***A PROJECT REPORT***

*E-Commerce Model*

*for online shopping system*

***By Smeet Kothari***

***NUID: 001740071***

*In partial fulfilment for the course*

*Of*

DATABASE MANAGEMENT AND DATABASE DESIGN

Introduction

1.

Electronic commerce or ecommerce is a term for any type of business, or commercial transaction that involves the transfer of information across the Internet. It covers a range of different types of businesses, from consumer based retail sites, through auction or music sites, to business exchanges trading goods and services between corporations. It is currently one of the most important aspects of the Internet to emerge.

Ecommerce allows consumers to electronically exchange goods and services with no barriers of time or distance. Electronic commerce has expanded rapidly over the past five years and is predicted to continue at this rate, or even accelerate. In the near future the boundaries between "conventional" and "electronic" commerce will become increasingly blurred as more and more businesses move sections of their operations onto the Internet.

Business to Business refers to electronic commerce between businesses rather than between a business and a consumer. Businesses often deal with hundreds or even thousands of other businesses, either as customers or suppliers. Carrying out these transactions electronically provides vast competitive advantages over traditional methods. When implemented properly, ecommerce is often faster, cheaper and more convenient than the traditional methods of bartering goods and services.

Electronic transactions have been around for quite some time in the form of Electronic Data Interchange or EDI. EDI requires each supplier and customer to set up a dedicated data link (between them), where ecommerce provides a cost-effective method for companies to set up multiple, ad links. Electronic commerce has also led to the development of electronic marketplaces where suppliers and potential customers are brought together to conduct mutually beneficial trade.

**1.2: Project Goals:**

In today’s world there is a growing emergence of e-commerce websites being used by customers to cater to their shopping needs, hence there is a need for management to realize that in order to support an emerging business with an online presence, a new and efficient database design will be required. The database needs to address some of the scenarios as described:

(1) Linking the customer details to the customer id and storing them to make the shopping experience easy and time saving.

(2) Giving the customers an option to select from various shipping options available.

(3) Keeping a track of privilege customers who can have access to certain things such as viewing the products.

(4) Enabling customer feedback on the products to better understand their needs.

(5) Updating the inventory to display the current status of products available.

(6) Enabling the option of tracking products that were returned or called back to improve servicing and enhancing the customer satisfaction.

**Roles:**

1: The user is a person who has several privileges on tables and can modify in timely manner. The different types of roles that are supported in an ecommerce model are:

1. Administrator:

The administrator is a person who keeps track of all the transactions. He has complete privilege over the system and can coordinate with the respective user for a particular modifications required

1. Manager:

Manager will interact with the customer and gather their needs and also he will keep an eye on the orders generated and shipped regularly to the respective warehouses.

1. Warehouse :

Staff employee has access to the inventory table to update the quantity of available products with consent from his manager. He will co-ordinate with the shipping employee and managers to process the shipments from the warehouse.

1. Shipping:

He will handle the shipping part and coordinates with the warehouses once the order is processed. He also keeps the track of orders that is whether it’s rejected or approved.

1. Customer:

Customer is the one who purchases the products. He can modify his shipping address, select shipping method, payment options.

3) Normalization:

Normalization is a process in the database design process which is done to ensure that the right data is in the right table. It helps to overcome the update problems. There are various forms of normalization namely 1NF, 2NF, 3NF, 4NF and 5NF. Another type of normalization is BCNF.

1NF ensures that there are no multi-valued attributes in any table.

2NF is used to eliminate any partial dependencies and ensure that all non-key columns depend on the entire primary key.

In Eco system model the primary key is auto incremented so that every row has a unique PK which behaves as an id or foreign key for another table. Hence the tables are already in 2NF.

3NF ensures that there are no transitive dependencies and that no non-key column depends on no other non-key column.

Normalization steps in Ecosystem model:

1. In person table the address is a composite attribute that is it may have multiple values like street, city, state etc.

Also the person could have more than one address which will violate the rules of 1NF hence the address is made as separate table.

1. The order table initially was designed to include billing. With the provision of a bill to be cleared in multiple transactions the billing could have multiple values and hence it has been made a separate entity to bring it to 1NF. The new billing table also has various columns for billing date, amount and card details to keep a track of the transaction details.

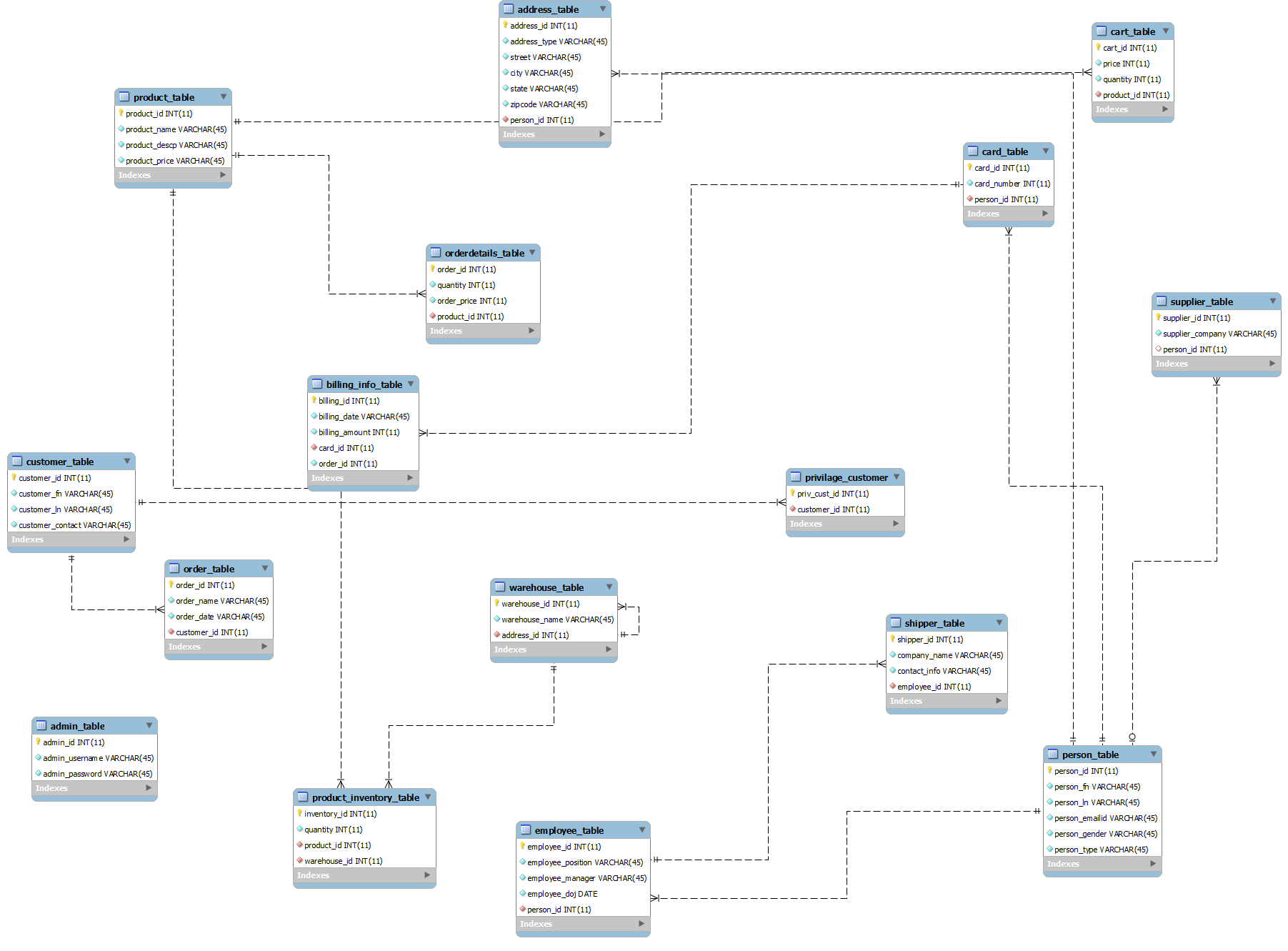
**4) Business Rules:**

Following business rules are required for an online ecosystem model:

1. Every person in the system is either a customer, supplier or employer.
2. Every person will have at least one address, he can have more than one.
3. Privilege rights are given to some customers only.
4. Every address will be associated with a person or a warehouse.
5. Every product must come from at least one warehouse but every warehouse may or may not produce a product.
6. Every order will have at least one product but a product may or may not belong to any orders.
7. Every product may or may not have a review and every customer may or may not give a review.
8. Every card must belong to at least one bill and one person.
9. Every bill will have one card.

10) Every order detail must have only one shipping id but one shipping id may have multiple orders.

1. EER- Diagram:



1. Views:
2. View for manager to see the product details:

Create view v\_product\_details as

SELECT

p.product\_id as product\_id,

p.product\_name as product\_name,

p.product\_desp as product\_description,

c.category\_name as category,

c.category\_descp as category\_description,

p.unit\_price as price,

p.unitweight as weight,

p.product\_size as size,

pi.product\_quantity as available\_quantity

from product\_t p

left join category\_t c

on p.category\_id = c.category\_id

left join product\_inventory pi

on p.product\_id = pi.product\_id;

1. View to see the order details of customer

create view order\_details as

SELECT

o.order\_number as order\_number,

o.customer\_id as cutomer\_id,

c.c\_fname as first\_name,

c.c\_lname as last\_name,

oi.product\_id as product\_id,

oi.quantity as quantity,

b.total\_amount as amount

from order\_t o

inner join customer\_t c

on c.customer\_id = o.customer\_id

inner join order\_table\_t oi

on o.order\_id = oi.order\_id

inner join billing b

on oi.order\_id = b.order\_id;

1. Stored Procedure:

**Procedure for inserting a customer/employee/supplier into the system**

Delimiter //

Create procedure p\_insert\_personInTable(ssn int(11), gender varchar(45), dob date, person\_type varchar(45), c\_fname varchar(45),c\_lname varchar(45),

employee\_doj date, company\_name varchar(45),shipment\_date date)

begin

SET @perosn\_id = (Select max(person\_id) from person\_t) + 1;

insert into person\_t

values (@person\_id, ssn, gender,dob,person\_type);

SET @perosn\_id = (Select max(person\_id) from person\_t);

if (person\_type = "Customer")

then

insert into customer\_t

values (null,c\_fname, c\_lname, @person\_id);

else if (person\_type = "Employee")

then

insert into employee\_t

values (null,@person\_id,employee\_doj, c\_fname );

else if (person\_type = "Supplier")

then

insert into supplier\_t

values (null,company\_name,@person\_id);

end if;

end if;

end if;

end; //

1. Triggers And Backups:
2. **Trigger to backup product Table.**

delimiter //

create trigger t\_productbackup1

before delete

on product\_t

for each row

begin

insert into t\_productbackup

select

product\_id,

product\_name,

product\_desp,

unit\_price,

unitweight,

product\_size,

category\_id

from product\_t;

end;//

1. **Trigger to backup order table**

delimiter //

create trigger t\_order\_backup

before delete

on order\_t

for each row

begin

set @order\_id = credit\_card\_torder\_id;

set @customer\_id = customer\_id;

insert into order\_backup

select

@order\_id,

cust.c\_fname,

cust.c\_lname,

cust.person\_id

from order\_t, customer\_t cust

where @customer\_id = cust.customer\_id;

end;

//

1. Privileges:

1 Create User ‘Manager’@%

GRANT SELECT, INSERT, UPDATE ON product\_t TO ‘Manager’ @%

GRANT SELECT, INSERT, (warehouse\_id, employee\_id), UPDATE ON product\_t TO ‘Manager’ @%

GRANT SELECT, INSERT, UPDATE ON product\_t TO ‘Manager’ @%

GRANT SELECT, INSERT, UPDATE ON product\_inventory TO ‘Manager’ @%

GRANT SELECT, INSERT, UPDATE ON customer\_t TO ‘Manager’ @%

GRANT SELECT, INSERT, UPDATE ON order\_t TO ‘Manager’ @%

2 Create User ‘Customer’@%

GRANT SELECT ON s\_product\_t TO ‘Customer’@%

GRANT SELECT ON s\_product\_category TO ‘Customer’@%

GRANT SELECT, INSERT ON s\_billing TO ‘Customer’@%

GRANT SELECT, INSERT (card\_id, card\_no) ON s\_card\_t TO ‘Customer’@%

GRANT SELECT (shipment\_date, shipment\_type) ON s\_shipper TO ‘Customer’@%

GRANT SELECT, INSERT, UPDATE (c\_fname, c\_lname, gender, email\_id, phone, street, city, state, zipcode) ON s\_person\_t