DATA WAREHOUSING UISING IBM CLOUD DB2 WAREHOUSE

Creating a data model for IBM Db2 Warehouse involves designing the structure of your database to represent the data you want to store and manage. Data modeling typically includes defining tables, columns, relationships, and constraints. Below is an example of a simplified data model for a hypothetical e-commerce system using Db2 Warehouse.

Entities:

1. Customers
2. Products
3. Orders
4. OrderDetails

Attributes:

1. Customers:
   - CustomerID (Primary Key)
   - FirstName
   - LastName
   - Email
   - Phone
   - Address
2. Products:
   - ProductID (Primary Key)
   - ProductName
   - Description
   - Price
   - Category
3. Orders:
   - OrderID (Primary Key)
   - CustomerID (Foreign Key referencing Customers)
   - OrderDate
   - Status (e.g., In Progress, Shipped, Delivered)
4. OrderDetails:
   - OrderDetailID (Primary Key)
   - OrderID (Foreign Key referencing Orders)
   - ProductID (Foreign Key referencing Products)
   - Quantity
   - Subtotal

Relationships:

1. One-to-Many Relationship between Customers and Orders:
   - Each customer can place multiple orders.
   - CustomerID in Orders is a foreign key referencing CustomerID in Customers.
2. One-to-Many Relationship between Orders and OrderDetails:
   - Each order can have multiple order details.

- OrderID in OrderDetails is a foreign key referencing OrderID in Orders.

3. One-to-Many Relationship between Products and OrderDetails:
   - Each product can be part of multiple order details.
   - ProductID in OrderDetails is a foreign key referencing ProductID in Products.

This data model represents a basic e-commerce system where customers can place orders, and each order consists of multiple products with associated quantities and subtotals. You can expand and refine this model based on the specific needs and complexity of your application.

To implement this data model in IBM Db2 Warehouse, you would use SQL statements to create the necessary tables, relationships, and constraints. Here's an example of SQL statements to create the tables:

```
-- Create Customers table

CREATE TABLE Customers (

    CustomerID INT PRIMARY KEY,

    FirstName VARCHAR(50),

    LastName VARCHAR(50),

    Email VARCHAR(100),

    Phone VARCHAR(20),

    Address VARCHAR(255)

);



-- Create Products table

CREATE TABLE Products (

    ProductID INT PRIMARY KEY,

    ProductName VARCHAR(100),

    Description VARCHAR(255),

    Price DECIMAL(10, 2),

    Category VARCHAR(50)

);
```

```
-- Create Orders table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    Status VARCHAR(20),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- Create OrderDetails table
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    Subtotal DECIMAL(10, 2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

This is just a starting point. You would need to consider indexing, constraints, and additional features as per your specific requirements and performance considerations.