in collaboration with

Softwarica College of IT & E-commerce | Coventry University

Assignment Title:

CW2 Replacement Coursework for Exam

Coursework Type:

Individual/Practical

Module Name:

Enterprise Information Systems

Intake:

September/November 2019

Submitted By

CU ID: 10173183

College ID: 190199

Name: Gauri Shankar Sharma

Softwarica College in collaboration with
Coventry University
Assessment Submission and Declaration Form
PLEASE COMPLETE SECTIONS IN BLOCK CAPITALS

Group work If group work ALL student names and IDs must be added below- on behalf of all members; Name.. ID....................... Name.. ID....................... Name.. ID....................... Name.. ID....................... Name.. ID.......................	**Surname:** SHARMA **First Name:** GAURI SHANKAR **Word Count:** 1096	
Student number (ID): 10173183	**Attempt:** FIRST: □ RESIT: □	
Assignment Due Date: 11/09/2020	**Module Code:** STW104KM	
Program Title: BSC (HONS)COMPUTING		
Module Title: Enterprise Information Systems		
Name of Supervisor or Tutor (if applicable): Achyut Timsina	**Individual Work:** □	**Group Work:** □
Assessment Title and Type (i.e. essay, journal, CD, Dissertation)	CW2 Replacement Coursework for Exam	

I have read the Softwarica College rules and regulations on the submission of academic work and in particular the sections concerning misconduct in assessment, including plagiarism, collusion and cheating. I certify that this assignment is the result of my own (or group) work and contains no unreferenced material from another source and does not contravene any part of the College's rules and regulations.

I acknowledge that in submitting this work I am declaring that I (or my group) are fit to be assessed and that a deferral may not be requested following hand in.

I confirm that an electronic version of the item to be assessed where appropriate) is available and will be made available to the College by the specified deadline via Moodle.

In respect of group assignments, the submission of this work is made on the basis that all group members are jointly and severally responsible for the work presented for assessment and that by handing in this item for assessment, all group members acknowledge and confirm the statements above and that ALL student names and ID numbers for the group are listed.

Student(s) Signature:	**College Stamp:**

Table of Contents

Introduction

The main goal of this report is to provide an optimal solution and management procedure of an online booking system for Itahari Arts which is an arts company that handles, Organize, plays and musicals. This system keeps track of all the Loyalty scheme Levels, Customer Details, Bookings, Tickets, Venues. For this, following activities will be done as given in the scenario.

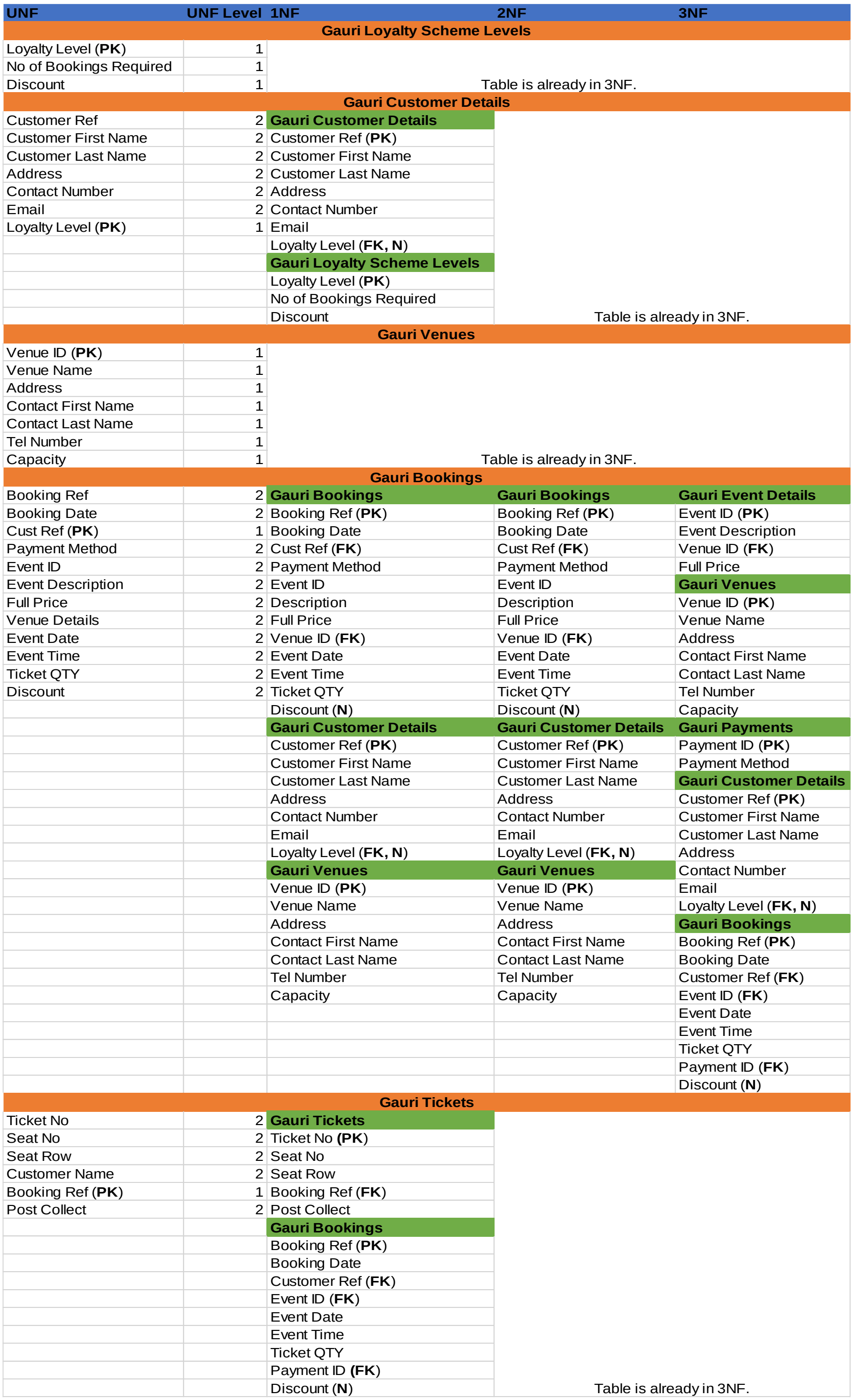
- Normalization of the given data will be done.
- An ER diagram will be made for the system.
- All normalized table will be created in SQL.
- All data will be entered inside the tables.
- SQL queries as given in the scenario will be written.
- A perfect cloud computing service and software development model will be suggested.

Section A

Normalization

Before rushing to the ER diagram, normalization of the given data has been shown below in a table.

Table 1:Normalization

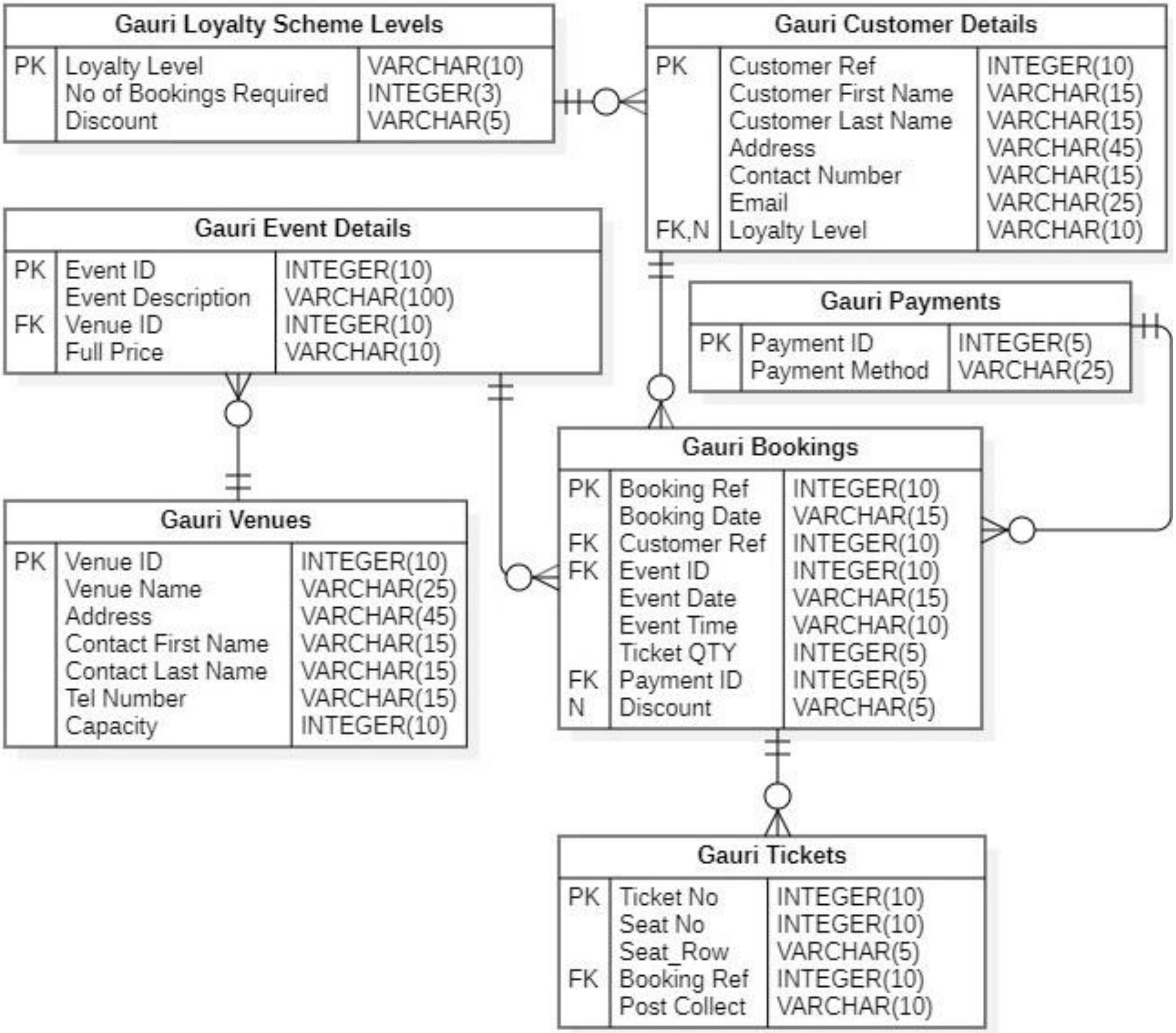


UNF	UNF Level	1NF	2NF	3NF
Gauri Loyalty Scheme Levels				
Loyalty Level **(PK)**	1			
No of Bookings Required	1			
Discount	1		Table is already in 3NF.	
Gauri Customer Details				
Customer Ref	2	**Gauri Customer Details**		
Customer First Name	2	Customer Ref **(PK)**		
Customer Last Name	2	Customer First Name		
Address	2	Customer Last Name		
Contact Number	2	Address		
Email	2	Contact Number		
Loyalty Level **(PK)**	1	Email		
		Loyalty Level **(FK, N)**		
		Gauri Loyalty Scheme Levels		
		Loyalty Level **(PK)**		
		No of Bookings Required		
		Discount	Table is already in 3NF.	
Gauri Venues				
Venue ID **(PK)**	1			
Venue Name	1			
Address	1			
Contact First Name	1			
Contact Last Name	1			
Tel Number	1			
Capacity	1		Table is already in 3NF.	
Gauri Bookings				
Booking Ref	2	**Gauri Bookings**	**Gauri Bookings**	**Gauri Event Details**
Booking Date	2	Booking Ref **(PK)**	Booking Ref **(PK)**	Event ID **(PK)**
Cust Ref **(PK)**	1	Booking Date	Booking Date	Event Description
Payment Method	2	Cust Ref **(FK)**	Cust Ref **(FK)**	Venue ID **(FK)**
Event ID	2	Payment Method	Payment Method	Full Price
Event Description	2	Event ID	Event ID	**Gauri Venues**
Full Price	2	Description	Description	Venue ID **(PK)**
Venue Details	2	Full Price	Full Price	Venue Name
Event Date	2	Venue ID **(FK)**	Venue ID **(FK)**	Address
Event Time	2	Event Date	Event Date	Contact First Name
Ticket QTY	2	Event Time	Event Time	Contact Last Name
Discount	2	Ticket QTY	Ticket QTY	Tel Number
		Discount **(N)**	Discount **(N)**	Capacity
		Gauri Customer Details	**Gauri Customer Details**	**Gauri Payments**
		Customer Ref **(PK)**	Customer Ref **(PK)**	Payment ID **(PK)**
		Customer First Name	Customer First Name	Payment Method
		Customer Last Name	Customer Last Name	**Gauri Customer Details**
		Address	Address	Customer Ref **(PK)**
		Contact Number	Contact Number	Customer First Name
		Email	Email	Customer Last Name
		Loyalty Level **(FK, N)**	Loyalty Level **(FK, N)**	Address
		Gauri Venues	**Gauri Venues**	Contact Number
		Venue ID **(PK)**	Venue ID **(PK)**	Email
		Venue Name	Venue Name	Loyalty Level **(FK, N)**
		Address	Address	**Gauri Bookings**
		Contact First Name	Contact First Name	Booking Ref **(PK)**
		Contact Last Name	Contact Last Name	Booking Date
		Tel Number	Tel Number	Customer Ref **(FK)**
		Capacity	Capacity	Event ID **(FK)**
				Event Date
				Event Time
				Ticket QTY
				Payment ID **(FK)**
				Discount **(N)**
Gauri Tickets				
Ticket No	2	**Gauri Tickets**		
Seat No	2	Ticket No **(PK)**		
Seat Row	2	Seat No		
Customer Name	2	Seat Row		
Booking Ref **(PK)**	1	Booking Ref **(FK)**		
Post Collect	2	Post Collect		
		Gauri Bookings		
		Booking Ref **(PK)**		
		Booking Date		
		Customer Ref **(FK)**		
		Event ID **(FK)**		
		Event Date		
		Event Time		
		Ticket QTY		
		Payment ID **(FK)**		
		Discount **(N)**	Table is already in 3NF.	

Normalization is a database design technique whose main purpose is to eliminate duplicate and repeating data. Therefore, normalization is a process of dividing larger tables into smaller ones and linking them using relationships to reduce data redundancy and to eliminate undesirable Insertion, Update and Deletion Anomalies. 1NF, 2NF, 3NF are basic database normal forms (Guru99, 2020).

ER Diagram

An entity relation model for the proposed system for Itahari Arts has been shown below using bottom-up approach which is fully normalized to 3rd normal form.



Gauri Loyalty Scheme Levels

PK	Loyalty Level	VARCHAR(10)
	No of Bookings Required	INTEGER(3)
	Discount	VARCHAR(5)

Gauri Customer Details

PK	Customer Ref	INTEGER(10)
	Customer First Name	VARCHAR(15)
	Customer Last Name	VARCHAR(15)
	Address	VARCHAR(45)
	Contact Number	VARCHAR(15)
	Email	VARCHAR(25)
FK,N	Loyalty Level	VARCHAR(10)

Gauri Event Details

PK	Event ID	INTEGER(10)
	Event Description	VARCHAR(100)
FK	Venue ID	INTEGER(10)
	Full Price	VARCHAR(10)

Gauri Payments

PK	Payment ID	INTEGER(5)
	Payment Method	VARCHAR(25)

Gauri Bookings

PK	Booking Ref	INTEGER(10)
	Booking Date	VARCHAR(15)
FK	Customer Ref	INTEGER(10)
FK	Event ID	INTEGER(10)
	Event Date	VARCHAR(15)
	Event Time	VARCHAR(10)
	Ticket QTY	INTEGER(5)
FK	Payment ID	INTEGER(5)
N	Discount	VARCHAR(5)

Gauri Venues

PK	Venue ID	INTEGER(10)
	Venue Name	VARCHAR(25)
	Address	VARCHAR(45)
	Contact First Name	VARCHAR(15)
	Contact Last Name	VARCHAR(15)
	Tel Number	VARCHAR(15)
	Capacity	INTEGER(10)

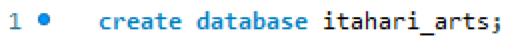
Gauri Tickets

PK	Ticket No	INTEGER(10)
	Seat No	INTEGER(10)
	Seat_Row	VARCHAR(5)
FK	Booking Ref	INTEGER(10)
	Post Collect	VARCHAR(10)

Figure 1: ER Diagram

Creating and Showing Tables

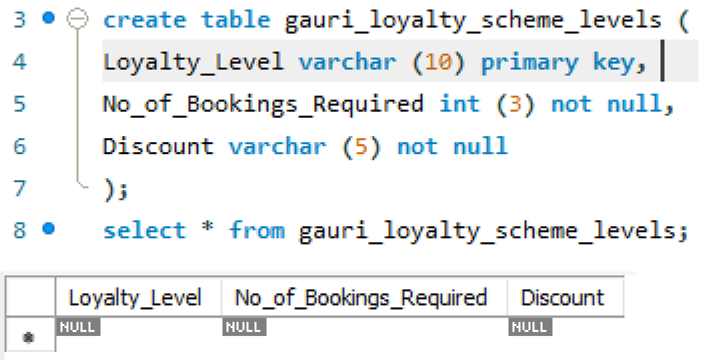
Database



```
create database itahari_arts;
```

Figure 2: Creating Database

Gauri Loyalty Scheme Levels

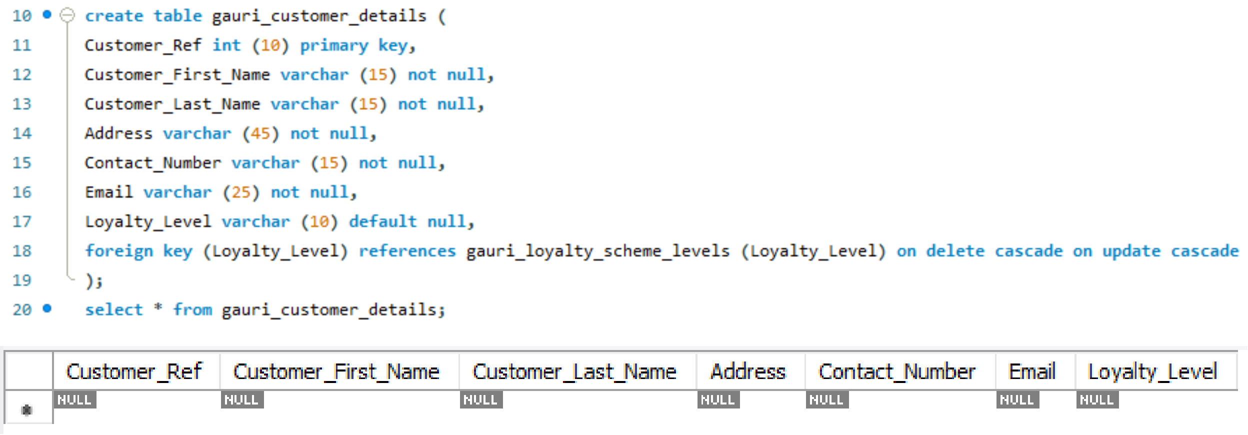


```
create table gauri_loyalty_scheme_levels (
Loyalty_Level varchar (10) primary key,
No_of_Bookings_Required int (3) not null,
Discount varchar (5) not null
);
select * from gauri_loyalty_scheme_levels;
```

Loyalty_Level	No_of_Bookings_Required	Discount
NULL	NULL	NULL

Figure 3: Creating and showing Gauri Loyalty Scheme Levels Table

Gauri Customer Details

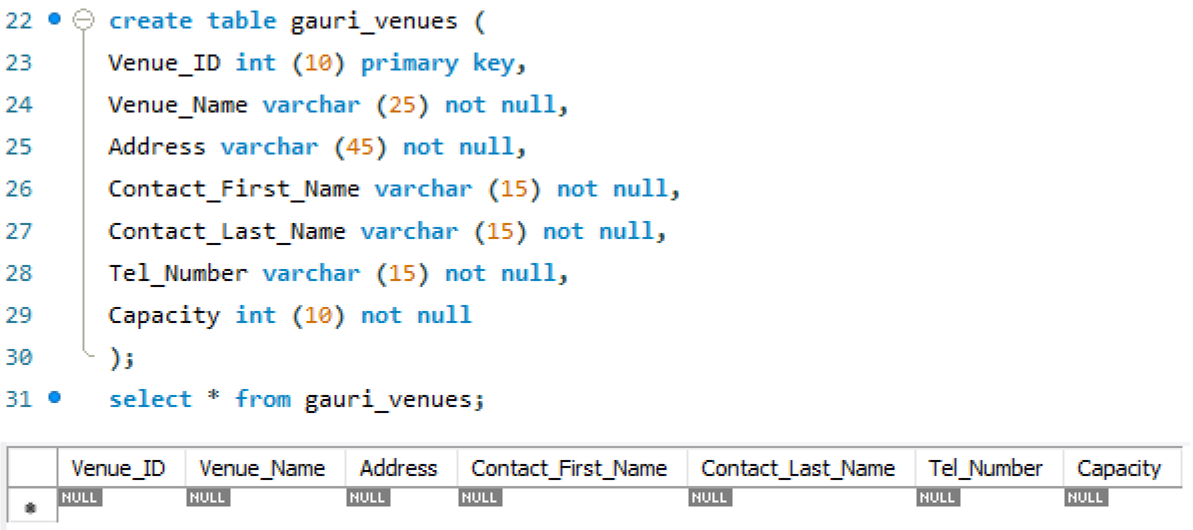


```
create table gauri_customer_details (
Customer_Ref int (10) primary key,
Customer_First_Name varchar (15) not null,
Customer_Last_Name varchar (15) not null,
Address varchar (45) not null,
Contact_Number varchar (15) not null,
Email varchar (25) not null,
Loyalty_Level varchar (10) default null,
foreign key (Loyalty_Level) references gauri_loyalty_scheme_levels (Loyalty_Level) on delete cascade on update cascade
);
select * from gauri_customer_details;
```

Customer_Ref	Customer_First_Name	Customer_Last_Name	Address	Contact_Number	Email	Loyalty_Level
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 4:Creating and showing Gauri Customer Details Table

Gauri Venues

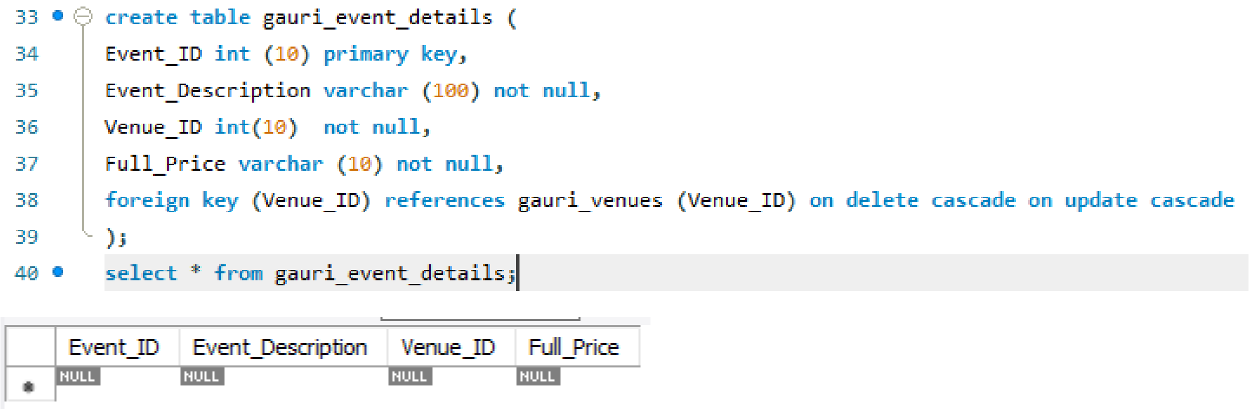


```
create table gauri_venues (
Venue_ID int (10) primary key,
Venue_Name varchar (25) not null,
Address varchar (45) not null,
Contact_First_Name varchar (15) not null,
Contact_Last_Name varchar (15) not null,
Tel_Number varchar (15) not null,
Capacity int (10) not null
);
select * from gauri_venues;
```

	Venue_ID	Venue_Name	Address	Contact_First_Name	Contact_Last_Name	Tel_Number	Capacity
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 5:Creating and showing Gauri Venues Table

Gauri Event Details

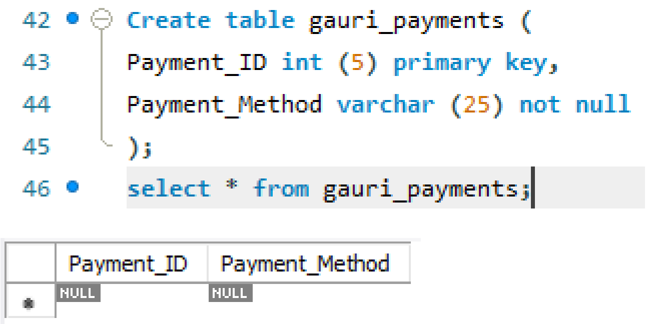


```
create table gauri_event_details (
Event_ID int (10) primary key,
Event_Description varchar (100) not null,
Venue_ID int(10)  not null,
Full_Price varchar (10) not null,
foreign key (Venue_ID) references gauri_venues (Venue_ID) on delete cascade on update cascade
);
select * from gauri_event_details;
```

	Event_ID	Event_Description	Venue_ID	Full_Price
*	NULL	NULL	NULL	NULL

Figure 6:Creating and showing Gauri Event Details table

Gauri Payments

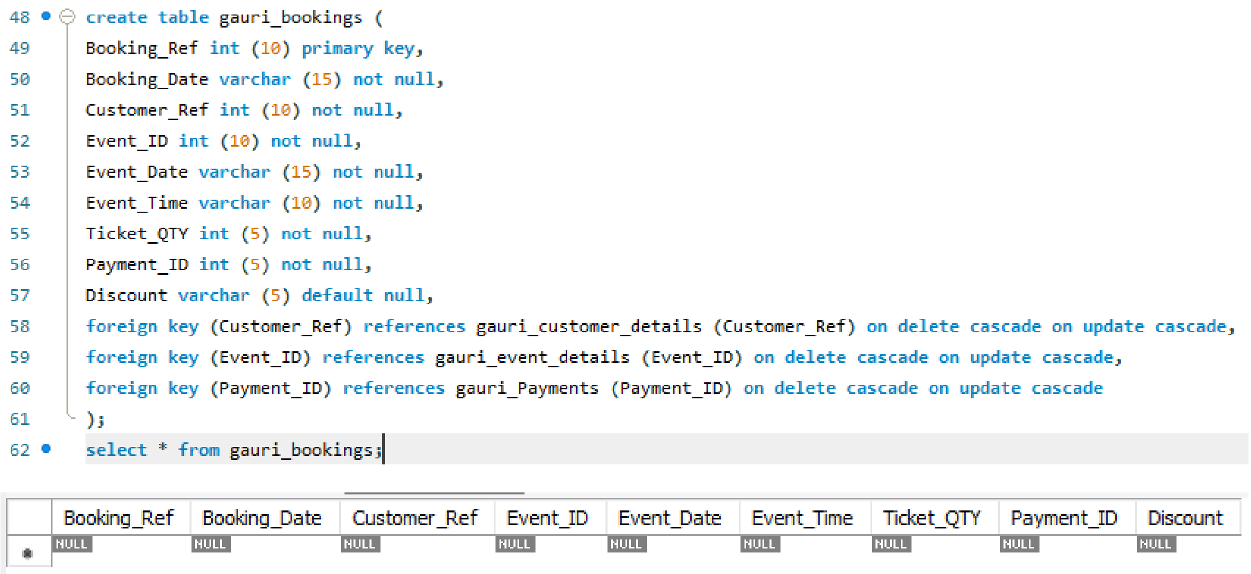


```
Create table gauri_payments (
Payment_ID int (5) primary key,
Payment_Method varchar (25) not null
);
select * from gauri_payments;
```

	Payment_ID	Payment_Method
*	NULL	NULL

Figure 7:Creating and showing Gauri Payments Table

Gauri Bookings

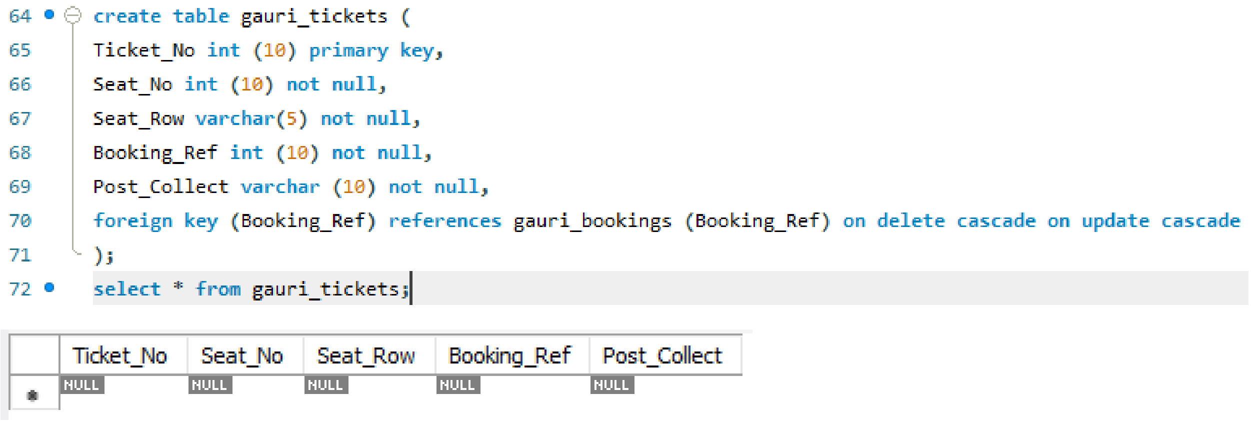


```
create table gauri_bookings (
Booking_Ref int (10) primary key,
Booking_Date varchar (15) not null,
Customer_Ref int (10) not null,
Event_ID int (10) not null,
Event_Date varchar (15) not null,
Event_Time varchar (10) not null,
Ticket_QTY int (5) not null,
Payment_ID int (5) not null,
Discount varchar (5) default null,
foreign key (Customer_Ref) references gauri_customer_details (Customer_Ref) on delete cascade on update cascade,
foreign key (Event_ID) references gauri_event_details (Event_ID) on delete cascade on update cascade,
foreign key (Payment_ID) references gauri_Payments (Payment_ID) on delete cascade on update cascade
);
select * from gauri_bookings;
```

Booking_Ref	Booking_Date	Customer_Ref	Event_ID	Event_Date	Event_Time	Ticket_QTY	Payment_ID	Discount
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 8:Creating and showing Gauri Bookings Table

Gauri Tickets



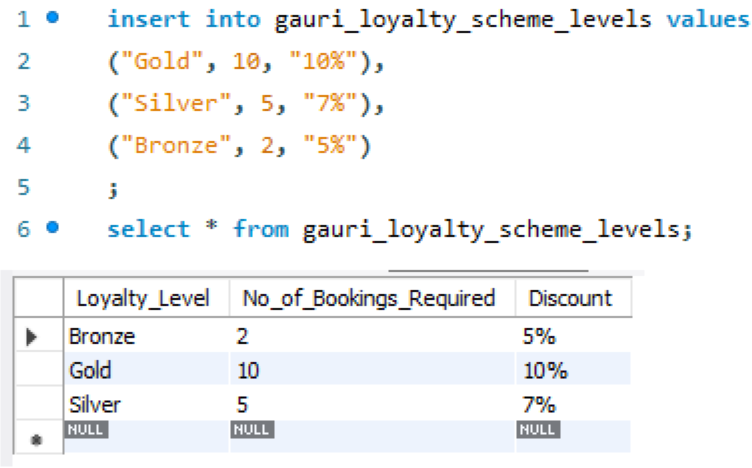
```
create table gauri_tickets (
Ticket_No int (10) primary key,
Seat_No int (10) not null,
Seat_Row varchar(5) not null,
Booking_Ref int (10) not null,
Post_Collect varchar (10) not null,
foreign key (Booking_Ref) references gauri_bookings (Booking_Ref) on delete cascade on update cascade
);
select * from gauri_tickets;
```

Ticket_No	Seat_No	Seat_Row	Booking_Ref	Post_Collect
NULL	NULL	NULL	NULL	NULL

Figure 9:Creating and showing Gauri Tickets Table

Inserting and Showing Data

Gauri Loyalty Scheme Levels

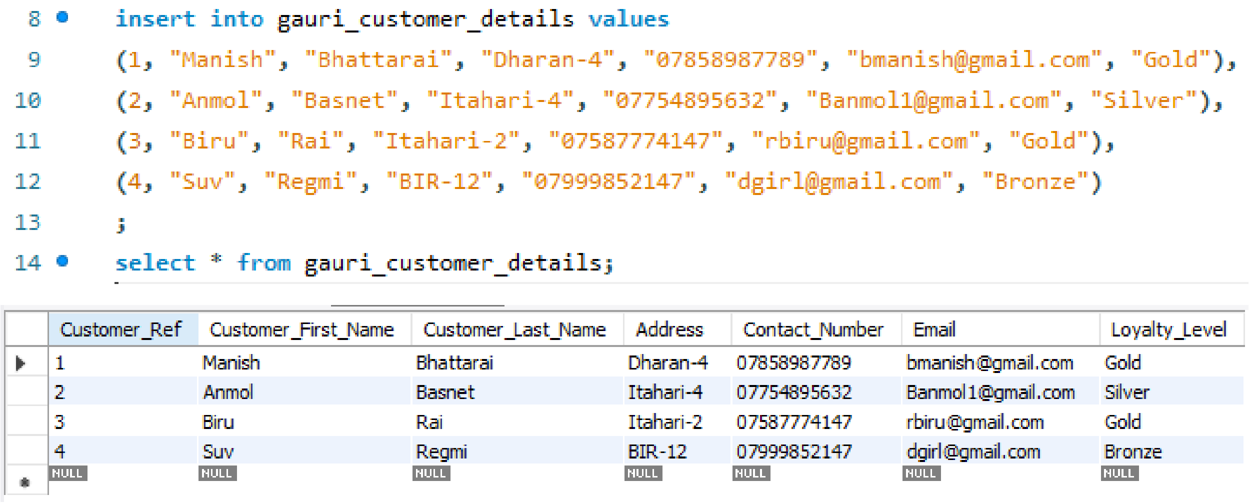


```
insert into gauri_loyalty_scheme_levels values
("Gold", 10, "10%"),
("Silver", 5, "7%"),
("Bronze", 2, "5%")
;
select * from gauri_loyalty_scheme_levels;
```

Loyalty_Level	No_of_Bookings_Required	Discount
Bronze	2	5%
Gold	10	10%
Silver	5	7%
NULL	NULL	NULL

Figure 10: Inserting and Showing Data of Gauri Loyalty Scheme Levels Table

Gauri Customer Details

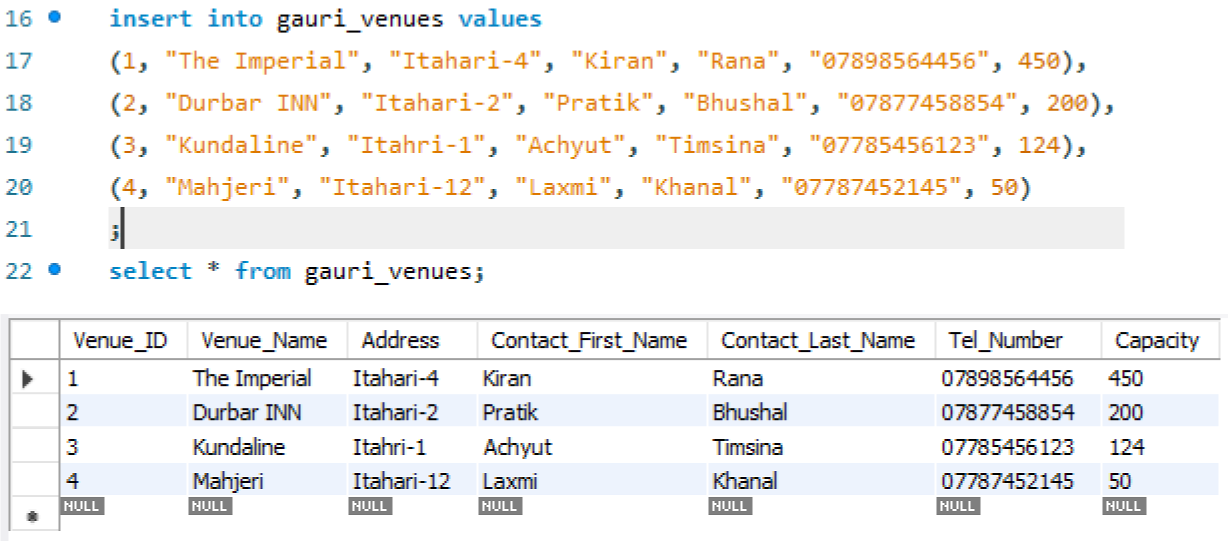


```
insert into gauri_customer_details values
(1, "Manish", "Bhattarai", "Dharan-4", "07858987789", "bmanish@gmail.com", "Gold"),
(2, "Anmol", "Basnet", "Itahari-4", "07754895632", "Banmol1@gmail.com", "Silver"),
(3, "Biru", "Rai", "Itahari-2", "07587774147", "rbiru@gmail.com", "Gold"),
(4, "Suv", "Regmi", "BIR-12", "07999852147", "dgirl@gmail.com", "Bronze")
;
select * from gauri_customer_details;
```

Customer_Ref	Customer_First_Name	Customer_Last_Name	Address	Contact_Number	Email	Loyalty_Level
1	Manish	Bhattarai	Dharan-4	07858987789	bmanish@gmail.com	Gold
2	Anmol	Basnet	Itahari-4	07754895632	Banmol1@gmail.com	Silver
3	Biru	Rai	Itahari-2	07587774147	rbiru@gmail.com	Gold
4	Suv	Regmi	BIR-12	07999852147	dgirl@gmail.com	Bronze
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 11:Inserting and Showing Data of Gauri Customer Details Table

Gauri Venues

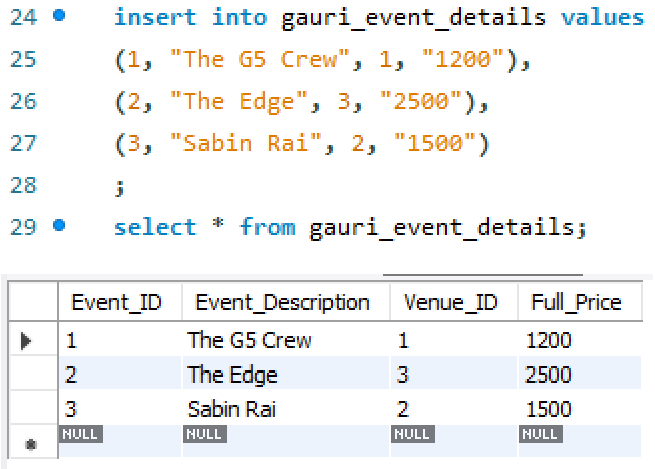


```
insert into gauri_venues values
(1, "The Imperial", "Itahari-4", "Kiran", "Rana", "07898564456", 450),
(2, "Durbar INN", "Itahari-2", "Pratik", "Bhushal", "07877458854", 200),
(3, "Kundaline", "Itahri-1", "Achyut", "Timsina", "07785456123", 124),
(4, "Mahjeri", "Itahari-12", "Laxmi", "Khanal", "07787452145", 50)
;
select * from gauri_venues;
```

Venue_ID	Venue_Name	Address	Contact_First_Name	Contact_Last_Name	Tel_Number	Capacity
1	The Imperial	Itahari-4	Kiran	Rana	07898564456	450
2	Durbar INN	Itahari-2	Pratik	Bhushal	07877458854	200
3	Kundaline	Itahri-1	Achyut	Timsina	07785456123	124
4	Mahjeri	Itahari-12	Laxmi	Khanal	07787452145	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 12:Inserting and Showing Data of Gauri Venues Table

Gauri Event Details

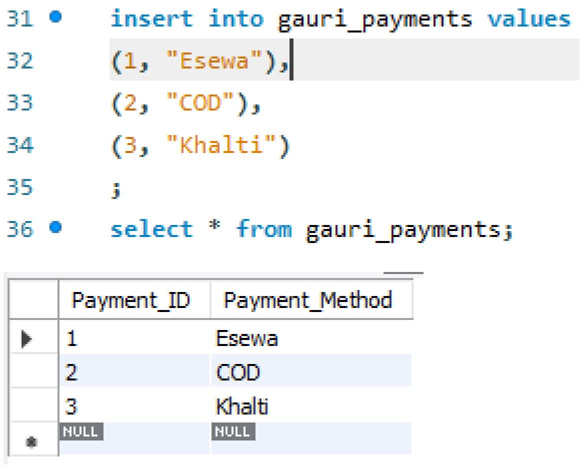


```
insert into gauri_event_details values
(1, "The G5 Crew", 1, "1200"),
(2, "The Edge", 3, "2500"),
(3, "Sabin Rai", 2, "1500")
;
select * from gauri_event_details;
```

Event_ID	Event_Description	Venue_ID	Full_Price
1	The G5 Crew	1	1200
2	The Edge	3	2500
3	Sabin Rai	2	1500
NULL	NULL	NULL	NULL

Figure 13:Inserting and Showing Data of Gauri Event Details Table

Gauri Payments

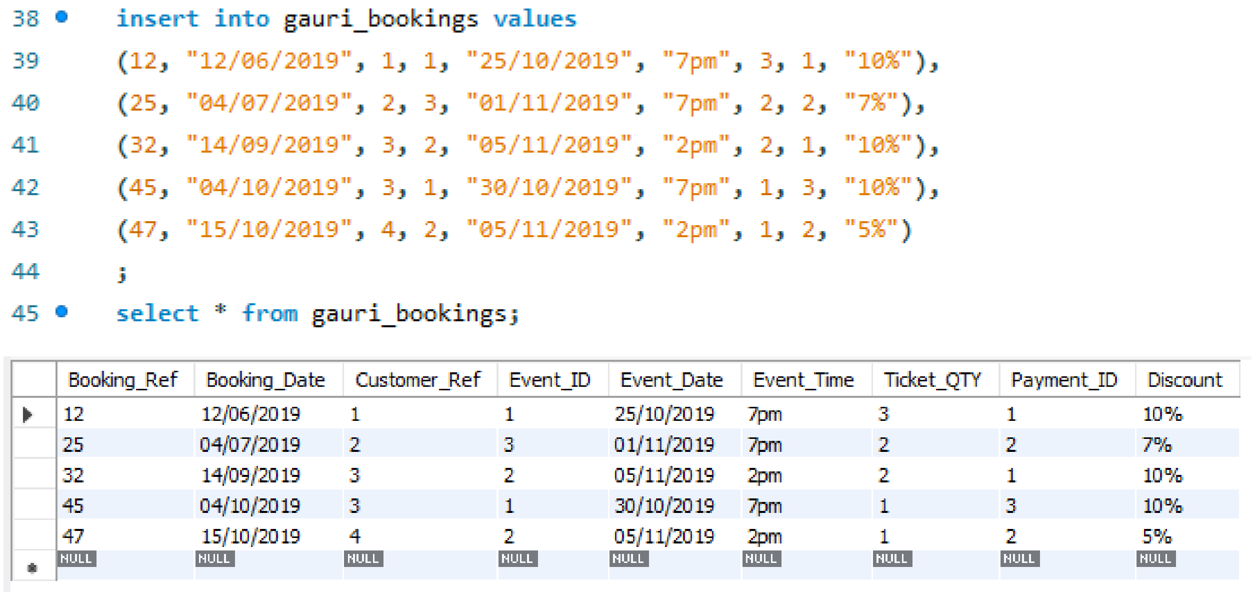


```sql
insert into gauri_payments values
(1, "Esewa"),
(2, "COD"),
(3, "Khalti")
;
select * from gauri_payments;
```

	Payment_ID	Payment_Method
▶	1	Esewa
	2	COD
	3	Khalti
✱	NULL	NULL

Figure 14:Inserting and Showing Data of Gauri Payments Table

Gauri Bookings

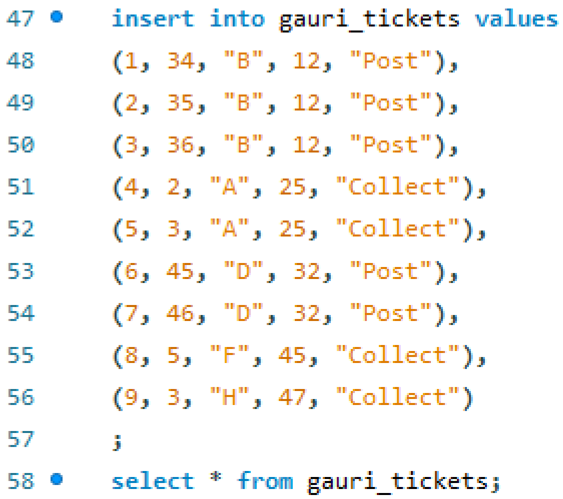


```sql
insert into gauri_bookings values
(12, "12/06/2019", 1, 1, "25/10/2019", "7pm", 3, 1, "10%"),
(25, "04/07/2019", 2, 3, "01/11/2019", "7pm", 2, 2, "7%"),
(32, "14/09/2019", 3, 2, "05/11/2019", "2pm", 2, 1, "10%"),
(45, "04/10/2019", 3, 1, "30/10/2019", "7pm", 1, 3, "10%"),
(47, "15/10/2019", 4, 2, "05/11/2019", "2pm", 1, 2, "5%")
;
select * from gauri_bookings;
```

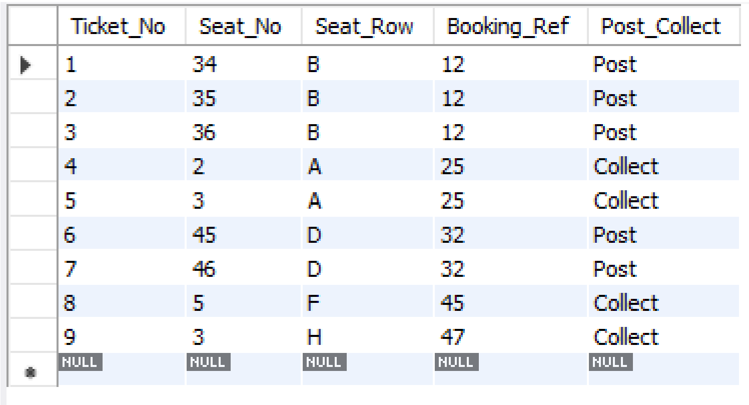
	Booking_Ref	Booking_Date	Customer_Ref	Event_ID	Event_Date	Event_Time	Ticket_QTY	Payment_ID	Discount
▶	12	12/06/2019	1	1	25/10/2019	7pm	3	1	10%
	25	04/07/2019	2	3	01/11/2019	7pm	2	2	7%
	32	14/09/2019	3	2	05/11/2019	2pm	2	1	10%
	45	04/10/2019	3	1	30/10/2019	7pm	1	3	10%
	47	15/10/2019	4	2	05/11/2019	2pm	1	2	5%
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 15:Inserting and Showing Data of Gauri Bookings Table

Gauri Tickets



```
insert into gauri_tickets values
(1, 34, "B", 12, "Post"),
(2, 35, "B", 12, "Post"),
(3, 36, "B", 12, "Post"),
(4, 2, "A", 25, "Collect"),
(5, 3, "A", 25, "Collect"),
(6, 45, "D", 32, "Post"),
(7, 46, "D", 32, "Post"),
(8, 5, "F", 45, "Collect"),
(9, 3, "H", 47, "Collect")
;
select * from gauri_tickets;
```

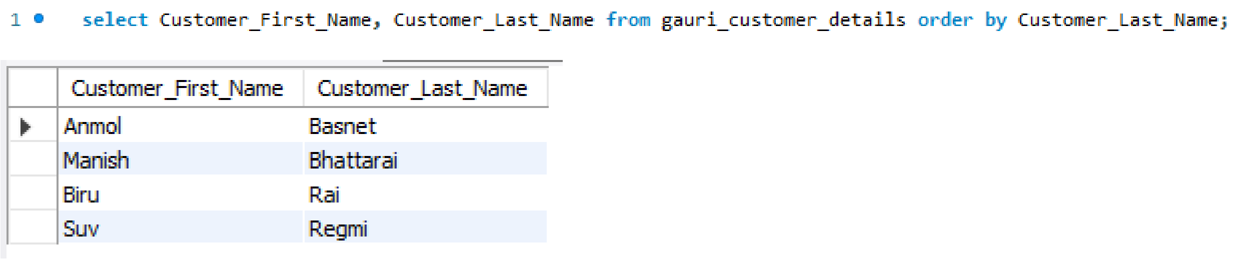


Ticket_No	Seat_No	Seat_Row	Booking_Ref	Post_Collect
1	34	B	12	Post
2	35	B	12	Post
3	36	B	12	Post
4	2	A	25	Collect
5	3	A	25	Collect
6	45	D	32	Post
7	46	D	32	Post
8	5	F	45	Collect
9	3	H	47	Collect
NULL	NULL	NULL	NULL	NULL

Figure 16:Inserting and Showing Data of Gauri Tickets Table

Writing Queries

Selecting the first name and surname of customers in alphabetical order of surname

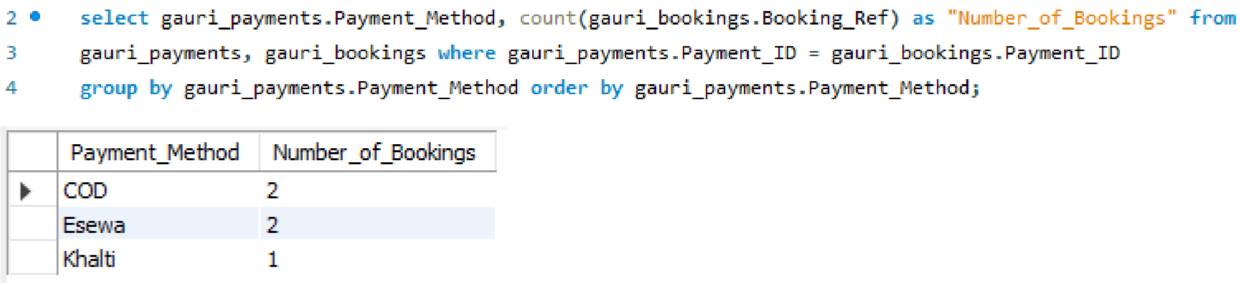


```sql
select Customer_First_Name, Customer_Last_Name from gauri_customer_details order by Customer_Last_Name;
```

Customer_First_Name	Customer_Last_Name
Anmol	Basnet
Manish	Bhattarai
Biru	Rai
Suv	Regmi

Figure 17:SQL Script and table

Showing the number of all the bookings according to their Payment Methods

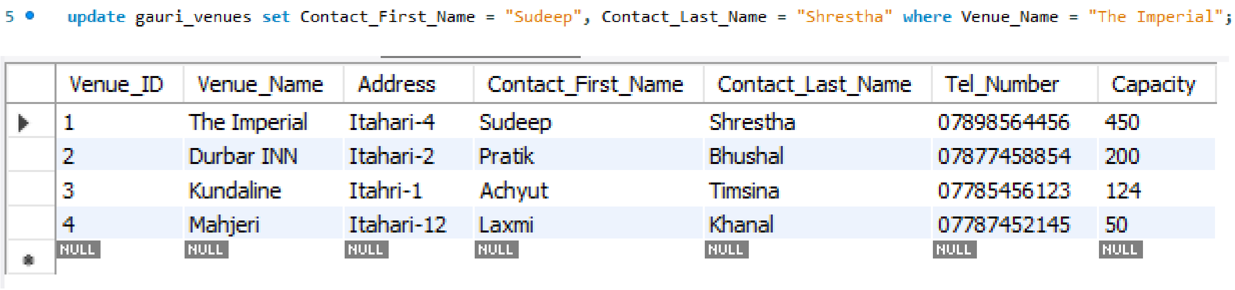


```sql
select gauri_payments.Payment_Method, count(gauri_bookings.Booking_Ref) as "Number_of_Bookings" from
gauri_payments, gauri_bookings where gauri_payments.Payment_ID = gauri_bookings.Payment_ID
group by gauri_payments.Payment_Method order by gauri_payments.Payment_Method;
```

Payment_Method	Number_of_Bookings
COD	2
Esewa	2
Khalti	1

Figure 18: SQL Script and table

Changing the contact name for ‘The Imperial’ to Sudeep Shrestha

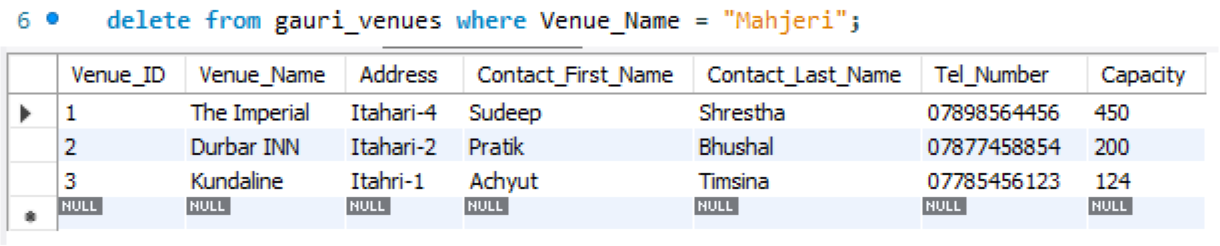


```sql
update gauri_venues set Contact_First_Name = "Sudeep", Contact_Last_Name = "Shrestha" where Venue_Name = "The Imperial";
```

Venue_ID	Venue_Name	Address	Contact_First_Name	Contact_Last_Name	Tel_Number	Capacity
1	The Imperial	Itahari-4	Sudeep	Shrestha	07898564456	450
2	Durbar INN	Itahari-2	Pratik	Bhushal	07877458854	200
3	Kundaline	Itahri-1	Achyut	Timsina	07785456123	124
4	Mahjeri	Itahari-12	Laxmi	Khanal	07787452145	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 19: SQL Script and table

Deleting the venue ‘Mahjeri’



```sql
delete from gauri_venues where Venue_Name = "Mahjeri";
```

Venue_ID	Venue_Name	Address	Contact_First_Name	Contact_Last_Name	Tel_Number	Capacity
1	The Imperial	Itahari-4	Sudeep	Shrestha	07898564456	450
2	Durbar INN	Itahari-2	Pratik	Bhushal	07877458854	200
3	Kundaline	Itahri-1	Achyut	Timsina	07785456123	124
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 20: SQL Script and table

Section B

Cloud Computing Service

Cloud computing is the delivery of different kind of computing services like software, storage, servers, database with the help of internet which helps the infrastructure run more efficiently and effectively. Cloud computing can be public, private and hybrid. Cloud computing services are listed below

1. Infrastructure as a service (IaaS),
2. Platform as a service (PaaS),
3. Software as a service (SaaS) (Azure, 2020).

As Itahari Arts is arts company and the bookings are done by the receptionist, IaaS will be better for Itahari Arts because it is mainly for Sysadmins and Itahri Arts does not seem to be a larger company in which the application is used by the receptionist or the administrator which will do the booking. It will avoid the expense and complexity of buying and management of Itahari Arts's infrastructure by providing a reliable, stable and supportable infrastructure. IaaS is very flexible and has enhanced scalability so that Itahari Arts will pay only what they will use, Itahari Arts will have full control over their application environment and deployment. Itahari Arts can purchase resource as they need. The main reason behind suggesting IaaS is because IaaS's cost varies depending on consumption so that Itahari Arts will get cloud computing service in as much less as they want and they will have full management of their application by themselves and its services will be highly scalable.

SaaS is on-demand service, platform independent and accessible via web browser or light weight client applications. PaaS is based on a programming language execution environment and it is domain for developers. I did not suggest SaaS because of Lack of control, Security and data concerns, Limited range of applications, Connectivity requirement, Performance and PaaS because of Data Security, Integration, Changes from vendor.

Software Development Life Cycle

Software development life cycle (SDLC) is a kind of loop which defines step by step processes involved in developing a software and it provides a detailed plan for building, testing, deploying an maintaining the software for production of high-quality product. Requirement gathering & Analysis, Design, Implementation & Coding, Testing, Deployment and Maintenance are different phases involved in SDLC. Waterfall Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model are different SDLC Models (Help, 2020).

For Itahari Arts, V-Shaped SDLC Mode will be better because V-Shaped Model is better for small to medium projects in which requirements are defined clearly and fixed. This model is also known as Verification and Validation Model where development and testing go parallel. Different phases of V-Shaped Model are listed below.

1. Verification Phase:

i. **Requirement Analysis:**

   All requirements are gathered around and then reviewing and analysis of the requirements are done.

ii. **System Design:**

   System is design and documented in a design document after the requirements are cleared.

iii. **High-Level Design:**

   Design of modules is done which defines the functionality between two modules.

iv. **Low-Level Design:**

   Design of individual components is done.

v. **Coding:**

   Development of coding is done.

2. Validation Phase:

i. **Unit Testing:**

Unit testing on individual components of Low-Level design is done using unit test cased.

ii. **Integration Testing:**

Integration Testing is done on the integrated modules of the High-Level design.

iii. **System Testing:**

The whole system and its functionality are tested.

iv. **Acceptance Testing:**

Testing of the requirements is done in the customer's environment.

Pros:

1. Simple and easy to understand.
2. Time saving due to parallel Verification and Validation which success over the waterfall model.
3. Gives high quality product due to systematic and disciplined.
4. Well for small project where requirements are easily understood.

Cons:

1. Not good for ongoing projects.
2. Cost to high if the requirement change is needed at later stage.
3. No production of early prototype of the software.
4. Rigid and least flexible.

References

Azure, M., 2020. *Azure.microsoft.com.* [Online]
Available at: https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#cloud-deployment-types
[Accessed 23 August 2020].

Guru99, 2020. *Guru99.com.* [Online]
Available at: https://www.guru99.com/database-normalization.html
[Accessed 21 August 2020].

Help, S. T., 2020. *Softwaretestinghelp.com.* [Online]
Available at: https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/
[Accessed 23 August 2020].