**BUAN 6320.009 – Database Foundations for Business Analytics - F22**

**PROJECT REPORT- GROUP 11**

**Art Gallery Management System**

**Logo

Description automatically generated**

Submitted By:

Chikitha Reddy

Gautam Iyer

Poorwa Kunwar

Rohan Shivakumar Kalyanshettar

Sai Rakshith Chithreddy

Submitted to

Professor K Srikanth

December 12, 2022

Executive Summary:

This project focuses on building an art gallery database management system. By creating a database for an art gallery with 17 tables encompassing data related to artwork inventory and business transactions, an accurate data model was built using MySQL 8.0 Workbench. The project highlights the use of automated processes to fetch data efficiently through the use of queries, joins and stored programs (functions, procedures and triggers) so as to holistically manage and enhance current business requirements.

Business Proposal:

We are a company that offers services to small and medium scale businesses to manage their data more precisely and allow them to get real-time updates about their data. We specialize in creating databases for businesses and training them on how to manage it. With a driven team of problem solvers, we believe in giving the best of services to our clients and enhancing their current business processes.

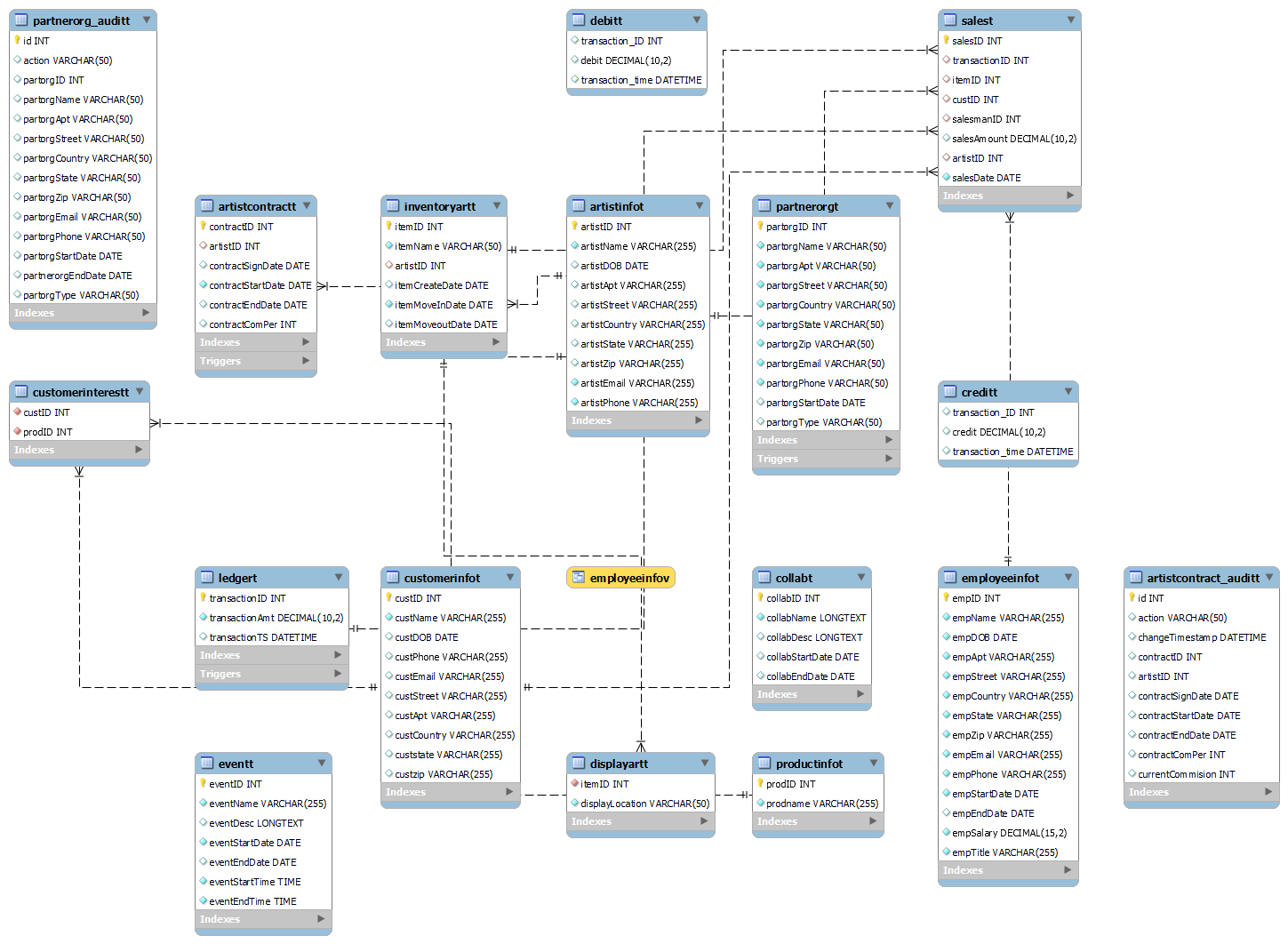
Current Issues:

* Inability to keep track of visitors, visitors’ preferences, and artwork collections
* Inability to implement automation in terms of cash flow management and keeping proper track of the same
* Inability to implement cost-effective marketing campaigns
* Inability to identify trends and customer behavior

Our Solutions:

* Automation of artwork inventory management and tracking
* Streamlining the process of adding and updating artwork information along with both employee and artist information
* Ability to view and search artwork by artist, category, or other criteria
* Automation of sales and cashflow management process
* Automation of customer relationship management (CRM) processes
* Streamlining the process of maintaining or updating the artist’s contract when due
* Ability to create reports to analyze the performance of the employees in terms of the sales made during a particular time duration
* Ability to analyze the advantages of doing a strategic tie-up with another art galleries in order to improve the reach of the organization

**Database** **Schema**:



TABLES

|  |  |  |
| --- | --- | --- |
| **SL. NO.** | **TABLE NAME** | **TABLE DESCRIPTION** |
| 1 | ArtistInfoT | Record of Artist’s personal information |
| 2 | ArtistContractT | Details of an artist’s contract with the art gallery |
| 3 | EmployeeInfoT | Record of the gallery’s employees |
| 4 | PartnerOrgT | Record of partner organisations – janitorial, catering and easte disposal |
| 5 | CustomerInfoT | Details of Customer’s personal information |
| 6 | ProductInfoT | Record of gallery’s souveniers, art pieces and newsletters |
| 7 | InventoryArtT | Details of an artwork- artists and the date of installments |
| 8 | CustomerInterestT | A Customer’s enquiry about product so as to implement recommender systems |
| 9 | DisplayArtT | Record of an item’s location |
| 10 | LedgerT | Record of every transaction made every day |
| 11 | CreditT | Record of cash inflow   * Automatically updates credit entry from ledger |
| 12 | DebitT | Record of cash outflow   * Automatically updates debit entry from ledger |
| 13 | CollabT | Record of events hosted by the art gallery in collaboration with other institutes |
| 14 | SalesT | Record of sales made by salesman to a customer |
| 15 | Artistcontract\_auditT | Automatically updates an artists contract renewal |
| 16 | partnerOrg\_auditT | Automatically updates a partner organisation’s last date with the gallery |
| 17 | EventT | Record of events hosted by the art gallery |

INDEXES:

Indexes are usually created to retrieve data from the database more quickly than otherwise.

We have 2 indexes in this database:

1. **salesmanID: FK in salesT :**

Since, in a sales table in order to analyze the performance of employees with the highest sale each month to reward them with bonuses we have created an index on the salesman id column

CREATE INDEX salesmanIDI

ON SalesT(salesmanID);

1. **artistID: FK in salesT:**

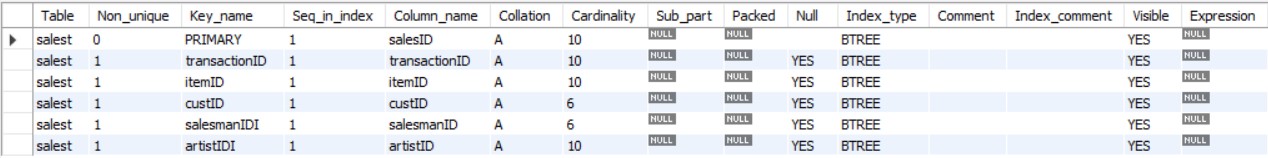
Similarly, in the sales table in order to analyze the performance of the artists with the highest/lowest sales each month, we have created an index on the artist id column

CREATE INDEX artistIDI

ON SalesT(artistID);

Query:

Show Index from SalesT;



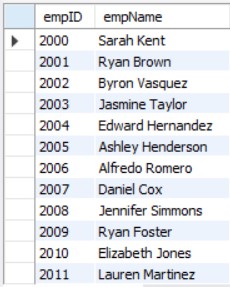
Views:

A view can select certain columns and/or rows from a table with permissions set on the view instead of the underlying tables. This allows surfacing only the data that a user needs to see.

So, in line with the above statement, we have created a view on the Employee information table where the other employees can only see the employee's name and ID thereby ensuring all the sensitive information about all the employees is secured and can only be accessed by the upper management.

EmployeeInfoV;

Select \* from EmployeeInfoV;

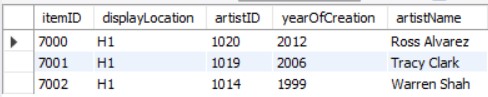


Queries:

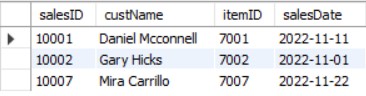
A Query is a mechanism of retrieving information from a database.

We have curated a total of 16 queries to achieve a truly automated system for analysis:

1. Query to display all the art pieces along with the names of the artists who created them and the year in which they were created that are displayed in hall 1 (DisplayArtT and InventoryArtT)



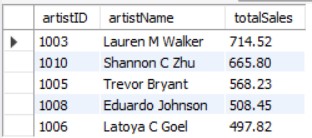
1. Display all the sales made in the month of Nov 2022 (salesT and inventoryArtT and customerInfoT)



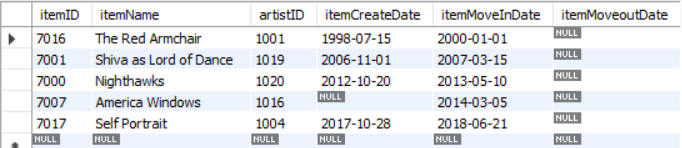
1. To find the employee of the month November (max total sales) in the year 2022 (SalesT and EmployeeInfoT)



1. Best performing (Top 5) artists in the year 2022



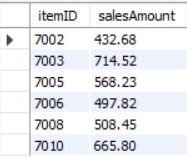
1. Least performing art pieces (Top 5 DESC) - stayed in the inventory for over a year



1. List all the customers that belong to a particular area (zipcode = x) and are interested in newsletters (productID = 3005)

Assuming our gallery is in texas, forthworth area and you want to select customers that live in the area with zipcode starting from 75

1. List of all the paintings that are priced higher than the average price of all the paintings sold in the gallery (subquery)



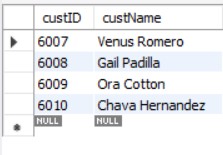
1. List months of 2022 in descending order of the total sales



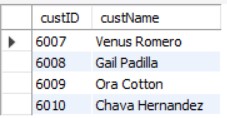
1. List average/total sales for the month of November 2022



1. List of customers who have not purchased anything from the gallery



1. List the customers who have not purchased anything from the gallery for the last five years



1. Display sale with 3rd highest sales amount for each year



1. Calculate the total commission earned by an artist given the artist ID = 1001



1. Calculate average yearly sales made by an artist and the commission he earned for it over the years (artist ID = 1001)



1. Calculate average yearly sales made by a salesman (salesmanID = 2006)



Stored Programs:

Stored programs are used to execute a set of instructions multiple times and to automate processes. They are used to create functions, triggers, and stored procedures, which are all ways of reusing code and making it easier to execute complex tasks.

We have 2 functions, 3 stored procedures and 4 triggers for our database

Functions:

1. totalPurchase()

* Returns the total purchase made by the customer based on the customerID.
* Returns ‘Invalid ID’ if the customerID is not present in the database



1. totalSales()

* Returns the total sales made by an artist based on the artistID.
* Returns ‘Invalid ID’ if the artistID is not present in the database



Stored Procedures:

1. GetCustomerLevel

Returns Classifies customers into three levels – Silver, Gold and Platinum based on their purchase amount or expenditure



1. GetArtistLevel

Classifies artists into three levels – Rookie, Novice, Expert and Genius based on their sales.



1. GetPlanner

Shows all the events happening on a certain date



Triggers:

A trigger is a named MySQL object that activates when an event occurs in a table. Triggers are a particular type of stored procedure associated with a specific table.

1. artistcontract\_updateT:

This is a pre-Update Trigger on the artistContractT table.

It stores all old values in artistcontract\_auditT, when an artist renews his contract

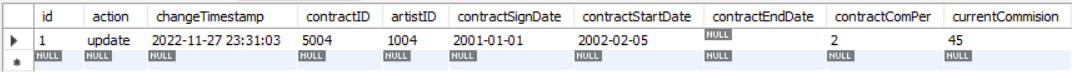
It Stores action value as ‘update’ and changeTimestamp as now()

select \* from artistcontractt;

update artistcontractt

set contractcomper = 45

where contractid = 5004;



1. pastPartnerOrg\_deleteT:

This is a pre-Delete Trigger on the partnerOrgT table.

It stores old values in partnerOrg\_auditT, when a partner organization is removed from the database.

It also stores action value as ‘delete’ and partOrgEndDate as curdate().

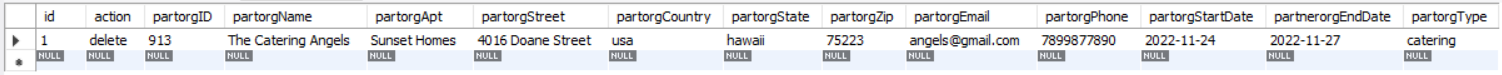
select \* from partnerorgt;

delete

from partnerorgt

where partorgID = 913;

select \* from partnerorg\_auditt;



1. ledger\_creditT:

This is a post-Insert Trigger on the ledgerT table. It Checks new.transactionAmt

If new.transactionAmt > 0 then inserts an entry into the credit, else inserts entry into the debit.

The new entry includes:

* + - Transaction ID from the ledgerT
    - Transaction amount from the ledgerT
    - Timestamp using now()

1. salesT:

This is a pre insert trigger on salesT table.

