UNIVERSITY OF GHANA

Department of Computer Engineering

SCHOOL OF ENGINEERING SCIENCES

COLLEGE OF BASIC AND APPLIED SCIENCES

FIRST SEMESTER 2022/2023 ACADEMIC YEAR

COURSE CODE:CPEN 211 DATABASE SYSTEMS DESIGN.

COURSE INSTRUCTOR: JOHN KORANKYE ASIAMAH

NAME: FAROUK SEDICK

ID NUMBER: 10947554

PROJECT 2

DATABASE DESIGN FOR AIRPORT SHELL FILLING STATION

INTRODUCTION

From the case study given, it is shown that Airport Shell is a large filling station near an international airport. Hence, many transactions occur at the filling station everyday. Some transactions that may occur include customers buying fuel and/or purchasing stuffs from the filling station's supermarket.

Due to a possible large number of people visiting the filling station everyday, it would be much more secure, effective and efficient to automate some operations and also audit transactions.

PROJECT OVERVIEW

➢ The database project was implemented using the following tools;

- Postgres – A database system
- PgAdmin – A postgres GUI
- SQL – A powerful relational database query language

➢ The project contains;

- Seven (7) Tables
- Five (5) Trigger functions
- Eight (8) Standard Functions

➢ With the tables, they are;

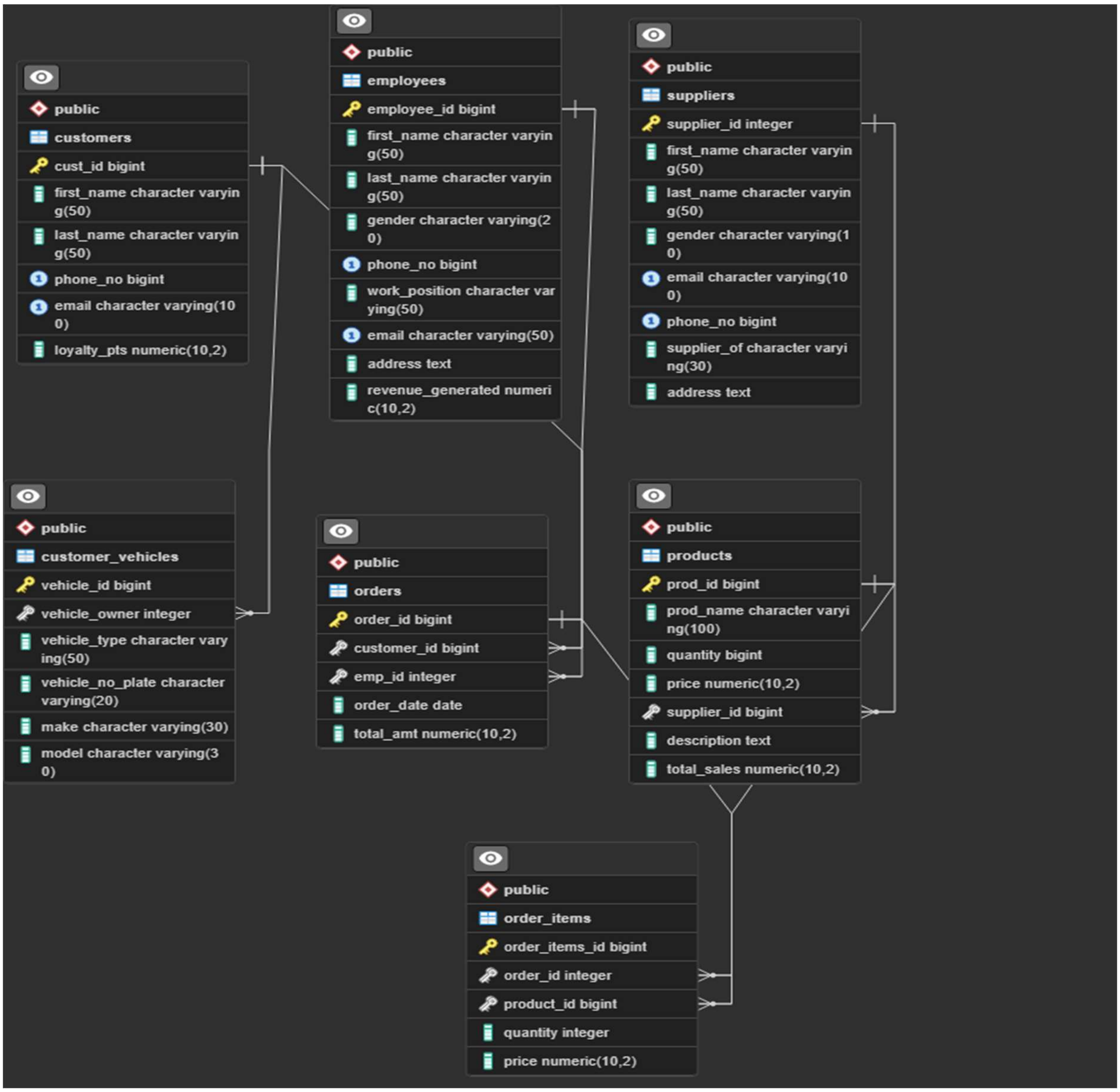
- Customers
- Customer_vehicles
- Employees
- Suppliers
- Products
- Orders
- Order_items

➢ The trigger functions are;

- Update_loyalty_pts()
- Update_order_total()
- Update_prod_total_sales()
- Update_product_quantities()
- Update_revenue_generated_by_salesperson()

- The standard functions are;
  - Avg_order_total()
  - Calculate_order_cost(order_id integer)
  - Customer_order_history(cust_id integer)
  - Employee_sales(emp_id integer)
  - Product_revenue(prod_id bigint)
  - Top_10_customers()
  - Top_selling_products(first_x_products integer)
  - Total_revenue()

Entity-Relationship Diagram (ERD)



public
customers
cust_id bigint
first_name character varying(50)
last_name character varying(50)
phone_no bigint
email character varying(100)
loyalty_pts numeric(10,2)

public
employees
employee_id bigint
first_name character varying(50)
last_name character varying(50)
gender character varying(20)
phone_no bigint
work_position character varying(50)
email character varying(50)
address text
revenue_generated numeric(10,2)

public
suppliers
supplier_id integer
first_name character varying(50)
last_name character varying(50)
gender character varying(10)
email character varying(100)
phone_no bigint
supplier_of character varying(30)
address text

public
customer_vehicles
vehicle_id bigint
vehicle_owner integer
vehicle_type character varying(50)
vehicle_no_plate character varying(20)
make character varying(30)
model character varying(30)

public
orders
order_id bigint
customer_id bigint
emp_id integer
order_date date
total_amt numeric(10,2)

public
products
prod_id bigint
prod_name character varying(100)
quantity bigint
price numeric(10,2)
supplier_id bigint
description text
total_sales numeric(10,2)

public
order_items
order_items_id bigint
order_id integer
product_id bigint
quantity integer
price numeric(10,2)

ENTITY SETS AND RELATIONSHIPS

The relationship between entity sets and their descriptions are explained below;

- Customers: In this entity set, each customer has a unique customer id (cust_id) which serves as a primary key. The identity details of customers such as first name and phone number are also stored in this table. Each customer is given a default loyalty points of 0 when he or she has never placed an order. The loyalty_pts column helps management know the topmost customers and also aid in creation of promotions and discounts.

- Customer Vehicles : In this entity set, the details of customers with cars are stored. The vehicle number plate and other essential info about customers' cars are stored here. The cars are referenced to the owners in the Customers table. This table helps security track and identify theft of customers' cars.

- Employees : In this entity set, each employee has a unique employee id which serves as a primary key. The identity details of employers such as first name and phone number are also stored in this table. Also, the description of work of each employee is also stored here, example, salesperson. Each employee is given a default revenue_generated value of 0 when he or she has never attended to an order. The revenue_generated column helps management track how much revenue each employee generates.

- Suppliers : In this entity set, suppliers of products to the filling station are recorded. Each supplier has a unique supplier id which serves as a primary key. The identity details of suppliers such as first name and phone number are also stored in this table. The details of what each supplier supplies are also recorded in this table. This aids management in knowing who and what is being supplied to the filling station.

- Products: In this entity set, each product sold has a unique product id which serves as a primary key.The product name, quantity and price are also recorded in this table. Each product references its supplier in the Suppliers table. The total sales of a particular product are also recorded here.

- Orders : In this table, each order initiated by a customer is recorded. Each order has a unique order id which serves as a primary key. Each order also references the customer placing the order, the employee serving the customer, the date in which the order took place and the total amount paid by the customer.

- Order Items : In this table, the items ordered by a customer are recorded when an order is initiated. The quantity ordered and the price of each ordered item is recorded. Each order item references an order and product from the Orders and Products tables respectively.

BACKGROUND LOGIC OF DATABASE

Trigger Functions

- Update_loyalty_pts(); This trigger function automatically increases the loyalty points of customers by the amount spent whenever they place an order. It is called and executed after INSERT or UPDATE statement is executed on the Orders table.

- Update_order_total(); This trigger function automatically calculates the total amount of an order whenever a customer places one. It is called and executed after INSERT or UPDATE statement is executed on the Order Items table.

- Update_prod_total_sales(); This trigger function automatically updates the total sales of a product whenever a customer orders an item. It is called and executed after INSERT or UPDATE statement is executed on the Order Items table.

- Update_product_quantity(); This trigger function automatically decreases the quantity of a specific product when a customer orders for an item of that product. It is called and executed after INSERT or UPDATE statement is executed on the Order Items table.

- Update_revenue_generated_by_salesperson(); This trigger function automatically updates the revenue_generated column of employees whenever a salesperson issues order. It is called and executed after INSERT or UPDATE statement is executed on the Orders table.

Standard Functions

- Top_10_customers() ; Retrieves the top 10 customers of the filling station based on how much they spend at the filling station.

- Top_selling_products(first_x_products integer) ; Retrieves the number of specified top selling products of the filling station. It accepts a limit number as a parameter.

- Avg_order_total(); Calculates and returns average total amount of orders.

- Calculate_order_cost(order_id integer) ; Calculates and returns cost of an order.

- Customer_order_history(cust_id integer) ; Retrieves order history of a customer by accepting the id of the customer as a parameter.

- Employee_sales(emp_id integer) ; Calculates and returns total revenue generated by an employee by accepting the id of the employee as a parameter.

- Product_revenue(prod_id bigint) ; Calculates and returns total revenue gained from sold items of a specific product by accepting the id of the product as a parameter.

- Total_revenue() ; Calculates and retrieves total revenue generated by the filling station.

CONCLUSION

With the database implemented above, the operations of the Airport Shell filling station will run smoothly as expected by management. It automates lots of processes with no error. It is also secure and efficient.