**Project:** Creating a database for ‘A Varnin Story’ business using PostgreSQL9.6.

CIS 556 – DBMS – Final Project

Vinyasa Kottapally

Date – 12-08-2020

A Varnin Story – A clothing sales business operated from Facebook and Instagram *handles. Products are obtained from various suppliers and are sold in different countries. Payments are obtained via bank transfer in USA and India.*

**Data model *–*** *ER Model*

**Purpose of the database –**

1.Organize and store customers and potential customers information.

2. Modify data concurrently.

3. extract analytical information to make assist in decision making.

4. Define multiple tables with complex information and retrieve all related data easily.

**List of Tasks that I intended to deliver on project checkpoint day.**

1. Client profile for whom the database is designed.

2. Database rules

3. High Level project description

4. ER diagrams describing data model (‘attributes’, schema etc.)

5. Sample data (‘This is a data from a real time clothing business that I operate and some of the data has been modified for privacy concern. The idea is to cover all possible known corner cases along with some Realtime data)

6. DDL statements

**List of major Entities with primary keys**

∙ Customers

∙ Products

∙ Shipping

∙ Orders

∙ Payments

∙ supplier

**Database Rules**

* All the tables (Customers, Products, Shipping, Orders, Payments) contain precise data and will be identified by one primary key
* Shipping is assumed to be from single service and delivery date is a derived attribute from shipping date. (Products are assumed to be shipped in Flat rate boxes across the USA )

**Cardinality Constraints**

Orders – payments (1-1);

∙ Each order must be associated with one payment

∙ Each payment must be associated with one order

Products-Shipping (M-1);

∙ Each product can have at most 1 tracking number

∙ Each tracking number can be associated with multiple products

Orders-Customers (M-N);

∙ Each order must have at least one customer associated

∙ Each customer may or may not have any orders associated.

∙ Even if a customer has orders’, there can be more than one order for each customer

Customers – products (M-N);

∙ Each product can be associated with exactly one customer

∙ Each customer can be associated with one or more products. Participation is not mandatory as customers and products are databases which hold much more data than purchases products

**Key Constraints**

1.order Price >0

2. check (‘(‘age>=21); and (‘age<=100););

3. check(‘ (‘country =USA ); or (‘country = India);

***Entities and Attributes***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Customer | Payments | Orders | Products | Shipping | Purchase | Supplier |
| cid | ref\_num | order\_id | pid | pid | cid | sup\_id |
| name | account | cid | type | shipping\_date | pid | sup\_name |
| city | country | ref\_num | fabric | delivery\_date |  |  |
| phone\_num | paid\_amt | order\_price | color | shipping cost |  |  |
| age |  |  | purchase | tracking\_num |  |  |
| gender |  |  |  |  |  |  |

# ER Diagram- A Varnin Story

*e*

have

>0,N

Orders

Customers

M

=1

M



purchase

have

=1

N

M



Products

have

Payments

<=1

>0



Supplier

sent

Shipping

=1

# Source Code for creating a database.

create table customer(

cid int primary key,

name varchar(20) not null,

city varchar(20) not null,

phone\_num char(10) not null,

age int check ((age>=21) and (age<=100)),

gender char(1)

);

create table payments(

ref\_num int primary key,

account varchar(10) not null,

country varchar(20) check( (country ='usa' ) or (country = 'india')),

paid\_amt bigint

);

Create table orders (

order\_id int primary key,

cid int references customer(cid) not null,

ref\_num int references payments(ref\_num) not null, order\_price bigint check (order\_price>0) not null, discount int

);

create table products(

pid varchar(7) primary key,

type varchar(15) not null,

fabric varchar(15) not null,

color varchar(10) not null,

purchase\_price bigint

);

create table shipping(

pid varchar(7) references products(pid) not null, shipping\_date varchar(10) not null,

 delivery\_date varchar(10),

 price bigint,

 tracking\_num int

 );

create table purchase (

 cid int references customer (cid) not null,

 pid varchar(7) references products(pid) not null  );

create table Supplier (

 sup\_id int primary key,

 sup\_name varchar(20)

 );

*NOTE – Data inserted into the database is attached in a different file.*

# Queries executed

1. Find the names of customers who ordered products less than 200$ and more than 100$

select c.*name, o*.order\_price from orders as o

join customer as c

on o.cid = c.cid

where order\_price >100 and order\_price<200

order by order\_price

|                name | order\_price |

|---------------------|-------------|

|              Sarada |         120 |

|           chandana  |         121 |

|              priya  |         134 |

| georgette paithani  |         144 |

|             lavanya |         145 |

|            sridevi  |         157 |

|               avani |         163 |

|             shilpa  |         165 |

|               kusum |         167 |

|               vani  |         169 |

|               Rekha |         175 |

|          sravanthi  |         185 |

|              akila  |         186 |

|            deepika  |         187 |

|             sravya  |         193 |

|              neeha  |         195 |

|             shwetha |         195 |

2. Write a query to list down most popular cities from which orders were placed.

select  c.city ,count(c.city)

from

orders as o left join

customer as c

on o.cid = c.cid

group by c.city

order by count(c.city) desc

|           city | count |

|----------------|-------|

|      wisconson |     3 |

|          texas |     2 |

|     Los Angles |     2 |

|      hyderabad |     2 |

|     washington |     1 |

|           novi |     1 |

|          india |     1 |

|        chittor |     1 |

|        khammam |     1 |

|         dallas |     1 |

|      Kalamazoo |     1 |

|         denver |     1 |

|            usa |     1 |

|        Chicago |     1 |

|     south Lyon |     1 |

| north carolina |     1 |

|       Delaware |     1 |

|        Khammam |     1 |

|            SFO |     1 |

3. who is most probable to order another product from purchase history and timelines of the customer.

       -- I'm looking at monthly historical of 3 months data to establish pattern for purchases.

select \* into temp1

from

purchase as p

natural join

shipping as *s;*

select t1.cid into temp2

from

temp1 as t*1, temp*1 as t2, temp1 as t3

where t1.pid > t2.pid and t2.pid > t3.pid

and t1.cid = t2.cid and t2.cid = *t3.cid;*

select customer.name from

temp2

left join

customer

on temp2.cid = customer.cid

4.. List the products are ordered the most in desc order.

select pr.type , count(age)

from purchase as p

left join

customer as c

on c.cid = p.cid

left join

products as pr

on pr.pid = p.pid

group by pr.type

order by count(age) desc

| **type** | **count** |
| --- | --- |
| Dress | 2 |
| Saree | 2 |
| Baby Clothes | 1 |