Concert Attendance Database

Chad McGuire

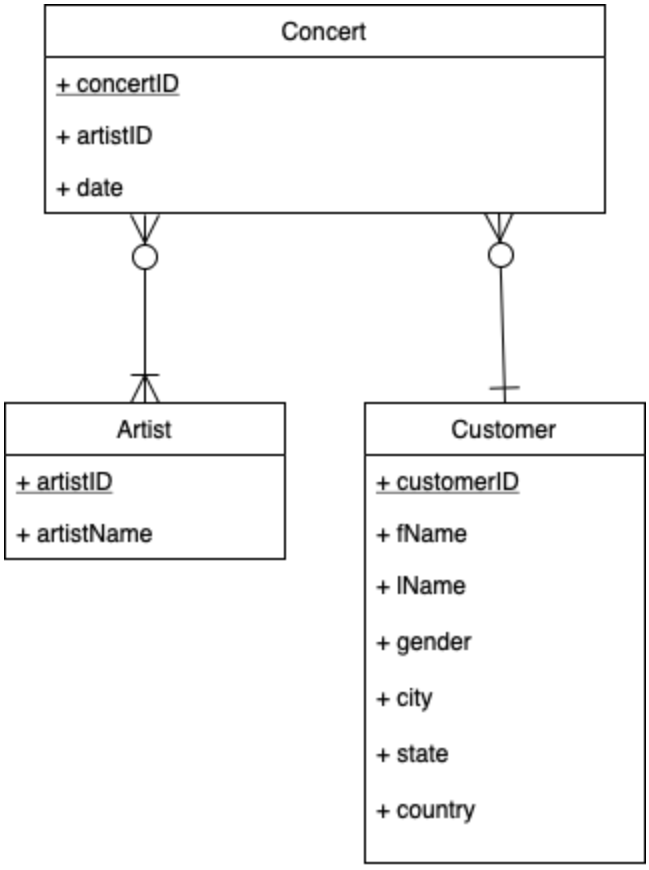
1. **Project Title:**

   System for the Tracking of Concert Attendance

2. **Outline:**

   This database stores information pertaining to concerts at a particular venue. The database stores the name of the artist performing and the date of the concert and then provides data relating to the fans in attendance. This database could be used by the concert venue or the artists to review data from previous events in order to better tailor marketing strategies for future performances.

3. **ER Diagram:**



4. **Table Schemas:**

   Artist (artistID, artistName)
   Concert (concertID, artistID, date)
   Customer (customerID, fName, lName, gender, city, state, country)
   ConcertCustomerIntersection (concertID, customerID)

5. **Explanation to Demonstrate Normalization:**

   The tables above are normalized and fit all discussed normal forms. The tables are in 1NF as they have no non-atomic values, 2NF as they have no partial key dependencies, 3NF as they have no transitive dependencies, and BCNF as every determinant is a candidate key. There are no multi-valued dependencies, so the tables are not concerned with 4NF and 5NF.

6. **DDL To Create Tables:**

```
CREATE TABLE Artist (
artistID INT NOT NULL AUTO_INCREMENT,
artistName CHAR(40) NOT NULL,
PRIMARY KEY (artistID)
);

CREATE TABLE Concert (
concertID INT NOT NULL AUTO_INCREMENT,
artistID INT NOT NULL,
date DATE NOT NULL,
PRIMARY KEY (concertID)
);

ALTER TABLE Concert
ADD FOREIGN KEY fk_artistID(artistID)
REFERENCES Artist(artistID)
ON DELETE CASCADE ON UPDATE CASCADE;
```

```
CREATE TABLE Customer (
customerID INT NOT NULL AUTO_INCREMENT,
fName CHAR(20) NOT NULL,
lName CHAR(20) NOT NULL,
gender ENUM('m','f'),
city CHAR(20),
state CHAR(20),
country CHAR(20),
PRIMARY KEY (customerID)
);

CREATE TABLE ConcertCustomerIntersection (
concertID INT NOT NULL,
customerID INT NOT NULL,
PRIMARY KEY (concertID, customerID)
);

ALTER TABLE ConcertCustomerIntersection
ADD FOREIGN KEY fk_concertID(concertID)
REFERENCES Concert(concertID)
ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE ConcertCustomerIntersection
ADD FOREIGN KEY fk_concertID(concertID)
REFERENCES Concert(concertID)
ON DELETE CASCADE ON UPDATE CASCADE;
```

7. Instructions for Using the System:

a.) INSERTS:

Inserting data to Artist:
(NOTE TO USER: to look insert different artists → change names in quotes as shown below)

```
INSERT INTO Artist (artistName)
VALUES
('Kanye West'),
('Ed Sheeran'),
('Khalid'),
('Chance the Rapper')
('Travis Scott'),
('Chris Stapleton'),
('Luke Bryan'),
('Drake');
```

Inserting data to Concert:
(NOTE TO USER: to add different dates → change artistID (which must correlate to inserted artists) and dates as shown below)

```
INSERT INTO Concert (artistID, date)
VALUES
(1, '2019-08-02'),
(2, '2019-08-03'),
(3, '2019-08-09'),
(4, '2019-08-10'),
(5, '2019-08-16'),
(6, '2019-08-17'),
(7, '2019-08-23'),
(8, '2019-08-24');
```

Inserting data to Customer:

(NOTE TO USER: to add different customers → change customer info and enter in format shown below)

```
INSERT INTO Customer (fName, lName, gender, city, state, country)
VALUES
('Bob', 'Jones', 'm', 'Anchorage', 'Alaska', 'USA'),
('Adam', 'Smith', 'm', 'Montgomery', 'Alabama', 'USA'),
('Mary', 'Heinz', 'f', 'Little Rock', 'Arkansas', 'USA'),
('Frank', 'Johnson', 'm', 'San Francisco', 'California', 'USA'),
('Chad', 'McGuire', 'm', 'Tallahassee', 'Florida', 'USA'),
('Hannah', 'Jones', 'f', 'Denver', 'Colorado', 'USA'),
('Linda', 'Zola', 'f', 'Atlanta', 'Georgia', 'USA'),
('Ella', 'Fernandez', 'f', 'Honolulu', 'Hawaii', 'USA'),
('Stephen', 'Parker', 'm', 'Chicago', 'Illinois', 'USA'),
('Wei', 'Li', 'm', 'Beijing', NULL, 'China'),
('Juan', 'De Soto', 'm', 'Lima', NULL, 'Peru'),
('Miguel', 'Garcia', 'm', 'La Romana', NULL, 'Dominican Republic'),
('Michael', 'Longworth', 'm', 'London', NULL, 'England'),
('Emma', 'Michaels', 'f', 'Boston', 'Massachusetts', 'USA');
```

Inserting data to ConcertCustomerIntersection:

(NOTE TO USER: to add different customers to different concerts → enter concertID first, then customerID in same format as below)

```
INSERT INTO ConcertCustomerIntersection
VALUES
(2, 13),
(3, 12),
(1, 9),
(8, 8),
(5, 14),
(7, 10),
(4, 3),
(6, 11),
(7, 11),
```

```
(8, 1),
(8, 2),
(1, 4),
(3, 5),
(4, 9),
(7, 7),
(4, 5),
(2, 6);
```

b.) QUERIES (WHOLE TABLES)

Query entire Artist table:

```
SELECT *
FROM Artist;
```

```
+----------+------------------+
| artistID | artistName       |
+----------+------------------+
|        1 | Kanye West       |
|        2 | Ed Sheeran       |
|        3 | Khalid           |
|        4 | Chance the Rapper |
|        5 | Travis Scott     |
|        6 | Chris Stapleton  |
|        7 | Luke Bryan       |
|        8 | Drake            |
+----------+------------------+
8 rows in set (0.00 sec)
```

Query entire Concert table:

```
SELECT *
FROM Concert;
```

```
+-----------+----------+------------+
| concertID | artistID | date       |
+-----------+----------+------------+
|         1 |        1 | 2019-08-02 |
|         2 |        2 | 2019-08-03 |
|         3 |        3 | 2019-08-09 |
|         4 |        4 | 2019-08-10 |
|         5 |        5 | 2019-08-16 |
|         6 |        6 | 2019-08-17 |
```

```
|          7 |         7 | 2019-08-23 |
|          8 |         8 | 2019-08-24 |
+------------+-----------+------------+
8 rows in set (0.00 sec)
```

Query entire Customer table:

```
SELECT *
FROM Customer;
```

```
+------------+---------+-----------+--------+---------------+--------------+--------------------+
| customerID | fName   | lName     | gender | city          | state        | country            |
+------------+---------+-----------+--------+---------------+--------------+--------------------+
|          1 | Bob     | Jones     | m      | Anchorage     | Alaska       | USA                |
|          2 | Adam    | Smith     | m      | Montgomery    | Alabama      | USA                |
|          3 | Mary    | Heinz     | f      | Little Rock   | Arkansas     | USA                |
|          4 | Frank   | Johnson   | m      | San Francisco | California   | USA                |
|          5 | Chad    | McGuire   | m      | Tallahassee   | Florida      | USA                |
|          6 | Hannah  | Jones     | f      | Denver        | Colorado     | USA                |
|          7 | Linda   | Zola      | f      | Atlanta       | Georgia      | USA                |
|          8 | Ella    | Fernandez | f      | Honolulu      | Hawaii       | USA                |
|          9 | Stephen | Parker    | m      | Chicago       | Illinois     | USA                |
|         10 | Wei     | Li        | m      | Beijing       | NULL         | China              |
|         11 | Juan    | De Soto   | m      | Lima          | NULL         | Peru               |
|         12 | Miguel  | Garcia    | m      | La Romana     | NULL         | Dominican Republic |
|         13 | Michael | Longworth | m      | London        | NULL         | England            |
|         14 | Emma    | Michaels  | f      | Boston        | Massachusetts | USA               |
+------------+---------+-----------+--------+---------------+--------------+--------------------+
14 rows in set (0.00 sec)
```

c.) SPECIFIC QUERIES AND DATA LOOKUPS

List all artists performing:

```
SELECT artistName
FROM Artist;
```

```
+-------------------+
| artistName        |
+-------------------+
| Kanye West        |
| Ed Sheeran        |
| Khalid            |
| Chance the Rapper |
| Travis Scott      |
| Chris Stapleton   |
| Luke Bryan        |
| Drake             |
+-------------------+
8 rows in set (0.00 sec)
```

List artist's name and the date they are performing:

```
SELECT a.artistName, c.date
FROM Artist AS a JOIN Concert AS c
ON a.artistID = c.artistID;
```

```
+------------------+------------+
| artistName       | date       |
+------------------+------------+
| Kanye West       | 2019-08-02 |
| Ed Sheeran       | 2019-08-03 |
| Khalid           | 2019-08-09 |
| Chance the Rapper | 2019-08-10 |
| Travis Scott     | 2019-08-16 |
| Chris Stapleton  | 2019-08-17 |
| Luke Bryan       | 2019-08-23 |
| Drake            | 2019-08-24 |
+------------------+------------+
8 rows in set (0.00 sec)
```

List the number of people in attendance on each date:

```
SELECT Concert.date, COUNT(Customer.customerID) AS attendance
FROM Concert
JOIN ConcertCustomerIntersection ON Concert.concertID =
ConcertCustomerIntersection.concertID
JOIN Customer ON Customer.customerID =
ConcertCustomerIntersection.customerID
GROUP BY Concert.date;
```

```
+------------+------------+
| date       | attendance |
+------------+------------+
| 2019-08-02 |          2 |
| 2019-08-03 |          2 |
| 2019-08-09 |          2 |
| 2019-08-10 |          3 |
| 2019-08-16 |          1 |
| 2019-08-17 |          1 |
| 2019-08-23 |          3 |
| 2019-08-24 |          3 |
+------------+------------+
8 rows in set (0.00 sec)
```

Check diagnostics on where each person at specific concert is from
(in this case ‘Kanye West’ concert):
(NOTE TO USER: to look at a different artist → change artistName in quotes)

```
SELECT Customer.city, Customer.state, Customer.country
FROM Customer
JOIN ConcertCustomerIntersection ON Customer.customerID =
ConcertCustomerIntersection.customerID
JOIN Concert ON Concert.concertID =
ConcertCustomerIntersection.concertID
JOIN Artist ON Artist.artistID = Concert.artistID
WHERE Artist.artistName = 'Kanye West';
```

```
+---------------+------------+---------+
| city          | state      | country |
+---------------+------------+---------+
| San Francisco | California | USA     |
| Chicago       | Illinois   | USA     |
+---------------+------------+---------+
2 rows in set (0.00 sec)
```

Check diagnostics on number of each gender at specific concert
(in this case ‘Luke Bryan’):
(NOTE TO USER: to look at a different artist → change artistName in quotes)

```
SELECT Customer.gender, COUNT(*) AS genderCount
FROM Customer
JOIN ConcertCustomerIntersection ON Customer.customerID =
ConcertCustomerIntersection.customerID
JOIN Concert ON Concert.concertID =
ConcertCustomerIntersection.concertID
JOIN Artist ON Artist.artistID = Concert.artistID
WHERE Artist.artistName = 'Luke Bryan'
GROUP BY Customer.gender;
```

```
+--------+-------------+
| gender | genderCount |
+--------+-------------+
| m      |           2 |
| f      |           1 |
+--------+-------------+
2 rows in set (0.00 sec)
```

d.) DELETES

Delete an artist and their concert by artistName:
(NOTE TO USER: to delete a different artist → change artistName in quotes)

```
DELETE FROM Artist
WHERE artistName = 'Ed Sheeran';

Query OK, 1 row affected (0.00 sec)
```

Delete a concert by date:
(NOTE TO USER: to delete a different concert → change date in quotes)

```
DELETE FROM Concert
WHERE date = '2019-08-02';

Query OK, 1 row affected (0.00 sec)
```

Delete a customer by fName and lName:
(NOTE TO USER: to delete a different customer → change fName and lName in quotes)

```
DELETE FROM Customer
WHERE fName = 'Emma' AND lName = 'Michaels';

Query OK, 1 row affected (0.00 sec)
```