

**Module Title : DATABASES**

**Assignment Title : STELLA ARTS EVENT BOOKING SYSTEM**

**Examination Cycle : DECEMBER 2018**

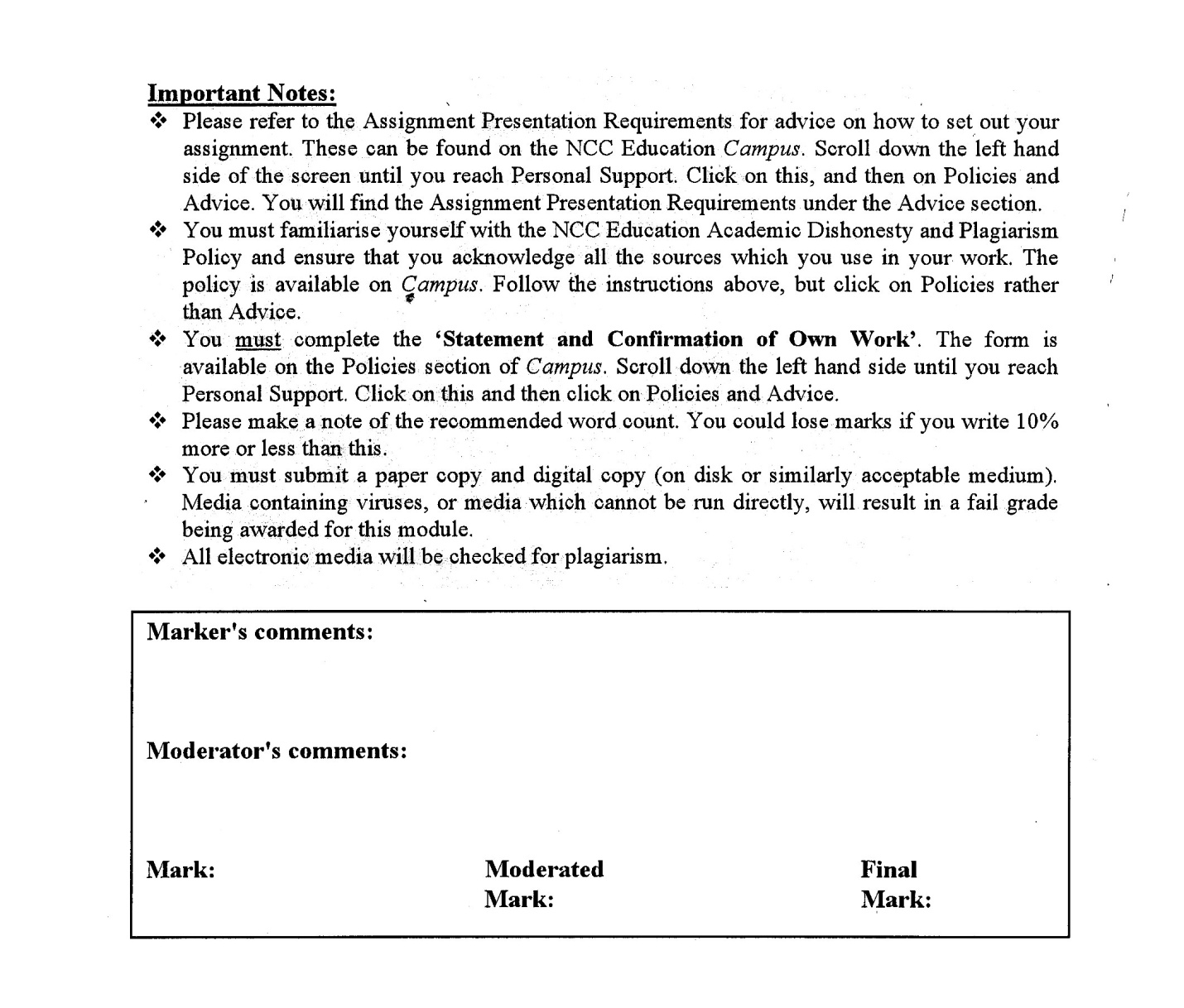
**Candidate Name : HTET YAMIN EI**

**Candidate No : 00177312**

**Centre Name : KMD COMPUTER CENTRE (YANGON)**

**Submission Date : 30 - SEPTEMBER - 2018**

**Important Notes:**

* Please refer to the Assignment Presentation Requirements for advice on how to set out your assignment. These can be found on the NCC Education *Campus*. Scroll down the left hand side of the screen until you reach Personal Support. Click on this, and then on Policies and Advice. You will find the Assignment Presentation Requirements under the Advice section.
* You must familiarize yourself with the NCC Education Academic Dishonesty and Plagiarism Policy and ensure that you acknowledge all the sources which you use in your work. The policy is available on *Campus*. Follow the instructions above, but click on Policies rather than Advice.
* You must complete the **‘Statement and Confirmation of Own Work’**. The form is available on the Policies section of *Campus*. Scroll down the left hand side until you reach Personal Support. Click on this and then click on Policies and Advice.
* Please make a note of the recommended word count. You could lose marks if you write 10%more or less than this.
* You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media which cannot be run directly, will result in a fail grade being awarded for this module.
* All electronic media will be checked for plagiarism.

****

**Statement and Confirmation of Own Work**

**Programmer/Qualification name: L4DC**

***All NCC Education assessed assignments submitted by students must have this statement as the cover page or it will not be accepted for marking. Please ensure that this statement is either firmly attached to the cover of the assignment or electronically inserted into the front of the assignment.***

**Student declaration**

**I have read and understood NCC Education’s Policy on Academic Dishonesty and Plagiarism.**

**I can confirm the following details.**

**Student ID/Registration number : 00177312**

**Name : HTET YAMIN EI**

**Centre Name : KMD COMPUTER CENTRE (YANGON)**

**Module Name : DATABASES**

**Module Leader : TAH MLAH MOO**

**Number of words : (1277) words**

**I confirm that this is my own work and that I have not plagiarized any part of it. I have also noted the assessment criteria and pass mark for assignments.**

**Due Date : 30 - SEPTEMBER - 2018**

**Student Signature : EI**

**Submission Date : 30 - SEPTEMBER - 2018**

**Table of Contents**

[**Introduction** 4](#_Toc525987594)

[**a)** **Entity Relationship Model** 6](#_Toc525987595)

[**Fully Normalized Model** 7](#_Toc525987596)

[***Figure 1 and 2: Customer Details and Loyalty Scheme Level*** 7](#_Toc525987597)

[***Figure 3: Booking*** 7](#_Toc525987598)

[***Figure 4: Tickets*** 8](#_Toc525987599)

[**b)** **How Normalization of the Data in the Scenario Contributed to the Finished ER Diagram** 10](#_Toc525987600)

[**c)** **Data Dictionary** 11](#_Toc525987601)

[***1.*** ***Loyalty*** 11](#_Toc525987602)

[***2.*** ***CustomersDetail*** 11](#_Toc525987603)

[***3.*** ***Venue*** 12](#_Toc525987604)

[***4.*** ***Event*** 12](#_Toc525987605)

[***5.*** ***Booking*** 13](#_Toc525987606)

[***6.*** ***Ticket*** 13](#_Toc525987607)

[**1.** **Creating Tables** 15](#_Toc525987608)

[***Loyalty*** 15](#_Toc525987609)

[***CustomersDetail*** 16](#_Toc525987610)

[***Venue*** 17](#_Toc525987611)

[***Event*** 18](#_Toc525987612)

[***Booking*** 19](#_Toc525987613)

[***Ticket*** 19](#_Toc525987614)

[**2.** **Entering Customers and Loyalty Level Data** 21](#_Toc525987615)

[***Loyalty*** 21](#_Toc525987616)

[***CustomersDetail*** 22](#_Toc525987617)

[**3.** **Entering Venue Data** 23](#_Toc525987618)

[***Venue*** 23](#_Toc525987619)

[**4.** **Entering Booking and Event Data** 24](#_Toc525987620)

[***Event*** 24](#_Toc525987621)

[***Booking*** 25](#_Toc525987622)

[**5.** **Entering Ticket Data** 26](#_Toc525987623)

[***Ticket*** 26](#_Toc525987624)

[**6.** **Selecting First Name and Surname of Customers in Order of Surname** 27](#_Toc525987625)

[**7.** **Selecting All Bookings by Event Date in Ascending Order** 27](#_Toc525987626)

[**8.** **Counting All Bookings Where Payment Method is Credit Card** 28](#_Toc525987627)

[**9.** **Selecting All Event Bookings Where Quantity of Tickets is Greater Than 1 Ordered by Description** 28](#_Toc525987628)

[**10.** **Selecting First Names, Surnames and Loyalty Level Who Pay by Credit Card Grouped by Loyalty Level** 29](#_Toc525987629)

[**11.** **Changing the Contact Name for ‘The Coliseum’ to ‘Mark Hibbart’** 29](#_Toc525987630)

[**12.** **Adding A New Venue** 30](#_Toc525987631)

[**Required Analysis and Assumptions** 32](#_Toc525987632)

[**Design Decisions and Justifications** 32](#_Toc525987633)

[**How the Requirements have been met** 33](#_Toc525987634)

[References 33](#_Toc525987635)

[**Candidate Checklist** 34](#_Toc525987636)

**Introduction**

Stella Arts, an arts company require a database system to keep tracks of the bookings made and to reduce pressures on the reception staffs. Therefore, a database system is created in the following steps. This includes 3 tasks such as design, data entry and data manipulation and assessment. They are as below.

**TASK – 1**

**Design**

1. **Entity Relationship Model**

Ticket

Venue

Booking

Loyalty

CustomersDetail

Event

\*

1

\*

1

1

\*

1

\*

\*

1

**Fully Normalized Model**

Normalization is a process of breaking down the data into separate relations. It relies on the conception of functional dependency. Normalization is made to overcome differences and duplication of the data. The 1st normal form is to remove repeating group information, the 2nd is to remove partial key dependencies and the 3rd is to remove non-key dependencies. Underlined words represent Primary Keys. The symbol \* represents Foreign Keys. The bolded words stand for entities.

***Figure 1 and 2: Customer Details and Loyalty Scheme Level***

|  |  |  |  |
| --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **Relation** |
| CustomerRef  FirstName  Surname  Address  ContactNumber  Email  LoyaltyLevel  No\_of\_Bookings\_Required  Discount | LoyaltyLevel  No\_of\_Bookings\_Required  Discount  CustomerRef  LoyaltyLevel\*  FirstName  Surname  Address  ContactNumber  Email | LoyaltyLevel  No\_of\_Bookings\_Required  Discount  CustomerRef\*  LoyaltyLevel\*  CustomerRef  FirstName  Surname  Address  ContactNumber  Email | **Loyalty**  **CustomersLoyalty**  **CustomersDetail** |

**ER Diagram**

***Figure 3: Booking***

1

\*

1

\*

CustomersDetail

CustomersLoyalty

Loyalty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** | **Relation** |
| BookingRef  BookingDate  CustomerRef  CustomerName  PaymentMethod  EventID  Description  FullPrice  VenueID  VenueName  EventDate  Time  TicketQTY  Discount | BookingRef  BookingDate  CustomerRef  CustomerName  PaymentMethod  EventID  Description  FullPrice  VenueID  VenueName  EventDate  Time  TicketQTY  Discount | BookingRef  BookingDate  CustomerRef  CustomerName  PaymentMethod  EventID  Description  FullPrice  VenueID  VenueName  EventDate  Time  TicketQTY  Discount | BookingRef  CustomerRef\*  EventID\*  VenueID\*  BookingDate  PaymentMethod  TicketQTY  CustomerRef  CustomerName  Discount  EventID  Description  FullPrice  EventDate  Time  VenueID  VenueName | **Booking**  **CustomersDetail**  **Event**  **Venue** |

**ER Diagram**

***Figure 4: Tickets***

\*

\*

1

1

\*

1

Venue

CustomersDetail

Event

Booking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** | **Relation** |
| TicketNo  SeatNo  Row  CustomerRef  CustomerName  BookingRef  BookingDate  Post\_Collect | TicketNo  SeatNo  Row  CustomerRef  CustomerName  BookingRef  BookingDate  Post\_Collect | TicketNo  SeatNo  Row  CustomerRef  CustomerName  BookingRef  BookingDate  Post\_Collect | TicketNo  BookingRef\*  CustomerRef\*  SeatNo  Row  Post\_Collect  BookingRef  BookingDate  CustomerRef  CustomerName | **Ticket**  **Booking**  **CustomersDetail** |

**ER Diagram**

1

\*

Booking

CustomersDetail

\*

Ticket

1

1. **How Normalization of the Data in the Scenario Contributed to the Finished ER Diagram**

The given first figure of the data in the scenario does not require normalization since it is already a simple figure. But the second one needs to be normalized because it is not as simple as the first figure. In the normalization of the second figure, the data are combined with those of the first figure. Then only it will be convenient for the normalization. In the 1st normal form, loyalty level is the repeating key. So, it is removed as a new entity including the attributes concerned with it and loyalty level becomes the foreign key in another entity. In the 2nd normal form, the customer details can also be removed as another table. As a result, there are 3 entities. But there are a few justifications for this result in the task 3.

The given third figure is even more complicated than the first and the second one. Thus, normalization is required. In the 1st normal form, the whole key “BookingRef” is regarded as a repeating key because of the given scenario of the assignment. The attribute “BookingRef” itself is enough to be a primary key. If one of the other attributes are considered as the single key, the relation will be become “many to many” relation. To reconsider in all probable ways, it is not possible to have such relation. Because there may be a lot of bookings made by one customer but a booking is only for one customer. In the same way, a booking is for one event only and an event can have hundreds of bookings. Also, a venue can have a lot of bookings and a booking is for one venue only. Therefore, having intersection entities in the normalization is not quite correct. That is why the normalization of the third figure does not have single key. So, there is no single key and thus, no partial key. Therefore, in the 3rd normal form, the entity “Booking” itself becomes the intersection entities having foreign keys. The result is the 4 tables in relation with one another.

The fourth figure is also a little bit complex. The whole key “TicketNo” is considered as a repeating key because of the given scenario. The “TicketNo” itself is enough to be a primary key since the data are unique. Just like the normalization of the third figure, only the data are separated in the 3rd normal form resulting in 3 entities. There are also a few justifications for the relation between these entities.

The fifth one is simple and do not have duplication of data. So, normalization is not necessary. Because of normalization, the attributes in the entities become more accurate. The data dictionary of each of the entities is as below.

1. **Data Dictionary**
2. ***Loyalty***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name: Loyalty  Primary Key: LoyaltyLevel  Foreign Key: - | | | |
| **Name** | **Type** | **Length** | **Description** |
| LoyaltyLevel | Varchar | 10 | Loyalty levels that customers can have |
| No\_of\_Bookings\_Required | Int | - | No of bookings needed for the specific loyalty level |
| Discount | Varchar | 5 | Discounts which customers can have according to their loyalty levels |

1. ***CustomersDetail***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name: CustomersDetail  Primary Key: CustomerRef  Foreign Key: - | | | |
| **Name** | **Type** | **Length** | **Description** |
| CustomerRef | Int | - | Identity of customers |
| FirstName | Varchar | 30 | First name of customers |
| Surname | Varchar | 30 | Surname of customers |
| Address | Varchar | 80 | Customers’ address |
| ContactNumber | Varchar | 20 | Customers’ contact number |
| Email | Varchar | 30 | Email of customers |

1. ***Venue***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name: Venue  Primary Key: VenueID | | | |
| **Name** | **Type** | **Length** | **Description** |
| VenueID | Int | - | Identity of Venues |
| Name | Varchar | 30 | Name of Venues |
| Address | Varchar | 80 | Name of the place where venues are situated |
| ContactName | Varchar | 30 | Contact name of the chiefs of venues |
| TelNumber | Varchar | 20 | Telephone number of the chiefs |
| Capacity | Int | - | Number of people that venues can hold |

1. ***Event***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name: Event  Primary Key: EventID  Foreign Key: - | | | |
| **Name** | **Type** | **Length** | **Description** |
| EventID | Int | - | Identity of Event |
| Description | Varchar | 50 | Description of Event |
| FullPrice | Varchar | 10 | Full price to pay to attend the event |
| EventDate | Varchar | 10 | Date of the event |
| Time | Varchar | 5 | Time of the event |
| VenueDetails | Varchar | 50 | Name of the venue where the event will take place |

1. ***Booking***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name: Booking  Primary Key: BookingRef  Foreign Key: EventID, VenueID | | | |
| **Name** | **Type** | **Length** | **Description** |
| BookingRef | Int | - | The unique booking number of customers |
| BookingDate | Varchar | 10 | Date of the Booking made |
| TicketQTY | Int | - | Quantity of tickets bought |
| EventID | Int | - | Identity of the event |
| VenueID | Int | - | Identity of the venue |

1. ***Ticket***

|  |  |  |  |
| --- | --- | --- | --- |
| Entity Name: Ticket  Primary Key: TicketNo  Foreign Key: BookingRef | | | |
| **Name** | **Type** | **Length** | **Description** |
| TicketNo  BookingRef\*  CustomerRef\*  SeatNo  Row  Post\_Collect | Int | - | The unique number of the ticket booked |
| BookingRef | Int | - | The unique booking number of customers |
| SeatNo | Int | - | The Seat number of the ticket |
| Row | Varchar | 10 | The row number of the ticket |
| CustomerName | Varchar | 50 | Name of the customer |
| Post\_Collect | Varchar | 10 | Type of ticket whether it is to be posted or collected |

**TASK – 2**

**Data Entry and**

**Data Manipulation**

1. **Creating Tables**

***Loyalty***

create table Loyalty

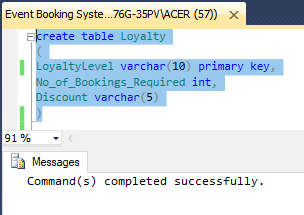
(

LoyaltyLevel varchar(10) primary key,

No\_of\_Bookings\_Required int,

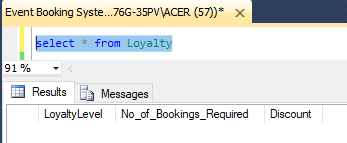
Discount varchar(5)

)



**Result**

select \* from Loyalty



***CustomersDetail***

create table CustomersDetail

(

CustomerRef int primary key,

FirstName varchar(30),

Surname varchar(30),

Address varchar(80),

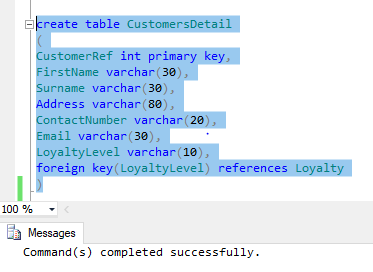
ContactNumber varchar(20),

Email varchar(30),

LoyaltyLevel varchar(10),

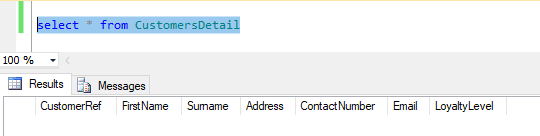
foreign key(LoyaltyLevel) references Loyalty

)



**Result**

select \* from CustomersDetail



***Venue***

create table Venue

(

VenueID int primary key,

Name varchar(30),

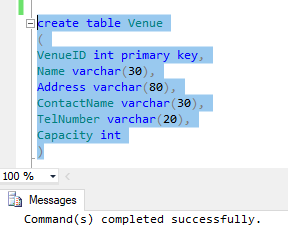
Address varchar(80),

ContactName varchar(30),

TelNumber varchar(20),

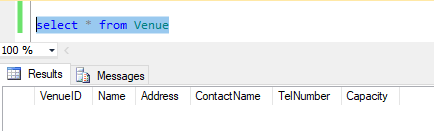
Capacity int

)



**Result**

select \* from Venue



***Event***

create table Event

(

EventID int primary key,

Description varchar(50),

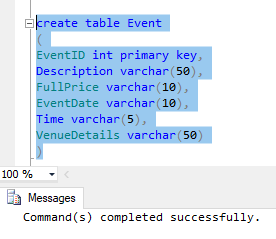
FullPrice varchar(10),

EventDate varchar(10),

Time varchar(5),

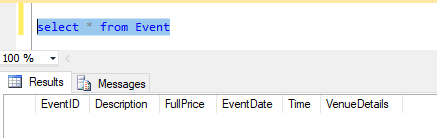
VenueDetails varchar(50)

)



**Result**

select \* from Event



***Booking***

create table Booking

(

BookingRef int primary key,

BookingDate varchar(10),

CustomerRef int,

foreign key(CustomerRef) references CustomersDetail,

PaymentMethod varchar(20),

TicketQTY int,

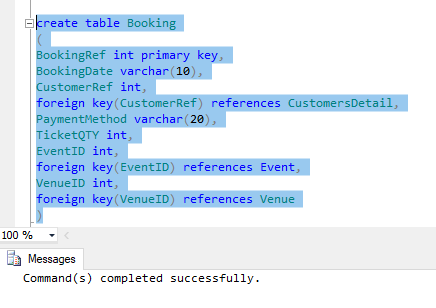
EventID int,

foreign key(EventID) references Event,

VenueID int,

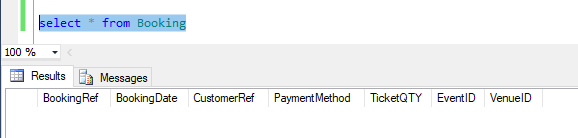
foreign key(VenueID) references Venue

)



**Result**

select \* from Booking



***Ticket***

create table Ticket

(

TicketNo int primary key,

SeatNo int,

Row varchar(10),

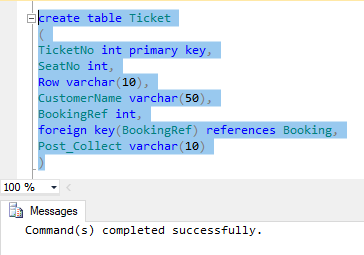
CustomerName varchar(50),

BookingRef int,

foreign key(BookingRef) references Booking,

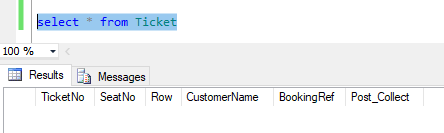
Post\_Collect varchar(10)

)



**Result**

select \* from Ticket



1. **Entering Customers and Loyalty Level Data**

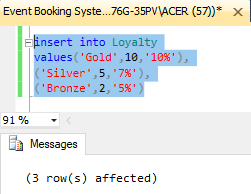
***Loyalty***

insert into Loyalty

values('Gold',10,'10%'),

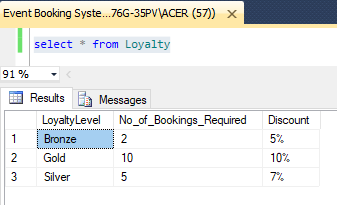
('Silver',5,'7%'),

('Bronze',2,'5%')



**Result**

select \* from Loyalty



***CustomersDetail***

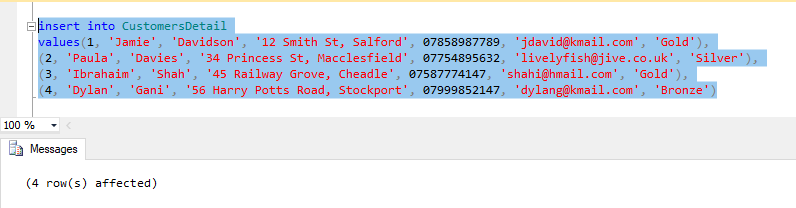
insert into CustomersDetail

values(1, 'Jamie', 'Davidson', '12 Smith St, Salford', 07858987789, 'jdavid@kmail.com', 'Gold'),

(2, 'Paula', 'Davies', '34 Princess St, Macclesfield', 07754895632, 'livelyfish@jive.co.uk', 'Silver'),

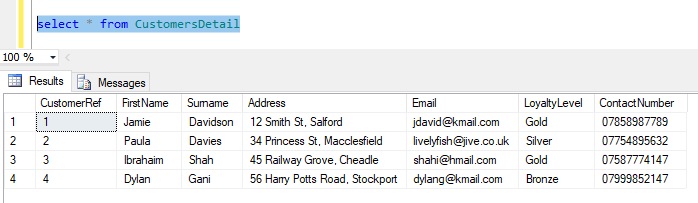
(3, 'Ibrahaim', 'Shah', '45 Railway Grove, Cheadle', '07587774147, 'shahi@hmail.com', 'Gold'),

(4, 'Dylan', 'Gani', '56 Harry Potts Road, Stockport', 07999852147, 'dylang@kmail.com', 'Bronze')



**Result**

select \* from CustomersDetail



1. **Entering Venue Data**

***Venue***

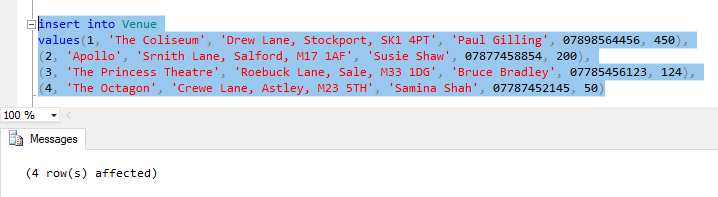
insert into Venue

values(1, 'The Coliseum', 'Drew Lane, Stockport, SK1 4PT', 'Paul Gilling', 07898564456, 450),

(2, 'Apollo', 'Srnith Lane, Salford, M17 1AF', 'Susie Shaw', 07877458854, 200),

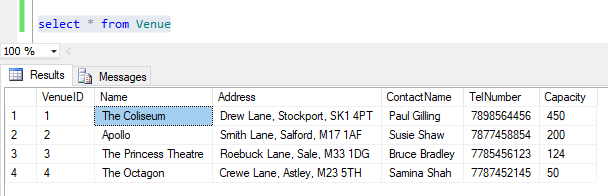
(3, 'The Princess Theatre', 'Roebuck Lane, Sale, M33 1DG', 'Bruce Bradley', 07785456123, 124),

(4, 'The Octagon', 'Crewe Lane, Astley, M23 5TH', 'Samina Shah', 07787452145, 50)



**Result**

select \* from Venue



1. **Entering Booking and Event Data**

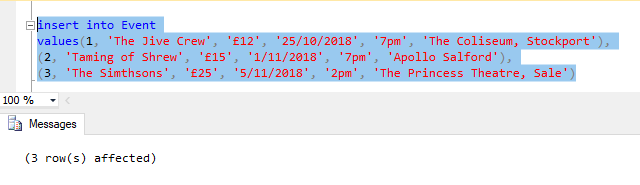
***Event***

insert into Event

values(1, 'The Jive Crew', '£12', '25/10/2018', '7pm', 'The Coliseum, Stockport'),

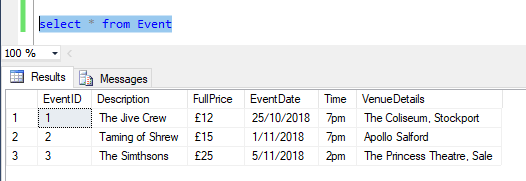
(2, 'Taming of Shrew', '£15', '1/11/2018', '7pm', 'Apollo Salford'),

(3, 'The Simthsons', '£25', '5/11/2018', '2pm', 'The Princess Theatre, Sale')



**Result**

select \* from Event



***Booking***

insert into Booking

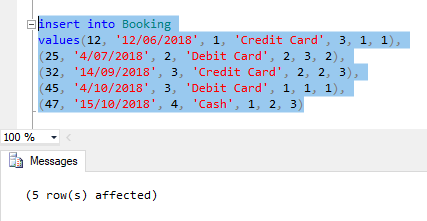
values(12, '12/06/2018', 1, 'Credit Card', 3, 1, 1),

(25, '4/07/2018', 2, 'Debit Card', 2, 3, 2),

(32, '14/09/2018', 3, 'Credit Card', 2, 2, 3),

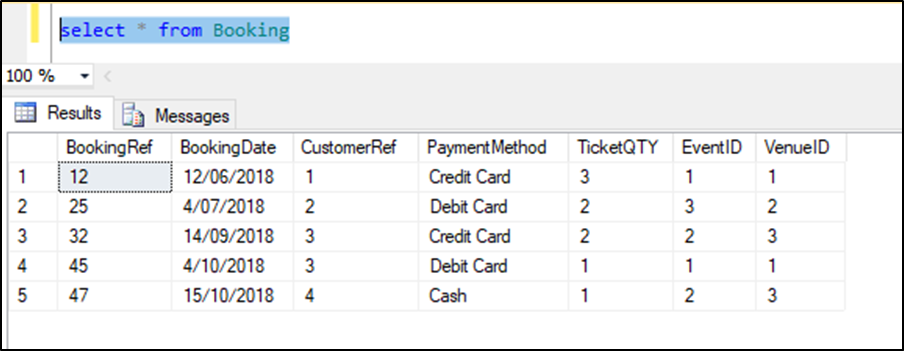
(45, '4/10/2018', 3, 'Debit Card', 1, 1, 1),

(47, '15/10/2018', 4, 'Cash', 1, 2, 3)



**Result**

select \* from Booking



1. **Entering Ticket Data**

***Ticket***

insert into Ticket

values(1, 34, 'B', 'Jamie Davidson', 12, 'Post'),

(2, 35, 'B', 'Jamie Davidson', 12, 'Post'),

(3, 36, 'B', 'Jamie Davidson', 12, 'Post'),

(4, 2, 'A', 'Paula Davies', 25, 'Collect'),

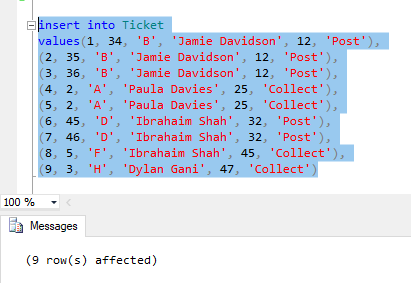
(5, 2, 'A', 'Paula Davies', 25, 'Collect'),

(6, 45, 'D', 'Ibrahaim Shah', 32, 'Post'),

(7, 46, 'D', 'Ibrahaim Shah', 32, 'Post'),

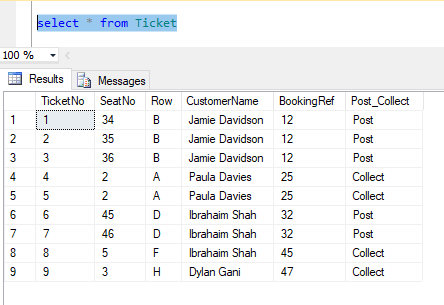
(8, 5, 'F', 'Ibrahaim Shah', 45, 'Collect'),

(9, 3, 'H', 'Dylan Gani', 47, 'Collect')



**Result**

select \* from Ticket

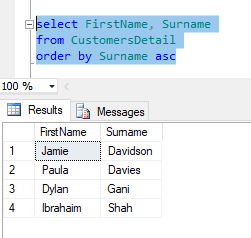


1. **Selecting First Name and Surname of Customers in Order of Surname**

select FirstName, Surname

from CustomersDetail

order by Surname asc



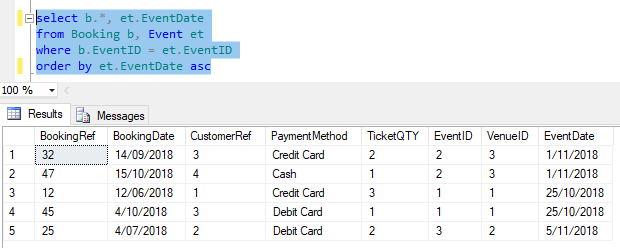
1. **Selecting All Bookings by Event Date in Ascending Order**

select b.\*, et.EventDate

from Booking b, Event et

where b.EventID = et.EventID

order by et.EventDate asc

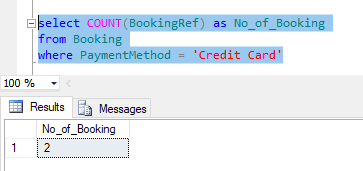


1. **Counting All Bookings Where Payment Method is Credit Card**

select COUNT(BookingRef) as No\_of\_Booking

from Booking

where PaymentMethod = 'Credit Card'



1. **Selecting All Event Bookings Where Quantity of Tickets is Greater Than 1 Ordered by Description**

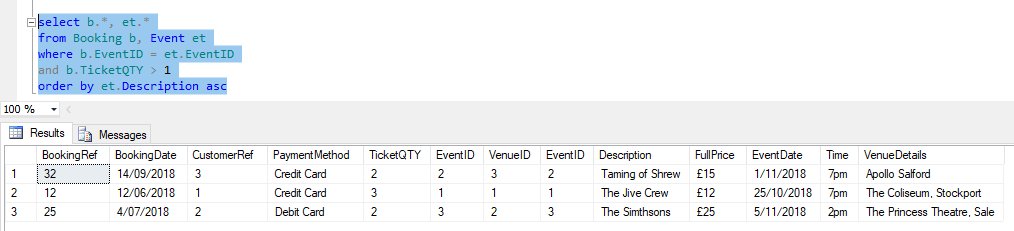
select b.\*, et.\*

from Booking b, Event et

where b.EventID = et.EventID

and b.TicketQTY > 1

order by et.Description asc



1. **Selecting First Names, Surnames and Loyalty Level Who Pay by Credit Card Grouped by Loyalty Level**

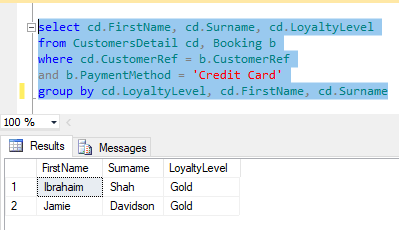
select cd.FirstName, cd.Surname, cd.LoyaltyLevel

from CustomersDetail cd, Booking b

where cd.CustomerRef = b.CustomerRef

and b.PaymentMethod = 'Credit Card'

group by cd.LoyaltyLevel, cd.FirstName, cd.Surname

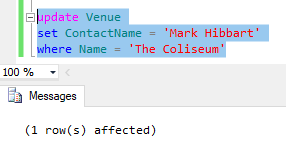


1. **Changing the Contact Name for ‘The Coliseum’ to ‘Mark Hibbart’**

update Venue

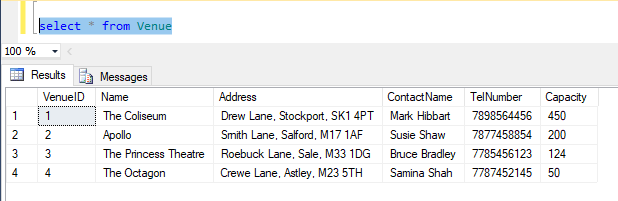
set ContactName = 'Mark Hibbart'

where Name = 'The Coliseum'



**Result**

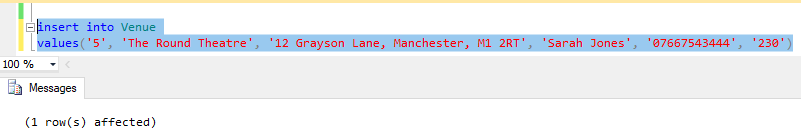
select \* from Venue



1. **Adding A New Venue**

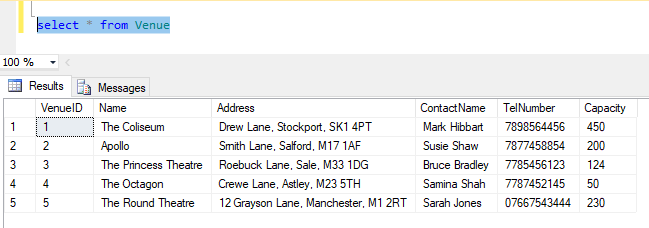
insert into Venue

values('5', 'The Round Theatre', '12 Grayson Lane, Manchester, M1 2RT', 'Sarah Jones', '07667543444', '230')



**Result**

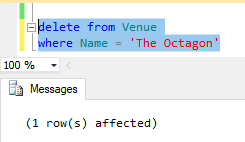
select \* from Venue



**Deleting the Venue ‘The Octagon’**

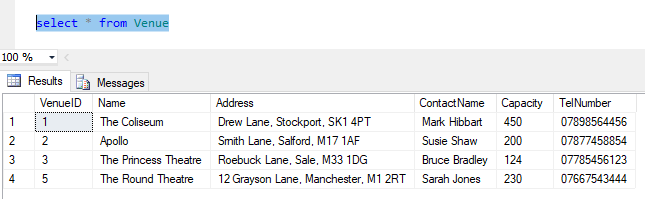
delete from Venue

where Name = 'The Octagon'



**Result**

select \* from Venue



**Task – 3**

**Assessment**

**Required Analysis and Assumptions**

The arts company called Stella Arts would like to have a database system to keep track of the bookings and to reduce pressure on the reception staff. Also, an online booking system and management information which relates to bookings for the business planning purposes are desired. Thus, a database system is built according to the requirements as possible.

In the task 1 of the assignment, it asks for the entity relationship model, a fully normalized model, how the data are normalized and a data dictionary. In the task 2 of the assignment, it expects the data to be created, inserted, updated and deleted. In the task 3 of the assignment, it demands for an assessment including the requirement analysis, assumptions, design decisions, justification and how the requirements have been met. Thus, assumptions for this assignment are that the task 1 is about deciding what attributes the database system should have, the task 2 is about creating a database system and the last task 3 is about writing a documentation of an assessment.

**Design Decisions and Justifications**

There are a few differences between the top down technique and the bottom up technique of entity relationship model. In the top down technique, there is no entity called “CustomersLoyalty” but in the normalization form, it exists. For the customer, there is only one loyalty level for him and in one specific loyalty level, there are more than one customer. Therefore, those two entities can be connected without an intersection entity. But if the customer has just joined, there is no loyalty level for that customer. When inserting the data for the newly joined customer, there will be no data in the “LoyaltyLevel”. It will only show “NULL” as the value. Thus, the entity “CustomerLoyalty” is considered as a further development in the future for those who has just joined for the Stella Arts Company.

Another one of the differences is that the entities “Tickets” and “CustomersDetail” are connected in the normalization form but they are not connected in the top down technique. In the database system, the entities are not connected because the “CustomersDetail” table is linked with “Booking” table and the “Booking” table is linked with “Ticket” table. So, it is not that necessary for the “CustomersDetail” table and “Ticket” table to be connected. If the “Ticket” table and “CustomersDetail” table are connected, the database system will be more complicated. In justification, there is no relation between those two entities to reduce complexity in the database system.

Since there is no relation between the two entities, there will be no foreign key “CustomerRef” in the “Ticket” table. Thus, only the attribute “CustomerName” exists in that table. To sum up, the database system is created according to those design decisions and justifications.

Only after finishing task 2, all the telephone number in the database does not display the first zero in data which all the telephone numbers have in common. Therefore, the columns are dropped and then added again. Since there is no data inserted in that columns, the values became “NULL” in those columns. Thus, the data are inserted again by the update statement. In this time, all the numbers are written with the open inverted comma and close inverted comma for the first zero in the telephone numbers to be displayed in the database. Then only the telephone numbers are displayed rightfully.

**How the Requirements have been met**

For the task 1, an entity relationship model is drawn with both the top down technique and bottom up technique and explanation of how normalization of the data has contributed to finish the ER diagram is also written. The required data dictionary is also entered.

In the task 2, all of the requirements can be done with SQL statements. In SQL, there are Data Definition Language (DDL) and Data Manipulation Language (DML). DDL is used for the purpose of defining the database objects, for example, tables and columns. The primary operator of DDL is the create statement. DML is used for the purpose of performing actions on the existed data. With those two languages, the SQL statements are written for the creating, inserting, selecting, updating and deleting of the data. And the requirements for the task 3 is completed.

(Benyon-Davies, 2003) (Connolly, T and Begg, 2009) (Holler, J Ramesh, V and T Toppi, 2010)

# **References**

Benyon-Davies, 2003. *Database Systems.* 3rd ed. s.l.:s.n.

Connolly, T and Begg, 2009. *Database System : A practical approach to Design, Implementation and Management.* 5th ed. s.l.:s.n.

Holler, J Ramesh, V and T Toppi, 2010. *Modern Database Management.* 10th ed. s.l.:s.n.

## **Candidate Checklist**

Please use the following checklist to ensure that your work is ready for submission.



Have you read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensured that you have acknowledge all the sources that you have used in your work?

Have you completed the 'Statement and Confirmation of Own Work' form and attached it to your assignment? **You must do this**.



Have you ensured that your work has not gone over or under the recommended word count by more than 10%?



Have you ensured that your work does not contain viruses and can be run directly?

