Final Report: Best Buy

1. Project Overview:

Best Buy, as described by their company mission, is a “provider of technology products, services, and solutions.” Best Buy purchases the latest and greatest products in innovative technology from it’s suppliers, the major tech giants. From there as vaguely stated previously they provide all services and sales of their products such as marketing, repairs, servicing, customer care and product advising.

2. User Requirements:

- **Customer Supertype (Disjoint):**
- **For Customer :**
    - Access to purchase inventory in-store or online.
    - Customers have access to find transactions through an employee.
- CustomerID: (Unique Identifier):
        - Helps Identify and organize each customer
- Customer Name:
        - The Customer's Name who visits Best Buy
- Customer Address (Composite: Street, Unit Number, City, Zip, State):
        - The Home or Business location of the customer
- Phone Number(s)
        - Information to help contact the customer via verbal communication
- Email
        - Information to help contact the customer via digital messaging
- Customer Type
- **Subtype: Rewards member, Guest**
- **For Rewards Member:**
    - Customers can use their points to get an amount off any purchase.
    - To get rewards, you must be a member and have purchase history.
    - Employers use this system to get customers to continue shopping with them.
- MemberID: (Unique Identifier):
        - Find out what member number they are to be found in the system.
- Reward Balance
        - Balance of points that have been accumulated.
- MemberSince
        - When the customer became a member for rewards
- **For Guest:**
    - Access to purchase inventory, but customer will not accumulate any points. Will not keep records of purchases.
- (No Attributes)

- **For Employee:**
    - Access to purchase history of a customer
    - Access to resources to help find a customer's needs.
    - Find out inventory for products
    - Use knowledge for recommendations
    - Help release inventory (store pick-up, point-of-sale)
    - Have access to find reports of a store's performance.
    - Employees have access to all order numbers.
- EmployeeID: (Unique Identifier):
        - Helps Identify and organize each employee

Employee Address (Composite: Street, Unit Number, City, Zip, State)

- **For Salary:**
    - A system for employees to determine how they will get paid for the hours they worked.
    - Higher role employees get paid at a fixed amount.
        - (Derived: Hours * Hourly pay)
- Salary ID (Unique Identifier)
        - Specifies if employee is hourly payed or on salary
- Role (ex. Manager)
        - Distinguishes employee's title; advisor, manager, supervisor
- Status (Seasonal vs. Full time)
        - Specifies if employee is full time employee, part time, or seasonal
- Benefit
        - States what benefits each employee has: Dental, Vision, 401k, etc.
- **For Order:**
    - Keep track of overall purchases that were purchased online or in-store.
    - An order number is generated when revenue is recognized.
- OrderID: (Unique identifier):
        - Order number that is given when order is placed.
- Fulfillment (Pickup vs Delivery)
        - Bestbuy offers pickup (curbside) or ship to home for orders.
- Date
        - Date the product or service was purchased
- Billing Address (Composite: Street, Unit Number, City, State, Zip)
        - The customer's billing address when order is placed.
- Shipping Address (Composite: Street, Unit Number, City, State, Zip)
        - Location that service or product is arriving at.
- **For Order Line (Associative Entity):**
    - Warehouse employees have access to receiving products from vendors, which are counted in the store's inventory.
- Quantity
        - States number of products
- **For Product:**

  - Inventory that is for sale.
  - Products can be purchased by guests or members.
  - The product is counted and added into the store’s inventory.
  - Inventory can be ordered online or in-store.
- ProductID: (Unique Identifier):
    - Helps Identify and organize each product
- Manufacturer
    - Identifies the company that made the product
- Serial number
    - UPC of the product to track product back to manufacture origin
- Price
    - Specifies price of product
- Warranty/Protection plan?
    - Identifies if product has a warranty or not
- **For Payment:**
  - Currency is exchanged for a product or service.
- PaymentID: (Unique Identifier):
- Payment Type (Apple pay, Credit card, etc.)
    - Specifies what type of payment method is being used (card type, cash, rewards,etc.)
- Card Details (Composite: Expiration, Card number, CVD)
- **For Supplier/Vendor:**
  - Access to sales to determine popularity of a product.
  - Have access to request more inventory for stores that need more stock.
- SupplierID: Identifier
    - The vendor that is providing products to the store. Apple, Samsung, Sony, etc
- Address (Composite)
    - Address of the vendor or supplier
- Company name
    - Specifies the company being used
- Manufactured product
    - Specifies product being made by manufacturer
- **For Store:**
  - Location where employees can help customers with products
  - Stores allow customers to purchase products and take them home same-day.
  - Vendors/Supplier’s location to drop off inventory and display their models.
- Store ID: Identifier
- Address (Composite)
    - Identifies location of store
- Hours
    - Specifies the operating hours of the store
- **For Stock:**

    - Accurate count for stock, which helps determine the quantity available for customers to purchase.
- StockID: Identifier
- Quantity
        - Identifies number of products store has
- **District Manager:**
    - Access to hiring new employees.
    - Find out status of employees (on vacation, sick time, etc)
    - Access to store reports and labor hours.
    - Access to financial information.
    - Access to modify existing employee/customer patterns.
- Manager ID=unique identifier
- Name
        - Name of district manager
- Store ID
    - Identifies which store they are over
- Location
    - Identifies the location they are managing
- **Maintenance:**
    - Best Buy Services that are purchased to help with the life of a product.
    - Allows employees to go to the client's home to perform a service.
- ServiceID=unique identifier
- Service Type (install TV, appliance, computer service, car installation)
        - Specifies what type of service was done such as: installation, IT service, etc.
- OperatingHours
        - Hours that the repair/service will take.
- Cost
        - Specifies cost of Maintenance
- Location(In-home vs store)
        - Distinguishes if maintenance was done in store or at home

UPDATED user requirements: Previously they were entity descriptions. User Requirements were changed to be about what the user wants or what the business wants to do.

3. Business Rules:

- **Customer (Disjoint rule for Rewards member and guest):**
    - Customer can be a Rewards Member or a Guest
    - A Customer can place one or many Orders, but must have a minimum of one.
    - A Customer can have one or many Payment Methods, but must have a minimum of one.
    - A Customer can purchase one or many Products, but must purchase at least one Product.
    - A Customer can purchase none, one, or many Maintenance requests.

    - A Customer can visit one or many Stores.
- **Rewards Member**
    - A Rewards Member can not be a Guest.
- **Guest**
    - A Guest cannot be a Rewards Member
- **Employee**
    - One employee can have one salary min and max one salary.
    - An Employee can work at min 1 Store and max 1 store.
    - An Employee is supervised by one and only one District Manager
    - An Employee can manage one and only one Stock.
    - An Employee can work at one and only one Maintenance Department.
- **Salary**
    - A Salary can have min 1 Employee and max many Employees
    - A Salary can have min 1 District Manager and many District Managers.
- **Order**
    - An order can only be placed by min one customer and max 1 customer
    - An order can have one or many suppliers
    - An order can have one or many products.
    - An order can have one and only one order line
- **Order Line (Associative Entity)**
    - An order line can have one or many orders
    - An order line can have one or many products
- **Payment**
    - A payment method can have min 1 customer and max 1 customer
- **Supplier**
    - A supplier can have one or many Orders.
    - A supplier can make one or more Products.
- **Store**
    - A Store can have one or many Employees
    - A Store can have one and only one District Manager.
    - A Store can sell none, one, or many Products.
    - A Store can have none, one, or many Customers.
    - A Store can have one and only one Stock.
- **Stock**
    - Stock can have none, one, or many Products.
    - A Stock can have one or many Stores.
    - Stock can be managed by one or more Employees.
    - Stock can be verified by one and only one District Manager.
- **District Manager**
    - A District Manager can have one or many Stores.
    - A District Manager supervises one or many Employees.
    - A District Manager can verify one or more Stocks.
    - A District Manager can have one and only one Salary.

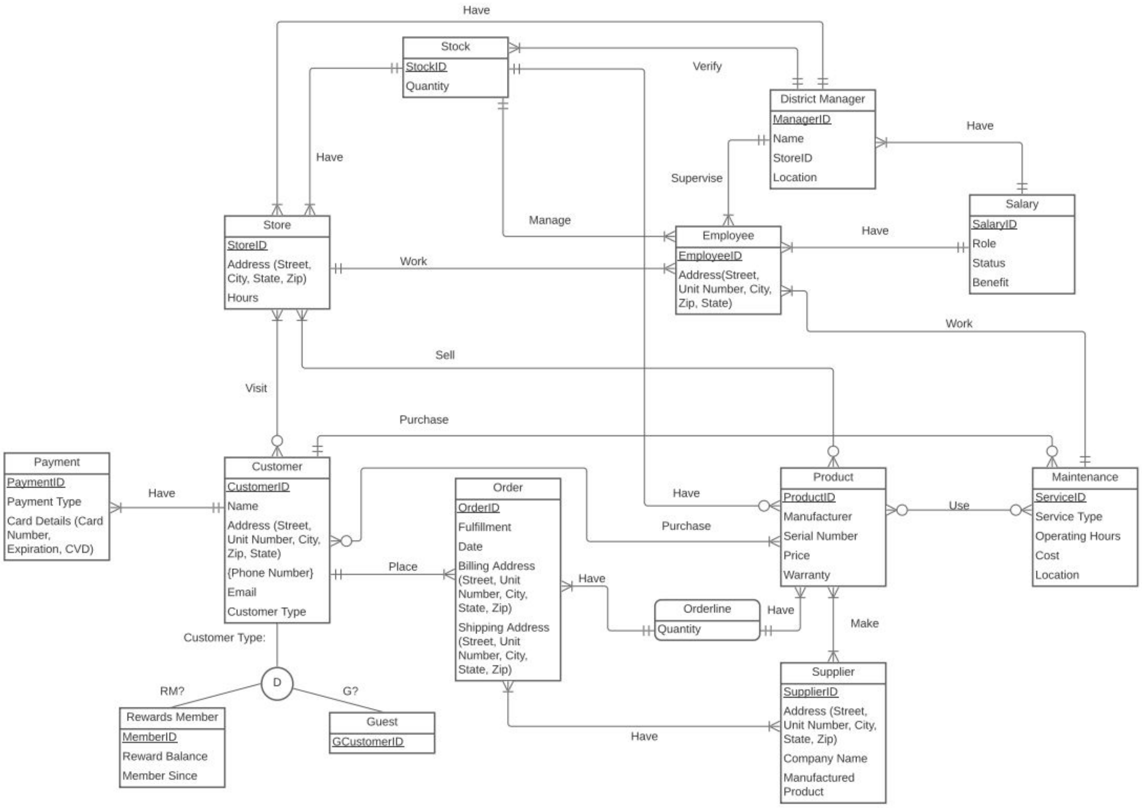
- **Maintenance**
    - Maintenance can use none, one, or many Products.
    - Maintenance can be purchased by one, and only one Customer.
    - Maintenance can have minimum of 1, or many Employees.
- **Product**
    - A Product be made from more than one Supplier, but must have a minimum of one.
    - A Product does not need a Maintenance order, but can have many.
    - A Product can be purchased by none, one, or many Customers.
    - A Product can be part of none or one Order Line
    - A product has one and only one Stock
    - A Product can be sold in one or many Stores

4. <u>Data Outputs (Reports)</u>

- Employee Hours Report
    - How many hours an employee has worked
- Customer Purchase Report
    - Looks up previous transactions that a customer has performed.
- Labor hours for Services
    - Keeps track of how many service hours fulfilled by an employee.
- Pending pick up report
    - Online orders or pick up orders that are pending to be picked up.
- Inventory Report
    - Shows stock for inventory that we have in store.
- Merchandising Report
    - Price tags that need to be updated, product that needs to be removed (discontinued products)
- Order from Supplier/Vendor Report
    - Inbound orders, for restocks, new products
- Personal Time Off Balance Report
    - Sick hours, vacation hours
- Customer Scheduling Report
    - Appointments for consultations, services
- Revenue Employee Report
    - How much revenue an individual employee has made the store in a day, month, year.
- Revenue Store Report
    - How much revenue the entire store has made in a day, month, year
- Employee Report

- ○ List of employees that work in the store

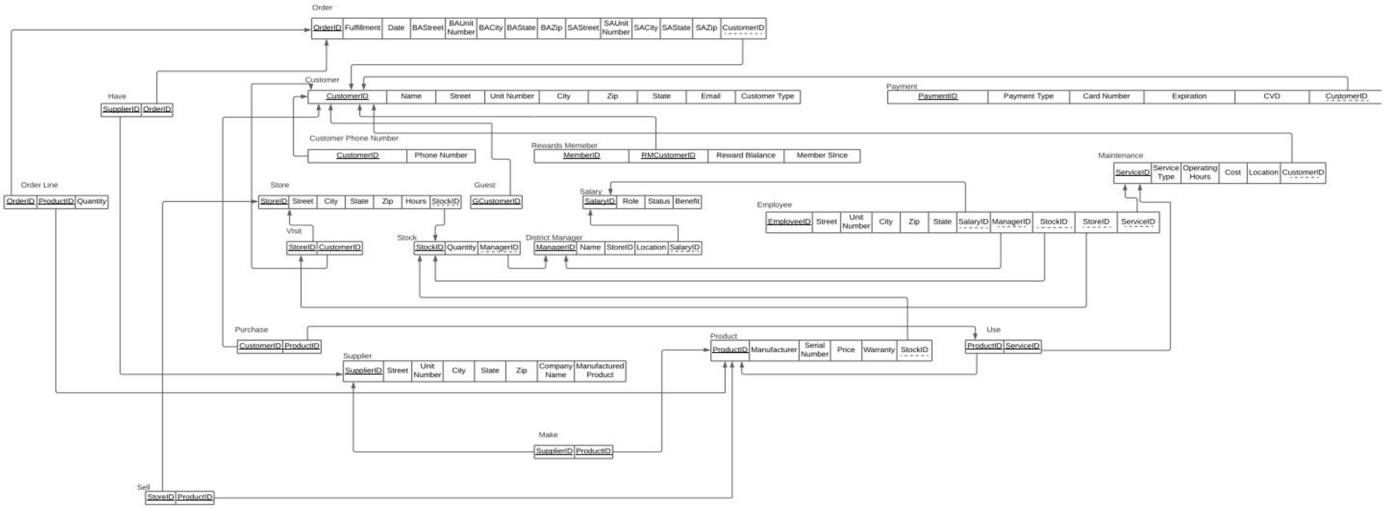
5. Conceptual Model



UPDATE:Previously we had Guest without it's own set of attributes. If we wanted to include "Guest" as a subtype we needed to have distinct attributes which is why we created a new attribute "GCustomerID."

UPDATE: Previously we had improper formatting for the binary relationships for the Orderline Associative entity. We improperly converted the many to many relationships. It has now been properly updated with 2 one to many relationships.

6. Logical Model



*A better view of our Logical Model can be found on our slideshow presentation

7. Database Implementation

DML Code:

Customer Table Insert statement:

```
INSERT INTO customer (customer.customerID, customer.Name, customer.Street,
customer.Unit_Number, customer.City, customer.Zip, customer.State, customer.Email,
customer.Customer_Type) VALUES(2, "Emmett Hart", "7318 East Washington Street", 745,
"Royersford", 19468, "Pennsylvania", "ehart@gmail.com", "Guest"),
(3, "Jeanette Harper", "9002 Roosevelt Street", 01, "Malvern", 19355, "Pennsylvania",
"jean.h@outlook.com", "Guest"),
(4, "Natasha Edwards", "8925 Old Iroquois Ave.", 16, "Boca Raton", 33428, "Florida",
"n_Edwards@hotmail.com", "Guest"),
(5, "Casey Martinez", "55 Shore Ave.", 601, "Laurel", 20707, "Maryland",
"c.martinez@yahoo.com", "Rewards Member"),
(6, "Louise Morton", "7024 Miles Ave.", 555, "Gulfport", 39503, "Mississippi",
"lmorton@yahoo.com", "Rewards Member"),
(7, "Frankie Neal", "716 Myrtle Street", 74, "Lake Jackson", 77566, "Texas",
"frankie_n@aol.com", "Guest"),
(8, "Mike Andrews", "2 Bedford Dr.", 123, "Norwood", 02062, "Maryland",
"M_Andrews@aol.com", "Rewards Member"),
(9, "Jean Maldonado", "32 Meadow Lame", 33, "Carpentersville", 60110, "Illinois",
"jean.m34@hotmail.com", "Rewards Member"),
(10, "Jeanne Figueroa", "513 Westport Avenue", 465, "Mason City", 50401, "Iowa",
"jf777@outlook.com", "Rewards Member"),
(11, "Randall Morales", "785 Shipley Ave.", 22, "Tualatin", 97062, "Oregon",
"randall.mor01@gmail.com", "Guest"),
(12, "Pauline Nichols", "7257 Pennington Lane", 812, "Hopkins", 55343, "Minnesota",
"Nichols.p@gmail.com", "Guest"),
(13, "Kim James", "26 Selby Circle", 26, "North Olmsted", 44070, "Ohio", "kjames@gmail.com",
"Rewards Member"),
(14, "Omar Alvarez", "890 Schoolhouse Ave.", 05, "Barberton", 44203, "Ohio",
"omar.alvarez@aol.com", "Guest"),
(15, "Cory Wong", "724 Brickyard Street", 632, "Hoboken", 07030, "New Jersey",
"c_wong@aol.com", "Guest"),
(16, "Vincent Malone", "463 Greenrose Street", 98, "Chesterfield", 23832, "Virginia",
"vincent_malone@outlook.com", "Guest"),
(17, "Harold Cole", "7365 W. Alderwood Drive", 365, "Staunton", 24401, "Virginia",
"h.cole42@outlook.com", "Guest"),
(18, "Lloyd Collins", "7974 S. Rockcrest Drive", 974, "Sanducky", 44870, "Ohio",
"llcollins@mac.com", "Rewards Member"),
(19, "Alvin Garner", "8284 Logan St.", 222, "Cumberland", 02864, "Rhode Island",
"alvin.g@comcast.net", "Guest"),
(20, "Jeanne Figueroa", "561 Woodland Street", 16, "Bowie", 20715, "Maryland",
"j.fig635@verizon.net", "Guest"),
(21, "Randall Morales", "9068 Wild Horse St.", 68, "Oxnard", 93035, "California",
"randall_m@mac.com", "Rewards Member"),
(22, "Pauline Nichols", "8897 East Bayport Avenue", 512, "Canandaigua", 14424, "New York",
"pnichols@gmail.com", "Rewards Member"),
(23, "Kim James", "7201 Devon Ave.", 102, "Oak Forest", 60452, "Illinois",
"kim.james@outlook.com", "Guest"),
(24, "Omar Alvarez", "685 Grand Street", 09, "Douglasville", 30134, "Georgia",
"omar_alvarez@verizon.net", "Guest"),
(25, "Cory Wong", "65 Ramblewood St.", 232, "Rego Park", 11374, "New York",
"c.wong@comcast.net", "Guest"),
(26, "Vincent Malone", "8301 Franklin Street", 103, "San Antonio", 78213, "Texas",
"vincent-malone@mac.com", "Guest"),
(27, "Harold Cole", "28 Country Street", 288, "Lorton", 22079, "Virginia", "h.cole@gmail.com",
"Guest"),
(28, "Lloyd Collins", "759 Logan St.", 59, "Beaver Falls", 15010, "Pennsylvania",
"llcollins19@yahoo.com", "Rewards Member"),
(29, "Alvin Garner", "52 E. Lancaster Street", 418, "Reidsville", 27320, "North Carolina",
"alvin_garner003@yahoo.com", "Rewards Member"),
(30, "Jeanne Figueroa", "39 Holly Street", 841, "North Augusta", 29481, "South Carolina",
"j.figueroa@hotmail.com", "Guest"),
(31, "Randall Morales", "40 Brewery St.", 366, "Hillsboro", 97124, "Oregon",
"r_morales@hotmail.com", "Rewards Member"),
(32, "Pauline Nichols", "792 Newcastle Lane", 55, "Millville", 08332, "New Jersey",
"pauline.nichols@yahoo.com", "Guest"),
(33, "Kim James", "90 Morris Court", 500, "Belmont", 02478, "Massachusetts",
"KJames411@outlook.com", "Guest"),
(34, "Omar Alvarez", "618 N. Joy Ridge Street", 411, "Forest Hills", 11375, "New York",
"o.alvarez@gmail.com", "Rewards Member"),
(35, "Cory Wong", "8143 Elm Rd.", 314, "Hickory", 28601, "North Carolina",
"cory.wong@gmail.com", "Guest"),
(36, "Vincent Malone", "73 Devon St.", 510, "Saint Paul", 55104, "Minnesota",
"vince.malone@gmail.com", "Guest"),
(37, "Harold Cole", "819 Heritage Lane", 918, "Troy", 12180, "New York",
"harold_c551@hotmail.com", "Rewards Member"),
(38, "Lloyd Collins", "7806 South Constitution Dr.", 806, "Cockeysville", 21030, "Maryland",
"llyod.collins@mac.com", "Rewards Member"),
(39, "Alvin Garner", "71 Pulaski Rd.", 67, "Reading", 01867, "Maryland",
"alvin-garner@mac.com", "Guest"),
(40, "Devin Mack", "8 Water Lane", 08, "Reynoldsburg", 43068, "Ohio", "d.mack@verizon.net",
"Guest");
```

Payment table insert statements:

```
INSERT INTO payment (payment.paymentID, payment.payment_type, payment.card_number,
payment.expiration, payment.cvd, payment.customerID) VALUES(222602, "Debit",
486030571038, 05/25, 902, 1),
(241797, "Debit", 4971475460795447, 05/22, 890, 2),
(275009, "Debit", 4073032894227492, 02/25, 786, 3),
(251483, "Credit", 488625348412022, 01/25, 382, 4),
(214627, "Apple Pay", 343056331388841, 11/29, 364, 5),
(229180, "Credit", 348577603428085, 04/26, 977, 6),
(205773, "Credit", 6011439279929081, 12/22, 706, 7),
(291880, "Apple Pay", 349920443947170, 03/29, 398, 8),
(291291, "Debit", 6011480931385692, 10/27, 864, 9),
(229802, "Debit", 6011415185654237, 01/27, 882, 10),
(212481, "Debit", 346617510136068, 10/22, 951, 11),
(208195, "Credit", 370363110122797, 08/21, 820, 12),
(270387, "Credit", 4326095806139654, 05/24, 391, 13),
(258841, "Apple Pay", 374453226517208, 05/22, 700, 14),
(205154, "Credit", 6011412559773370, 03/22, 450, 15),
(281343, "Debit", 5387845717099014, 06/27, 830, 16),
(238488, "Debit", 5517680705790016, 03/22, 851, 17),
(233837, "Apple Pay", 6011842060282703, 09/25, 435, 18),
(294195, "Apple Pay", 6011850085636100, 10/23, 921, 19),
(277694, "Debit", 5402690894654087, 01/22, 990, 20),
(226578, "Credit", 4181296477198763, 12/25, 698, 21),
(202450, "Credit", 5162107957114958, 03/24, 556, 22),
(211451, "Credit", 6011844805408457, 12/24, 659, 23),
(222630, "Credit", 6011723732468807, 01/24, 147, 24),
(240272, "Credit", 5429796693054095, 05/22, 134, 25),
(237933, "Debit", 5109178935453780, 11/26, 933, 26),
(275835, "Apple Pay", 4636485126263158, 02/22, 941, 27),
(294925, "Apple Pay", 5463545528109872, 05/25, 672, 28),
(212371, "Apple Pay", 345002461445322, 03/24, 679, 29),
(242954, "Debit", 346902767737976, 08/23, 742, 30),
(205240, "Debit", 374912847012517, 01/25, 783, 31),
(208834, "Credit", 4221174944083581, 10/22, 712, 32),
(266933, "Credit", 4487206235222756, 11/25, 768, 33),
(257160, "Debit", 4069846930619792, 03/21, 106, 34),
(281758, "Debit", 5523488318487588, 07/26, 292, 35),
(253530, "Debit", 371756547492072, 06/22, 714, 36),
(201820, "Debit", 4291484938270432, 09/21, 211, 37),
(249282, "Credit", 4258949028368750, 01/25, 552, 38),
(254370, "Apple Pay", 5122959288806754, 07/25, 706, 39),
(233724, "Apple Pay", 5180497431921245, 01/26, 569, 40);
```

Supplier Insert statements:

```
INSERT INTO supplier (supplier.supplierID, supplier.street, supplier.unit_number, supplier.city,
supplier.state, supplier.zip, supplier.company_name, supplier.manufactured_product)
VALUES(104525, "8264 Myrtle Rd.", 38, "Clifton Park", "New York", 12065, "Quado", "laptop"),
(189950, "903 Greenview Street", 401, "New Orleans", "Louisiana", 70115, "Zoojo", "air pods"),
(105439, "9 Wilson St.", 115, "Norristown", "Pennsylvania", 19401, "Dexo", "Hard Drive"),
(130305, "93 Central Street", 120, "Westford", "Massachusetts", 01886, "US", "Graphics Card"),
(160454, "7179 Circle Street", 11, "Logansport", "Indiana", 46947, "Leetude", "USB"),
(104248, "57 Henry Smith St.", 86, "Glenside", "Pennsylvania", 19038, "Sufend", "SSD"),
(178891, "463 Edgewood Drive", 600, "Wyoming", "Michigan", 49506, "Symil", "Monitor"),
(182614, "9298 Chapel Road", 27, "Riverdale", "Georgia", 30274, "Finity", "Keyboard"),
(155295, "349 Oak Street", 707, "Charlottesville", "Virginia", 22901, "Kilium", "Mouse"),
(170291, "806 Pumpkin Hill Road", 32, "Buffalo Grove", "Illinois", 60089, "Yagen", "Printer"),
(104742, "9424 Mayflower St.", 274, "Dearborn Heights", "Michigan", 48127, "Isovu", "Fax
Machine"),
(152753, "72 Pineknoll St.", 779, "Longwood", "Florida", 32779, "Divalane", "Shredder"),
(118844, "7884 Gates Lane", 451, "Hopkinsville", "Kentucky", 42240, "Movist", "Tablet"),
(137752, "22 School St.", 347, "Patchogue", "New York", 11772, "Autofy", "Desktop"),
(102630, "66 Forest Drive", 103, "Hazleton", "Pennsylvania", 18201, "Protoloo", "HDMI");
```

Maintenance Insert statements:

```
INSERT INTO maintenance (maintenance.serviceID, maintenance.service_type,
maintenance.operating_hours, maintenance.cost, maintenance.location,
maintenance.customerID) VALUES(96330, "IT", "0.5 hours", 20, "Store", 10),
(90961, "Installation", "2 hours", 75, "In-Home", 22),
(99973, "Installation", "0.25 hours", 20, "Store", 40),
(92750, "Installation", "1.5 hours", 45, "Store", 8),
(98643, "IT", "0.4 hours", 2, "Store", 14),
(93298, "Computer Service", "2 hours", 60, "In-Home", 30),
(99374, "IT", "1 hours", 50, "In-Home", 26),
(93873, "IT", "1.5 hours", 55, "Store", 32),
(93724, "Computer Service", "0.3 hours", 20, "Store", 19),
(94454, "Computer Service", "0.25 hours", 20, "Store", 9),
(95203, "Appliance", "1.5 hours", 100, "In-Home", 31),
(92586, "Appliance", "0.4 hours", 75, "In-Home", 6),
(99775, "Computer Service", "0.15 hours", 54, "Store", 11), (96592, "Appliance", "1.5 hours",
120, "In-Home", 34),
(92083, "IT", "1 hours", 88, "In-Home", 1)
```

Salary Insert Statements:

```
INSERT INTO salary (salary.salaryID, salary.role, salary.status_, salary.benefit)
VALUES(54911, "District Manager", "Full Time", "Health Dental Vision 401k"),
(40356, "District Manager", "Full Time", "Health Dental Vision 401k"),
(42174, "District Manager", "Full Time", "Health Dental Vision 401k"),
(43537, "District Manager", "Full Time", "Health Dental Vision 401k"),
(31420, "District Manager", "Full Time", "Health Dental Vision 401k"),
(61421, "District Manager", "Full Time", "Health Dental Vision 401k"),
(52186, "District Manager", "Full Time", "Health Dental Vision 401k"),
(43139, "District Manager", "Full Time", "Health Dental Vision 401k"),
(60446, "District Manager", "Full Time", "Health Dental Vision 401k"),
(41784, "District Manager", "Full Time", "Health Dental Vision 401k"),
(45044, "District Manager", "Full Time", "Health Dental Vision 401k"),
(43010, "District Manager", "Full Time", "Health Dental Vision 401k"),
(55044, "District Manager", "Full Time", "Health Dental Vision 401k"),
(62072, "District Manager", "Full Time", "Health Dental Vision 401k"),
(63601, "District manager", "Full time", "Health Dental Vision 401k"),
(39164, "Manager", "Full time", "Health Dental Vision 401k"),
(37653, "Advisor", "Part Time", "Health Dental Vision"),
(35407, "Supervisor", "Part Time", "Dental Vision"),
(31658, "Team Member", "Seasonal", "Vision"),
(37696, "Advisor", "Seasonal", "Vision"),
(30105, "Team Member", "Part Time", "Dental Vision"),
(35244, "Supervisor", "Seasonal", "Vision"),
(35221, "Supervisor", "Part Time", "Health Dental Vision"),
(38041, "Manager", "Full Time", "Health Dental Vision 401k"),
(33968, "3rd Key", "Full Time", "Health Dental Vision"),
(31381, "Team Member", "Seasonal", "Vision"),
(38416, "Manager", "Full Time", "Health Dental Vision 401k"),
(37751, "Advisor", "Seasonal", "Vision"),
(30421, "Team Member", "Part Time", "Dental Vision"),
(37315, "Advisor", "Seasonal", "Vision");
```

Make Insert statement:

```
INSERT INTO make (make.supplierID, make.productID) VALUES(104525, 27822),
(189950, 25617),
(105439, 26292),
(130305, 20891),
(160454, 22906),
(104248, 27992),
(178891, 27917),
(182614, 20435),
(155295, 23958),
(170291, 23019),
(104742, 22006),
(152753, 22228),
(118844, 21929),
(137752, 27874),
(102630, 23526);
```

Orders Insert Statements:

```
INSERT INTO orders (orders.orderID, orders.fulfillment, orders.Date_, orders.bastreet,
orders.baunit_number, orders.bacity, orders.bastate, orders.bazip, orders.sastreet,
orders.saunit_number, orders.sacity, orders.sastate, orders.sazip, orders.customerID)
VALUES(1346, "Pickup", 12/23/2016, "562 West Ivy St.", 10, "Jonesboro", "Georgia", 30236,
"562 West Ivy St.", 10, "Jonesboro", "Georgia", 30236, 1), (1893, "Pickup", 12/25/2018, "7318
East Washington Street", 745, "Royersford", "Pennsylvania", 19468, "7318 East Washington
Street", 745, "Royersford", "Pennsylvania", 19468, 2),
(1154, "Pickup", 1/21/2020, "9002 Roosevelt Street", 01, "Malvern", "Pennsylvania", 19355, "9002 Roosevelt Street", 01, "Malvern", "Pennsylvania", 19355, 3),
(1862, "Pickup", 3/6/2017, "8925 Old Iroquois Ave.", 16, "Boca Raton", "Florida", 33428, "8925 Old Iroquois Ave.", 16, "Boca Raton", "Florida", 33428, 4),
(1704, "Delivery", 3/28/2020, "55 Shore Ave.", 601, "Laurel", "Maryland", 20707, "55 Shore Ave.", 601, "Laurel", "Maryland", 20707, 5),
(1873, "Delivery", 2/16/2016, "7024 Miles Ave.", 555, "Gulfport", "Mississippi", 39503, "7024 Miles Ave.", 555, "Gulfport", "Mississippi", 39503, 6),
(1757, "Pickup", 5/7/2017, "716 Myrtle Street", 74, "Lake Jackson", "Texas", 77566, "716 Myrtle Street", 74, "Lake Jackson", "Texas", 77566, 7),
(1063, "Delivery", 12/5/2020, "2 Bedford Dr.", 123, "Norwood", "Maryland", 02062, "2 Bedford Dr.", 123, "Norwood", "Maryland", 02062, 8),
(1972, "Delivery", 9/27/2019, "8897 East Bayport Avenue", 512, "Canandaigua", "New York", 14424, "8897 East Bayport Avenue", 512, "Canandaigua", "New York", 14424, 22),
(1039, "Delivery", 4/22/2017, "9068 Wild Horse St.", 68, "Oxnard", "California", 93035, "9068 Wild Horse St.", 68, "Oxnard", "California", 93035, 21),
(1837, "Pickup", 9/16/2018, "561 Woodland Street", 16, "Bowie", "Maryland", 20715, "561 Woodland Street", 16, "Bowie", "Maryland", 20715, 20),
(1586, "Pickup", 11/6/2018, "8284 Logan St.", 222, "Cumberland", "Rhode Island", 02864, "8284 Logan St.", 222, "Cumberland", "Rhode Island", 02864, 19),
(1179, "Delivery", 9/21/2020, "7974 S. Rockcrest Drive", 974, "Sanducky", "Ohio", 44870, "7974 S. Rockcrest Drive", 974, "Sanducky", "Ohio", 44870, 18),
(1227, "Pickup", 9/16/2019, "890 Schoolhouse Ave.", 05, "Barberton", "Ohio", 44203, "890 Schoolhouse Ave.", 05, "Barberton", "Ohio", 44203, 14),
(1957, "Delivery", 12/21/2016, "724 Brickyard Street", 632, "Hoboken", "New Jersey", 07030, "724 Brickyard Street", 632, "Hoboken", "New Jersey", 07030, 15),
(1321, "Pickup", 5/12/2017, "463 Greenrose Street", 98, "Chesterfield", "Virginia", 23832, "463 Greenrose Street", 98, "Chesterfield", "Virginia", 23832, 16),
(1867, "Pickup", 6/11/2017, "7365 W. Alderwood Drive", 365, "Staunton", "Virginia", 24401, "7365 W. Alderwood Drive", 365, "Staunton", "Virginia", 24401, 17),
(1851, "Pickup", 11/10/2020, "26 Selby Circle", 26, "North Olmsted", "Ohio", 44070, "26 Selby Circle", 26, "North Olmsted", "Ohio", 44070, 13),
(1810, "Pickup", 6/4/2018, "7257 Pennington Lane", 812, "Hopkins", "Minnesota", 55343, "7257 Pennington Lane", 812, "Hopkins", "Minnesota", 55343, 12),
(1705, "Delivery", 6/24/2018, "785 Shipley Ave.", 22, "Tualatin", "Oregon", 97062, "785 Shipley Ave.", 22, "Tualatin", "Oregon", 97062, 11),
(1733, "Delivery", 1/14/2020, "513 Westport Avenue", 465, "Mason City", "Iowa", 50401, "513 Westport Avenue", 465, "Mason City", "Iowa", 50401, 10),
(1203, "Delivery", 11/28/2017, "32 Meadow Lame", 33, "Carpentersville", "Illinois", 60110, "32 Meadow Lame", 33, "Carpentersville", "Illinois", 60110, 9),
(1394, "Delivery", 2/26/2019, "7201 Devon Ave.", 102, "Oak Forest", "Illinois", 60452, "7201 Devon Ave.", 102, "Oak Forest", "Illinois", 60452, 23),
(1369, "Delivery", 11/11/2018, "685 Grand Street", 09, "Douglasville", "Georgia", 30134, "685 Grand Street", 09, "Douglasville", "Georgia", 30134, 24),
(1609, "Delivery", 4/19/2020, "65 Ramblewood St.", 232, "Rego Park", "New York", 11374, "65
Ramblewood St.", 232, "Rego Park", "New York", 11374, 25),
(1735, "Pickup", 2/2/2019, "8301 Franklin Street", 103, "San Antonio", "Texas", 78213, "8301
Franklin Street", 103, "San Antonio", "Texas", 78213, 26),
(1918, "Delivery", 1/18/2019, "28 Country Street", 288, "Lorton", "Virginia", 22079, "28 Country
Street", 288, "Lorton", "Virginia", 22079, 27),
(1700, "Delivery", 12/12/2020, "759 Logan St.", 59, "Beaver Falls", "Pennsylvania", 15010, "759
Logan St.", 59, "Beaver Falls", "Pennsylvania", 15010, 28),
(1033, "Delivery", 6/13/2017, "52 E. Lancaster Street", 418, "Reidsville", "North Carolina",
27320, "52 E. Lancaster Street", 418, "Reidsville", "North Carolina", 27320, 29),
(1970, "Delivery", 12/26/2018, "39 Holly Street", 841, "North Augusta", "South Carolina", 29481,
"39 Holly Street", 841, "North Augusta", "South Carolina", 29481, 30),
(1311, "Pickup", 10/13/2017, "7806 South Constitution Dr.", 806, "Cockeysville", "Maryland",
21030, "7806 South Constitution Dr.", 806, "Cockeysville", "Maryland", 21030, 38),
(1420, "Pickup", 6/14/2019, "8 Water Lane", 08, "Reynoldsburg", "Ohio", 43068, "8 Water Lane",
08, "Reynoldsburg", "Ohio", 43068, 40),
(1698, "Pickup", 3/13/2019, "71 Pulaski Rd.", 67, "Reading", "Maryland", 01867, "71 Pulaski
Rd.", 67, "Reading", "Maryland", 01867, 39),
(1092, "Delivery", 12/9/2017, "90 Morris Court", 500, "Belmont", "Massachusetts", 02478, "90
Morris Court", 500, "Belmont", "Massachusetts", 02478, 33),
(1649, "Delivery", 4/22/2018, "55 Shore Ave.", 601, "Laurel", "Maryland", 20707, "55 Shore
Ave.", 601, "Laurel", "Maryland", 20707, 5),
(1540, "Delivery", 6/17/2017, "792 Newcastle Lane", 55, "Millville", "New Jersey", 08332, "792
Newcastle Lane", 55, "Millville", "New Jersey", 08332, 32),
(1366, "Pickup", 9/6/2017, "819 Heritage Lane", 918, "Troy", "New York", 12180, "819 Heritage
Lane", 918, "Troy", "Mew York", 12180, 37),
(1675, "Pickup", 8/14/2020, "40 Brewery St.", 366, "Hillsboro", "Oregon", 97124, "40 Brewery
St.", 366, "Hillsboro", "Oregon", 97124, 31),
(1966, "Delivery", 8/3/2019, "2 Bedford Dr.", 123, "Norwood", "Maryland", 02062, "2 Bedford
Dr.", 123, "Norwood", "Maryland", 02062, 8),
(1767, "Pickup", 8/17/2019, "73 Devon St.", 510, "Saint Paul", "Minnesota", 55104, "73 Devon
St.", 510, "Saint Paul", "Minnesota", 55104, 36),
(1493, "Pickup", 3/15/2017, "618 N. Joy Ridge Street", 411, "Forest Hills", "New York", 11375,
"618 N. Joy Ridge Street", 411, "Forest Hills", "New York", 11375, 34),
(1812, "Pickup", 10/22/2018, "8143 Elm Rd.", 314, "Hickory", "North Carolina", 28601, "8143
Elm Rd.", 314, "Hickory", "North Carolina", 28601, 35),
(1620, "Delivery", 1/29/2020, "8301 Franklin Street", 103, "San Antonio", "Texas", 78213, "8301
Franklin Street", 103, "San Antonio", "Texas", 78213, 26);
```

District manager Insert Statements:

```
INSERT INTO district_manager (district_manager.managerID, district_manager.name,
district_manager.storeID, district_manager.location, district_manager.salaryID) VALUES(475,
"Lynette Henderson", 2089, "Ohio", 54911),
(402, "Christopher Dawson", 2225, "Florida", 40356),
(489, "Billy Carson", 2976, "Georgia", 42174),
(444, "Kent Fleming", 2144, "Mississippi", 43537),
(456, "Rodolfo Carpenter", 2999, "Maryland", 41420),
(422, "Jill Bowers", 2700, "Texas", 61421),
(481, "Blake Reynolds", 2580, "Iowa", 52186),
(405, "Jessi Burke", 2397, "Oregon", 43139),
(411, "Lynda Turner", 2010, "Minnesota", 60446),
(499, "Martha Webster", 2111, "New Jersey", 41784),
(480, "Sammy Morris", 2466, "Virginia", 45044),
(487, "Dewey Williams", 2371, "Pennsylvania", 43010),
(465, "Wade Franklin", 2859, "California", 55044),
(446, "Kathryn Anderson", 2509, "New York", 62072),
(435, "Anna Edwards", 2131, "Illinois", 63601);
```

(*Edit)Product insert statements:

```
INSERT INTO product (product.productID, product.manufacturer, product.serial_number,
product.price, product.warranty, product.stockID) VALUES(27822, "Quado", 88965778, 450,
"Warranty", 36885),
(25617, "Zoojo", 37248463, 800, "Warranty", 36885),
(26292, "Dexo", 53893535, 488, "No Warranty", 36885),
(20891, "US", 99583342, 222, "No Warranty", 36885),
(22906, "Leetude", 44442924, 25, "No Warranty", 36885),
(27917, "Symil", 27499658, 92, "Warranty", 36885),
(20435, "Finity", 33576987, 37, "No Warranty", 36885),
(23958, "Kilium", 53989868, 20, "No Warranty", 36885),
(23019, "Yagen", 92888764, 175, "Warranty", 36885),
(22006, "Isovu", 33284683, 500, "Warranty", 36885),
(22228, "Divalane", 27655334, 75, "No Warranty", 36885),
(21929, "Movist", 68765648, 300, "Warranty", 36885),
(27874, "Autofy", 46327372, 600, "Warranty", 36885),
(23526, "Protoloo", 67365259, 15, "No Warranty", 36885)
```

Orderline insert Statements:

```
INSERT INTO order_line (order_line.orderID, order_line.productID, order_line.quantity)
VALUES(1493, 27874, 10),
(1812, 23526, 15),
(1620, 22906, 26);
```

Store insert statements:

```
INSERT INTO store (store.storeID, store.street, store.city, store.state, store.zip, store.hours,
store.stockID) VALUES(2089, "9416 8th Dr.", "Findlay", "Ohio", 45840, "9am-9pm", 36715),
(2225, "3 Pheasant Dr.", "Bonita Springs", "Florida", 34135, "9am-9pm", 34763),
(2976, "733 Fremont Ave.", "Hinesville", "Georgia", 31313, "8am-7pm", 36885),
(2144, "9336 Bowman Court", "Fort Dodge", "Mississippi", 38041, "9am-10pm", 37539),
(2999, "178 San Pablo Ave.", "Brockton", "Maryland", 20707, "9am-10pm", 38739),
(2700, "9015 Miller Road", "Norwich", "Texas", 77566, "9am-10pm", 37852),
(2580, "88 S. Delaware Drive", "Anaheim", "Iowa", 50401, "8am-7pm", 39603),
(2397, "9015 Miller Road", "Norwich", "Oregon", 97062, "9am-9pm", 36415),
(2010, "3 Old York Street", "Hastings", "Minnesota", 55033, "8am-7pm", 30410),
(2111, "185 Logan St.", "Little Falls", "New Jersey", 07424, "8am-7pm", 33143),
(2466, "559 Helen Rd.", "Winchester", "Virginia", 22601, "9am-10pm", 39725),
(2371, "8363 South Fremont Lane", "Wilkes Barre", "Pennsylvania", 18702, "9am-9pm", 30800),
(2859, "95 Trusel Lane", "Bakersfield", "California", 93306, "9am-9pm", 32230),
(2509, "8609 Park St.", "Centereach", "New York", 11720, "9am-9pm", 39032),
(2131, "7014 Brickell St.", "Northbrook", "Illinois", 60062, "9am-10pm", 38488);
```

Customer phone number insert statements:

```
INSERT INTO customer_phone_number (customer_phone_number.customerID,
customer_phone_number.phone_number) VALUES(1, 9036098961), (2, 52064246030), (3,
3157716283), (4, 8285338173), (5, 7135407505), (6, 8147063906), (7, 3203745238), (8,
8597451141), (9, 3476549011), (10, 9096290882), (11, 8174935558), (12, 2816620499), (13,
3213548663), (14, 6055632328), (15, 5868053965), (16, 7044670788), (17, 4434752786), (18,
9168962249), (19, 4122582031), (20, 4846818306), (21, 2678508081), (22, 2488161326), (23,
7654815997), (24, 6305217056), (25, 9286274203), (26, 8012182116), (27, 7196942955), (28,
9187234590), (29, 6174373033), (30, 8154301036), (31, 2568449455), (32, 5075825184), (33,
3138977899), (34, 2162895009), (35, 5868064997), (36, 6108732205), (37, 928272545), (38,
8012187258), (39, 7196945745), (40, 9187234402);
```

Visit insert statements:

```
INSERT INTO visit(visit.storeID, visit.customerID) VALUES(77521, 004), (46225, 016), (23771,
022);
```

Rewards Member insert statements:

```
INSERT INTO rewards_member (rewards_member.memberID,
rewards_member.RMcustomerID, rewards_member.reward_balance,
rewards_member.member_since) VALUES(219, 5, 10863, 9/10/2003),
(279, 6, 10367, 3/24/2004),
(247, 8, 10639, 6/14/2012),
(234, 9, 10036, 4/26/2016),
(250, 10, 10686, 7/18/2006),
(205, 13, 10501, 1/6/2017),
(252, 18, 10906, 8/26/2019),
(248, 21, 10002, 8/22/2014),
(239, 22, 10947, 3/31/2006),
(262, 28, 10917, 11/15/2003),
(241, 29, 10658, 10/8/2005),
(218, 31, 10809, 5/26/2014),
(265, 34, 10591, 7/24/2017),
(217, 37, 10305, 9/28/2018),
(257, 38, 10609, 8/3/2006);
```

Have Insert Statement:

```
INSERT INTO have (have.supplierID, have.orderID) VALUES(104525, 1970), (104525, 1311),
(189950, 1893), (189950, 1033), (189950, 1420), (105439, 1154), (105439, 1700), (105439,
1698), (130305, 1862), (130305, 1918), (130305, 1092), (160454, 1704), (160454, 1735),
(160454, 1649), (104248, 1873), (104248, 1609), (104248, 1540), (178891, 1757), (178891,
1369), (178891, 1366), (182614, 1063), (182614, 1394), (182614, 1675), (155295, 1972),
(155295, 1203), (155295, 1966), (170291, 1039), (170291, 1733), (170291, 1767), (104742,
1837), (104742, 1705), (152753, 1586), (152753, 1810), (118844, 1179), (118844, 1851),
(137752, 1227), (137752, 1867), (102630, 1957), (102630, 1321), (104525, 1346)
```

Purchase Insert Statements:

```
INSERT INTO purchase (purchase.customerID, purchase.productID) VALUES(1, 27822), (2,
25617), (3, 26292), (4, 20891), (5, 22906), (6, 27992), (7, 27917), (8, 20435), (9, 23958), (10,
23019), (11, 22006), (12, 22228), (13, 21929), (14, 27874), (15, 23526), (16, 23526), (17,
27874), (18, 21929), (19, 22228), (20, 22006), (21, 23019), (22, 23958), (23, 20435), (24,
27917), (25, 27992), (26, 22906), (27, 20891), (28, 26292), (29, 25617), (30, 27822), (31,
20435), (32, 27992), (33, 20891), (34, 27874), (35, 23526), (36, 23019), (37, 27917), (38,
27822), (39, 26292), (40, 25617);
```

Uses Insert Statements:

```
INSERT INTO uses (uses.productID, uses.serviceID) VALUES(27822, 90961),
(25617, 99973),
(26292, 92750),
(20891, 98643),
(22906, 93298),
(27992, 99374),
(27917, 93873),
(20435, 93724),
(23958, 94454),
(23019, 95203),
(22006, 92856),
(22228, 99775),
(21929, 965920),
(27874, 92083),
(23526, 96330);
```

Stock Insert Statements:

```
INSERT INTO stock (stock.stockID, stock.quantity, stock.managerID) VALUES(36885, 216,
489),
(39032, 62, 446),
(38739, 31, 456),
(38488, 169, 435),
(33143, 89, 499),
(34763, 50, 402),
(37852, 204, 422),
(36715, 49, 475),
(37539, 101, 444),
(39603, 201, 481),
(39725, 183, 480),
(30410, 108, 411),
(30800, 30, 487),
(32230, 200, 465),
(36415, 119, 405);
```

Sell Insert Statements:

```
INSERT INTO sell (sell.storeID, sell.productID) VALUES(2089, 27822),
(2225, 25617),
(2976, 26292),
(2144, 20891),
(2999, 22906),
(2700, 27992),
(2580, 27917),
(2397, 20435),
(2010, 23958),
(2111, 23019),
(2466, 22006),
(2371, 22228),
(2859, 21929),
(2509, 27874),
(2131, 23526);
```

Visit Insert Statements:

```
INSERT INTO visit (visit.storeID, visit.customerID) VALUES(2089, 14),
(2225, 4),
(2976, 1),
(2144, 6),
(2999, 8),
(2700, 26),
(2580, 10),
(2397, 31),
(2010, 12),
(2111, 32),
(2466, 17),
(2371, 2),
(2859, 21),
(2509, 34),
(2131, 9);
```

Employee insert statements:

```
INSERT INTO employee VALUES(321, "9078 Foxrun Dr.", 32, "Dacula", 30019, "Georgia",
39164, 489, 36885, 2976, 90961),
(325, "8941 Brewery St.", 05, "Yorktown Heights", 10598, "New York", 37653, 446, 39032,
2509, 99973),
(311, "7924 Cemetery Street", 197, "Westford", 01886, "Maryland", 35407, 456, 38739, 2999,
92750),
(366, "16 W. Monroe Street", 91, "Wilmette", 60091, "Illinois", 31658, 435, 38488, 2131, 98643),
(391, "2 Augusta Dr.", 714, "Hickory", 07030, "New Jersey", 37696, 499, 33143, 2111, 93298),
(349, "34 South St.", 39, "Natchez", 33428, "Florida", 30105, 402, 34763, 2225, 99374),
(344, "4 Stillwater Lane", 72, "Pewaukee", 77566, "Texas", 35244, 422, 37852, 2700, 93873),
(306, "391 Union St.", 244, "Staunton, 44070, "Ohio", 35221, 475, 36715, 2089, 93724),
(386, "366 Holly Drive", 550, "Owatonna", 39503, "Mississippi", 38041, 444, 37539, 2144,
94454),
(314, "7558 Hill Road", 503, "Des Moines", 50310, "Iowa", 33968, 481, 39603, 2580, 95203),
(333, "9255 Glendale Lane". 61, "Hopewell", 23860, "Virginia", 31381, 480, 39725, 2466,
92586),
(343, "453 Yukon Avenue", 106, "Anoka", 55303, "Minnesota", 38416, 411, 30410, 2010,
99775),
(329, "5 John Dr.", 42, "King of Prussia" 19406, "Pennsylvania", 37751, 487, 30800, 2371,
96592),
(301, "7973 Henry Avenue", 373, 93555, "Ridgecrest", "California", 3421, 465, 32230, 2859,
92083),
(309, "97 Garfield Lane", 851, "Mason City", 97062, "Oregon", 37315, 405, 36415, 2397,
96330);
```

<u>**DDL Code:**</u>

Creation of the Customer table:

```
Create table Customer(customerID int(11) not null, Name varchar(50), Street varchar(50),
Unit_Number int(11), City varchar(50), Zip varchar(5), State varchar(50), Email varchar(50)
Customer_Type varchar(50), Constraint customer_pk Primary key (customer.customerID);
```

Creation of the District Manager table:

```
Create table district_manager(managerID int(11) not null, name varchar(50), storeID int(11),
location varchar(50), salaryID int(11), Constraint district_manager_pk Primary Key
(district_manager.managerID), Constraint district_manager_fk Foreign Key
(district_manager.SalaryID) References Salary (Salary.salaryID) on update cascade;
```

Creation of the Salary table:

```
Create table salary(salaryID int(11) not null, role varchar(50), status_ varchar(50), benefit
varchar(50), Constraint salary_pk Primary Key (salary.salaryID);
```

Creation of the Stock table:

```
Create table stock(stockID int(11) not null, quantity int(255), managerID int(11), constraint
stock_pk Primary Key (stock.stockID), constraint stock_fk Foreign Key (stock.managerID)
references district_manager (district_manager_managerID) on update cascade;
```

Creation of the Store table:

Create table store(storeID int(11) not null, street varchar(50), city varchar(50), state varchar(50), zip int(6), hours varchar(50), stockID int(11), constraint store_pk Primary Key (store.storeID), constraint store_fk Foreign Key (store.stockID) references stock (stock.stockID) on update cascade;

8. Queries and Views (Answer Business Questions):

Queries:

1. select avg(order_line.quantity) from order_line INNER JOIN customer WHERE customer.State = "California";
2. select product.manufacturer, product.price, product.productID from product LEFT JOIN order_line on product.productID = order_line.productID WHERE product.price >= 40;
3. SELECT AVG(maintenance.operating_hours) FROM maintenance WHERE maintenance.customerID IN (SELECT customer.customerID FROM customer WHERE maintenance.cost < 100)
4. Select product.manufacturer, product.price from product where product.price = (Select MIN(product.price) from product) UNION Select product.manufacturer, product.price from product where product.price = (Select Max(product.price) from product);
5. Select customer.*, orders.orderID from customer NATURAL JOIN orders where orders.orderID = "1346";

VIEWS:

1. create VIEW all_orders as (select orders.*, customer.Name, customer.Street, customer.Unit_Number, customer.City, customer.Zip, customer.State, customer.Email, customer.Customer_Type FROM orders LEFT JOIN customer ON orders.customerID = customer.customerID);
2. create VIEW sold_items as (select product.productID, order_line.quantity FROM order_line RIGHT join product on product.productID = order_line.productID)