**Online shopping Website Database**

**Database Specification: Purpose, Business Problems addressed and Business Rules**

Group 11:

* Snehal Pathak
* Sowmya Benakappa
* Aleesha Reez
* Sahana Kannahallimole Somanna

**Database Purpose:**

The purpose of the database is to maintain the data for products, suppliers, distributors and customers. Customers will be able to purchase the products; Admins will be able to maintain products, distributors and supplier details.

**Business Problems Addressed:**

* Online shopping website enables customers to shop at their comfort reducing overhead of maintaining physical stores
* Allows administrative staff to anticipate merchandise inventory needed for on demand clothes
* Support team helps to handle all customer queries
* Allows administrative staff to generate detailed reports including product suppliers and distributors
* Allow customers to search, sort and add products to cart
* Allows customers to view products based on their preferences (e.g. sizes, types, categories)
* Customer can place order and view order summary for the same
* Admin will take care of adding products, suppliers and distributors

**Business Rules:**

* A person can have only one address
* A customer can add one or many products to cart
* A customer can have zero or one cart
* A customer can ask zero or more customer queries
* A customer can place zero or many orders
* A customer can be Premium member based on no. of orders (No. of Order > 3)
* An admin can add zero or many suppliers, distributors, products and product details
* An admin can address zero or many customer queries
* Supplier and distributor can have only one address
* Supplier can have one or many products
* An order has only one shippingId
* An order can have one or many products
* A product will have one or many sizes and colors

**Design Requirements (Credit to Professor Simon Wang):**

* Use Crow’s Foot Notation.
* Specify the primary key fields in each table by specifying PK beside the fields.
* Draw a line between the fields of each table to show the relationships between each table. This line should be pointed directly to the fields in each table that are used to form the relationship.
* Specify which table is on the one side of the relationship by placing a one next to the field where the line starts.
* Specify which table is on the many side of the relationship by placing a crow’s feet symbol next to the field where the line ends.

**Design Decisions:**

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Why Entity Included** | **How Entity is Related to Other Entities** |
| **Person** | The purpose of Person entity is to avoid redundancy data for admin and customer.  It includes basic information of a person (e.g. firstName, lastName, contactNumber, emailId, address) | An address can be associated to one person. ‘addressId’ is added as foreign key to have one to one relationship. |
| **Customer** | One of the primary purposes of the database is to collect information about the customer. | A customer can be a person so it will include the person details hence included personId as foreign key. It is related as one to one relationship. |
| **Admin** | Another key function of the database is to collect admin data. Admin functionality includes add and maintain product inventory, suppliers and distributors information. Admin will also address to the customer queries. | An admin can be a person so it will include the person details hence included personId as foreign key. It is related as one to one relationship. |
| **Supplier** | Supplier entity is used to keep track of all the suppliers who supply products. | It is admin’s responsibility to add supplier’s details. ‘adminId’ is included as a foreign key to get to know which admin is responsible for the specific supplier. One admin can manage many suppliers and is related with one to many relationships.  An address can be associated to one supplier. ‘addressId’ is added as foreign key to have one to one relationships. |
| **Distributor** | Distributor entity is used to keep track of all the distributors who delivers the products. | It is admin’s responsibility to add distributor’s details. ‘adminId’ is included as a foreign key to get to know which admin is responsible for the specific distributor. One admin can manage many distributors and it is related with one to many relationships.  An address can be associated to one distributor. ‘addressId’ is added as foreign key to have one to one relationships. |
| **Cart** | Cart entity holds product information for a customer. | As zero or many products can be added to cart, it relates to product details entity as one to many relationships.  A customer will add products to the cart. To track which cart belongs to which customer, ‘customerId’ is added as a foreign key. A customer can have only zero or one cart so it is related as one to one relationship. |
| **Product** | Being the key entity, Product entity holds the product information (e.g. productName, description, price, etc.) | It is admin’s responsibility to add product details. ‘adminId’ is included as a foreign key to get to know which admin is responsible for the specific product. One admin can manage many products and it is related with one to many relationships. |
| **ProductDetails** | ProductDetails entity captures the various details of a product like colors, sizes, quantities and categories | ‘productDetailsId’ is the surrogate key which represents product information and it is related as one to one relationship with products, size, color, and category.  ‘supplierId’ is added as a foreign key to track which supplier supplies each product and is related as one to one relationship. |
| **Order** | As one of the key entities, the Order entity holds the information about the orders placed by customer including customer information, shipping details and total amount payable. | Order entity will track the orders placed by the customer and each order is associated to only one customer hence it is related as one to one relationship.  ‘shippingId’ is added as a foreign key to track the shipping details of a particular order and it is related as one to one relationship. |
| **OrderDetails** | OrderDetails entity provides the complete information about the product that are ordered by the customer. | An order can hold many products so an ‘orderId’ and ‘productDetailsId’ are added as a foreign key to get the information of a product in that order and it is related as one to one relationship. |
| **ShippingDetails** | ShippingDetails entity holds the order information that is been shipped including distributor details. | ‘distributorId’ is added as a foreign key to track which distributor delivers each product and is related as one to one relationship. |
| **Address** | Address entity holds the address information of person, suppliers and distributors. | The Address entity provides the atomic decomposition of the address.  An address can be associated to one person, supplier and distributor. |
| **Size** | Size entity holds the information about different sizes of the products. | One size can be related to many products; hence it is related as one to many. |
| **Color** | Color entity holds the information about different colors of the products. | One color can be related to many products; hence it is related as one to many. |
| **Categories** | Categories entity holds the information about different categories of the products. | One category can be related to many products; hence it is related as one to many. |
| **Support** | Support entity is included to address any customer queries. This entity includes the support person name, customer query and isResolved flag to know if the query has resolved or not. | ‘adminId’ is included as a foreign key to know which admin has addressed to a customer query and is one to one relationship.  ‘customerId’ is included to track which customer has asked the query and is related as one to one relationship. |
| **CustomerStatus** | This entity holds the premium customers only. For ex. If No of orders > 3 then the customer status is Premium Member. This helps in determining if the customer is premium or not. | ‘customerId’ is included as both primary and foreign key and it is related as one to one relationship and optional participation for customer. |