

CSE 202 Fundamentals of Database Management Systems

Project Title

Design of an online retail store system

Group 30 Members

Mehul Singh 2020081
Shivam Jindal 2020125
Shreyanshu Sharma 2020130
Kanishk Singh 2020073

Scope of Project

The design of an online retail store system replicates the database management of retail stores with online shopping availability such as Big Bazaar, and is worked on and studied to understand the different aspects involving the storage of products, the interaction of employees, the company itself and the customer among various other objectives.

Stakeholders

Stakeholders are a person or a group being directly or indirectly affected by the activity of an organisation. Stakeholders may be directly or indirectly involved.

1. Customers: External stakeholders who can shop on the online portal
2. Employees: Internal stakeholders that manage the concerns of the customers through customer care department, delivery of the required goods as per the customer and the management and maintenance of
3. Suppliers: External stakeholders that provide the products to the company to be provided and delivered to the customers.

Entities and their attributes

1. Customer: Has customer ID(cust_ID), customer name(cust_name) and contact_no as the attributes, and is involved in Uses relationship with Account and Helps relationship with Employee
2. Account: Denoting the portal account of the customer, has delivery address(delivery_address) and email address(email_address) of the customer as entities, and is involved in manages Cart and Products
3. Cart: Denoting the cart of the customer, has cart ID(cart_id), total cost of the cart products(total_cost) and quantities(quantity) as the entities
4. Products: Represents products available on the platform, has product ID(product_ID), category(category), stock of product available(Availability) and price of product(price) as the attributes, , and is involved in Manages relationship with Cart, Has relationship with Online_portal and supplies relationship with Supplier

5. Supplier: Denotes the supplier/manufacturer of products, has supplier SSN(supplier_reg_no), supplier name(supplier_name) and supplier contact number(supplier_contact_no) as attributes, , and is involved in supplier relationship with Products
6. Online_portal: Represents online website portal of the company, has domain address(domain) as attribute and involved in Maintains relationship with Employee
7. Employee: Represents employees of the establishment, has employee ID(emp_ID), employee name(emp_name), designation of the employee(designation), employee D.O.B(emp_DOB) and salary(salary) as the attributes, and is involved in Helps relationship with Customer, Maintains relationship with Online_portal, and Delivers relationship with Products
8. Payment: Represents the payment of fee portal side of the store, has payment ID number(payment_ID) and mode of payment i.e UPI, Net banking etc (payment_method) as the attributes, , and is involved in Pays relationship with Cart

Relational Schema

customer = (customer_id, first_name, last_name, contact_no)

account = (email_address, house_no, locality, landmark, city, state)

payment = (payment_id, payment_method, cart_id)

Employee = (emp_id, first_name, last_name, designation, emp_DOB, salary)

online_portal = (domain)

Products = (product_id, product_name, category, availability, price)

supplier = (supplier_reg_no, supplier_name, supplier_contact_no)

cart = (cart_id, total_price)

uses = (user_login_id, user_password, email_address, cust_id)

maintains = (domain, emp_id, admin_username, admin_password)

supplies = (supply_id, amount, date_of_supply, quantity, product_id, supplier_reg_no)

helps = (query_no, query_type, cust_id, emp_id)

delivers = (delivery_id, date_of_delivery, emp_id, product_id)

manages = (email_address, product_id, cart_id, quant)

Grants

```
CREATE USER 'jeffrey'@'localhost' IDENTIFIED BY 'jeffrey';
grant all on mn.products to 'jeffrey'@'localhost';
grant insert on mn.customer to 'jeffrey'@'localhost';
grant select on mn.customer to 'jeffrey'@'localhost';
grant all on mn.manages to 'jeffrey'@'localhost';
grant all on mn.cart to 'jeffrey'@'localhost';
grant all on mn.manages to 'jeffrey'@'localhost';
```

Views

```
create view clothing_products as select p.product_name, s.supplier_name
from supplier S inner join products P inner join supplies SP
on P.product_id=SP.product_id and SP.supplier_reg_no=S.supplier_reg_no
where p.category="Clothing";
```

```
create view young_emp as select emp_id, first_name, TIMESTAMPDIFF(YEAR,
emp_dob, CURDATE()) as age, emp_salary
from employee
where TIMESTAMPDIFF(YEAR, emp_dob, CURDATE())<25 and emp_salary>25000 ;
```

```
create view clothing_products as select p.product_name, s.supplier_name
from supplier S inner join products P inner join supplies SP
on P.product_id=SP.product_id and SP.supplier_reg_no=S.supplier_reg_no
where p.category="Clothing";
```

```
create view products_suppliers as select p.product_name,
s.supplier_name
from supplier S inner join products P inner join supplies SP
on P.product_id=SP.product_id and
SP.supplier_reg_no=S.supplier_reg_no;
```

SQL Queries

#query 1

to find all products which are supplied by the supplier with registration no=18243 which are running low and need restocking
select P.product_id,P.product_name,P.price
from supplier S inner join products P inner join supplies SP
on P.product_id=SP.product_id and
SP.supplier_reg_no=S.supplier_reg_no
where S.supplier_reg_no=18243 and P.availability<50;

#query 2

to sort number of deliveries by the month they were delivered
select date_format(date_of_delivery, '%Y-%m') as date_of_delivery,
count(delivery_id) as
number_of_deliveries
from delivers
group by date_of_delivery
order by date_of_delivery desc;
update account set city="New Delhi" where landmark ="";

#query 3

to find the first names of all the customers whose last name is null and they live in New Delhi
select c.first_name
from customer c inner join uses u inner join account a
on c.cust_id=u.cust_id and u.email_address=a.email_address
where c.last_name = "" and a.city="New Delhi";

#query 4

to show the delivery details of every jewellery delivery with the date of the product's supply starting from the latest supplied product

```
select d.delivery_id, p.product_name, sp.date_of_supply
from delivers d inner join products p inner join supplies sp
on d.product_id=p.product_id and p.product_id=sp.product_id
where p.category="Jewellery"
order by date_of_supply desc;
```

#query 5

#to find the delivery agents of all the jewellery deliveries, made in the last decade, ordered by the most expensive to least expensive delivery

```
select e.first_name,e.emp_id, p.price, d.delivery_id,
d.date_of_delivery
from employee e inner join delivers d inner join products p
on e.emp_id=d.emp_id and d.product_id=p.product_id
where p.category="Jewellery" and year(d.date_of_delivery)>2010
order by p.price desc;
```

```
update delivers set
date_of_delivery=DATE_FORMAT(date_of_delivery,'2011-%m-%d')
where year(date_of_delivery)<2000;
```

#query 6

#view showing the product names and it's supplier's name

```
create view clothing_products as select p.product_name,
s.supplier_name
from supplier S inner join products P inner join supplies SP
on P.product_id=SP.product_id and
SP.supplier_reg_no=S.supplier_reg_no
where p.category="Clothing";
select * from clothing_products;
```

#query 7

to find out all employee whose age are less than 25 and have salary>25000

#to nominate employees for the best employee award ;)

create view young_emp as select emp_id, first_name, TIMESTAMPDIFF(YEAR, emp_dob, CURDATE()) as age, emp_salary from employee

where TIMESTAMPDIFF(YEAR, emp_dob, CURDATE())<25 and emp_salary>25000 ;

select first_name from young_emp;

#query8

#to delete records from delivery where date of delivery is more than 15 years old

delete from delivers where TIMESTAMPDIFF(YEAR, date_of_delivery, CURDATE())>15 ;

#query9

#to display cart with pricing and product with total price equalling to greater than Rs. 30,000

select c.cart_id, c.total_price, p.product_name from cart c inner join manages m inner join products p on c.cart_id=m.cart_id and m.product_id=p.product_id where c.total_price>30000;

#query10

#view of products and suppliers

create view products_suppliers as select p.product_name, s.supplier_name

from supplier S inner join products P inner join supplies SP on P.product_id=SP.product_id and SP.supplier_reg_no=S.supplier_reg_no;

select * from products_suppliers;


```
select product_name, count(supplier_name) from  
products_suppliers  
group by product_name  
having count(supplier_name)>2;
```

Embedded SQL Queries

1. Searching by product price (High to Low)
2. Searching by product price (Low to High)
3. User Login Page
4. Searching using a search bar for product with name entered by user

Indexing

```
create index price_sort on Products(price ASC);  
create index availability_sort on Products(availability ASC);  
create index cat_sort on Products(category ASC);  
create index sal_sort on Employee(salary ASC);
```

Triggers

```
create trigger total_cost  
after INSERT  
on manages  
for each row  
update cart  
set total_price=total_price + new.quant*(select price  
                                         from products  
                                         where new.product_id=products.product_id);
```

```
create trigger quant_avail
after INSERT
on manages
for each row
update products
set availability=availability-new.quant
where new.product_id=products.product_id;
select * from products where product_id = 17248;
```

```
create trigger lowercase_email
after INSERT
on Online_portal
for each row
update online_portal
set new.domain=LOWER(new.domain);
```

INDIVIDUAL CONTRIBUTION

Kanishk Singh:

1. Updating relational schema
2. SQL Queries
3. Embedded Queries
4. Indexing
5. Documentation/Answers in write-up

Shivam Jindal:

1. Front End coding
2. Embedded Queries
3. SQL Queries
4. Updating relational schema
5. Error Handling

Shreyanshu Sharma:

1. Updating relational schema
2. Views and Grants
3. SQL Queries
4. Embedded Queries

Mehul Singh:

1. Updating relational schema
2. Triggers
3. SQL Queries
4. Embedded Queries
5. Error Handling