Order Management Schema details

This document captures the scenario of simple order management functionality of an online retail store.

Typical purchase scenario: A customer places an order for N products specifying quantity for each line item of the order. Every product belongs to a product class (or category). All products ordered in one order, are shipped to customer's address (in India or Outside) by a shipper in one shipment. Order can be paid using either Cash, Credit Card or Net Banking.

There can be customers who may not have placed any order. Few customers would have cancelled their orders (As a whole order, no cancellation of individual item allowed). Few orders may be 'In process' status. There can also be products that were never purchased.

Shippers use optimum sized cartons (boxes) to ship an order, based on the total volume of all products and their quantities. Dimensions of each product (L, W, H) is also stored in the database. To keep it simple, all products of an order are put in one single appropriately sized carton for shipping.

Project MYSQL

You are hired by a chain of online retail stores "Reliant retail limited". They provided you with orders database and seek answers to the following queries as the results from these queries will help the company in making data driven decisions that will impact the overall growth of the online retail store. All the questions come under MYSQL and the queries should be executed in MYSQL. (SQL script- orders.sql)

The best practice to understand the tables is to run each of them one by one and try to understand each table columns and its values. Also, try to compare tables to understand if there is any primary key / common columns on which we can join and get some more understanding about data.

For example:

What does Product_Quanity_Available in product table mean? Is it the total products (including orders placed i.e. sold (product_quantity in order_items table) and not sold?

How can we figure out this?

Let's run a query to see all the columns in product table, then run a query to see all columns of order_items table. Now, join both the tables and check for different conditions like if product_quantity_available equal to; less than or greater than product_quantity. Here, we found that all the conditions are satisfied which assures us that product_quantity_available contains only those products which are currently available excluding the once which are sold.

This is one way to understand the tables as in Industry we might not have complete information.