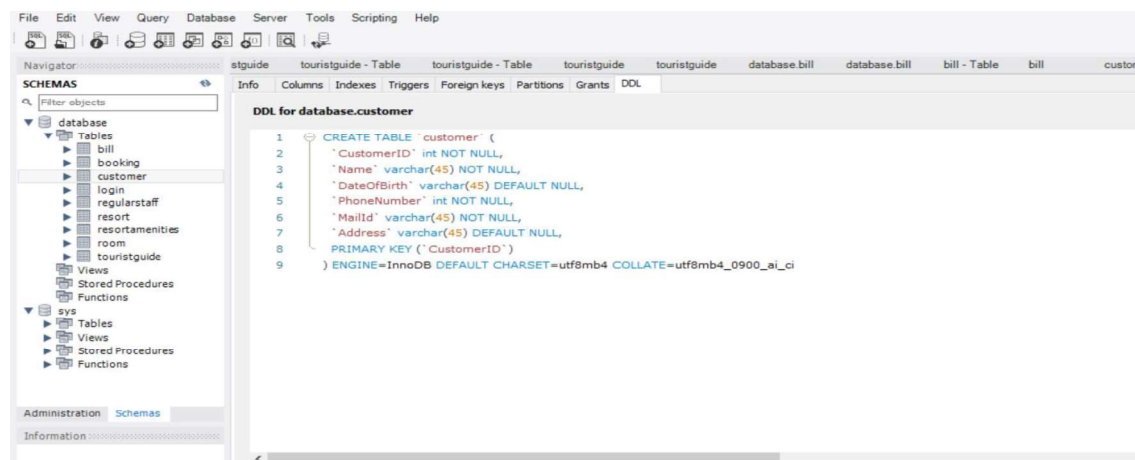


The resort management system has the following tables-

Customer, Login, Room, Resort, ResortAmenities, Bill, Booking, RegularStaff, TouristGuide and are created using the following DDL queries.

Customer Table –



Inserting Records into the table –

```
INSERT INTO customer(CustomerID, Name, DateOfBirth, PhoneNumber, MailId, Address) VALUES  
(89, JOHN, 09/23/1994, 67868716, JOHN@GMAIL.COM, BOSTON),  
(90, MARY, 06/16/1997, 890657892, MARY@GMAIL.COM, BOSTON),  
(91, PAUL, 12/12/1999, 876427894, PAUL@GMAIL.COM, BOSTON),  
(92, ROSE, 10/02/2000, 564379932, ROSE@GMAIL.COM, BOSTON)
```

resort resortamenities regularstaff database.regularstaff regularstaff touristguide database.touristguide database.touristguide tourist

1 • SELECT * FROM customer;

Result Grid

CustomerID	Name	DateOfBirth	PhoneNumber	MailId	Address
89	JOHN	09/23/1994	678686716	JOHN@gmail.com	boston
90	MARY	06/16/1997	890657892	mary@gmail.com	boston
91	PAUL	12/12/1999	876427894	paul@gmail.com	boston
92	ROSE	10/02/2000	564379932	rose@gmail.com	boston

LOGIN TABLE –

Navigator: database Schemas Filter objects

database

- Tables
 - bill
 - booking
 - customer
 - login
 - regularstaff
 - resort
 - resortamenities
 - room
 - touristguide
- Views
- Stored Procedures
- Functions

sys

- Tables
- Views
- Stored Procedures
- Functions

Table touristguide touristguide database.bill database.bill bill - Table bill customer database.customer

Info Columns Indexes Triggers Foreign keys Partitions Grants DDL

DDL for database.login

```

1 CREATE TABLE `login` (
2   `LoginID` int NOT NULL,
3   `Password` varchar(45) NOT NULL,
4   `CustomerID` int NOT NULL,
5   PRIMARY KEY (`LoginID`),
6   UNIQUE KEY `LoginID_UNIQUE` (`LoginID`),
7   KEY `CustomerID_idx` (`CustomerID`),
8   CONSTRAINT `CustomerID` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`CustomerID`)
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
  
```

INSERTING RECORDS –

INSERT INTO login(LoginID, Password, CustomerID) VALUES

(34, bsr, 92),

(56, der, 89),

(78, ght, 90)

The screenshot shows a database client interface with a query window containing the SQL statement: `SELECT * FROM login;`. Below the query window, the 'Result Grid' displays the following data:

LoginID	Password	CustomerID
34	bsr	92
56	der	89
78	ght	90
NULL	NULL	NULL

Bill Table-

The screenshot shows a database client interface with the 'DDL' tab selected for the 'bill' table. The DDL for the 'bill' table is as follows:

```

1 CREATE TABLE 'bill' (
2   'BillID' int NOT NULL,
3   'CustID' int NOT NULL,
4   'Amount' int DEFAULT NULL,
5   'PaymentMethod' varchar(45) DEFAULT NULL,
6   'PaymentDate' varchar(45) DEFAULT NULL,
7   PRIMARY KEY ('BillID'),
8   UNIQUE KEY 'BillID_UNIQUE' ('BillID'),
9   UNIQUE KEY 'CustomerID_UNIQUE' ('CustID'),
10  CONSTRAINT 'CustID' FOREIGN KEY ('CustID') REFERENCES 'customer' ('CustomerID')
11 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

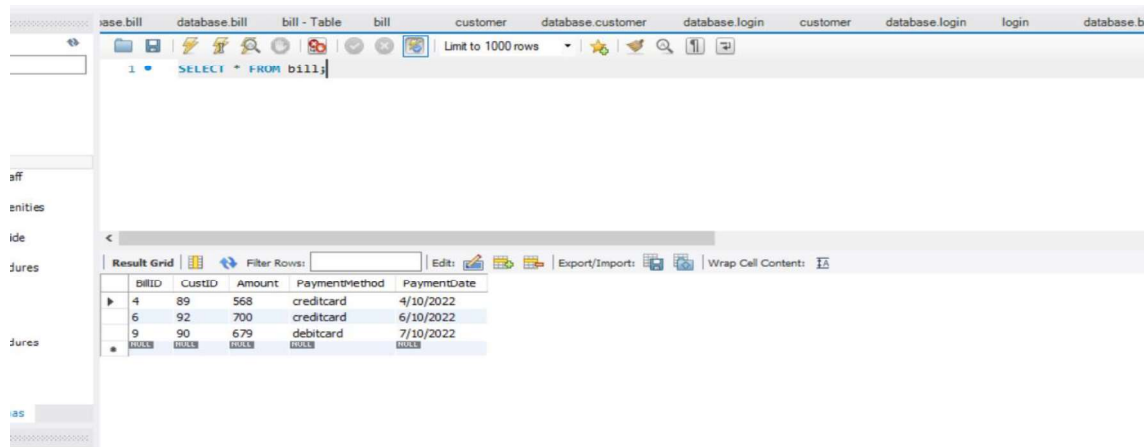
```

Inserting records-

INSERT INTO bill(BillID, CusID, Amount, PaymentMethod, PaymentDate) VALUES
(4, 89, 568, creditcard, 4/10/2022),

(6, 92, 700, creditcard, 6/10/2022),

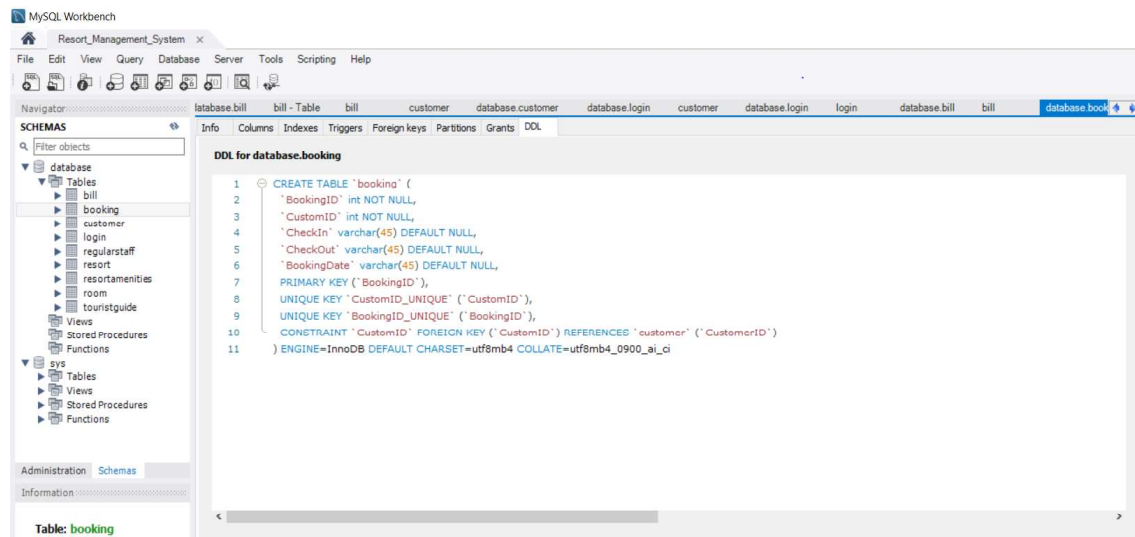
(9, 90, 679, debitcard, 7/10/2022)



The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, query execution, and data management. The main window displays a query result for the 'bill' table. The query is 'SELECT * FROM bill;'. The result grid shows the following data:

	BillID	CustID	Amount	PaymentMethod	PaymentsDate
▶	4	89	568	creditcard	4/10/2022
▶	6	92	700	creditcard	6/10/2022
▶	9	90	679	debitcard	7/10/2022

Booking Table-



The screenshot shows the MySQL Workbench interface with the 'database.booking' table selected. The DDL for the 'booking' table is displayed in the main window:

```
1 CREATE TABLE `booking` (  
2   `BookingID` int NOT NULL,  
3   `CustomID` int NOT NULL,  
4   `CheckIn` varchar(45) DEFAULT NULL,  
5   `CheckOut` varchar(45) DEFAULT NULL,  
6   `BookingDate` varchar(45) DEFAULT NULL,  
7   PRIMARY KEY (`BookingID`),  
8   UNIQUE KEY `CustomID_UNIQUE` (`CustomID`),  
9   UNIQUE KEY `BookingID_UNIQUE` (`BookingID`),  
10  CONSTRAINT `CustomID` FOREIGN KEY (`CustomID`) REFERENCES `customer` (`CustomerID`),  
11 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Inserting records-

INSERT INTO booking(BookingID, CustomID, CheckIn, CheckOut, BookingDate)

VALUES-

(5, 89, 4/7/2022, 4/10/2022,3/25/2022),

(6, 92, 6/1/2022,6/10/2022,5/28/2022),

(7, 90, 7/7/2022,7/10/2022,7/1/2022)

The screenshot shows a database management tool interface. At the top, there are tabs for different databases: 'resortamenities', 'regularstaff', 'database.regularstaff', 'regularstaff', 'touristguide', 'database.touristguide', and 'database.touristguide'. Below the tabs is a toolbar with various icons and a 'Limit to 1000 rows' dropdown. The main area displays a SQL query: `SELECT * FROM booking;`. Below the query, there is a 'Result Grid' showing the results of the query. The grid has five columns: 'BookingID', 'CustomID', 'CheckIn', 'CheckOut', and 'BookingDate'. The results are as follows:

BookingID	CustomID	CheckIn	CheckOut	BookingDate
5	89	4/7/2022	4/10/2022	3/25/2022
6	92	6/1/2022	6/10/2022	5/28/2022
7	90	7/7/2022	7/10/2022	7/1/2022
NULL	NULL	NULL	NULL	NULL

Room Table-

The screenshot shows a database management tool interface. On the left, there is a 'SCHEMAS' panel with a tree view showing the database structure. The 'database' folder is expanded, showing various tables and views. The 'room' table is selected. On the right, there is a 'DDL for database.room' panel showing the DDL for the 'room' table. The DDL is as follows:

```
1 CREATE TABLE `room` (  
2   `ResortID` int NOT NULL,  
3   `BookingID` int NOT NULL,  
4   `RoomNumber` int DEFAULT NULL,  
5   `RoomType` varchar(45) DEFAULT NULL,  
6   `Price` float DEFAULT NULL,  
7   PRIMARY KEY (`ResortID`),  
8   UNIQUE KEY `ResortID_UNIQUE` (`ResortID`),  
9   CONSTRAINT `Res_ID` FOREIGN KEY (`ResortID`) REFERENCES `resort` (`ResortID`)  
10  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

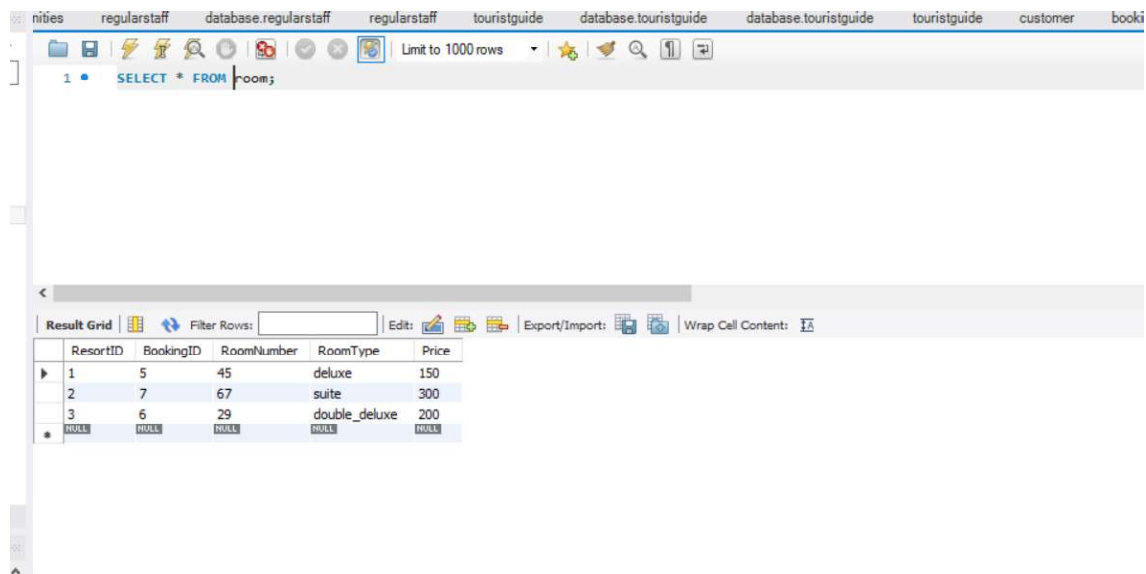
Inserting Records:

INSERT INTO room(ResortID, BookingID,RoomNumber, RoomType, Price) VALUES

(1, 5, 45, deluxe, 150),

(2, 7, 67, suite, 300),

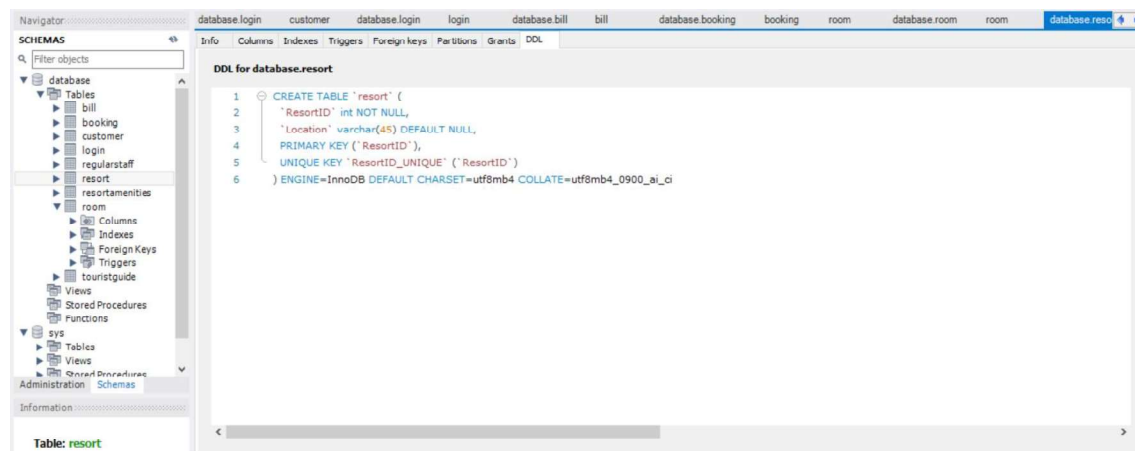
(3, 6, 29, double_deluxe, 200)



The screenshot shows a database query tool interface. At the top, there's a toolbar with various icons and a text input field containing the SQL query: `SELECT * FROM room;`. Below the toolbar, there's a "Result Grid" section. It includes a "Filter Rows:" input field, an "Edit:" button, and an "Export/Import:" button. The main area displays a table with the following data:

ResortID	BookingID	RoomNumber	RoomType	Price
1	5	45	deluxe	150
2	7	67	suite	300
3	6	29	double_deluxe	200
NULL	NULL	NULL	NULL	NULL

Resort table-



The screenshot shows a database tool interface with a "SCHEMAS" panel on the left and a "DDL for database.resort" panel on the right. The "SCHEMAS" panel shows a tree view of the database structure, including tables, columns, indexes, foreign keys, triggers, and views. The "DDL for database.resort" panel displays the following SQL code:

```
1 CREATE TABLE `resort` (  
2   `ResortID` int NOT NULL,  
3   `Location` varchar(45) DEFAULT NULL,  
4   PRIMARY KEY (`ResortID`),  
5   UNIQUE KEY `ResortID_UNIQUE` (`ResortID`)  
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

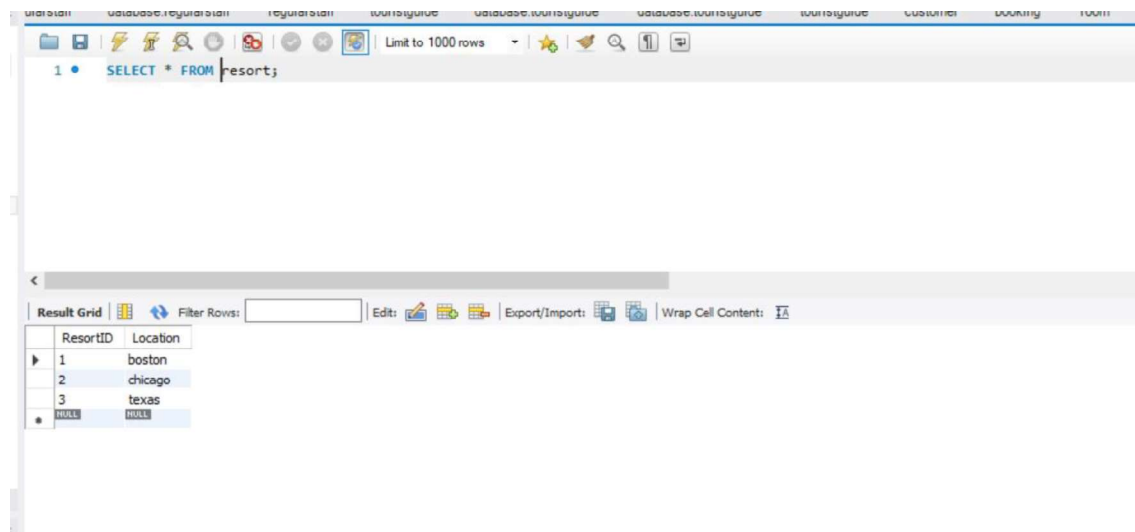
Inserting Records:

INSERT INTO resort(ResortID,Location) VALUES

(1, boston),

(2,chicago),

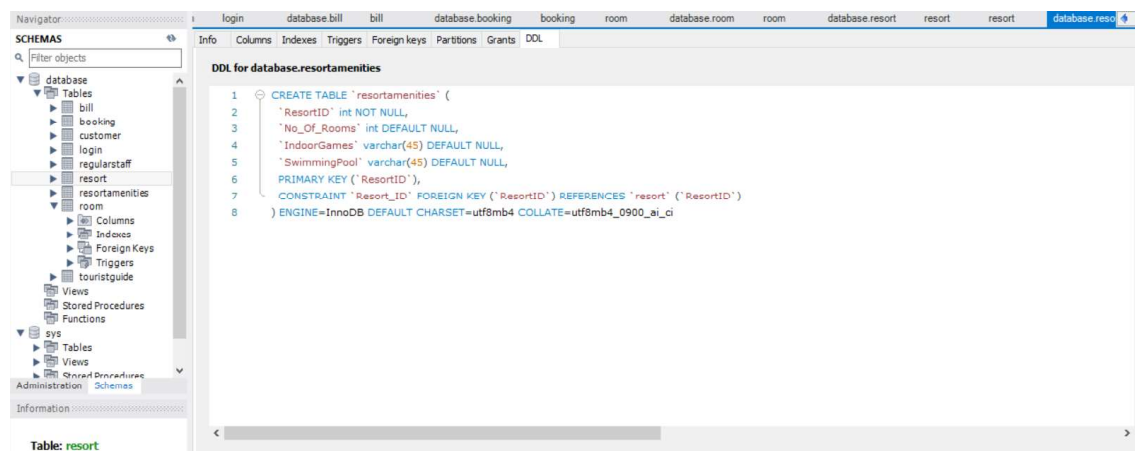
(3,texas)



The screenshot shows a database query tool interface. At the top, a SQL query is entered: `SELECT * FROM resort;`. Below the query, the results are displayed in a table with two columns: `ResortID` and `Location`. The results are as follows:

ResortID	Location
1	boston
2	chicago
3	texas
NULL	NULL

ResortAmenties Table-



The screenshot shows a database tool interface with a schema browser on the left and a DDL editor on the right. The schema browser shows the following structure:

- database
 - Tables
 - bill
 - booking
 - customer
 - login
 - regularstaff
 - resort
 - resortamenities
 - room
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - touristguide
 - Views
 - Stored Procedures
 - Functions
- sys
 - Tables
 - Views
- Administration
 - Schemas

The DDL editor shows the following SQL code for the `resortamenities` table:

```
1 CREATE TABLE `resortamenities` (  
2   `ResortID` int NOT NULL,  
3   `No_Of_Rooms` int DEFAULT NULL,  
4   `IndoorGames` varchar(45) DEFAULT NULL,  
5   `SwimmingPool` varchar(45) DEFAULT NULL,  
6   PRIMARY KEY (`ResortID`),  
7   CONSTRAINT `Resort_ID` FOREIGN KEY (`ResortID`) REFERENCES `resort` (`ResortID`)  
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

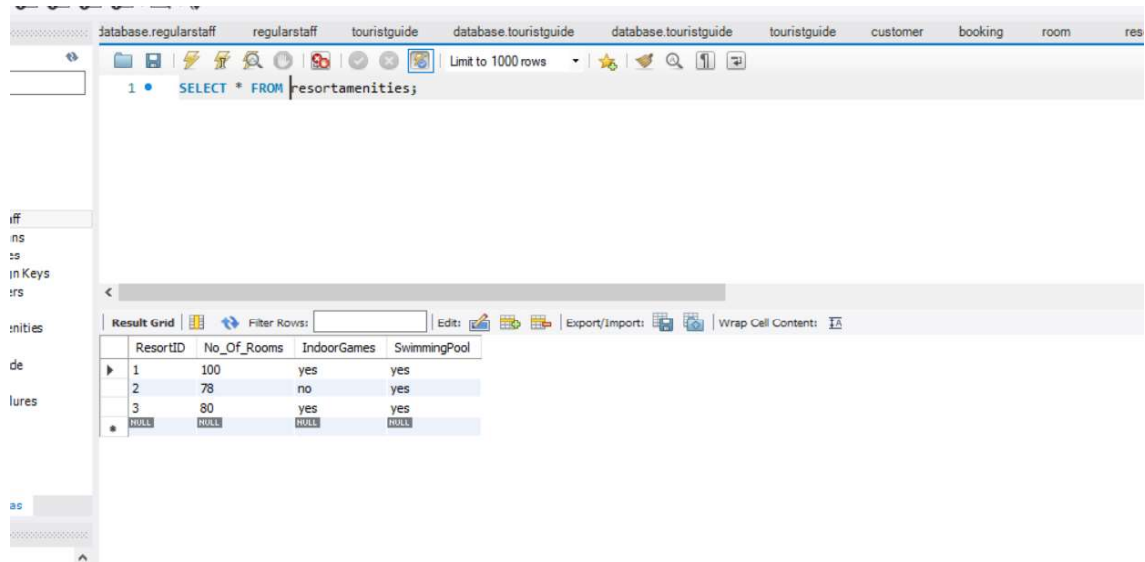
Inserting Records:

INSERT INTO resortamenities(ResortID,No_Of_Rooms,IndoorGames,SwimmingPool) VALUES

(1, 100,yes,yes),

(2,78,no,yes),

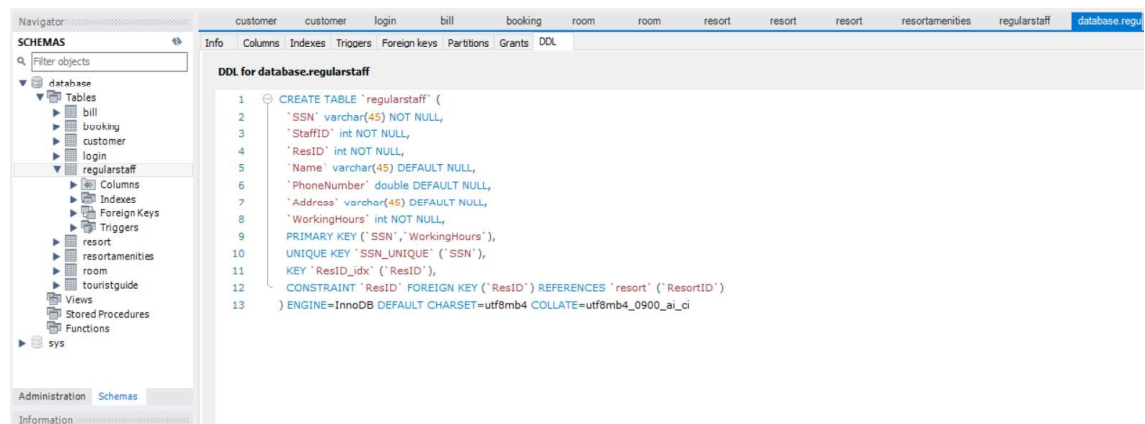
(3,80,yes,yes)



The screenshot shows a database management tool interface. The top menu bar includes 'database.regularstaff', 'regularstaff', 'touristguide', 'database.touristguide', 'database.touristguide', 'touristguide', 'customer', 'booking', 'room', and 'res'. The main query editor displays the SQL statement: `SELECT * FROM resortamenities;`. Below the query editor, the 'Result Grid' shows the following data:

ResortID	No_Of_Rooms	IndoorGames	SwimmingPool
1	100	yes	yes
2	78	no	yes
3	80	yes	yes
* NULL	NULL	NULL	NULL

RegularStaff Table-



The screenshot shows a database management tool interface. The left sidebar displays a tree view of the database schema, including tables like 'bill', 'booking', 'customer', 'login', 'regularstaff', 'resort', 'resortamenities', 'room', 'touristguide', 'Views', 'Stored Procedures', and 'Functions'. The main panel shows the DDL for the 'regularstaff' table:

```
1 CREATE TABLE `regularstaff` (  
2   `SSN` varchar(45) NOT NULL,  
3   `StaffID` int NOT NULL,  
4   `ResID` int NOT NULL,  
5   `Name` varchar(45) DEFAULT NULL,  
6   `PhoneNumber` double DEFAULT NULL,  
7   `Address` varchar(45) DEFAULT NULL,  
8   `WorkingHours` int NOT NULL,  
9   PRIMARY KEY (`SSN`,`WorkingHours`),  
10  UNIQUE KEY `SSN_UNIQUE` (`SSN`),  
11  KEY `ResID_idx` (`ResID`),  
12  CONSTRAINT `ResID` FOREIGN KEY (`ResID`) REFERENCES `resort` (`ResortID`)  
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Inserting Records:

INSERT INTO regularstaff(SSN,StaffID,RestID,Name,PhoneNumber,Address,WorkingHours) VALUES

(AAA_GG_SSSS,56,1,THANYA,9876453445,BOSTON,8),

(AAA_YG_SSNS,67,2,TINA,8764578903,CHICAGO,8),

(AYA_IG_USSS,87,3,NINA,9834257945,TEXAS,9)

SQL query: `SELECT * FROM regularstaff;`

SSN	StaffID	ResID	Name	PhoneNumber	Address	WorkingHours
AAA_GG_SSSS	56	1	THANYA	9876453445	BOSTON	8
AAA_YG_SSNS	67	2	TINA	8764578903	CHICAGO	8
AYA_IG_USSS	87	3	NINA	9834257945	TEXAS	9

TouristGuide Table-

DDL for database.touristguide

```

1 CREATE TABLE `touristguide` (
2   `SSN` varchar(45) NOT NULL,
3   `C_ID` int NOT NULL,
4   `R_ID` int NOT NULL,
5   `Staff_ID` int DEFAULT NULL,
6   `Name` varchar(45) DEFAULT NULL,
7   `PhoneNumber` double DEFAULT NULL,
8   `WorkingHours` int DEFAULT NULL,
9   PRIMARY KEY (`SSN`),
10  UNIQUE KEY `SSN_UNIQUE` (`SSN`),
11  KEY `C_ID_idx` (`C_ID`),
12  KEY `R_ID_idx` (`R_ID`),
13  CONSTRAINT `C_ID` FOREIGN KEY (`C_ID`) REFERENCES `customer` (`CustomerID`),
14  CONSTRAINT `R_ID` FOREIGN KEY (`R_ID`) REFERENCES `resort` (`ResortID`)
15 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

Inserting records –

INSERT INTO touristguide(SSN,C_ID, Staff_ID, Name, PhoneNumber, WorkingHours) VALUES

(AYA_CD_UFVS, 92, 1, 15, Harvey, 9569426945, 9),

(AYA_IG_DFRT, 90, 3, 13, LOUIS, 9894257945, 8),

(BBB_IG_USSS, 89, 2, 12, ROB, 6894211945, 7)

resortresortresortamenitiesregularstaffdatabase.regularstaffregularstafftouristguidedatabase.touri

Limit to 1000 rows

1 • SELECT * FROM touristguide;

Result Grid

Filter Rows:

SSNC_IDR_IDStaff_IDNamePhoneNumberWorkingHours

▶

AYA_CD_UFVS92115HARVEY95694269459

AYA_IG_DFRT90313LOUIS98942579458

BBB_IG_USSS89212ROB68942119457

•

NULLNULLNULLNULLNULLNULL

Wrap Cell Content: