TECHNICAL REPORT

DATABASE DESIGN AND MAINTENACE

Contents

Introduction

This database application is for our customer who owns music store in which thy sell music albums and songs to the customers. This app will provide them an analysis of data stored in form of records in different tables like customers, employee , users , artist , product and many more .

This application will provide safe and secure data structure system for users so they can store all their records safely and use this application to survey about their business activities .

The challenging thing to make this application was to decide table columns and implement tables in a way that we can give users an easy-to-use application and yet effective when it came to use data for business.

This application is also used to give users feature to get information about their profit and loss , most selling products reports so they can make changes in their business where it requires .

General Scenario

1.Display a list of clients that spent more than the average spent by client in the past month.

- We have made invoice table which shows the details like Items purchased, invoice date, customer (client), amount total, with use of this information (invoice date, customer, amount) we have implemented a query which can display the list of customers who s pent more than average in any period here in our case it's month.

2. The top sold products and least sold products over a week.

- We have made table for items sold in invoices and made the relation with invoice table in the form of one to many. This tables contains columns like item_id, item name , quantity , price which is useful to get the information like number of quantities sold and apart from that to get the number of products sold in a week we have used the relation between invoice items table and invoice table to get the invoice date. With use of invoice date, we can get the number of products sold in a week .

3. The maximum price of products in the same genre (for example, rock, pop, country, hip-hop). Use GROUP BY to list all the genres and their maximum price.

- The products table is representing music albums and single songs this table contains information like price, name, artist , song genre (category of product) which is stored in another table category and category table is in relation with product table .So, with the use of above information, we have displayed maximum price from each product category or genre which fulfills this requirement.

4. List how many customers the system has by location (Country, Province, and City), and then sort them.

- It is important to know information about customers to built good relationship with them, so we have made customers table with data like their location info like pin code , city , province , country .with the use of above information we have added a feature for our user to sort customers with location data .

5. List how many products the store has sold for a particular month.

- We have implemented the invoice table for each sales of product inside music store .This invoice table contains relation with invoice items table with data like product name , quantity , price . so, with using invoice date from invoice and using product quantity sold from invoice items we have displayed numbers of product sold in a particular month .

6. List how many distinct albums each singer has.

- For each product (albums , single songs) a relation with artist table is made so with the use of this information we have displayed the number of products is in relation with each artist and it will fulfill this requirement.

7. List how many copies of an album are available of a particular singer.

- For each album there is a filed called stock quantity in table of products and each album is attached with an artist or singer table so with this above data we have displayed stock quantity for each album with group by with singer.

Specific Scenarios

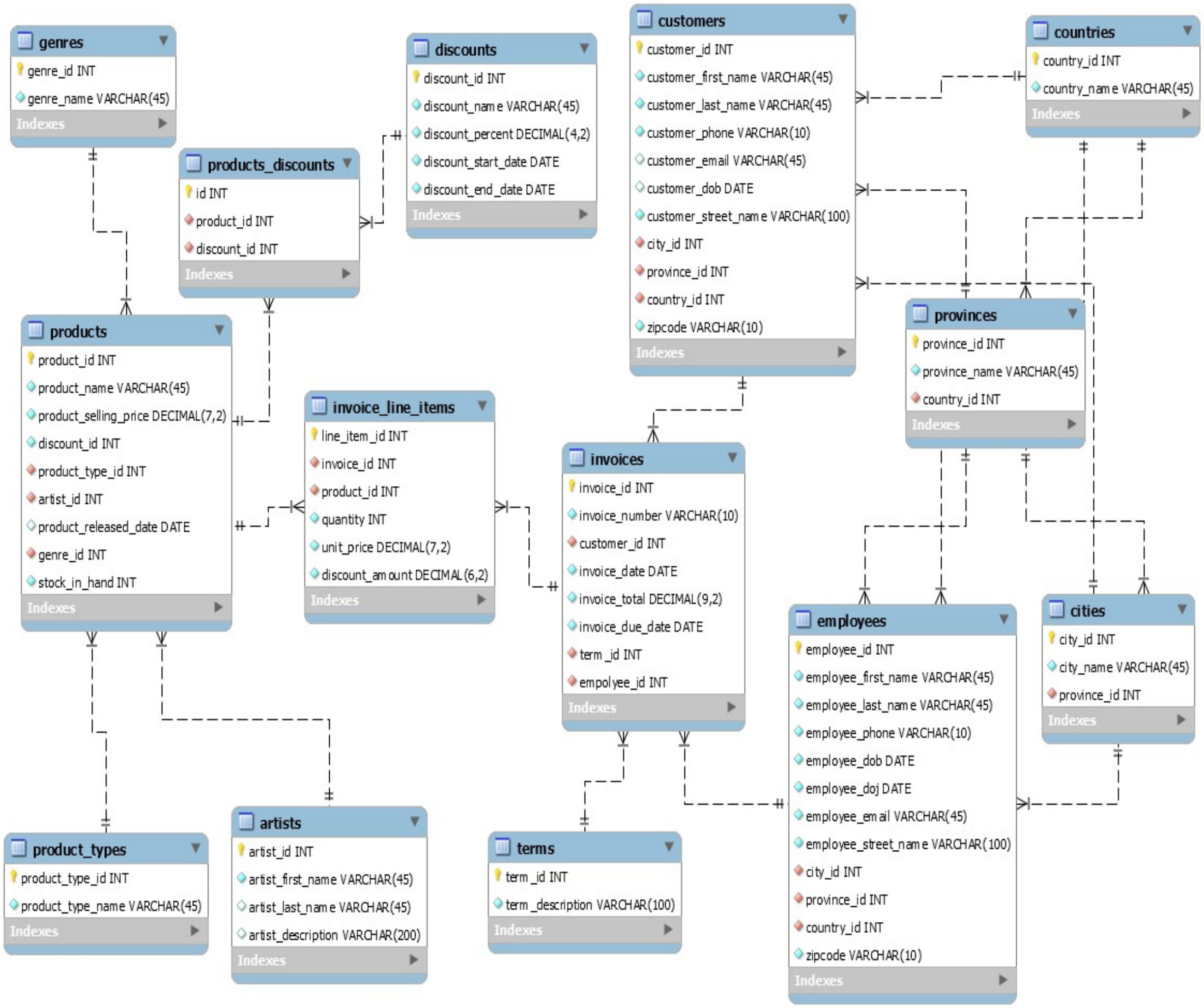
1.Top Trending Artist/s of the month .

- This feature displays the top trending artist of the month to decide this we have selected artist whom albums have been sold highest by the music store in the period on a month.

2. (Upcoming Birthdays) List all customers whose having their birthday in next month.

- It's important to build a good relationship with customers in every business. Customer always finds the discounts and sale attractive, So to provide a good offers on their birth day we have built a feature which
  Displays the upcoming birthdays of customers in next month.
- To add this feature, we have created scheduler which makes the list of customers whom birthday is in upcoming month, we have added customers birthday field in customers table to establish this feature.

Documentation



- This model diagram is created as based on user's requirement and some of the other relations and columns has been added by us to make sure we can add some more features in the future as well for users .

Details about each table in database :

1. **Customers** : Provides users to have information about their customers .
2. **Products** : Stores records for the albums and single songs into database .
3. **Invoices** : Stores all the billing details about sale orders for the music store .
4. **Artists** : Stores details about artists and the albums made by the artists .
5. **Employee** : This table stores all the information about employees working at music store
6. **Product type** : This table used to classify the albums and singles from the product table
7. **Invoice line items**: This table is related to invoice table and it’s used to store multiple products sold to the customers in invoices.
8. **Terms** : Stores all the payments terms for the customers invoices .

More details about tables are explained in appendix.

Conclusion

In conclusion our application stores the information about music store business .Customers plays the major role in this business ; they can buy many types of albums and single songs from music store. This app can get all the information about customer invoices and users can do analysis related to top selling products and other info like profit and loss .Our app also keep track on the customers interests like top trending songs and albums in period of times which will be helpful for our users to manage and plan business activities according to that .

This application is created based on requirement from our client , we have also provided two more features which can be useful for our clients .So this app will provide easy and effective access to the database of the music sore.

Appendix

Table breakdown

NN: Not Null

AI: Auto Incremental

UQ: Unique

Table Name:	**Countries**	
Description	a table created to store information related to countries.	
Columns Detail	**Type**	**Description**
country_name	VARCHAR(45) NN	The column country_name describes the name of the countries.
Primary Key	**Type**	**Description**
country_id	INT NN AI	created to give unique id to countries

Table Name:	**Provinces**	
Description	a table created to store information related to provinces in respective countries.	
Columns Detail	**Type**	**Description**
province_name	VARCHAR(45) NN	The column province_name describes the name of the provinces.

Primary Key	**Type**	**Description**
province_id	INT NN AI	created to give unique id to provinces.
Foreign Keys	**Type**	**Description**
country_id	INT NN	identifies the country id.

Table Name:	**Cities**	
Description	a table created to store information related to cities in the provinces.	
Columns Detail	**Type**	**Description**
city_name	VARCHAR(45) NN	The column city_name describes the name of the city.
Primary Key	**Type**	**Description**
city_id	INT NN AI	created to give unique id to cities.
Foreign Keys	**Type**	**Description**
province_id	INT NN	identifies the country id.

Table Name:	**Customers**	
Description	a table created to store information related to customers.	
Columns Detail	**Type**	**Description**
customer_first_name	VARCHAR(45) NN	First Name of the customer.
customer_last_name	VARCHAR(45) NN	Last Name of the customer.

customer_phone	VARCHAR(10) NN	Phone Number of the customer.
customer_email	VARCHAR(45) NN	Email ID of the customer.
customer_street_name	VARCHAR(45) NN	Street Address of the customer.
zipcode	VARCHAR(10) NN	Address zip code of the customer.
customer_dob	DATE NN	Date of birth of the customer.
Primary Key	**Type**	**Description**
customer_id	INT NN AI	Gives unique id to customers.
Foreign Keys	**Type**	**Description**
country_id	INT NN	identifies the country id.
province_id	INT NN	identifies the province id.
city_id	INT NN	identifies the city id.

Table Name:	**Product_types**	
Description	a table stores information whether a product is a Album or a Single.	
Columns Detail	**Type**	**Description**
product_type_name	VARCHAR(45) NN	Name of type of product
Primary Key	**Type**	**Description**
product_type_id	INT NN AI	Gives unique id to product types.

Table Name:	**Artists**	
Description	a table created to store information related to artists.	
Columns Detail	**Type**	**Description**
artist_first_name	VARCHAR(45) NN	First Name of the artist.
artist_last_name	VARCHAR(45) NN	Last Name of the artist.
artist_description	VARCHAR(200) NN	A small information about the artist.
Primary Key	**Type**	**Description**
artist_id	INT NN AI	Gives unique id to artists.

Table Name:	**Genres**	
Description	a table stores information related to genres type of the album or single.	
Columns Detail	**Type**	**Description**
genre_name	VARCHAR(45) NN	Name of the genre.
Primary Key	**Type**	**Description**
genre_id	INT NN AI	Gives unique id to genres.

Table Name:	**Products**
Description	a table created to store information related to products.

Columns Detail	**Type**	**Description**
product_name	VARCHAR(45) NN	Name of the product.
product_selling_price	DECIMAL(7,2) NN	Selling price of the product.
product_released_date	DATE NN	Release Date of the product.
stock_in_hand	INT NN	Inventory of the product.
Primary Key	**Type**	**Description**
product_id	INT NN AI	Gives unique id to products.
Foreign Keys	**Type**	**Description**
product_type_id	INT NN	identifies the product type id.
genre_id	INT NN	identifies the genre id.
artist_id	INT NN	identifies the artist id.

Table Name:	**Terms**	
Description	a table stores information agreed terms.	
Columns Detail	**Type**	**Description**
term_description	VARCHAR(100) NN	Information about the term whether the accepted term is 10/30/60 days.
Primary Key	**Type**	**Description**
term_id	INT NN AI	Gives unique id to terms.

Table Name:	**Employees**	
Description	a table created to store information related to employees.	
Columns Detail	**Type**	**Description**
Employee_first_name	VARCHAR(45) NN	First Name of the employee.
employee_last_name	VARCHAR(45) NN	Last Name of the employee.
employee_phone	VARCHAR(10) NN	Phone Number of the employee.
employee_email	VARCHAR(45) NN	Email ID of the employee.
employee_street_name	VARCHAR(45) NN	Street Address of the employee.
zipcode	VARCHAR(10) NN	Address zip code of the employee.
employee_dob	DATE NN	Date of birth of the employee.
Primary Key	**Type**	**Description**
employee_id	INT NN AI	Gives unique id to employees.
Foreign Keys	**Type**	**Description**
country_id	INT NN	identifies the country id.
province_id	INT NN	identifies the province id.
city_id	INT NN	identifies the city id.

Table Name:	**Invoices**	
Description	a table created to store information related to invoices.	
Columns Detail	**Type**	**Description**
invoice_number	VARCHAR(45) NN	Invoice Number of the purchase.
invoice_date	DATE NN	Purchase Date of the product.
invoice_total	DECIMAL(9,2) NN	Total Cost of the invoice.
invoice_due_date	DATE NN	Due Date of the invoice.
Primary Key	**Type**	**Description**
invoice_id	INT NN AI	Gives unique id to invoices.
Foreign Keys	**Type**	**Description**
customer_id	INT NN	identifies the customer id.
term_id	INT NN	Term ID of the invoice.
empolyee_id	INT NN	ID of the employee who entered the invoice data.

Table Name:	**invoice_line_items**	
Description	a table created to store information related to invoice items.	
Columns Detail	**Type**	**Description**
quantity	INT NN	Quantity of the item.
unit_price	DECIMAL(7,2) NN	Price of a single item.
discount_amount	DECIMAL(6,2) NN	Amount of discount that was appiled.
Primary Key	**Type**	**Description**
line_item_id	INT NN AI	Gives unique id to line items.
Foreign Keys	**Type**	**Description**
invoice_id	INT NN	identifies the invoice id of invoice line items.
product_id	INT NN	identifies the product id of invoice line items.

Table Name:	**discounts**	
Description	a table created to store information related to discounts.	
Columns Detail	**Type**	**Description**
discount_name	VARCHAR(45) NN	Name of the discount.
discount_start_date	DATE NN	Start Date of the discount.
discount_end_date	DATE NN	End Date of the discount.
discount_percent	DECIMAL(9,2) NN	Amount of discount that can be applied.

Primary Key	**Type**	**Description**
discount_id	INT NN AI	Gives unique id to discounts.

Table Name:	**products_discounts**	
Description	a table created to link products to discounts.	
Primary Key	**Type**	**Description**
id	INT NN AI	Gives unique id to products discounts relation.
Foreign Keys	**Type**	**Description**
discount_id	INT NN	identifies the discount id linked to the product.
product_id	INT NN	identifies the product id linked to the discount.