Term Paper
E-commerce Database Management

Name: Yogesh Kumar Dewangan
Roll no: 19111064
Branch: Biomedical Engineering 21/01/2022

What is a Database and DBMS?

Today we deal with a large amount of data. Almost every website and mobile application are generating lots of data every day and these data need to be handled. This can't be handled with simple software or tool. We need to store all these data and then manage the data with some dedicated system. Here comes the role of Database and DBMS.

You all must be familiar with the term Database and DBMS. So, is there any difference between these two terms or they are almost the same term?? No, they are not the same term. In this blog, we will be learning about the term Database and DBMS. So, let's get started.

Database

The database sometimes referred to as an electronic database, is an organized collection of logically related data that is stored in an efficient manner so that it can be easily accessed managed and updated. Let's divide the whole definition into parts and understand in an easier way:
Organised Collection:Data should be arranged in such a way that the user can easily process the data when required. Example: If we have some details about Computer specifications, then we can represent the specification in two ways:
Unorganized Data: Lenovo35000i58, HP55000i34. The computer specifications are provided here like its brand name, price generation and RAM but they are unorganized and it is not easy to process them. Here, you can't identify the difference between the generation of the computer i.e i5 and its RAM i.e. 8GB. So, we have to organize the data.

Organized Data: Lenovo/35000/i5/8, HP/55000/i3/4. Here, we have made a separation between each specification so that it can be easily distinguished and processed. Here, we can see that the generation and RAM of the system are separated by /änd this helps us in distinguishing between the two features.

Logically Related Data Logically related data means that the data should be relevant in some context. Example: If we are going to make a database for a customer then the database may include customer name, contact number, age, past orders, address, email id, etc. All these information are in the context of the customer. But the information like the number of siblings of the customer is out of context and logically not related to the customer database, though this information is related to the customer, we shouldn't include this data in our database.

DBMS - DataBase Management System

The software which is used to manage the database is called Database Management System(DBMS). It provides us with an interface or a tool, to perform various operations like creating the database,manipulating the database,storing and retrieving the data from the database,deleting data from the database, etc.
The changes in the database have to be made according to certain rules and these rules are defined in DBMS itself. A DBMS can limit what data the end-user sees and provides multiple views of the same database depending upon the user accessibility. For example, you can provide access to write on the database to some particular users only and for other users, you can provide the read access. The best part is that all you need to do is just use some DBMS software and you are good to go. Some popular DBMS software is MySQL, Oracle, SQLite, PostgreSQL, MariaDB etc.

Characterstics of DBMS

Real-World Entity: A DBMS uses real-world entities(object) to design its architecture. Example: A customer database uses customers as an entity and the phone number of the customer as an attribute. Relation-based Tables: Using DBMS we can form tables based on relations between various entities.

Query Language: DBMS comes equipped with query language which allows the users to store and retrieve the data. We can apply as many filtering options as required and get specific results.

Multiple Views: It provides multiple views of the same data depending upon the user. Example: In a university database, the accountant will have a different view of data than a student. The accountant may have access to the salary of teachers but students won't have that access.

Multiple Views: It provides multiple views of the same data depending upon the user. Example: In a university database, the accountant will have a different view of data than a student. The accountant may have access to the salary of teachers but students won't have that access.

Multiple Users: DBMS allows multiple users can access the data at the same time and work upon it parallelly.

ACID Properties: The transaction(a group of tasks) in DBMS follows the concept of ACID(Atomicity, Consistency, Isolation, Durability). Atomicity means either the transaction will happen or it will not happen. Consistency means the state of the database will be consistent before and after the transaction. In Isolation, one transaction will not affect the working of others. Durability means the database should be durable and should not be affected by some system failures or any other errors.

Ecommerce Database

As technology grows, the number of people participating in e-commerce purchases will grow along with it. Electronic commerce, known as e-commerce, is a type of business model that involves making transactions over the internet. Any store, business, or person who actively sells products online are considered to be apart of e-commerce. In 2018 alone, mobile conversions increased by 55 percent and are expected to reach 175.4 billion dollar in USD sales.

However, with the ever-growing industry, there is a system that is running the backend of each and every successful e-commerce site; a database. With the abundance of fast-growing data being transferred every day, sites need to be thoughtful about database best practices. By ensuring proper procedures, businesses can fully leverage an agile system to help store, organize, and structure data for an e-commerce site .If you have ever heard of the term "Cloud database" or "Database-backed" site, they are referring to a site that is using a database. The basic function of a database is to digitally store information that can be captured, retrieved, and distributed easily at a later time. Through this system, information such as transactional or systems oriented data can be organized and tracked based on chosen settings. This also gives businesses the ability to analyze and track information about the products, sales, and customers that have been input into the database.

Provides Structure:One of the biggest benefits of using a database for e-commerce is the addition of structure to vast amounts of shop data. No matter how big or small your online store is, an infrastructure is needed for all the gathered information to make sense and provide useful insights. When the data is structured, it can be accessed more efficiently by the e-commerce application.

Attracts an Audience: A database system helps e-commerce sites pinpoint potential customers based on compiled information. Marketing teams can use customer data that is stored within the database to create targeted lists that will be used for directing marketing efforts. The more information a marketing team has, the better they can identify and tailor communications with them. Not only will this method help to retain customers, but also helps to gain new customers as well.

Tracks Data:Databases are an integral part of the success of an online commerce site. With the ability to store, organize, and analyze immense amounts of data quickly, there is no wonder why databases are essential. Through organized and updated data, companies can quickly respond and update based on changing market conditions.

The following are common types of data that databases help to store and track for e-commerce sites:

Product Information;Databases help e-commerce sites to update and hold information regarding products. This can include detailed product descriptions, prices, specifications, product reference numbers, promotions, and availability. Updating this information through your database saves you from changing and publishing new data to every individual site page by uniformly keeping all pages up-to-date with the same info.

Customer Information;Having customer data to store and analyze is huge for e-commerce sites. Through databases, information such as contact details, names, spending patterns, and more can be held for later marketing use. For instance, personal demographic details are very relevant to buying potential. The collection of this type of data helps to create distinguished target markets and allows for improved customer communications, encouraging online commerce sites to focus on customers' wants and needs. When you can tell what a customer wants, issues and problems can be resolved easier.

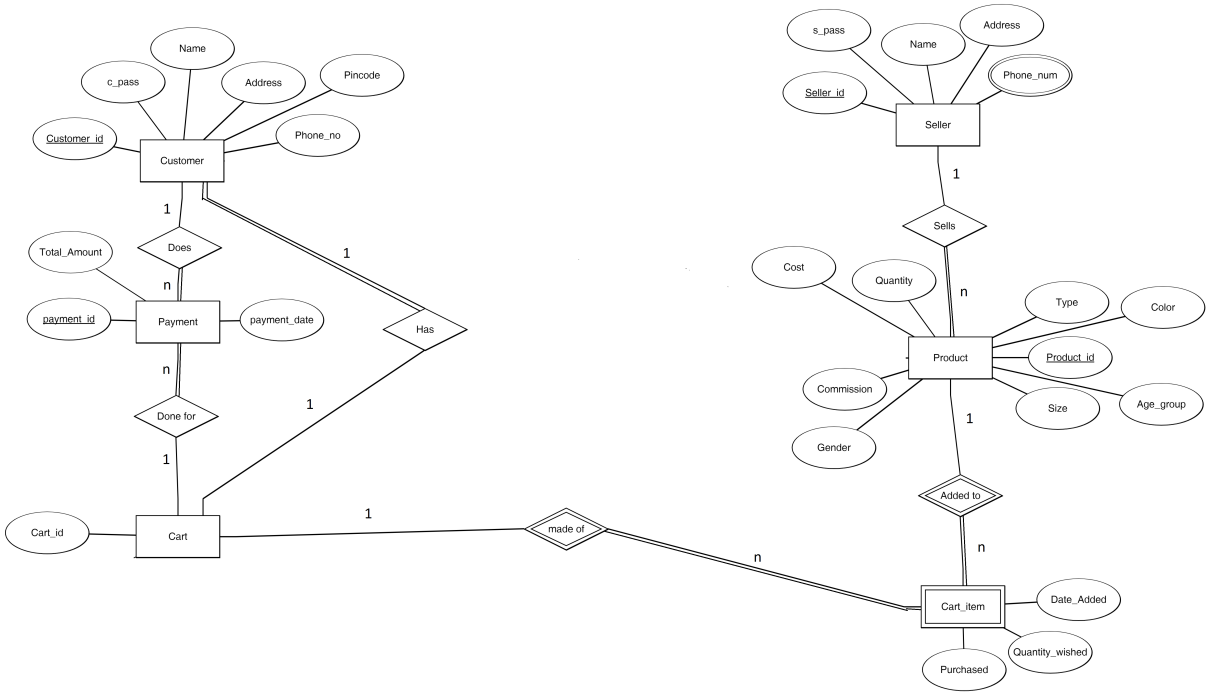
Transactional Information;Tracking and managing transactions are one of the most important jobs a database can do for an e-commerce site. Tracking every order throughout the sales funnel in an organized manner, along with the needed processing details, is key to keeping the business functioning. The database helps to keep the inventory up to date after each transaction, such as what is in or out of stock, billing, shipping statuses, purchase orders, and more.

Databases are crucial to an efficient and successful e-commerce site. Whether you have a small company with limited data or a big store with millions of products, databases help to digitally store and organize important data. With so much information needing to be processed, it would be impractical to not have a database in use.

Basic Structure

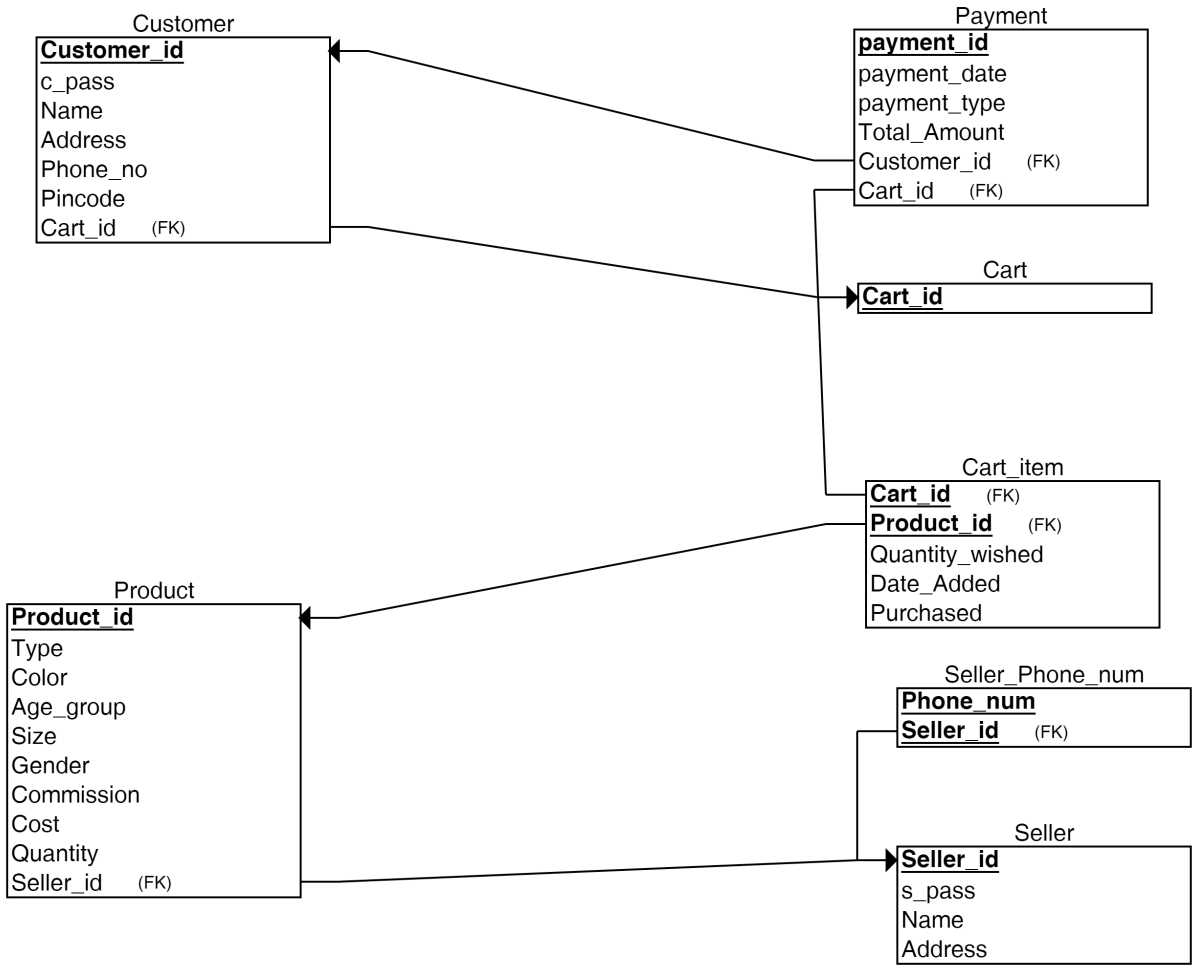
Functional Requirement
-A new user can register on the website.
-A customer can see details of the product present in the cart
-A customer can view his order history.
-Admin can start a sale with certain discount on every product.
-Customer can filter the product based on the product details.
-A customer can add or delete a product from the cart.
-A seller can unregister/ stop selling his product.
-A seller/ customer can update his details.
-Admin can view the products purchased on particular date.
-Admin can view number of products sold on a particular date.
-A customer can view the total price of product present in the cart unpurchased.
-Admin can view details of customer who have not purchased anything.
-Admin can view total profit earned from the website.

Entity Relation Diagram



Figuur 1:

Relational Database Schema



Figuur 2:

Implementation

Creating Tables

```
CREATE TABLE Cart
(
Cart_id VARCHAR(7) NOT NULL,
PRIMARY KEY(Cart_id)
);

CREATE TABLE Customer
(
Customer_id VARCHAR(6) NOT NULL,
c_pass VARCHAR(10) NOT NULL,
Name VARCHAR(20) NOT NULL,
Address VARCHAR(20) NOT NULL,
Pincode NUMBER(6) NOT NULL,
Phone_number_s number(10) NOT NULL,
PRIMARY KEY (Customer_id),
Cart_idVARCHAR(7)NOTNULL,
FOREIGNKEY(Cart_id)REFERENCESscart(Cart_id)
);
```

$CREATETABLESeller$
$($
$Seller\_idVARCHAR(6)NOTNULL,$
$s\_passVARCHAR(10)NOTNULL,$
$NameVARCHAR(20)NOTNULL,$
$AddressVARCHAR(10)NOTNULL,$
$PRIMARYKEY(Seller\_id)$
$);$

$CREATETABLESeller_Phone\_num$
$($
$Phone\_numNUMBER(10)NOTNULL,$
$Seller\_idVARCHAR(6)NOTNULL,$
$PRIMARYKEY(Phone\_num, Seller_id),$
$FOREIGNKEY(Seller\_id)REFERENCESSeller(Seller\_id)$
$ONDELETECASCADE$
$);$

$CREATETABLEPayment$
$($
$payment\_idVARCHAR(7)NOTNULL,$
$payment\_dateDATENOTNULL,$
$Payment\_typeVARCHAR(10)NOTNULL,$
$Customer\_idVARCHAR(6)NOTNULL,$
$Cart\_idVARCHAR(7)NOTNULL,$
$PRIMARYKEY(payment\_id),$
$FOREIGNKEY(Customer\_id)REFERENCESCustomer(Customer\_id),$
$FOREIGNKEY(Cart\_id)REFERENCESCart(Cart\_id),$
$total\_amountnumeric(6)$
$);$

$CREATETABLEProduct$
$($
$Product\_idVARCHAR(7)NOTNULL,$
$TypeVARCHAR(7)NOTNULL,$
$ColorVARCHAR(15)NOTNULL,$
$P\_SizeVARCHAR(2)NOTNULL,$
$GenderCHAR(1)NOTNULL,$
$CommissionNUMBER(2)NOTNULL,$
$CostNUMBER(5)NOTNULL,$
$QuantityNUMBER(2)NOTNULL,$
$Seller\_idVARCHAR(6),$
$PRIMARYKEY(Product\_id),$
$FOREIGNKEY(Seller\_id)REFERENCESSeller(Seller\_id)$
$ONDELETESETNULL$
$);$

$CREATETABLECart\_item$
$($
$Quantity\_wishedNUMBER(1)NOTNULL,$
$Date\_AddedDATENOTNULL,$
$Cart\_idVARCHAR(7)NOTNULL,$
$Product\_idVARCHAR(7)NOTNULL,$
$FOREIGNKEY(Cart\_id)REFERENCESCart(Cart\_id),$
$FOREIGNKEY(Product\_id)REFERENCESProduct(Product\_id),$
$Primarykey(Cart\_id, Product\_id)$
$);$

$altertableCart\_itemaddpurchasedvarchar(3)default'NO';$

Inserting Values

$insertintoCartvalues('crt1011');$

insert into Customer values('cid100','ABCM1235','rajat','G-453','632014',9893135876, 'crt1011');

insert into Seller values('sid100','12345','aman','delhi cmc');

insert into Product values('pid1001','jeans','red','32','M',10,10005,20,'sid100');

insert into Seller_Phone_num values('9943336206','sid100');

insert into Cart_item values(3,to_date('10-OCT-1999','dd-mon-yyyy'),'crt1011','pid1001','Y');

insert into Payment values('pmt1001',to_date('10-OCT-1999','dd-mon-yyyy'),'online','cid100','crt1011',NULL);

Queries

Basic Queries
If the customer wants to see details of product present in the cart
select * from product where product_id in(
select product_id from Cart_item where (Cart_id in (
select Cart_id from Customer where Customer_id='cid100'
))
and purchased='NO');

If a customer wants to see order history
select product_id,Quantity_wished from Cart_item where
(purchased='Y' and Cart_id in (select Cart_id from customer where Customer_id='cid101'));

Customer wants to see filtered products on the basis of size,gender,type
select product_id, color, cost, seller_id from product where
(type='jeans' and p_size='32' and gender='F' and quantity¿0)

If customer wants to modify the cart
delete from cart_item where (product_id='pid1001' and Cart_id
in (select cart_id from Customer where Customer_id='cid100'));

If a seller stops selling his product
delete from seller where seller_id = 'sid100';
update product set quantity = 00 where seller_id is NULL;

If admin want to see what are the product purchased on the particular date
select product_id from cart_item where (purchased='Y' and date_added='12-dec-2018');

How much product sold on the particular date
select count(product_id) count_pid,date_added from Cart_item where purchased='Y' group by(date_added);

If a customer want to know the total price present in the cart
select sum(quantity_wished * cost) total_payable from product p join cart_item c on p.product_id=c.product_id
where c.product_id in (select product_id from cart_item where cart_id in(select Cart_id from customer
where customer_id='cid101') and purchased='Y');

Show the details of the customer who has not purchased any thing
Select * from customer where customer_id not in (select customer_id from Payment);

Find total profit of the website from sales.
select sum(quantity_wished * cost * commission/100) total_profit from product p join cart_item c on

p.product_id=c.product_id where purchased='Y';

Example of a Good Database Design

Databases are vital tools for storing, managing, and retrieving information. They are also critical for building an e-commerce system. A well-structured database powers e-commerce and manages all the interactions within the system.

A good e-commerce database design includes:

Simple, functional database structure: The database table structure is simple but covers all the required functionality without compromising the user experience.

High performance: Database queries execute quickly to facilitate live customer interactions and support a frictionless shopping experience. Therefore, the selected database should have good indexing and performance optimization options.

High availability and scalability: A good database design is highly available with automatic snapshots and enables automatic scaling to support future platform growth as well as sudden traffic spikes.