

**Introduction to Database System**

**Fall 2023**

# **Mid Term Lab**

# **VERSION I**

**Time Allowed: 1.5 hours**

|  |  |
| --- | --- |
| Name: |  |
| Registration No |  |
| Serial Number (of attendance sheet) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ERD | DML | DDL | Total |
| Total Marks | 15 | 20 | 20 | 55 |
| Obtained Marks |  |  |  |  |

**Instructions:**

1. **This is a closed book, closed notes paper.**
2. **Understanding questions is part of the paper**. Therefore, no queries will be entertained during examination.
3. Schema file is uploaded on teams/portal. Use those files to attempt the DML section of paper.
4. Use proper indentation/formatting while writing queries. Not properly indenting will deduct 5% marks.
5. **You need to make an MS Word with your name and registration should be mentioned on each page.**
6. **You need to write only queries with output in MS Word file, you need to write query (text form) + its output table (picture) if any.**
7. **You will solve the DDL + DML section only in word docx.**

# **Part 1 ERD** **[10 Marks]**

**Design an Entity-Relationship Diagram (ERD) for an online bookstore database. The primary entities in this database are Books, Authors, Customers, Orders, and OrderDetails. (Marks=15)**

Starting with the **Books** entity, each book is uniquely identified by a BookID and contains information such as Title, ISBN (International Standard Book Number), Price, and Quantity in Stock. The relationship between Books and Authors is modeled as a Many-to-Many relationship since an author can write multiple books, and a book can have multiple authors.

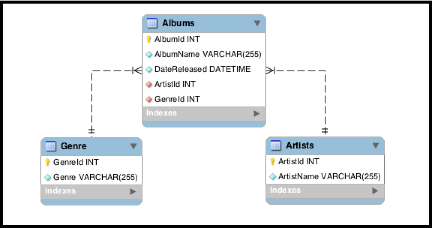
The **Authors** entity represents individuals who contribute to the creation of books. Each author is identified by an AuthorID and has attributes including FirstName, LastName, and Email. The Many-to-Many relationship between Books and Authors is captured through an associative table.

Moving on to the **Customers** entity, individuals who interact with the bookstore are represented. A Customer is identified by a CustomerID and includes details like FirstName, LastName, Email, Address, and Phone. The relationship between Customers and Orders is established as a One-to-Many relationship since a customer can place multiple orders, but each order is placed by a single customer.

The **Orders** entity signifies each transaction made by a customer. An Order is identified by an OrderID and includes a reference to the Customer who placed the order and the OrderDate. The One-to-Many relationship with OrderDetails reflects that an order can have multiple associated order details, but each order detail belongs to a specific order.

Finally, the **OrderDetails** entity captures the specifics of each product included in an order. It has attributes such as DetailID, Quantity, and Total Price (calculated as Price \* Quantity). The relationships with Books and Authors are defined to maintain the integrity of the database structure.

# **Part 2 DDL** **[20 Marks]**



**Q1.** Implement the above tables with its attributes. Make Primary Key in both tables.

**[10 Marks]**

**Q2.** Make a relation between Artists and Albums, Genre and Artist using alter command.

**[5 Marks]**

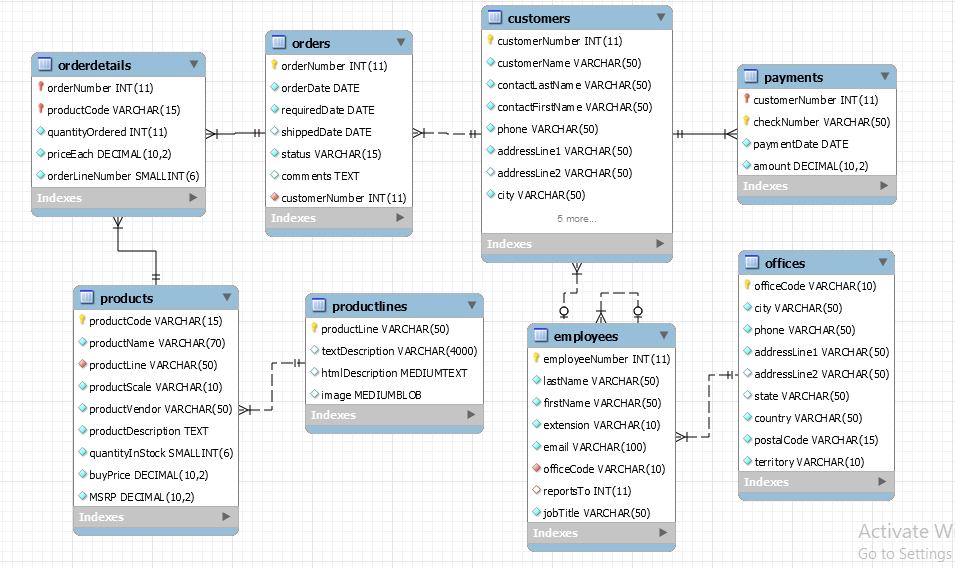
Note:

* ArtistID in Albums table, should be foreign key from Artists table.
* GenreID in Albums table, should be foreign key from Genre table.

**Q3.** Change the variable name of AlbumName to Name and datatype from varchar (255) to char in Albums table. **[5 Marks]**Note: Do these in one Query only.

**Part 3 DML** **[20 Marks]**

* **Import ClassicModel.sql schema.**



**Solve the following questions from the above schema:**

1. Retrieve every entry in the 'employees' table where the 'First\_name' field is composed of exactly five letters. (Like)
2. Show all products where product quantity is greater than 4000. (>)
3. Extract all data from a database where the individuals hold the position of 'sales representative'.
4. List every client name in the 'customers' table whose associated country is either France, Norway, or Denmark.
5. Show following columns CustomerNumber, Name and phone of "Customers" where country is NOT "USA".