Name: Sunkara Lakshmi Deepa

INFO 5707 Project Report

**Management System for Online Music Store**

**General Description**

The Online Music Store comprises of different types of music from all over the world.The Management System is needed in order to create a database for online music store due to the increasing demand for music from customer’s side. The database provides the accessibility to the customers where it will help them especially in choosing their genre. The online music store consists around 45,000 music items where the customers can buy them through the site and additionally they can know about the music things which are displayed in a site.

**User Requirements**

While developing a database, it is important to know about the users requirements. In this particular database the user require the details of the product he purchased through online ( Tracking details ), also he requires an email or contact support in order to enquire about the product information, availability of the product and etc. He must be provided with various payment options and etc.

**Business rules**

We must consider the following business rules

1. A login system is required in order to have Users ID, because by having this attribute we can able to access other details about users information (satisfies one to many rule)
2. A user can buy ‘N’ number of items.
3. User address information is required like (name, email address, telephone number and mailing address)
4. Sometimes the shipping address might be different hence we also require the delivery address.
5. An Album Id is also required in order to find out the detail like (title, cast, date of release and no of album copies available)
6. Order ID is also required to track the purchase item, and related details.

**Major Entities**

**PAYMENT DETAILS:** In this table we will have the information regarding (payment\_id (primary Key), order\_id(Foreign key), payment\_type, price\_of\_item, discount\_applied, finalprice\_of\_item)

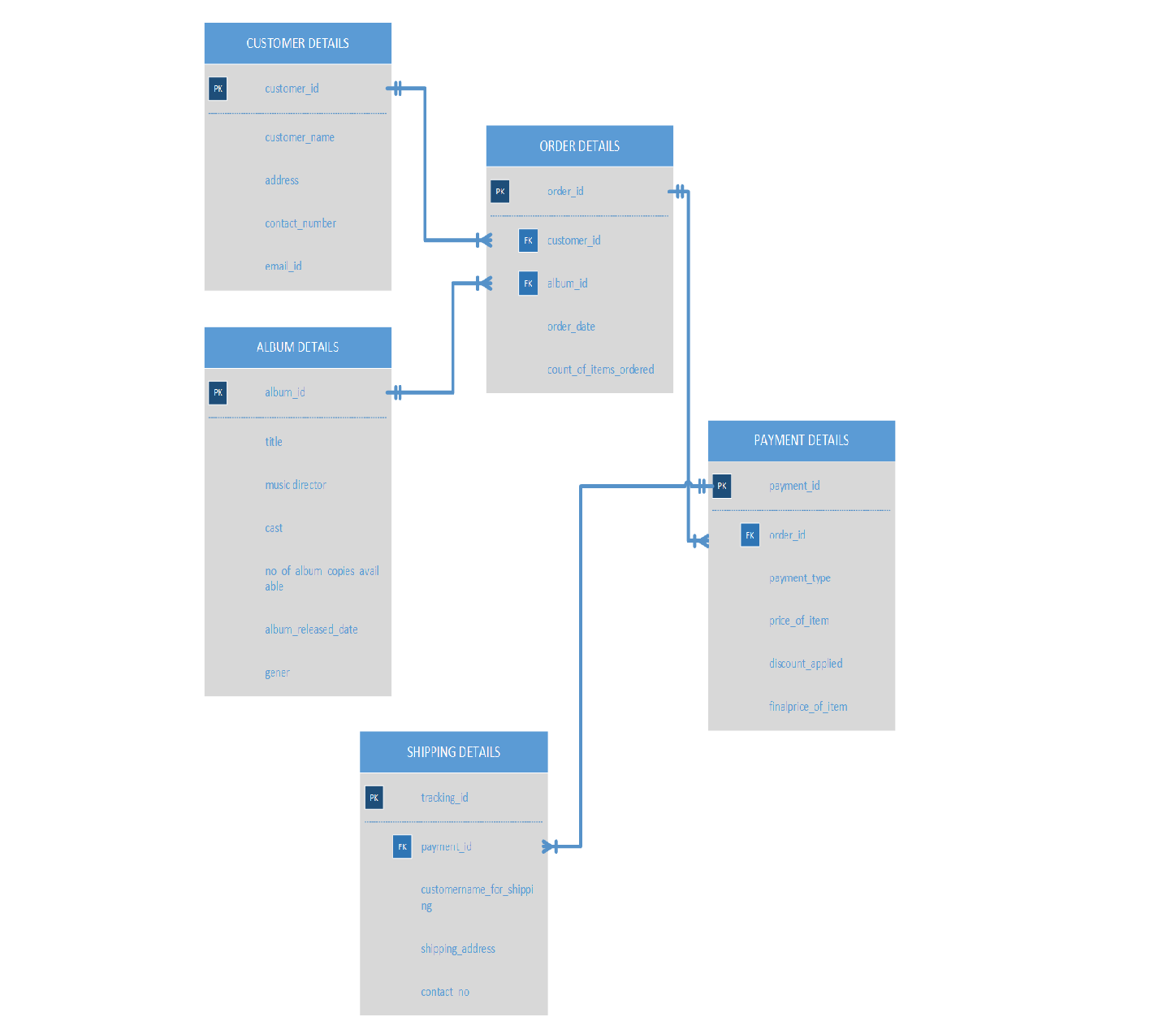
**ORDER DETAILS:** In this table we have the details of the customer's order like ( order\_id(primary key), customer\_id(foreign key), album\_id(foreign key), order\_date, count\_of\_items\_ordered)

**CUSTOMER DETAILS**: Here we have all the customer information like ( customer\_id(Primary key), customer\_name, address, contact\_number, email\_id)

**ALBUM DETAILS:** This table consists of the album details like( album\_id(Primary key), title, music director, cast, no\_of\_album\_copies\_available, album\_released\_date, gener)

**SHIPPING DETAILS**: The table of shipping details consists details like ( tracking\_id(Primary key), payment\_id(Foreign key), customername\_for\_shipping, shipping\_address, contact\_no)

**Entity Relationship Diagram(ER Diagram):**

****

**Data Dictionary**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TABLE NAME** | **ATTRIBUTENAME** | **CONTENTS** | **TYPE** | **FORMAT** | **REQUIRED** | **PK/FK** | **FK REFERENCE TABLE** |
| **CUSTOMER DETAILS** | **customer\_id** | **Customer’s ID Number** | **INT** | **###** | **Yes** | **PK** |  |
|  | **customer\_name** | **Customer’s Name** | **VARCHAR (30)** | **Xxxxxx** | **Yes** |  |  |
|  | **address** | **Customer’s Address** | **VARCHAR (60)** | **Xxxxxxxxxx** | **Yes** |  |  |
|  | **contact\_number** | **Customer’s**  **Number** | **INT** | **##########** | **Yes** |  |  |
|  | **email\_id** | **Customer’s Email ID** | **VARCHAR (30)** | **xxxx@xxx.xx** | **Yes** |  |  |
| **ALBUM DETAILS** | **album\_id** | **Albums’s ID Number** | **INT** | **####** | **Yes** | **PK** |  |
|  | **title** | **Album's Title** | **VARCHAR (30)** | **Xxxxxxx** | **Yes** |  |  |
|  | **music director** | **Album’s Music Director** | **VARCHAR (60)** | **Xxxxxxx** | **Yes** |  |  |
|  | **cast** | **Album’s cast names** | **VARCHAR (60)** | **Xxxxxxx** | **Yes** |  |  |
|  | **no\_of\_album\_copies\_available** | **Album copies** | **INT** | **###** | **Yes** |  |  |
|  | **album\_released\_date** | **Album release year** | **DATE** |  | **Yes** |  |  |
|  | **gener** | **Album Gener Details** | **VARCHAR (30)** | **Xxxxxx** | **Yes** |  |  |
| **ORDER DETAILS** | **order\_id** | **Order ID Number** | **INT** | **####** | **Yes** | **PK** |  |
|  | **customer\_id** | **Customer ID Number** | **INT** | **###** | **Yes** | **PK/FK** | **CUSTOMER DETAILS** |
|  | **album\_id** | **Album ID Number** | **INT** | **####** | **Yes** | **PK/FK** | **ALBUM DETAILS** |
|  | **order\_date** | **Ordered Item Date** | **DATE** |  | **Yes** |  |  |
|  | **count\_of\_items** | **No Of Ordered Items** | **INT** | **###** | **Yes** |  |  |
| **PAYMENT DETAILS** | **payment\_id** | **Payment ID Number** | **INT** | **####** | **Yes** | **PK** |  |
|  | **order\_id** | **Order ID Number** | **INT** | **####** | **Yes** | **PK/FK** | **ORDER DETAILS** |
|  | **payment\_type** | **Type of the Payment** | **VARCHAR (30)** | **Xxxxxxxx** | **Yes** |  |  |
|  | **price\_of\_item** | **Price of product** | **INT** | **####** | **Yes** |  |  |
|  | **discount\_applied** | **Discount percentage** | **INT** | **##%** | **Yes** |  |  |
|  | **finalprice\_of\_item** | **Price after Discount** | **INT** | **####** | **Yes** |  |  |
| **SHIPPING DETAILS** | **tracking\_id** | **Tracking ID Number** | **INT** | **#######** | **Yes** | **PK** |  |
|  | **payment\_id** | **Payment ID Number** | **INT** | **####** | **Yes** | **PK/FK** | **PAYMENT DETAILS** |
|  | **customername\_for\_shipping** | **Shipping Name** | **VARCHAR (30)** | **Xxxxxx** | **Yes** |  |  |
|  | **shipping\_address** | **Shipping Address** | **VARCHAR (60)** | **Xxxxxx** | **Yes** |  |  |
|  | **contact\_no** | **Shipping Mobile Number** | **INT** | **##########** | **Yes** |  |  |

**SQL Queries to create MCPL tables**

CREATE TABLE CUSTOMER DETAILS (

customer\_id INT NOTNULL UNIQUE,

customer\_name VARCHAR(30) NOTNULL,

address VARCHAR(60) NOTNULL,

contact\_number INT NOTNULL,

email\_id VARCHAR(30) NOTNULL,

PRIMARY KEY (customer\_id));

CREATE TABLE ALBUM DETAILS (

album\_id INT NOTNULL UNIQUE,

title VARCHAR(30) NOTNULL,

music director VARCHAR(60) NOTNULL,

cast VARCHAR(60) NOTNULL,

no\_of\_album\_copies\_available INT NOTNULL,

album\_released\_date DATE NOTNULL,

PRIMARY KEY( album\_id));

CREATE TABLE ORDER DETAILS (

order\_id INT NOTNULL UNIQUE,

customer\_id INT NOTNULL,

album\_id INT NOTNULL,

order\_date DATE NOTNULL,

count\_of\_items INT NOTNULL,

PRIMARY KEY(order\_id),

FOREIGN KEY(customer\_id) REFERENCES CUSTOMER DETAILS,

FOREIGN KEY(album\_id) REFERENCES ALBUM DETAILS);

CREATE TABLE PAYMENT DETAILS (

payment\_id INT NOTNULL UNIQUE,

order\_id INT NOTNULL,

payment\_type VARCHAR(30) NOTNULL,

price\_of\_item INT NOTNULL,

discount\_applied INT NOTNULL,

finalprice\_of\_item INT NOTNULL,

PRIMARY KEY(payment\_id),

FOREIGN KEY(order\_id) REFERENCES ORDER DETAILS);

CREATE TABLE SHIPPING DETAILS (

tracking\_id INT NOTNULL UNIQUE,

payment\_id INT NOTNULL,

customername\_for\_shipping VARCHAR(30) NOTNULL,

shipping\_address VARCHAR(60) NOTNULL,

contact\_no INT NOTNULL,

PRIMARY KEY (tracking\_id),

FOREIGN KEY(payment\_id) REFERENCES PAYMENT DETAILS);

**Sample queries**

**1. Mention the album names whose genre is equal to "Pop" or "Juke" ?**

SELECT [ALBUM DETAILS].title, [ALBUM DETAILS].gener FROM [ALBUM DETAILS] WHERE ((([ALBUM DETAILS].gener)="Pop" Or ([ALBUM DETAILS].gener)="Juke"));

**2. In which year "Silento" albums were released?**

SELECT [ALBUM DETAILS].title, [ALBUM DETAILS].[music director], [ALBUM DETAILS].album\_released\_date FROM [ALBUM DETAILS] WHERE ((([ALBUM DETAILS].[music director])="silento"));

**3. Mention the Customer names who got 0% discount on their purchase of album?**

SELECT [CUSTOMER DETAILS].customer\_name, [PAYMENT DETAILS].discount\_applied FROM ([CUSTOMER DETAILS] INNER JOIN [ORDER DETAILS] ON [CUSTOMER DETAILS].customer\_id = [ORDER DETAILS].customer\_id) INNER JOIN [PAYMENT DETAILS] ON [ORDER DETAILS].order\_id = [PAYMENT DETAILS].order\_id WHERE ((([PAYMENT DETAILS].discount\_applied)="0%"));

**4. According to the mentioned Tracking\_ID find out the order details?**

SELECT [SHIPPING DETAILS].tracking\_id, [CUSTOMER DETAILS].customer\_name, [ALBUM DETAILS].title, [ORDER DETAILS].order\_date FROM (([CUSTOMER DETAILS] INNER JOIN ([ALBUM DETAILS] INNER JOIN [ORDER DETAILS] ON [ALBUM DETAILS].album\_id = [ORDER DETAILS].album\_id) ON [CUSTOMER DETAILS].customer\_id = [ORDER DETAILS].customer\_id) INNER JOIN [PAYMENT DETAILS] ON [ORDER DETAILS].order\_id = [PAYMENT DETAILS].order\_id) INNER JOIN [SHIPPING DETAILS] ON [PAYMENT DETAILS].payment\_id = [SHIPPING DETAILS].payment\_id WHERE ((([SHIPPING DETAILS].tracking\_id)=[Enter Tracking ID]));

**Sample report**

1. The Album Details Report consists of all the Details related to the ALBUM present in the Online Music Store.

2. The Customer Details Report consists details of Online Music store Customers and also details about Customers who recently purchase albums from online music store.

3. The Order details report lists the order details and when the order is placed and count of the no of items purchased by the customer.

4. The payment details report present information on how the purchase is been made by the customer. ( Whether it is Online Payment or Cash On deliver).

5. Next the shipping details report gives the tracking details of the customer purchased order.

6. Finally the relationship diagram report represents the various relationships that exists between different tables.