# **Phase One**

# **Data and Functional Requirements for the COMPANY Database**

* 1. **Data Requirements**
* The music store database has Customers, each identified by a unique Username, Password, First\_Name, Last\_Name, City and Zip\_Code.
* The music store consists of a unique Albums, Artist, Soundtracks, and Order.
* Each album consists of a unique Album\_ID, Artist, Release\_Date, Length, Genre, Cost, and Soundtrack.
* Each Artist has a unique Name, Soundtrack, and Album. It can be assumed that some Artist can have on multiple albums.
* Each Soundtrack consists of a unique soundtrackID, soundtrack\_Name, Artist, Length, Release-Date, and Cost. It can be assumed that some soundtrack can have multiple artist.
* Each order consists of a unique OrderID, Username, Date, Price and Items-total. It can be assumed that the customer can order an album or multiple albums.
  1. **Functional Requirements**
     1. **Update Requirements**

1. Each customer can search for an Album from the database
2. Each customer can purchase an Album, from the database.
3. Each customer can add or remove an album from the order bag.
4. Each customer can create orders.
5. Each customer can create an account and the database can store the user’s authentication information, last name, first name, zip code and preferred payment method.
   * 1. **Retrieval Requirements**

….

1. The database should allow user to reset authentication information.
2. The database should allow user to add, delete and update the customer information.
3. The database should allow user to add, delete and update information about the album.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Phase Two**

# **Conceptual COMPANY Database Design using the ER Data Model**

**2.1 Initial Conceptual Design (ICD) of the COMPANY Database Initial Conceptual Design (ICD) of the online Music Store Database**

CUSTOMERS (Username, Password, FirstName, LastName, City, Zip-Code)

MUSIC-STORE (Albums, Artist, Soundtracks, Order)

ALBUM (Album\_ID, Artist, Release-Date, Length, Genre, Cost, Soundtrack

ARTIST (Name, Soundtrack, Album)

ORDER (OrderID, CustomerID, Date, Price, Item-Total)

SOUNDTRACK (Artist, Cost, Release-Date, Name, Length)

**2.3 Individual ER Diagram of the company DB**

ORDERS

ALBUM

CUSTOMERS

m

makes

1

Purchase\_items

n

m

n

contains

Has

ARTIST

1

Has

m

1

Contains

Featured-artist  
“  
|==

]  
|=[

\/}

\

]

]

m m m

SOUNDTRACK  
  
|==

]  
|=[

\/}

\

]

]

n

Contains

* 1. **Assumptions**

….

* Some Soundtracks can have multiple artist.
* Some Artist can be on multiple albums
* A customer can order a single album or multiple albums

# **Phase Three**

# **Relational COMPANY Database Schema Design using the Relational Data Model**

**3.1**

CUSTOMERS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Username | password | City | FirstName | LastName | Zip-code |

ORDERS

|  |  |  |  |
| --- | --- | --- | --- |
| orderID | userName | Date | Total\_price |

ALBUMS-ORDER

|  |  |
| --- | --- |
| orderID | albumID |

ARTIST

|  |  |  |
| --- | --- | --- |
| name | Soundtrack | Album |

ALBUMS

|  |  |  |  |
| --- | --- | --- | --- |
| album\_id | name | release-date | cost |

SOUNDTRACK

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| soundtrackID | Artist-name | length | Release-date | Soundtrack\_name | albumID |

FEATURED-ARTIST

|  |  |
| --- | --- |
| soundtrackID | name |

* 1. **Update Requirements**

1. Each customer can search for an Album from the database

Assuming search word = “Spicy”

SELECT \* FROM ALBUMS WHERE NAME LIKE “%Spicy%”

1. Each customer can purchase an Album, from the database.

INSERT INTO

1. Each customer can add or remove an album from the order.

AFTER GETTING THE ORDER\_ID: ASSUMING Order\_id = 123 & Album\_id = 321

ADDING ALBUM:

INSERT INTO ALBUMS\_ORDERED (Order\_id, Album\_id) VALUES (123, 321);

DELETING ALBUM:

DELETE FROM ALBUMS\_ORDERED WHERE Order\_id = 123 AND Album\_id = 321;

1. Each customer can create orders.

ADD ORDER: ASSUMING ORDER\_ID IS AUTO\_GENERATED

INSERT INTO ORDERS (Username, Date, Final\_price) VALUES (“Some User”, “29/05/20 12:30 PM”, “13.50”);

REMOVE FROM ORDER: ASSUMING ORDER\_ID = 123

DELETE FROM ORDERS WHERE OrderID = 123;

1. Each customer can create an account and the database can store the user’s authentication information, last name, first name, zip code and preferred payment method.

INSERT INTO CUSTOMERS (Username, password, city, firstName, lastName, zipCode) VALUES (“Keneisha”, “12345678”, “Wiggan”, “Kay”, “Neisha”, 09402);

**3.4 Retrieval Requirements**

….

1. The database should allow user to reset password information.

UPDATE CUSTOMERS SET password = 1234 WHERE username = “Keneisha”;

1. The database should allow user to add, delete and update the customer information.

ADD: SAME AS UPDATE QUESTION 6

DELETE: DELETE FROM CUSTOMERS WHERE username = “Keneisha”;

\* If we’re updating City, it’ll be like:

UPDATE: UPDATE CUSTOMERS SET city = “Orangeburg” WHERE username = “Keneisha”;

1. The database should allow user to add, delete and update information about the album.

“SAME OPERATIONS AS 8 EXCEPT” CUSTOMERS will be ALBUMS “AND” WHERE username = “Keneisha” will be WHERE Album\_ID = 1 (“Some ID”)