Rent_items-:

	Field_Name	Type (Size)	Constraints
1	rent_id	Int	Primary Key
2	Rent_date	Datetime	NOT NULL
3	Album_id	Int	Foreign Key
4	Quantity	Int	NOT NULL
5	Client_id	Int	Foreign Key
6	Return_Date	Datetime	NOT NULL
7	current_status	Varchar(45)	NOT NULL

PRIMARY KEY:

rent_id is the primary key which is unique and not null. This column is to uniquely identify each of the other columns in the table.

FOREIGN KEY:

1. Album_id : The album table has a one to many relationship with this table. Only the album id present in the original table can be present in the order_items table.
2. Client_id: The clients table has a one to many relationship with this table.Only clients present in the clients table can be included in the order_items table.

CONSTRAINTS:

1. rent_date:

   There should be an rent date for each of the entries. It cannot be null.

2. Return_date:

   There should be a return date for each of the entries. It cannot be null.

3. Quantity:

   The amount of items that has been rented for a particular album. It cannot be null. Any change in this column will effect the product_remaining column.

4. Current_status:

   If the item that has been rented has been returned, it's due or overdue. Cannot be null.

TRIGGERS:

1. trigger_new_client_rent_items:

   This trigger is used to see if the client that has already rented wants to rent again. If he does then it will prompt to delete the row for the client first and then insert into the table.

2. trigger_rent_product_remaining:

   This trigger is used to see if the client wants to rent an item where the quantity of items wanted exceeds the quantity of items available. If so then it will not insert into the table and throw and exception. Otherwise it will insert and go to the album table and change the total amount.

PROCEDURES:

3. procedure_rent_expiry:

   This is a procedure that checks if a product has been rented has exceeded it's return date and still haven't been returned. If it's due then it will be changed to overdue.

4. procedure_return_rented:

   This is a stored procedure that takes an rent_id as an input. Goes to the rent_table and change the status to returned if it's due or overdue.

Song-:

	Field_Name	Type (Size)	Constraints
1	Song_id	Int	Primary Key

2	Album_id	Int	Foreign Key
3	Song_Rating_id	Int	Foreign Key
4	Genre_id	Int	Foreign Key
5	Song_name	Varchar(45)	NOT NULL
6	Singer_name	Varchar(45)	NOT NULL

Primary Key:

Song_id is the primary key to uniquely identify the columns of the table.

Foreign Key:

Album_id: albums table has a one to many relationship with the song table. One album can contain many songs.

Song_rating_id:

Song ratings id has a one to one relationship with the song table. Each song has a song_rating. Song_rating can be null.

Genre_id :

Genre table has one to many relationship with song table. One genre can have many songs in it.

Constraint:

Song_name:

Name of the song for a particular album. It cannot be null.

Singer_name:

Name of the singer of the song. It cannot be null.