          1          2          5          1
         3          5          3          1          1
         4          3          4          2          2
         5          2          5          4          4
         1          5          5          1          3
         2          2          6          3          2
         3          1          7          2          5
         4          4          8          4          3
         5          4          9          3          4

10 rows selected.

SQL>
```

Figure 39Displaying TicketInfo table

12. Queries

12.1 Information Queries

12.1.1 List all customers, old and current

```
SQL> SELECT People_Name, People_CustomerType FROM People, Customer
  2    WHERE People.People_ID = Customer.People_ID;

PEOPLE_NAME               PEOPLE_CUS
------------------------- ----------
Manish Nepali             Old
Ayusha Rai                New
Chloe Martinez            Old
Nijan Shakya              Old
Samir Thapa               New

SQL>
```

Figure 40 Information Query -1

12.1.2 List all customers with all their addresses.

```
SQL> SELECT People_ID, People_Name, People_Country, People_Province, People_City, People_Street, People_StreetNo
  2  FROM People
  3  WHERE People_Type='Customer';

 PEOPLE_ID PEOPLE_NAME               PEOPLE_COUNTRY  PEOPLE_PRO PEOPLE_CITY     PEOPLE_STREET   PEOPLE_STREETNO
---------- ------------------------- --------------- ---------- --------------- --------------- ---------------
         1 Manish Nepali             Nepal           StateNo3   Lalitpur        Jawalakhel                    3
         2 Ayusha Rai                Nepal           StateNo2   Nawalaprasi     Sashwatdham                   9
         5 Chloe Martinez            Nepal           StateNo2   Bhaktapur       Thimi                         1
         9 Nijan Shakya              Nepal           StateNo4   Simara          Arunmarg                      5
        10 Samir Thapa               Nepal           StateNo2   Bhaktapur       Sano Thimi                    4

SQL>
```

Figure 41Information Query - 2

12.1.3 For a given usher, find all the shows he/she ushered or will usher and the amount he/she got/will get for ushering the show.

```
SQL> SELECT People_Name, People_Type, Play_Name, People_StaffSalary
  2  FROM People, PlayInfo, Staff, Play
  3  WHERE People.People_ID=Staff.People_ID
  4  AND People.People_ID = Play.People_ID
  5  AND PlayInfo.Play_ID = Play.Play_ID
  6  AND Staff.People_StaffType = 'Usher';

PEOPLE_NAME              PEOPLE_TYPE          PLAY_NAME                 PEOPLE_STAFFSALARY
------------------------ -------------------- ------------------------- ------------------
Prabal Shrestha          Employee             Henry                                   6000
Ashina Nepal             Employee             Black Rose                              7000

SQL>
```

Figure 42Information Query - 3

12.1.4 List all customers that are also ushers.

```
SQL> SELECT People_Name, People_Type, People_StaffType
  2  FROM People, Staff
  3  WHERE People.People_ID = Staff.People_ID
  4  AND Staff.People_StaffType = 'Usher';

PEOPLE_NAME              PEOPLE_TYPE          PEOPLE_STA
------------------------ -------------------- ----------
Prabal Shrestha          Employee             Usher
Ashina Nepal             Employee             Usher

SQL>
```

Figure 43Information Query - 4

12.2Transaction Queries

12.2.1 List all ushers that attended a show that had a lunch in a given place.

```
SQL> SELECT People_Name, People_StaffType, Play_ID, Meal_Type, Accomodation_Type
  2  FROM People, Staff, TicketInfo, Meal
  3  WHERE People.People_ID = Staff.People_ID
  4  AND People.People_ID = TicketInfo.People_ID
  5  AND Meal.Meal_ID = TicketInfo.Meal_ID
  6  AND Staff.People_StaffType = 'Usher';

PEOPLE_NAME              PEOPLE_STA    PLAY_ID MEAL_TYPE  ACCOMODATION_TY
------------------------ ---------- ---------- ---------- --------------
Prabal Shrestha          Usher               5 Breakfast  Gold
Ashina Nepal             Usher               1 Dinner     Premium

SQL>
```

Figure 44 Transaction Query - 1

12.2.2 List the shows that will have breakfast at a given place on a given date

```
SQL> SELECT Play_Name, Meal_Type, Meal_Price, Booking_Date
  2  FROM PlayInfo, Meal, TicketInfo, Booking
  3  WHERE PlayInfo.Play_ID = TicketInfo.Play_ID
  4  AND Booking.Booking_ID = TicketInfo.Booking_ID
  5  AND Meal.Meal_Type = 'Breakfast'
  6  AND Booking_Date='2018-11-11';

PLAY_NAME                MEAL_TYPE  MEAL_PRICE BOOKING_DAT
------------------------ ---------- ---------- -----------
Athens                   Breakfast         100 2018-11-11
Athens                   Breakfast         200 2018-11-11
A Dolls House            Breakfast         100 2018-11-11
A Dolls House            Breakfast         200 2018-11-11

SQL>
```

Figure 45Transaction Query - 2

12.2.3 List all employees that have worked as an usher or will work as an usher for a show or who have attended or will attend a show.

```
SQL> SELECT People_Name, People_Type, People_StaffType
  2  FROM People, Staff
  3  WHERE People.People_ID = Staff.People_ID;

PEOPLE_NAME                PEOPLE_TYPE          PEOPLE_STA
-------------------------- -------------------- ----------
Prabal Shrestha            Employee             Usher
Neha Kunwar                Employee             Actor
Ashok Sharma               Employee             Actor
Ashina Nepal               Employee             Usher
Oshin Shrestha             Employee             Actor

SQL>
```

Figure 46Transaction Query - 3

12.2.4 List all customers booked for a show starting later or on a given date.

```
SQL> SELECT People_Name, People_Type, Booking_Date, Booking_Type
  2  FROM People, Booking, TicketInfo
  3  WHERE People.People_ID = TicketInfo.People_ID
  4  AND Booking.Booking_ID = TicketInfo.Booking_ID
  5  AND Booking.Booking_Type = 'Early'
  6  AND Booking.Booking_Date = '2018-11-12';

PEOPLE_NAME                PEOPLE_TYPE          BOOKING_DAT BOOKING_TY
-------------------------- -------------------- ----------- ----------
Prabal Shrestha            Employee             2018-11-12  Early
Chloe Martinez             Customer             2018-11-12  Early

SQL>
```

Figure 47Transaction Query - 4

13. Critical Evaluation

a. Further discussion on learning experience

The coursework required much research and self-learning. We had to research about the theatre company. We has to search about its history, its business activities and operations and also its business rules. With the reference to these research, we had to create entities for further processing. The scenario decided the entities that will be included in the further process of normalisation. The process required consulting with the tutor many times. After the creation of scenario, building relationships between the entities was also difficult part but our tutors helped us sincerely which eased the process. Then after, normalisation was another challenging part of the assessment. Since there was a big number of attributes to be included, it became difficult to arrange them. Even after the process of normalisation started, many problems arose such as selecting the repeating groups, finding the partial or transitive dependencies. After the normalisation process, creating table in Oracle was not tough but populating the table was a tough task.

Lots of problem arose while performing the assigned tasks. The first difficulty was when we had to research about the real world scenario of theme. It was difficult finding out real world data related to theatres. The process of creating scenario was confusing as there were many requirements to be fulfilled. While normalising the data, we met with problems during separation of super type and sub type entity. Populating the tables was also tough.

Most of the problems were tackled by consulting with the tutor. The tutor helped us to clarify the requirements of the assessment which made it easier for us to fulfil the assigned task. Proper research through the internet helped us gain better knowledge about super type and sub type entities which was one of the most important part of the project,

b. Critical assessment of coursework

Even if we had a hard time fulfilling the coursework, the assigned tasks were really helpful. The coursework covered most of the topics that were studied during the lecture. This helped us to recall the materials studied. The coursework also gave us a basic idea about real life scenario of implementation of database. We also realised that database can be really helpful if it can associated with the other modules that we are studying in this semester. Use of database in java programming can be useful to build the system that is much more efficient that the one without use of database. Database also plays a vital role in software engineering.