



Wedding Planner System

10.08.2023

Emily Esther Kurian (0759532)
Keya Bhadreshkumar Adhyaru (0762103)
AMOD 5450 (Summer, 2023)
Trent University

Index


1. Introduction
2. List of Entity
3. ER Model
4. Relational Data Model
5. SQL commands to create tables and input data
6. SQL Queries to retrieve data
7. Conclusion

I. Introduction

The Wedding Planner Database is a comprehensive and organized collection of data designed to facilitate the management and organization of weddings. The database serves as a centralized repository of information, offering wedding planners, couples, and vendors the necessary tools to ensure a seamless and memorable wedding experience.

II. List of Entity

- Users: This entity represents the individuals involved in the wedding, such as the bride and groom. It contains their personal details, contact information, and other relevant attributes.
- Bride and Groom: This is a subclass of the "Users" entity, representing the central figures of the wedding. It has additional attributes specific to the bride, such as the bridal shower date and number of bridesmaids.

- 
- Groom: This is a subclass of the "Users" entity, representing the central figures of the wedding. It has additional attributes specific to the groom, such as the number of groomsmen.
 - Vendors: This entity captures information about various categories of wedding vendors, including florists, caterers, photographers, and more. Each vendor is associated with a unique vendor ID.
 - Vendor_list: This entity contains details about vendors, such as their contact information, city, postal code, and portfolio link.
 - Vendor_pic: This weak entity stores images or galleries related to vendors, linked to their respective entries in the "Vendor_list" table.
 - Appointment: This entity records appointments made between users and vendors, specifying the date and time of the appointment.
 - Connections: This entity represents a recursive relationship between users, allowing them to be connected to each other, perhaps representing friendships or relationships.

2. Bride (Subclass of Users):

User_id (Foreign Key to Users table)

Bridesmaid_No

Bridalshower_Date

3. Groom (Subclass of Users):

User_id (Foreign Key to Users table)

Groomsmen_No

4. Vendors:

vendorId (Primary Key)

vendorName

5. Vendor_list:

venList_id (Primary Key)

Vendor_id (Foreign Key to Vendors table)

Contact_num

City

Postal_code

Portfolio_link

Vendor_Name

6. Vendor_pic: (Weak Entity)

venList_id (Primary Key, Foreign Key to Vendor_list table)

gallery

7. Appointment:

Appoint_id (Primary Key)
Vendor_type_id (Foreign Key to Vendors table)
Vendor_id (Foreign Key to Vendor_list table)
Appoin_Date
Appoin_Time
User_id (Foreign Key to Users table)

8. Connections: (Recursive Relationship with Users table)

connection_id (Primary Key)
user1_id (Foreign Key to Users table)
user2_id (Foreign Key to Users table)

V) SQL commands to create tables and input data

```
--  
--  
-- Table structure for table `Users`  
--  
CREATE TABLE `Users` (  
  `User_id` int(11) NOT NULL,  
  `First_name` varchar(50) DEFAULT NULL,  
  `Last_name` varchar(50) DEFAULT NULL,  
  `Contact_num` char(10) DEFAULT NULL,  
  `Email` varchar(100) DEFAULT NULL,  
  `DOB` date DEFAULT NULL,  
  `Wedding_date` date DEFAULT NULL,  
  `Wedding_Location` varchar(100) DEFAULT NULL
```

```

) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

--

-- Dumping data for table `Users`
--

INSERT INTO `Users` (`User_id`, `First_name`, `Last_name`, `Contact_num`,
`Email`, `DOB`, `Wedding_date`, `Wedding_Location`) VALUES
(1, 'Alice', 'Johnson', '1234567890', 'alice@example.com', '1990-05-15',
'2023-09-20', 'Garden Venue A'),
(2, 'John', 'Smith', '9876543210', 'john@example.com', '1988-08-10',
'2023-09-20', 'Garden Venue A'),
(3, 'Emily', 'Davis', '5555555555', 'emily@example.com', '1992-12-20',
'2023-10-05', 'Beach Venue B'),
(4, 'Michael', 'Brown', '9999999999', 'michael@example.com', '1991-06-25',
'2023-11-15', 'Vineyard Venue C'),
(5, 'Sophia', 'Wilson', '1111111111', 'sophia@example.com', '1995-02-28',
'2023-12-30', 'Castle Venue D')

-- -----
--

-- Table structure for table `Vendors`
--

CREATE TABLE `Vendors` (
  `vendorId` int(11) NOT NULL,
  `vendorName` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`vendorId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

--

```

```
-- Dumping data for table `Vendors`
--
INSERT INTO `Vendors` (`vendorId`,`vendorName`) VALUES
(1, 'Flower Shop'),
(2, 'Catering Services'),
(3, 'Photography Studio');

--
-- Table structure for table `Vendor_List`
--

CREATE TABLE `Vendor_List` (
  `venList_id` int(11) NOT NULL,
  `Vendor_id` int(11) DEFAULT NULL,
  `Contact_num` varchar(20) DEFAULT NULL,
  `City` varchar(100) DEFAULT NULL,
  `Postal_code` varchar(10) DEFAULT NULL,
  `Portfolio_Link` varchar(200) DEFAULT NULL,
  `Vendor_Name` varchar(50) DEFAULT NULL,
  PRIMARY KEY (venList_id),
  FOREIGN KEY (vendor_id) REFERENCES Vendors(vendorId)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
```



```
--  
-- Dumping data for table `Vendor_list`  
--  
INSERT INTO `Vendor_list` (`venList_id`, `Vendor_id`, `Contact_num`, `City`,  
`Postal_code`, `Portfolio_Link`, `Vendor_Name`) VALUES  
(1, 1, '111-222-3333', 'Peterborough', '12345', 'www.flowershopa.com', 'Peter The  
Florist'),  
(2, 2, '444-555-6666', 'York', '56789', 'www.cateringservicesa.com', 'Pearl  
catering'),  
(3, 3, '777-888-9999', 'Vaughan', '98765', 'www.photographystudioa.com', 'Moments  
Wedding Pictures'),  
(4, 1, '777-888-9999', 'Kitchner', '58695', 'www.flowershopb.com', 'Blossom  
Boutique'),  
(5, 1, '777-888-9999', 'Mississauga', '48562', 'www.flowershopb.com', 'The flower  
emporium'),  
(6, 2, '777-888-9999', 'Hamilton', '26587', 'www.cateringservicesb.com', 'Chef  
Creations Catering'),  
(7, 3, '777-888-9999', 'Vaughan', '65894', 'www.photographystudiob.com', 'Perfect  
Pixel');  
  
-- Table structure for table `Appointment`  
--  
CREATE TABLE `Appointment` (  
  `Appoint_id` int(11) NOT NULL,  
  `Vendor_type_id` int(11) DEFAULT NULL,  
  `Vendor_id` int(11) DEFAULT NULL,  
  `Appoin_Date` date DEFAULT NULL,  
  `Appoin_Time` time DEFAULT NULL,  
  `User_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (Appoint_id),
```

```
FOREIGN KEY (Vendor_type_id) REFERENCES Vendor_list(venList_id),
FOREIGN KEY (Vendor_id) REFERENCES Vendors(vendorId),
FOREIGN KEY (User_id) REFERENCES Users(User_id)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
--
-- Dumping data for table `Appointment`
--
INSERT INTO `Appointment` (`Appoint_id`, `Vendor_type_id`, `Vendor_id`,
`Appoin_Date`, `Appoin_Time`, `User_id`) VALUES
(1, 1, 1, '2023-08-10', '10:00:00', 1),
(2, 2, 2, '2023-08-15', '12:30:00', 2),
(3, 3, 3, '2023-08-20', '15:00:00', 1);
--
-- Table structure for table `Vendor_pic`
--
CREATE TABLE `Vendor_pic` (
  `venList_id` int(11) NOT NULL,
  `gallery` varchar(200) DEFAULT NULL,
  PRIMARY KEY (venList_id),
  FOREIGN KEY (venList_id) REFERENCES Vendor_list(venList_id)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;
--
-- Dumping data for table `Vendor_pic`
```

```
--  
  
INSERT INTO `Vendor_pic` (`venList_id`, `gallery`) VALUES  
(1, '/home/user/vendor/vendor1'),  
(2, '/home/user/vendor/vendor2'),  
(3, '/home/user/vendor/vendor3');  
  
-- -----  
  
--  
  
-- Table structure for table `Bride`  
--  
  
CREATE TABLE `Bride` (  
  `User_id` int(11) NOT NULL,  
  `Bridesmaid_No` int(11) DEFAULT NULL,  
  `Bridalshower_Date` date DEFAULT NULL,  
  PRIMARY KEY (User_id),  
  FOREIGN KEY (User_id) REFERENCES Users(User_id)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;  
  
--  
  
-- Dumping data for table `Bride`  
--  
  
INSERT INTO `Bride` (`User_id`, `Bridesmaid_No`, `Bridalshower_Date`) VALUES  
(1, 4, '2023-08-25'),  
(3, 10, '2023-12-15'),  
(5, 6, '2023-09-10');
```

```
-----  
  
--  
-- Table structure for table `Connections`  
--  
  
CREATE TABLE `Connections` (  
  `connection_id` int(11) NOT NULL,  
  `user1_id` int(11) DEFAULT NULL,  
  `user2_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (connection_id),  
  FOREIGN KEY (user1_id) REFERENCES Users(User_id),  
  FOREIGN KEY (user2_id) REFERENCES Users(User_id)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;  
  
--  
-- Dumping data for table `Connections`  
--  
  
INSERT INTO `Connections` (`connection_id`, `user1_id`, `user2_id`) VALUES  
(1, 1, 2);  
  
-----  
  
--  
-- Table structure for table `Groom`
```

```
--  
  
CREATE TABLE `Groom` (  
  `User_id` int(11) NOT NULL,  
  `Groomsmen_No` int(11) DEFAULT NULL,  
  PRIMARY KEY (User_id),  
  FOREIGN KEY (User_id) REFERENCES Users()User_id  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;  
  
--  
  
-- Dumping data for table `Groom`  
  
--  
  
INSERT INTO `Groom` (`User_id`, `Groomsmen_No`) VALUES  
(2, 5),  
(4, 10);  
  
-----
```

Vi. SQL queries to retrieve data:

1. Three queries for one table

a. Retrieve users with a specific last name:

```
SELECT * FROM Users WHERE Last_name = 'Davis';
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	User_id	First_name	Last_name	Contact_num	Email	DOB	Wedding_date	Wedding_location
<input type="checkbox"/> Edit Copy Delete	3	Emily	Davis	5555555555	emily@example.com	1992-12-20	2023-10-05	Beach Venue B

b. Retrieve users who have 'gmail' as their email provider:

```
SELECT * FROM `Users` where Email LIKE '%gmail.com';
```

Extra options

	User_id	First_name	Last_name	Contact_num	Email	DOB	Wedding_date	Wedding_location
<input type="checkbox"/> Edit Copy Delete	4	Michael	Brown	9999999999	michael@gmail.com	1991-06-25	2023-11-15	Vineyard Venue C
<input type="checkbox"/> Edit Copy Delete	5	Sophia	Wilson	1111111111	sophia@gmail.com	1995-02-28	2023-12-30	Castle Venue D

c. Retrieve users who have their weddings after 1st October, 2023:

```
SELECT * FROM `Users` where Wedding_date > '2023-10-01';
```

	User_id	First_name	Last_name	Contact_num	Email	DOB	Wedding_date	Wedding_location
<input type="checkbox"/> Edit Copy Delete	3	Emily	Davis	5555555555	emily@example.com	1992-12-20	2023-10-05	Beach Venue B
<input type="checkbox"/> Edit Copy Delete	4	Michael	Brown	9999999999	michael@gmail.com	1991-06-25	2023-11-15	Vineyard Venue C
<input type="checkbox"/> Edit Copy Delete	5	Sophia	Wilson	1111111111	sophia@gmail.com	1995-02-28	2023-12-30	Castle Venue D

☐ Check all | With selected: Edit Copy Delete Export

2. Three queries for two table

a. Retrieve all the information of brides only:

```
SELECT u.*, b.*
FROM Users u
JOIN Bride b ON u.User_id = b.User_id;
```

User_id	First_name	Last_name	Contact_num	Email	DOB	Wedding_date	Wedding_location	User_id	Bridesmaid_No	Bridalshower_Date
1	Alice	Johnson	1234567890	alice@example.com	1990-05-15	2023-09-20	Garden Venue A	1	4	2023-08-25
3	Emily	Davis	5555555555	emily@example.com	1992-12-20	2023-10-05	Beach Venue B	3	10	2023-12-15
5	Sophia	Wilson	1111111111	sophia@gmail.com	1995-02-28	2023-12-30	Castle Venue D	5	6	2023-09-10

☐ Show all | Number of rows: 25 ▼ | Filter rows:

b. Retrieve brides and their bridesmaid count:

```
SELECT u.First_name, u.Last_name, b.Bridesmaid_No FROM Users u JOIN Bride b ON
u.User_id = b.User_id;
```

Extra options

First_name	Last_name	Bridesmaid_No
Alice	Johnson	4
Emily	Davis	10
Sophia	Wilson	6

- c. Retrieve all appointments along with the corresponding user's first name and last name:

```
SELECT a.Appoint_id, a.Appoin_Date, a.Appoin_Time, u.First_name, u.Last_name
FROM Appointment a
JOIN Users u ON a.User_id = u.User_id;
```

Extra options

Appoint_id	Appoin_Date	Appoin_Time	First_name	Last_name
1	2023-08-10	10:00:00	Alice	Johnson
2	2023-08-15	12:30:00	John	Smith
3	2023-08-20	15:00:00	Alice	Johnson

3. Three queries for three table

- a. Retrieve all appointments with the corresponding user and vendor information:

```
SELECT a.Appoint_id, u.First_name, u.Last_name, v.vendorName, vl.Vendor_Name
a.Appoin_Date, a.Appoin_Time
FROM Appointment a
JOIN Users u ON a.User_id = u.User_id
JOIN Vendor_list vl ON a.Vendor_id = vl.venList_id
JOIN Vendors v ON vl.Vendor_id = v.vendorId;
```

Appoint_id	First_name	Last_name	vendorName	Vendor_Name	Appoin_Date	Appoin_Time
1	Alice	Johnson	Flower Shop	Peter The Florist	2023-08-10	10:00:00
2	John	Smith	Catering Services	Pearl catering	2023-08-15	12:30:00
3	Alice	Johnson	Photography Studio	Moments Wedding Pictures	2023-08-20	15:00:00

☐ Show all | Number of rows: 25 ▼ | Filter rows: Search this table | Sort by key: None ▼

b. Retrieve brides' information along with their appointment details:

```
SELECT u.First_name, u.Last_name, u.Email, a.Appoin_Date, a.Appoin_Time
FROM Users u
JOIN Bride b ON u.User_id = b.User_id
JOIN Appointment a ON u.User_id = a.User_id;
```

First_name	Last_name	Email	Appoin_Date	Appoin_Time
Alice	Johnson	alice@example.com	2023-08-10	10:00:00
Alice	Johnson	alice@example.com	2023-08-20	15:00:00

c. Retrieve all appointments with user and vendor details ordered by appointment date in ascending order:

```
SELECT a.Appoint_id, a.Appoin_Date, a.Appoin_Time, u.First_name AS User_FirstName,
u.Last_name AS User_LastName, v.Vendor_Name, v.City
FROM Appointment a
JOIN Users u ON a.User_id = u.User_id
JOIN Vendor_list v ON a.Vendor_id = v.venList_id
ORDER BY a.Appoin_Date ASC;
```

Appoint_id	Appoin_Date ▲ 1	Appoin_Time	User_FirstName	User_LastName	Vendor_Name	City
1	2023-08-10	10:00:00	Alice	Johnson	Peter The Florist	Peterborough
2	2023-08-15	12:30:00	John	Smith	Pearl catering	York
3	2023-08-20	15:00:00	Alice	Johnson	Moments Wedding Pictures	Vaughan

Vii. Conclusion:

Virtual Wedding planner helps users to explore vendors for their big day. They can view the portfolio of the vendors and further can book the appointment to discuss things. These database involves eight different entities which have normal relationship, weak relationship and recursive relationship with each other. The report showcases the ER diagram of the database. Later, the attributes and key constraints are also described in the document. SQL queries to create tables and then inserting data into tables is presented. Lastly, several queries are performed to extract desired data followed by screenshots of the output.