

DBMS_PROJECT SUBMISSION

full project

<<https://github.com/harshkb/ClassroomShoppers>>

E-R Diagram

Relational Schema

Submitted by

Anubhav Jain 15114014

Aman Agarwal 15114006

Devesh Masane 15114023

Dhanraj Sahu 15114024

Harsh Kumar Bansal 15114033

Database report Part2

Queries for ClassroomShopper projects

For header file.

```
SELECT Category_id,categoryName FROM category;SELECT  
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;
```

For index file

```
SELECT Category_id,categoryName FROM category;SELECT  
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT productName,  
newPrice, smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart natural join  
Product WHERE Cart.email_id = ?;SELECT sum(newPrice * quantity) as total FROM Cart  
natural join Product WHERE Cart.email_id = ?",[user_id,user_id]
```

For Account-order

```
SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT order_no,
timeStamp, status, newPrice FROM Orders natural join Product WHERE Orders.email_id = ?;
SELECT firstName, second_name, email_id, contactNo FROM USER WHERE email_id =
?;SELECT sum(newPrice * quantity) as total FROM Cart natural join Product WHERE
Cart.email_id = ?;SELECT productName, newPrice, smallImage, quantity, (newPrice * quantity)
as subtotal FROM Cart natural join Product WHERE Cart.email_id = ?
```

For Account Address

```
SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT address,
isdefault FROM Shipping WHERE email_id = ? ; SELECT firstName, second_name, email_id,
contactNo FROM USER WHERE email_id = ? ;SELECT sum(newPrice * quantity) as total
FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT productName, newPrice,
smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart natural join Product WHERE
Cart.email_id = ?
```

For About_us

```
SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category; SELECT sum(newPrice
* quantity) as total FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT
productName, newPrice, smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart
natural join Product WHERE Cart.email_id = ?
```

For wishlist

```
"SELECT productName, newPrice, smallImage FROM based_on_searches natural join Product
WHERE based_on_searches.email_id = ?;SELECT Category_id,categoryName FROM
category;SELECT Subcategory_id,subCategoryName,Category_id FROM
Sub_Category;SELECT sum(newPrice * quantity) as total FROM Cart natural join Product
WHERE Cart.email_id = ?;SELECT productName, newPrice, smallImage, quantity, (newPrice *
quantity) as subtotal FROM Cart natural join Product WHERE Cart.email_id = ?"
```

For FAQ

```
SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT sum(newPrice *
quantity) as total FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT
productName, newPrice, smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart
natural join Product WHERE Cart.email_id = ?
```

For Contact_us

```
SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT sum(newPrice *
```

quantity) as total FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT
productName, newPrice, smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart
natural join Product WHERE Cart.email_id = ?

For site_map

SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT sum(newPrice *
quantity) as total FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT
productName, newPrice, smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart
natural join Product WHERE Cart.email_id = ?

For Cart

SELECT productName, newPrice, smallImage, quantity, (newPrice * quantity) as subtotal
FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT sum(newPrice * quantity)
as total FROM Cart natural join Product WHERE Cart.email_id = ?;SELECT
Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category

For Listing

SELECT Category_id,categoryName FROM category;SELECT
Subcategory_id,subCategoryName,Category_id FROM Sub_Category;SELECT
productName,product_id,subCategoryName,stock FROM Product NATURAL JOIN
Sub_Category WHERE Category_id= ? and subCategory_id= ?;Select subCategoryName from
Sub_Category WHERE subCategory_id = ? AND Category_id = ? ;SELECT productName,
newPrice, smallImage, quantity, (newPrice * quantity) as subtotal FROM Cart natural join
Product WHERE Cart.email_id = ?;SELECT sum(newPrice * quantity) as total FROM Cart
natural join Product WHERE Cart.email_id = ?

For Create-Account

INSERT INTO newDatabase.USER set ?;

For Login-Account

SELECT firstName,second_name FROM newDatabase.USER WHERE email_id =
""+email_id+"" and password = ""+password+"";

For Cart

SELECT * from Cart WHERE email_id=?;
INSERT INTO newDatabase.Cart set ?;

For Checkout

SELECT * from Cart WHERE email_id=?;
INSERT INTO newDatabase.Orders set ?;

For Filter Products by Price

SELECT * from Product WHERE newPrice>=? AND newPrice <= ?';

All these queries are written in server.js file

We use Node.js for this project.

We Also added signup and login functionaity.

Database Tables for ClassroomShopper

```
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_newDatabase |
+-----+
| Cart                    |
| IndexProduct            |
| Orders                  |
| Product                 |
| Region                  |
| Review                  |
| Seller                   |
| Sells                    |
| Shipping                |
| Sub_Category            |
| USER                    |
| based_on_searches       |
| category                |
+-----+
13 rows in set (0.00 sec)

mysql>
```

To execute code;

install nodejs , type this in terminal

->sudo apt-get install nodejs

->sudo apt-get install nodejs-legacy

->sudo apt-get install npm

open terminal of mockshopper directory

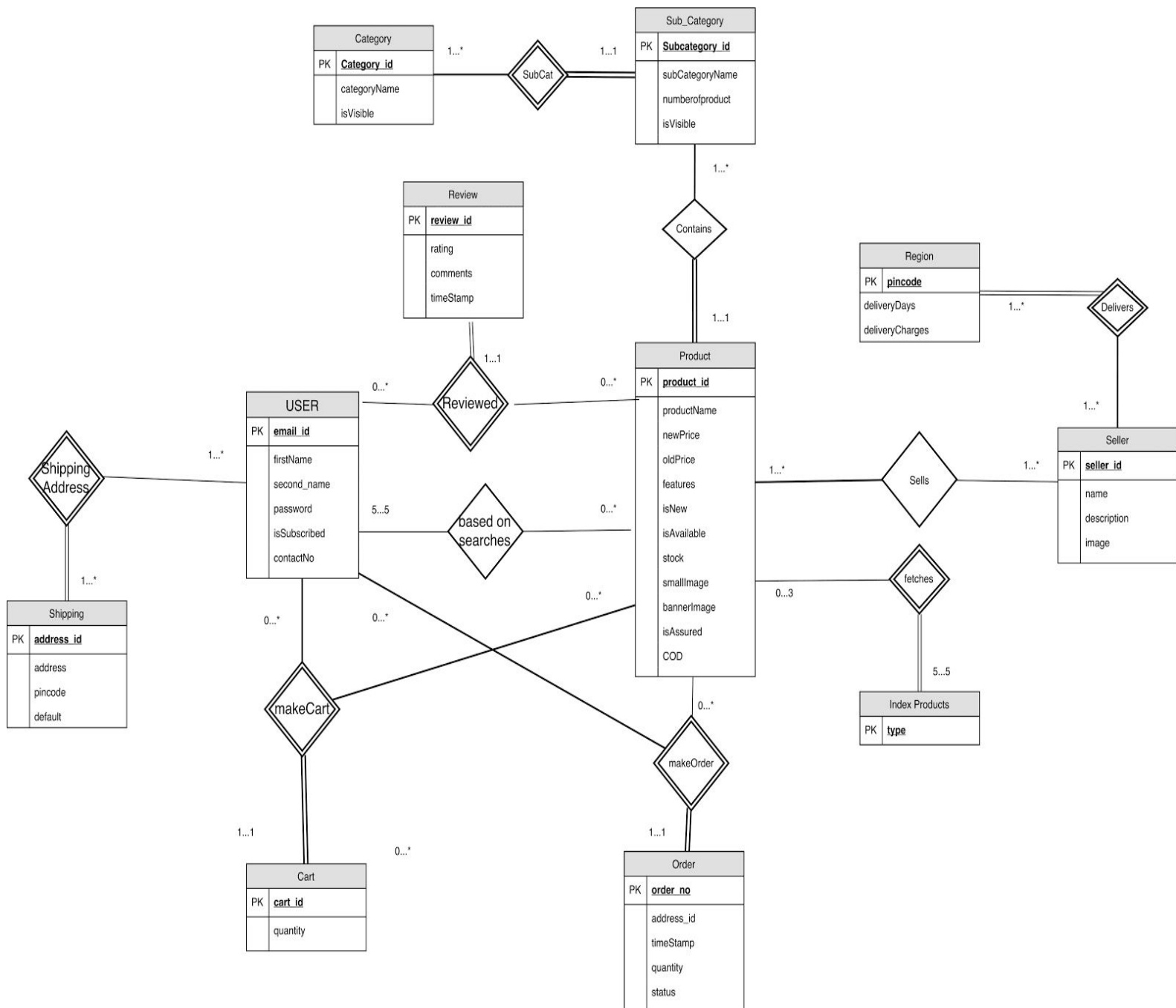
type ->npm install

type->node server.js

Open your browser at localhost:3000.

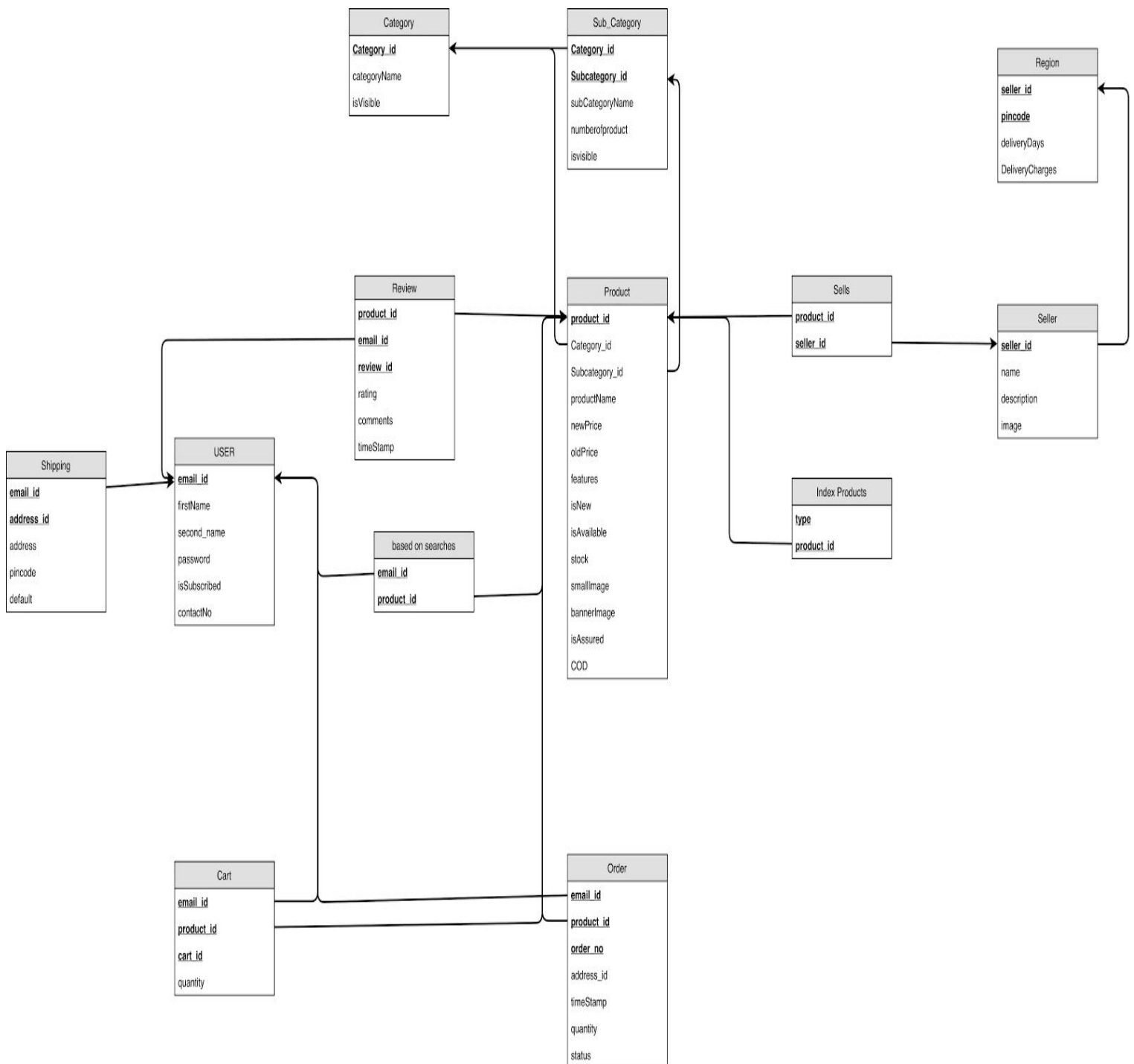
Report Of DataBase Structure

E-R Diagram



Link:- [ClassroomShoppers/dataStructureDocumentation/mockShoppersERD.pdf](#)

Relational Schema



Link :- ClassroomShoppers/dataStructureDocumentation/mockShopperSchema.pdf

ATTRIBUTES AND NOTATION

Attribute	Notation
Category_id	A
categoryName	B
isVisible	C
Subcategory_id	D
SubCategoryName	E
numberofproducts	F
isVisible	G
product_id	H
productName	I
newPrice	J
oldPrice	K
features	L
isNew	M
isAvailable	N
stock	O
COD	P
isAssured	Q
smallImage	R
bannerImage	S
isAssured	Q
seller_id	T
seller.name	U
description	V

image	W
pincode	X
deliveryDays	Y
deliveryCharges	Z
firstName	a
second_Name	b
email_id	c
password	d
isSubscribed	e
contactNo	f
address_id	g
address	h
pincode	i
default	j
review_id	k
review.timeStamp	l
rating	m
comments	n
cart_id	o
cart.quantity	p
order_no	q
address_id	r
order.timeStamp	s
order.quantity	t
status	u
type	v

Functional Dependencies of tables:

For Category table

$A \rightarrow BC$

$B \rightarrow AC$

Candidate Keys: A,B

For Sub_category table

$AD \rightarrow EFG$

$BD \rightarrow EFG$

$AE \rightarrow DFG$

$BE \rightarrow DFG$

Candidate Keys: AD, BD, AE, BE.

For Product table

$H \rightarrow IJKLMNOPQRS$

Candidate Keys: H

For Seller

$T \rightarrow UVW$

Candidate Keys: T

For Region table

$TX \rightarrow YZ$

Candidate keys: T,X

For USER table

$c \rightarrow abdef$

Candidate Keys: c

For Shipping table

$cg \rightarrow hi$

$h \rightarrow i$

Candidate Keys: cg

For Review table

$cHl \rightarrow kmn$

$cHk \rightarrow lmn$

Candidate Keys: cHl, cHk

For Cart table

$cHo \rightarrow p$

Candidate Keys: cHo

For Order table

$cHq \rightarrow rstu$

$cHs \rightarrow qrtu$

Candidate Keys: cHq , cHs .

Steps for finding minimal cover of these dependencies:

Step 1: Decomposing RHS:

$A \rightarrow B$

$A \rightarrow C$

$B \rightarrow A$

$B \rightarrow C$

$AD \rightarrow E$

$AD \rightarrow F$

$AD \rightarrow G$

$BD \rightarrow E$

$BD \rightarrow F$

$BD \rightarrow G$

$AE \rightarrow D$

$AE \rightarrow F$

$AE \rightarrow G$

$BE \rightarrow D$

$BE \rightarrow F$

$BE \rightarrow G$

$H \rightarrow I$

$H \rightarrow J$

$H \rightarrow K$

$H \rightarrow L$

$H \rightarrow M$

$H \rightarrow N$

$H \rightarrow O$

$H \rightarrow P$

$H \rightarrow Q$

$H \rightarrow R$

$H \rightarrow S$

$T \rightarrow U$

$T \rightarrow V$

$T \rightarrow W$

$TX \rightarrow Y$

$TX \rightarrow Z$

$c \rightarrow a$

c->b
c->d
c->e
c->f
cg->h
cg->i
cg->j
h->i
cHl->k
cHl->m
cHl->n
cHk->l
cHk->m
cHk->n
cHo->p
cHq->r
cHq->s
cHq->t
cHq->u
cHs->r
cHs->q
cHs->t
cHs->u

Step 2: Cancelling redundant dependencies. Cancelled dependencies are:

A->C
AD->E
AD->F
AD->G
BD->F
BD->G
AE->D
AE->F
AE->G
cg->i
cHl->m
cHl->n
cHq->r
cHq->t
cHq->u

Step 3: Checking left side of the dependencies if they can be reduced.

Dependencies that are reduced :

BD→E to B→E
cHl→k to cH→k
cHk→l to cH→l
cHk→m to cH→m
cHk→n to cH→n
cHq→s to cH→s
cHs→q to cH→q
cHs→r to cH→r
cHs→t to cH→t
cHs→u to cH→u

Therefore final minimal cover:

A→B
B→ACE
BE→DFG
H→IJKLMNOPQRS
T→UVW
TX→YZ
c→abdef
cg→hj
h→i
cH→klmnqrstu
cHo→p

TABLES:

Category(A,B,C)
Sub_category(A,D,E,F,G)
Product(H,I,J,K,L,M,N,O,P,Q,R,S)
Seller(T,U,V,W)
Region(T,X,Y,Z)
Sells(T,H)
User(a,b,c,d,e,f)
Shipping(c,g,h,i,j)
Review(c,H,k,l,n,m)
Cart(c,H,o,p)
Order(c,H,q,r,s,t,u)
basedOnSearches(c,H)
indexProduct(v,H)

Final functional dependencies:

A->B
B->ACE
BE->DFG
H->IJKLMNOPQRS
T->UVW
TX->YZ
c->abdef
cg->hj
h->i
cH->klmnqrstu
cHo->p

Prime Attributes: A,B,D,E,H,T,U,V,W,X,c,g,l,k,o,q,s

Non Prime Attributes: C,F,G,I,J,K,L,M,N,O,P,Q,R,S,Y,Z,a,b,d,e,f,h,i,j,m,n,p,r,t,u,v

Normalization of tables:

1NF normalisation

All the above given tables are already in 1NF, because all are in atomic form.

2NF normalisation

All the tables are already in 2NF because there is no partial dependencies among FDs.

3NF normalisation

There is transitive dependency in table **Shipping(c,g,h,i,j)**

cg→h,

h→i (both h and i are non-prime attribute so it is transitive dependency)

So we decompose this table as.,

Shipping_address(c,g,h,j)

address_pincode(h,i)

All others Tables are already in 3NF because there is no transitive dependency.

After decomposition 3NF forms of all tables are:

Category(A,B,C)

Sub_category(A,D,E,F,G)

Product(H,I,J,K,L,M,N,O,P,Q,R,S)

Seller(T,U,V,W)

Region(T,X,Y,Z)

Sells(T,H)

User(a,b,c,d,e,f)

Shipping_address(c,g,h,j)

address_pincode(h,i)

Review(c,H,k,l,n,m)

Cart(c,H,o,p)

Order(c,H,g,r,s,t,u)

basedOnSearches(c,H)

indexProduct(v,H)

BCNF Normalisation

All the new formed tables are in BCNF because the left side of the dependencies of all the tables are superkeys.

Since there is no multi-valued dependency and join dependency, so the given tables are already in 5NF.

So, final tables are:

Category(A,B,C)
Sub_category(A,D,E,F,G)
Product(H,I,J,K,L,M,N,O,P,Q,R,S)
Seller(T,U,V,W)
Region(T,X,Y,Z)
Sells(T,H)
User(a,b,c,d,e,f)
Shipping_address(c,g,h,j)
address_pincode(h,i)
Review(c,H,k,l,n,m)
Cart(c,H,o,p)
Order(c,H,q,r,s,t,u)
basedOnSearches(c,H)
indexProduct(v,H)