**Databases: Modelling Exercise**

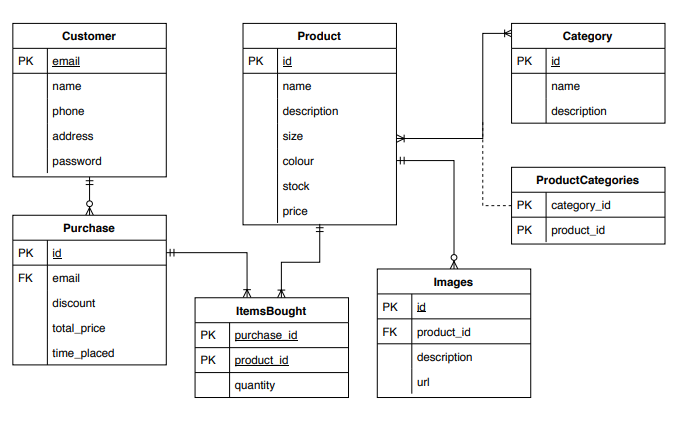
**ASOS**

The website I have chosen to model is ASOS, which is an online clothing retailer on which users can browse clothes by categories (e.g. “newest”, “men”, “shirts”) or search them by name, then filtering by size, colour and/or price.

Users can create accounts with an email and password, providing a name, address, and phone number. Through the account, users can see the items they’ve bought through their order history.

The main entities I have modelled are the customer, product, category, and images, with relational tables allowing customers to purchase products, and allowing products to belong to more than one category. The way it is modelled allows a customer to purchase multiple different products through one purchase id, instead of a single relational purchases table which would only allow one product to be bought at a time. The images table doesn’t need a relational table because each image can only belong to one product, therefore it simply needed a foreign key displaying the product id that it belongs to.

Modelling these as shown below gives the user the ability to create an account using their email address as their unique id, and by giving a password (which would be encrypted when stored in the database). They can then browse products by category or name, then adding them to their order and checking out, which produces a unique purchase id (which relates to a number of items in the ItemsBought table) detailing the total price including any discount, and the time the order was placed (this would likely feed in to another theoretical table which deals with delivery).



A few different examples of user cases are:

1) Viewing all the items ordered by a certain user in the past 6 months

SELECT Pur.email, Pur.id, Pro.name, Pur.time\_placed FROM Purchase Pur

JOIN ItemsBought ON Pur.id=ItemsBought.purchase\_id

JOIN Product Pro ON ItemsBought.product\_id=Pro.id

WHERE Pur.email = ? AND Pur.time\_placed >= DATE\_SUB(now(), INTERVAL 6 MONTH);

2) Browsing all men’s medium white t-shirts that are in stock, sorting by price

SELECT Pro.id, Pro.name FROM Product Pro

JOIN ProductCategories PC ON PC.product\_id=Pro.id

JOIN Category Cat ON Cat.id=PC.category\_id

WHERE Cat.name = ‘men’ AND Cat.name=’t-shirt’

AND size=’medium’ AND stock > 0

ORDER BY price ASC;

3) Selecting all the images of orange shirts to display on the website (this user is the person creating the website)

SELECT Im.url FROM Images Im

JOIN Product Pro ON Im.product\_id=Pro.id

JOIN ProductCategories PC ON PC.product\_id=Pro.id

JOIN Category Cat ON Cat.id=PC.category\_id

WHERE Cat.name LIKE ‘%shirt%’

AND Pro.colour=’orange’;