Database Systems – I

Inventory and Sales Management System

Project Partners
Ibraheem Bin Haseeb (FA21-BCS-052)
Ifrah Zulfiqar (FA21-BCS-064)

Index

Abstract

This report is for a Inventory and Sales Management System particularly designed for stores which sell various types of products in a day ranging from the littlest products like candies to large scale products like a 50kg sac of rice. The report is specifically formed after interviewing with a local store, collecting issues and problems they face on daily routine and how digitalizing their whole system can eliminate problems and boost their sales. The report demonstrates a design for a Relational Database System that can be implemented to eliminate daily life problems and easy the process with recommendations mentioned where necessary.

Domain Description

H. K Rice Shop is a shop situated near Bhalla Chowk, Lahore who have been dealing with various types of rice, spices, nuts, and many other small items that a household needs on a daily business.

Their supplier is from a village situated in the West of Punjab from which they purchase these rice. Other products like spices and nuts come from other regions of Punjab and Sindh which are then sold to locals and other big sharks.

During the times of COVID-19, their business took a hit and since then they have recovered quite a lot. To grow huge and exponentially they have taken into consideration to digitalize their business which is healthy both for their business and clients as it insures transparency and quality.

Current Situation

Currently the shop works manually where when the buy something for their inventory, they store the supplier name, amount, and roughly store their products on a register written by hand which is very time consuming and a tiring process for the worker there.

Besides when it comes to sales, they never really create any bills or store them in their register. This brings a lot of problems at the end of the month when audit is to be done to check if everything is up-to-date or not. Maybe there is a burglary but there is no way to tell and the damage will be done without any record on paper or anything.

Bills are only made when large amount of items are sold or something expensive is sold in which case bills are stored in the registers. But, this way of storing data only based on amounts or quantity is really bad for the store and results in loss and mishaps.

Requirements

In this part of the report we will discuss the requirements and needs for this software and make a case where one can see the need of a software to eliminate potential threats to a business and keep it in records. Why just a piece of well designed database is so important in scaling a business.

Records:

When the purchases are made for the inventory it is necessary and extremely important to keep the record of those purchases. These records are crucial to find which supplier is working the best for the business. Where can the shop find the best price for its business and which supplier does it deliveries on time and so on.

Storage:

Storing the inventory in database is again crucial step to take as it insures exactly what is available in the store instead of manually going and checking if the item is available or not. The seller can easily check its availability, price, and so on about the product that is being sold or questioned.

Sales:

When the products are sold, keeping the record of it has several advantages. Some of them can be listed as:

- Helps in auditing at the end of every month.
- Keeps the record of sales made at any point in time.
- Keeps the record of which product is needed more at any given point in time.
- Helps in calculating the total profit made in the business at any given point in time.

Functional Requirements

ID	Function	Entity	Priority
Records			
1	Insert/Update/Delete/View of the record(Time, Products, Supplier, etc.)	Purchase, Order	High
2	All the records for a given period	Purchase, Order	Normal
Storage			
3	Insert/View/Update/Delete of products from the storage	Products	High
4	View all the products	Products	High
5	View total number of products available in the stock	Products	Normal
6	Look up if a product is available or not	Products	High
7	Tell the user if product is low on stock	Products	Low
8	Best performing product for a given time	Order	Low
9	Most expensive product	Products	Low
10	Cheapest product	Products	Low
11	Products low in stock	Products	Low
Sales			
12	Insert/View/Update/Delete for the orders	Orders	High
13	All sales for a given period	Orders	High
14	Profit made for a given period	Orders	Normal

Non-Functional Requirements

- Employees should have their own login and password from which they can access the database and use it accordingly.
- A specific level of user's access will ensure the security risks in database.
- A final report will be available of given requirements.
- System must be designed in a fashion to have room for future growth.

Assumptions

Records:

Product entry can only be done by all level of employees. Entering the record is open to all the employees working in the shop. Products can be viewed by all the employees as well.

Updating or deletion of any record is restricted to only the manager to clear any security risk.

Storage:

Storage can be viewed by only the cashier or the manager. Other employees do not have access to do so.

Any changes such as update or delete will only be done by the manager or cashier. No other employee will be given the access to do so. Some view queries can be done by all the employees depending upon their security risk.

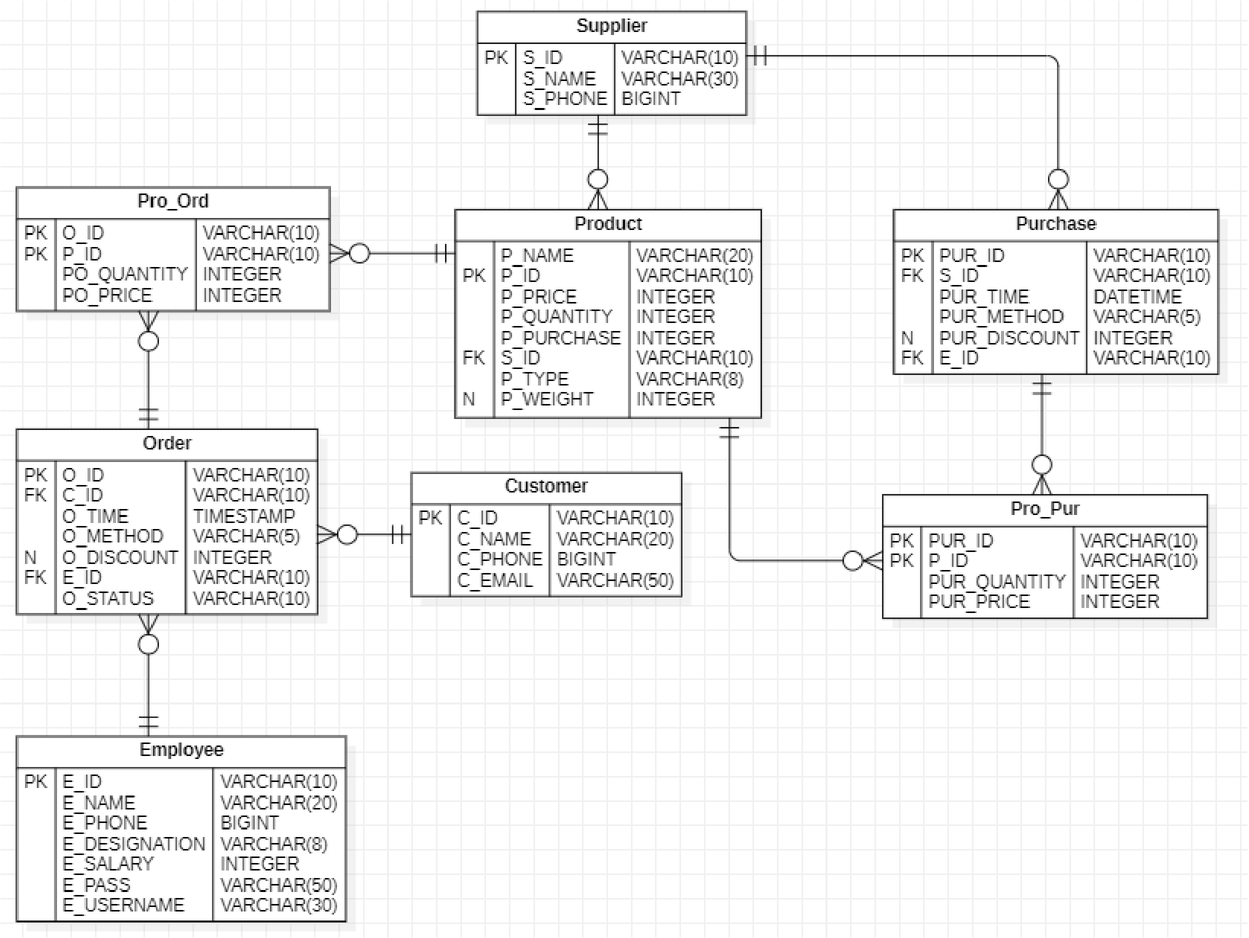
We assume that there are only two types of products that are available in the shop. One that is sold based on their weight and one that is sold based on their quantity. For example, rice is only sold based on their weight. Spices are sold per fixed price on their packet meaning if one packet of 50rs is sold, that is one packet sold but no dependent on its weight.

Sales:

Sales can be made by all the employees in the shop. Sales include Insertion in its record and updating in storage.

Sales can only be viewed by the manager as it is some information that should not be available to everyone. Manager can then view and make changes in their business as they feel like it.

Entity Relationship Diagram



	Supplier	
PK	S_ID	VARCHAR(10)
	S_NAME	VARCHAR(30)
	S_PHONE	BIGINT

	Pro_Ord	
PK	O_ID	VARCHAR(10)
PK	P_ID	VARCHAR(10)
	PO_QUANTITY	INTEGER
	PO_PRICE	INTEGER

	Product	
	P_NAME	VARCHAR(20)
PK	P_ID	VARCHAR(10)
	P_PRICE	INTEGER
	P_QUANTITY	INTEGER
	P_PURCHASE	INTEGER
FK	S_ID	VARCHAR(10)
	P_TYPE	VARCHAR(8)
N	P_WEIGHT	INTEGER

	Purchase	
PK	PUR_ID	VARCHAR(10)
FK	S_ID	VARCHAR(10)
	PUR_TIME	DATETIME
	PUR_METHOD	VARCHAR(5)
N	PUR_DISCOUNT	INTEGER
FK	E_ID	VARCHAR(10)

	Order	
PK	O_ID	VARCHAR(10)
FK	C_ID	VARCHAR(10)
	O_TIME	TIMESTAMP
	O_METHOD	VARCHAR(5)
N	O_DISCOUNT	INTEGER
FK	E_ID	VARCHAR(10)
	O_STATUS	VARCHAR(10)

	Customer	
PK	C_ID	VARCHAR(10)
	C_NAME	VARCHAR(20)
	C_PHONE	BIGINT
	C_EMAIL	VARCHAR(50)

	Pro_Pur	
PK	PUR_ID	VARCHAR(10)
PK	P_ID	VARCHAR(10)
	PUR_QUANTITY	INTEGER
	PUR_PRICE	INTEGER

	Employee	
PK	E_ID	VARCHAR(10)
	E_NAME	VARCHAR(20)
	E_PHONE	BIGINT
	E_DESIGNATION	VARCHAR(8)
	E_SALARY	INTEGER
	E_PASS	VARCHAR(50)
	E_USERNAME	VARCHAR(30)

Entities and Attributes

Format

- Every id must be of the format: (ABC000)
- Phone numbers are saved in the given format: (923210123456)

Supplier

S_ID is the primary key for the table which is in the form of varchar. We can use this id to fetch more information about suppliers such as their name and phone number. Supplier information is kept minimal as they don't play much of a role in day to day functioning. We have kept the supplier's name and supplier's phone number which is necessary incase of any mishaps.

Product

The product table is the most important table in the database as pretty much everything happening in the database revolves around the product. P_ID is kept as the primary key for the table in the form of varchar. Purchase column is the amount that was paid to buy this product and Price column is amount at which the product will be sold to the clients and public. Both the values are stored in integers as it is extremely unlikely to extend the integer size. Product type is a varchar the size of only 1 as its value will be either `Q` or `W`. Q is for quantity and W is for weight. Q is for quantity(products that are sold for their quantity as expressed below) and W is for weight(products that are sold for their weight).

S_ID is stored for the supplier id as the foreign key that can be used to fetch its information where necessary. Weight column is only for the products whose type is W otherwise it will be set to null.

Purchase

PUR_ID is the primary key for the table alongside some foreign keys such as E_ID to determine who entered this record in case of any problems that might occur. There is S_ID too for the supplier to keep the record of supplier within this record. Method column stores information about whether the amount was paid in cash or online. Discount column shows if any discount was available for the particular purchase and time is stored for when the purchase was made.

Order

Order table is also very important for this database as the sales made are all stored in this column. It stores a lot of information with very few columns inside. It stores similar information to purchase like time, method, etc. It stores some foreign keys to know which client was sold this product and which employee made the sale. It is possible to not store the client's id as it is not such an important information. In that case it will be set to null. Status column basically stores the information about whether the payment is done or is it due. In case of due, we will store this information aside so manager or cashier knows which client has to pay in some time in the future.

Customer

We store bare information about the clients like their names, phone number, email, etc. C_ID stores the primary key which can be used to fetch the client's information where needed.

Employee

Employee tables helps a lot in keeping track of crucial information. It is helpful if someone makes any sort of change in the database, the record will be stored which employee's account did it. We can keep track of misconduct in the database. Employee table stores basic information about the employee including their salary and designation. We have already established that their only three types of employees and they are as: Manager, Cashier, Worker. The password field will store employee's password with a special encryption algorithm.

Pro_Ord Table

Product order table defines relationship between product table and order table. It defines a `many-to-many` relationship between the two tables. It stores foreign keys for both of the tables alongside some extra information such as the price at which the product was sold because it is very much likely the product is sold at a different price and in the future the price is increased or decreased due to the world economics. So it helps in keeping a record of the prices of the past. With that we store how much discount is being given for this product and how much product is being sold in the quantity table. For primary key, we are using order id and product id together in the form of Composite primary key.

Pro_Pur Table

This table again defines a `many-to-many` relationship between product table and purchase table as we know one product can appear in many purchases from time to time and many purchases can be done of the same product. It uses a composite primary key in the form of pur_id and p_id which happens to be the foreign keys for their respective tables as well. Aside that we store the price and quantity in this table for the product because the price may fluctuate in the future.

Relationships and Labels

Record

- A supplier can be linked to many records of purchases.
- One record of purchase will have only one supplier.
- Many products can appear in many purchase records.

Storage

- A product will appear only once on the table.
- A product has only one supplier
- A supplier can provide many products.

Sales

- A product can appear in many orders.
- An order can have many products.
- An order can have only one employee but one employee can be on many orders.

Validation

In accordance with the user's requirements we can constitute that the designed Entity Relationship diagram is covering all the factors needed and making the database as robust as possible. It is covering the most minimum purchases made and keeping track of every little thing that is happening around for concreteness of the database. It helps in keeping everything on track reduce the errors to almost none. Digitalizing the system has made it as transparent as possible.

We have already made all the assumptions and little rules to follow throughout the database which work alongside the business rules of the shop. We have established all the relationships between the entities and can move onwards on verifying that our database is absolutely compatible with them. There will be no conflicts anywhere and smooth transaction of data will be made where necessary to ensure database integrity.

Requirements 1 and 2 are completed from the Purchase and Product table where user can enter the record and view the entries they have made. Only the authorized user can update and delete the information if and only if necessary.

Requirements 3-12 are fulfilled by product and supplier tables where we store the records of the products and see what we have in our inventory in case needed. We can make any type of query needed. Only the authorized user will have access to alter the data as it is not something very common that happens on a day-to-day basis. If it happens, it means that the database has failed to comply with the business rules of the store.

Requirements 12-15 are met by the order and product tables where their many-to-many relationship has greatly reduced our effort and helped us design a robust system that can work with huge and large amounts of data. Authorized users can make the fetch and alter requests as per needed.

Description

The entity relationship diagram was carefully designed to meet the business requirements and work in accordance to them. Failing to work with the business rules ultimately fails the database which is the waste of money, time, and many priceless resources. Therefore, this entity relationship diagram was formed while keeping in view all the transactions that happen within the boundaries of this business.

Diving into the cycle of this database, the first step for any business is to buy some inventory they sell. When they do so that inventory is recorded into the database into the form of purchase table and product entries are also made in the meantime. We can also enter the supplier information which is extremely helpful during the times of analytics and audits. User can see which supplier the best option of their business is continuing to work with them or resolve any issues if they find with some suppliers from the help of this carefully designed database.

Once the entry is made now its time to sell this product. Let’s say someone comes to the shop and wants to buy 10kg of banaspati rice. The process to make this transaction is simple as we can record some client’s information if it is necessary(not mandatory by database rules to get each client’s information), user can then search and find the needed product and add it to a temporary products list. User can add more if necessary and then it is time to finally make the transaction. If there is some discount, store wants to offer, they can do so by setting the discount amount and then finally making the changes in the products table(decrementing the quantity column) and added entries in orders and other linked tables.

At the end of the month or anytime whenever user wants, they can view the data and see how much profit or how many sales have been from a given time to given time. This is extremely helpful in seeing in what days, which product is sold more, which product performs better so they can work on their business rules to boost their sales accordingly.

Conclusion

This model attempts to fulfill all the business requirements and make a smooth mode of transaction for users to work. Incase of any problems or lacking, there is always room to improve and fix the issues to ensure database integrity. During the time of implementation, it is quite possible that some modifications will be needed to meet the business requirements.

Recommendations

The database is quite up to standards as it is right now but there is always some room for improvements. For starters, we can add an option to keep record of the expenses that comes when it comes to just running the store. We can keep track of the electricity and water bills. The rent that is paid each and every month. This helps in transparency and a proof incase of any problems.

If the store grows, we have to keep in view that maybe the store will have multiple branches, so we need to keep them all connected and keep track of things, maybe something is available in one branch but not in other so we can send our customers there for their ease instead of them finding the product and tiring themselves.

Word Count: 2828