ADVENTURE WORKS OLTP DATABASE
Analysis Report

Prepared by:
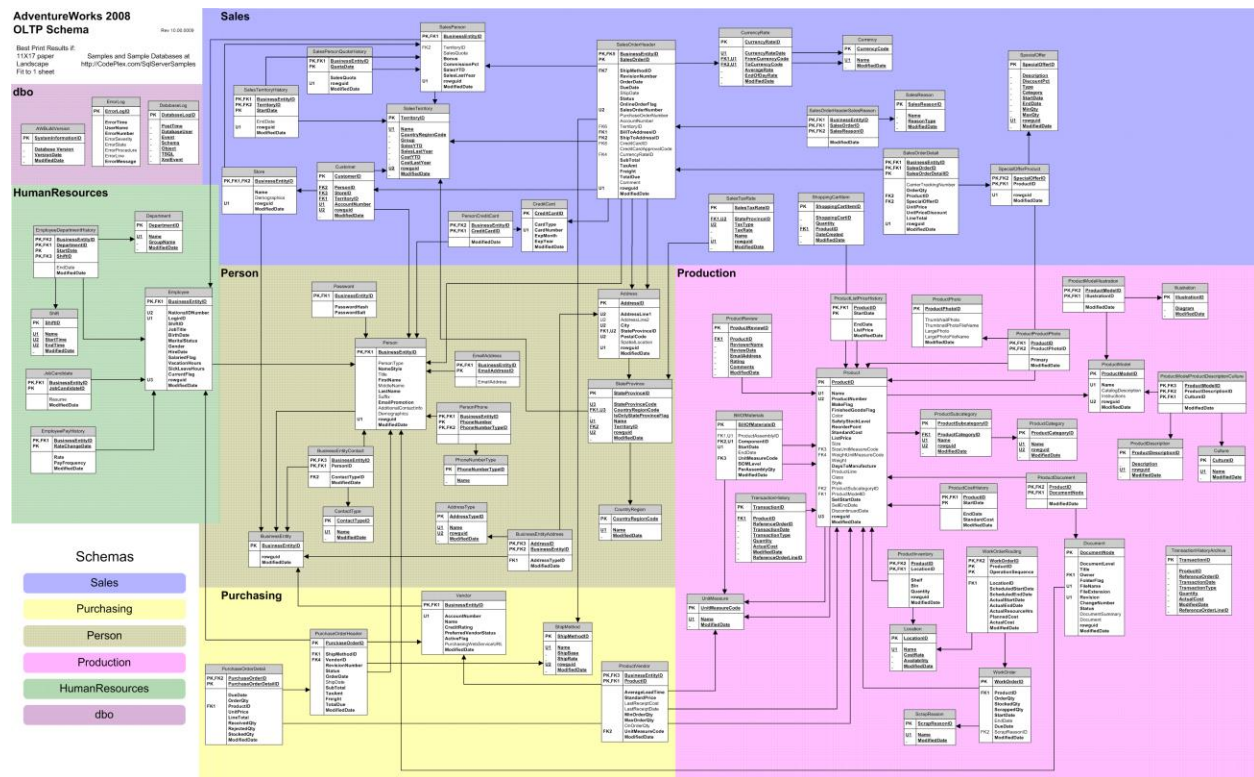
Zaki Kurdya,

BI/Data Engineering Intern

Z.Kurdiya@unrwa.org

Introduction

AdventureWorks database supports standard online transaction processing scenarios for a fictitious bicycle manufacturer - Adventure Works Cycles. Adventure Works Cycles, is a large, multinational manufacturing company. The company manufactures and sells metal and composite bicycles to North American, European, and Asian commercial markets. [1]

AdventureWorks database contains the following schemas:

1. **Person:** Contains information about individuals related to the business, such as employees, customers, and vendors.
2. **Human Resources:** Deals with employee data, including employee details, job titles, and departments.
3. **Sales:** Stores details about sales orders, including customer and product information.
4. **Production:** Represents the production process, including products, bills of materials, and work orders.
5. **Purchasing:** Manages purchase orders and vendor-related information.

AdventureWorks Entity Relationship (ER) Diagram – Version (15.0.4280.7)

---

[1] https://learn.microsoft.com/en-us/analysis-services/multidimensional-tutorial/analysis-services-tutorial-scenario

Under the hood of the database schemas

In each table, there's a **ModifiedDate** column to show the date and time the record was last updated.

• Person schema:
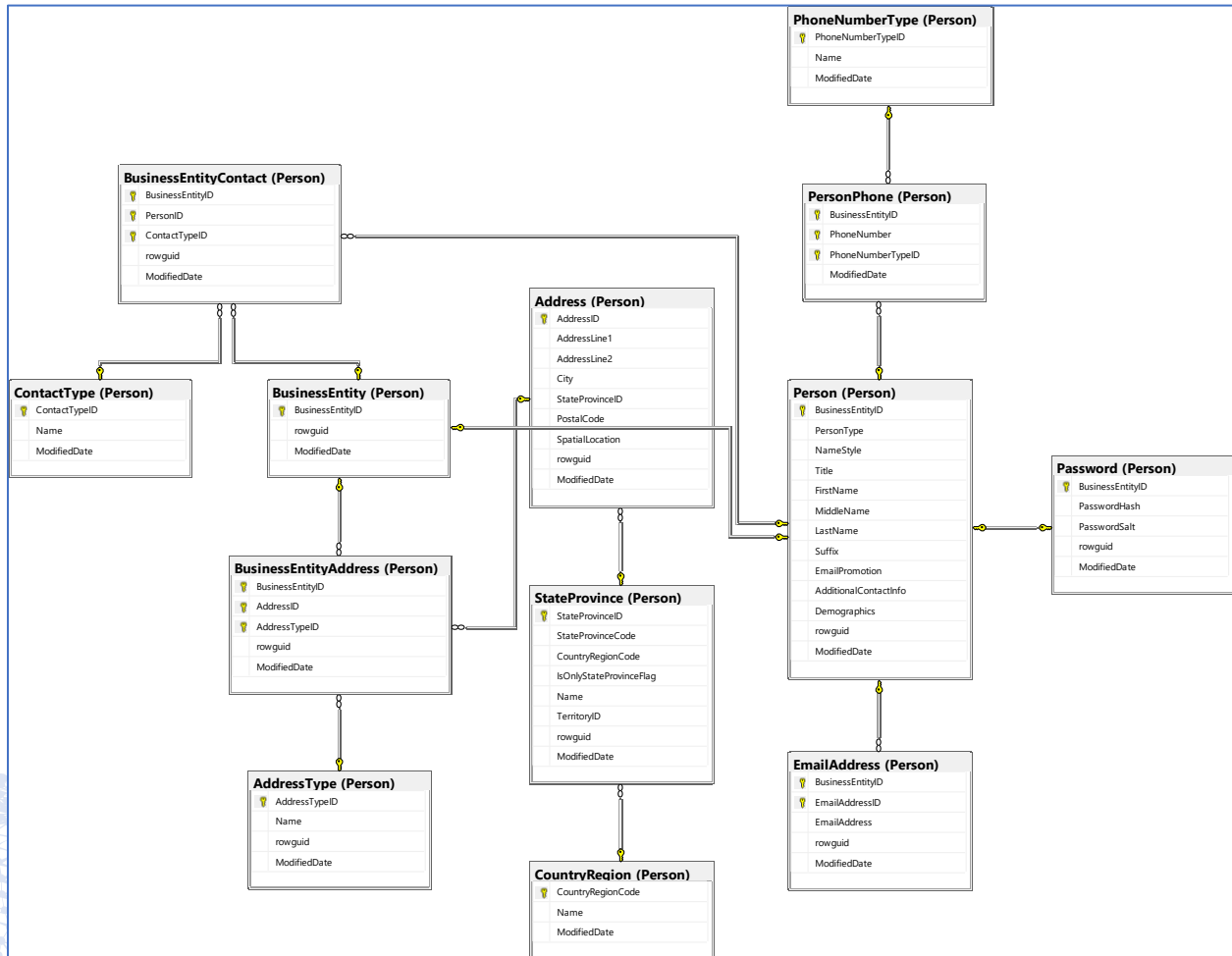
Person Schema Entity Relationship (ER) Diagram

In the AdventureWorks database, there are:

- 18484 individual (retail) customers.
- 17 sales persons.
- 273 employees (non-sales).
- 753 store contacts.
- 156 vendor contacts.
- 289 general contacts.

All employees are located in the United States.

Tables:

The **BusinessEntity** table is the source of the ID that connects vendors, customers, and employees with address and contact information.

The **Person** table contains these main columns:

- **PersonType:** Primary type of person: SC = Store Contact, IN = Individual (retail) customer, SP = Salesperson, EM = Employee (non-sales), VC = Vendor contact, GC = General contact.
- **NameStyle:** 0 = The data in FirstName and LastName are stored in Western style (first name, last name) order. 1 = Eastern style (last name, first name) order.
- **Title:** A courtesy title. For example, Mr. or Ms.
- **EmailPromotion:** 0 = Contact does not wish to receive e-mail promotions, 1 = Contact does wish to receive e-mail promotions from AdventureWorks, 2 = Contact does wish to receive e-mail promotions from AdventureWorks and selected partners.
- **Demographics:** Personal information such as hobbies, and income collected from online shoppers (stored in XML format), and it can be used later for further analysis.

The **ContactType** table is a lookup table containing the types of business entity contacts, for example, Sales Manager, Assistant Sales Agent, and Marketing Representative, and it contains 20 different contact types besides that the person may be assigned to multiple contact types for example "Owner and Purchasing Manager".

The **BusinessEntityContact** is a cross-reference table mapping stores, vendors, and employees to people, and it contains these keys:

- **BusinessEntityID:** Primary key. Foreign key to BusinessEntity.BusinessEntityID.
- **PersonID:** Primary key. Foreign key to Person.BusinessEntityID.
- **ContactTypeID:** Primary key. Foreign key to ContactType.ContactTypeID.

The **Password** table contains one-way hashed authentication information, it contains these columns:

- **PasswordHash:** Hashed password for the email account.
- **PasswordSalt:** Random value concatenated with the password string before the password is hashed.

The **EmailAddress** table contains emails for each person and the person may have multiple email addresses.

The **PersonPhone** and **PhoneNumberType** tables contain the phone number and phone number type for each person (Cell, Home, or Work), besides that the person may have multiple phone numbers, and some phone numbers are used by multiple persons.

The **BusinessEntityAddress** is a cross-reference table mapping customers, vendors, and employees to their addresses.

The **Address** table contains street address information for customers, employees, and vendors, and it contains these main columns:

- **StateProvinceID:** Unique identification number for the state or province. Foreign key to StateProvince table.
- **SpatialLocation:** Latitude and longitude of this address.

The **StateProvince** is a lookup table for state and province, that contains these main columns:

- **CountryRegionCode:** ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode.
- **TerritoryID:** ID of the territory in which the state or province is located. Foreign key to SalesTerritory.SalesTerritoryID.

The **CountryRegion** is a lookup table containing the ISO standard codes for countries and regions with their names.

Business logic:

The Person schema contains information about people associated with the company, such as employees and customers. At first, each person is assigned a BusinessEntityID from the BusinessEntity table (which can be considered an id generator), and then the rest of the person's details will be added, including the person type. Also, each person has an email address and password, which can be used for authentication in the backend system, and the person may have multiple phone numbers, which can be accessed from another table. To get the person's address, we can use the cross-reference table (BusinessEntityAddress), which provides the addresses associated with each person using the BusinessEntityID. In addition, after adding the person record, it will be assigned a contact type, and because the person may have multiple contact types, there's another cross-reference table called (BusinessEntityContact) that can be used to access the contact types.

- **Human Resources schema:**
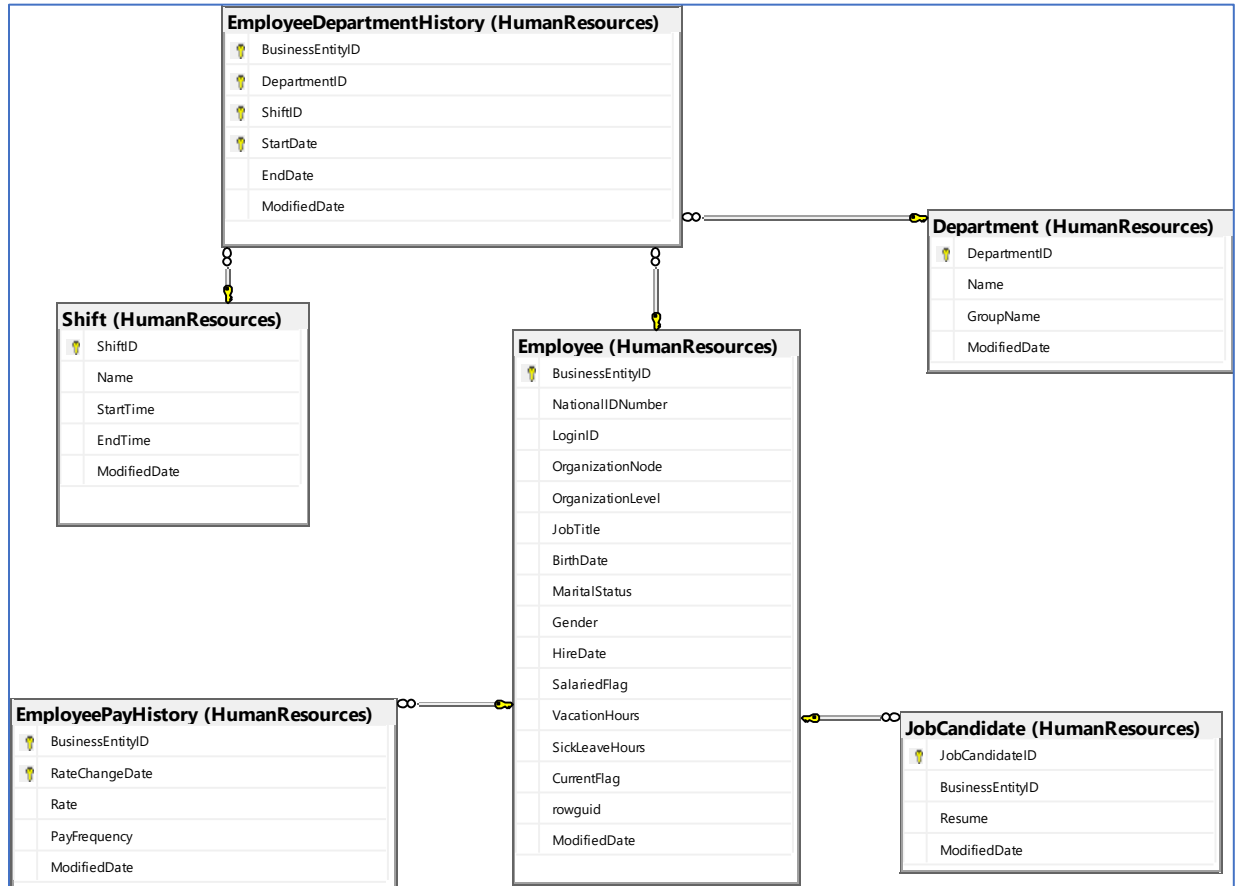
Human Resources Schema Entity Relationship (ER) Diagram

Tables:

The **EmployeeDepartmentHistory** is a junction table, such that, many employees may work in the same department and the employee may work in multiple departments.

BusinessEntityID	NumOfDepartments	DepartmentsID
4	2	1, 2
16	2	4, 5
224	2	7, 8
234	2	10, 16
250	3	4, 5, 13

The **Employee** table stores the employee information such as salary, department, and title. It contains these main columns:

- **BusinessEntityID:** Primary key for Employee records. Foreign key to Person.BusinessEntityID.
- **JobTitle:** Work titles such as Buyer or Sales Representative.
- **HireDate:** Employee hired on this date.
- OrganizationNode: Where the employee is located in the corporate hierarchy (for example it can be used to find the managers for each employee).

The **Department** is a lookup table containing the departments within the Adventure Works Cycles company.

It contains 16 different departments and 6 different groups.

The group contains multiple sub-departments, for instance: Sales and Marketing Group contains: Sales department and Marketing department.

The **Shift** is the work shift lookup table, it contains these shifts: day, evening, and night with their start and end times.

The **EmployeePayHistory** table contains employee pay history, and it contains these mains columns:

- **RateChangeDate:** Date the change in pay is effective.
- **Rate:** Salary hourly rate.
- **PayFrequency:** 1 = Salary received monthly, 2 = Salary received biweekly.

We can use this table later for further analysis to determine the time each employee take until his salary rate increase.

The **JobCandidate** table contains resumes submitted to the human resources department by job applicants in XML format (we can extract information from the resumes such as the education level and the work experience and use it in the analysis), also it contains the BusinessEntityID column which the employee identification number if the applicant was hired (Foreign key to Employee.BusinessEntityID).

Business logic:

The Human Recourses schema is concerned with employee data, such as employee information, job titles, and departments. First, the applicant begins the application and review process by submitting the resume, which is stored in the JobCandidate table. After the employee is hired, a new Person record is created, the related BusinessEntityID for that employee is assigned, and applicant employment details are stored in the Employee table. The applicant is also assigned to the appropriate Department, and the work shift and the initial salary rate are specified and logged in the database.

• Purchasing schema:
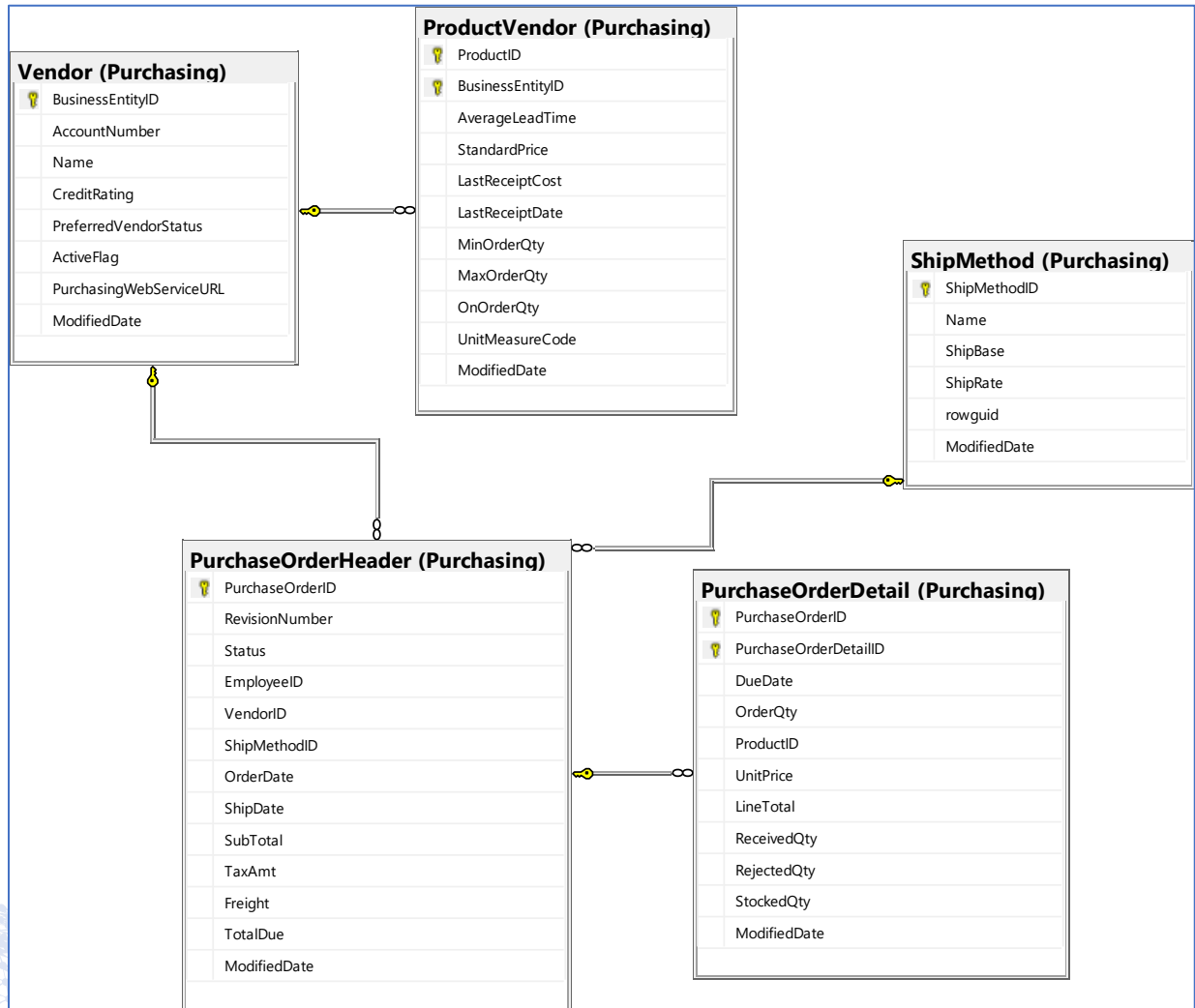
Purchasing Schema Entity Relationship (ER) Diagram

Tables:

The **ProductVendor** table is a cross-reference table mapping vendors with the products they supply, where the product can be supplied by multiple vendors and the vendor can supply multiple.

NumOfVendors	NumOfProducts
1	113
2	109
3	43

It contains these main columns:

- **AverageLeadTime:** The average time (in days) between placing an order with the vendor and receiving the purchased product.
- **StandardPrice:** The vendor's usual selling price.
- **LastReceiptCost:** The selling price when last purchased.
- **Min/Max OrderQty:** The minimum and maximum quantity that should be ordered.
- **OnOrderQty:** The quantity currently on order.
- **UnitMeasureCode:** The product's unit of measure, for example (Boxes, Cubic centimeters, Kilogram, and others).

The **Vendor** table contains the companies from whom Adventure Works Cycles purchases parts or other goods, it contains these main columns:

- **BusinessEntityID:** Primary key for Vendor records. Foreign key to BusinessEntity.BusinessEntityID.
- **Name:** Company name.
- **CreditRating:** 1 = Superior, 2 = Excellent, 3 = Above average, 4 = Average, 5 = Below average.
- **PreferredVendorStatus:** 0 = Do not use if another vendor is available. 1 = Preferred over other vendors supplying the same product.
- **ActiveFlag:** 0 = Vendor no longer used. 1 = Vendor is actively used.

The **ShipMethod** table is the shipping company lookup table, it contains the shipping company name (5 different companies), the minimum shipping charge, and the shipping charge per pound (0.45359237 kg). Also, the same shipment methods are used in the Sales when a new sales order is created.

The PurchaseOrder follows the Header/Detail model, which is a form of a one-to-many relationship, with the header table containing primary key fields that uniquely identify data in the detail table.

The **PurchaseOrderHeader** table contains the general purchase order information, it contains these main columns:

- **PurchaseOrderID:** Primary key (ID generated for each order).
- **Status:** Order current status. 1 = Pending; 2 = Approved; 3 = Rejected; 4 = Complete.
- **EmployeeID:** Employee who created the purchase order. Foreign key to Employee.BusinessEntityID.
- **VendorID:** Vendor with whom the purchase order is placed. Foreign key to Vendor.BusinessEntityID.
- **ShipMethodID:** Shipping method. Foreign key to ShipMethod.ShipMethodID.

- **OrderDate:** Purchase order creation date.

Also, it contains the tax amount and the shipping cost.

The **PurchaseOrderDetail** table contains the individual products associated with a specific purchase order, and it contains these main columns:

- **PurchaseOrderID:** Primary key. Foreign key to PurchaseOrderHeader.PurchaseOrderID.
- **PurchaseOrderDetailID:** Primary key (Identity).
- **ProductID:** Product identification number. Foreign key to Product.ProductID.
- **DueDate:** Date the product is expected to be received.
- **OrderQty:** Quantity ordered.
- **ReceivedQty:** Quantity actually received from the vendor.
- **RejectedQty:** Quantity rejected during the inspection (checking).
- **StockedQty:** Quantity accepted into inventory. Computed as ReceivedQty - RejectedQty.
- **UnitPrice:** Vendor's selling price of a single product.

Business logic:

Purchase orders and vendor-related information are managed via the Purchasing schema. Initially, vendor-related information besides the products they sell should be saved anytime a company initiates a new contract with that vendor, and then the predefined products should be associated with the vendors who can supply them. When an employee creates a new purchase order, we specify the vendor by checking the active and preferred status, as well as the shipment method and order quantity, and wait until the order is received before storing the product delivery details such as the received and rejected quantity.

- **Production schema:**
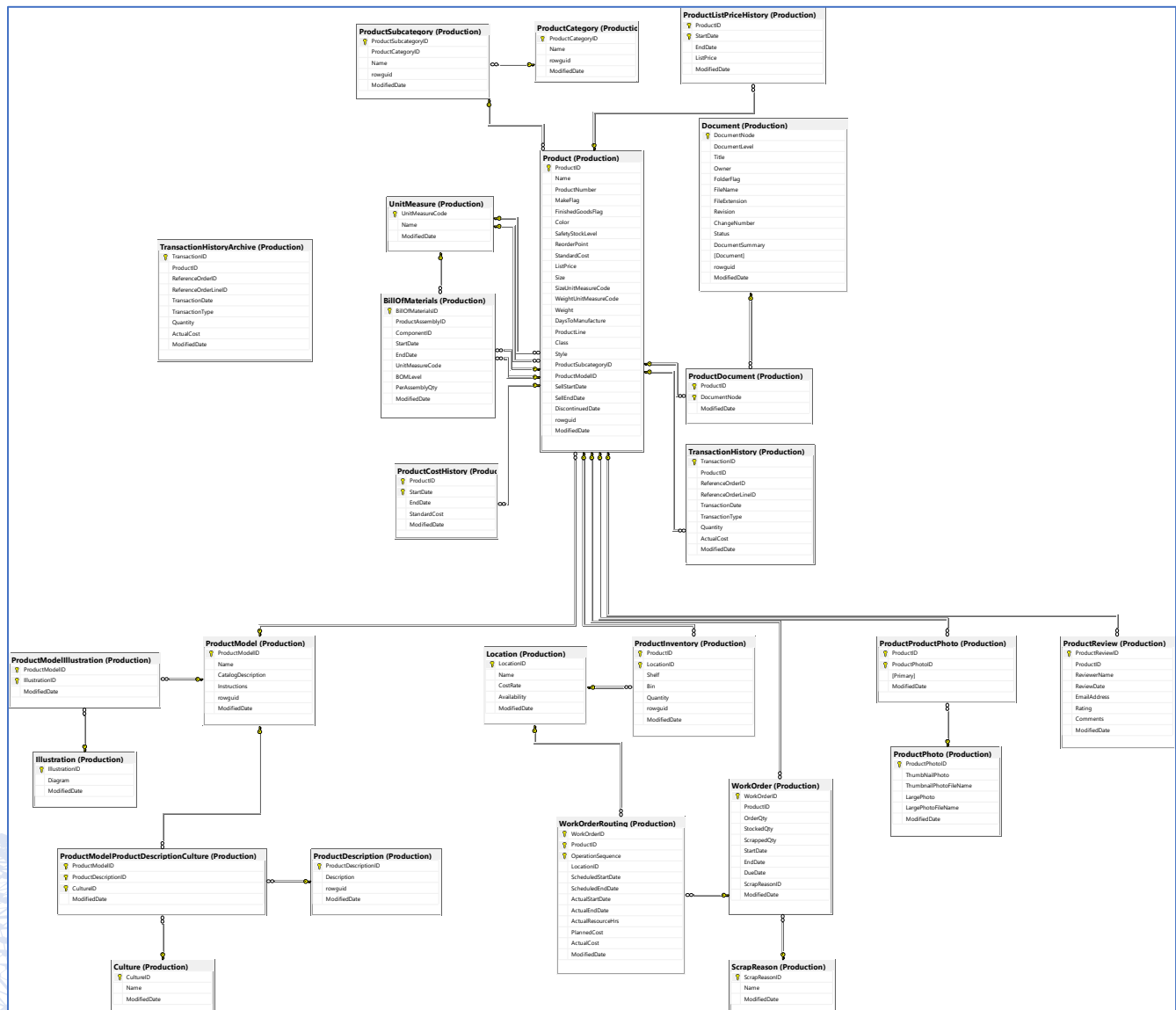
Production Schema Entity Relationship (ER) Diagram

Tables:

We can logically divide the tables in the production schema into three main categories: Products, Manufacturing, and Inventory.

1. Products:

The **Product** table contains the products sold or used in the manufacturing of sold products, it includes these main columns:

- **ProductID:** Primary key for Product records.
- **Name:** Name of the product.
- **MakeFlag:** 0 = Product is purchased, 1 = Product is manufactured in-house (default).
- **FinishedGoodsFlag:** 0 = Product is not a salable item. 1 = Product is salable.
- **SafetyStockLevel:** Minimum inventory quantity.
- **ReorderPoint:** Inventory level that triggers a purchase order or work order.
- **StandardCost:** Standard cost of the product.
- **Size/WeightUnitMeasureCode:** Unit of measure, foreign key to Production.UnitMeasure.
- **DaysToManufacture:** Number of days required to manufacture the product.
- **ProductLine:** R = Road, M = Mountain, T = Touring, S = Standard.
- **Class:** H = High, M = Medium, L = Low.
- **ProductSubcategoryID:** The product is a member of this product subcategory. Foreign key to ProductSubCategory.ProductSubCategoryID.
- **ProductModelID:** The product is a member of this product model. Foreign key to ProductModel.ProductModelID.
- **SellStartDate:** Date the product was available for sale.
- **SellEndDate:** The date the product was no longer available for sale.

The **ProductCategory** table is a high-level product categorization, it contains 4 categories: Bikes, Components, Clothing, and Accessories.

| CategoryName	NumOfProducts
Components	134
Bikes	97
Clothing	35
Accessories	29

The **ProductSubcategories** table contains 37 different sub-categories, for example, Mountain Bikes, Brakes, Headsets, and Locks.

The **ProductModel** table describes the product model classification, it contains these main columns:

- **Name:** Product model description, for example: Classic Vest, and ML Road Pedal.
- **CatalogDescription:** Detailed product catalog information in XML format.

The **ProductModelIllustration** table is a cross-reference table mapping product models and illustrations (the product may have multiple illustrations).

The **Illustration** table contains the bicycle assembly diagrams in XAML format (Illustrations used in manufacturing instructions), for example:
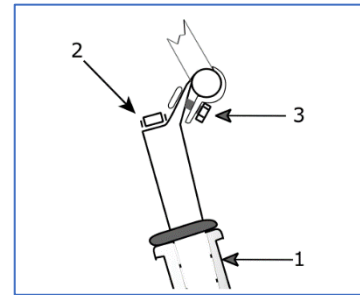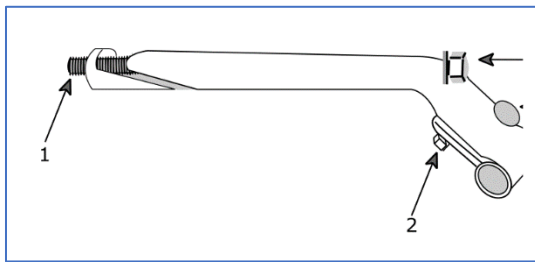
The **Culture** table is a Lookup table containing the languages in which some AdventureWorks data is stored, it contains the CultureID, for example, ar, en, and, fr. Also contains the culture Name, for example: Arabic, English, and, French.

The **ProductDescription** table contains the product descriptions in several languages, for example:

EN: Lightweight, wind-resistant, packs to fit into a pocket.

AR: علب خفيفة الوزن، ومقاومة للريح، تناسب حجم الجيب.

The **ProductModelProductDescriptionCulture** table is a cross-reference table mapping the product description and the language the description is written in.

The **ProductDocument** table is a cross-reference table mapping products to related product documents, where the product may have multiple documents and the document may belong to multiple products.

The **Document** table contains the product maintenance documents, it includes these main columns:

- **Title:** Title of the document.
- **Owner:** Employee who controls the document. Foreign key to Employee.BusinessEntityID.
- **Status:** 1 = Pending approval, 2 = Approved, 3 = Obsolete.
- **DocumentSummary:** Document abstract.

Also, it contains the file name and file extension.

The **ProductProductPhoto** table is a cross-reference table mapping products and product photos (the product may have multiple photos).

The **ProductPhoto** table contains the Product images as a binary file.

The **ProductReview** table contains the customer reviews of products they have purchased; it includes these main columns:

- **ProductID:** Product identification number. Foreign key to Product.ProductID.
- **ReviewerName:** Name of the reviewer.
- **EmailAddress:** Reviewer's email address (not associated with the person's email address).
- **Rating:** Product rating given by the reviewer. The scale is 1 to 5 with 5 as the highest rating.
- **Comments:** Reviewer's comments.

These reviews can be used later for further analysis.

The **UnitMeasure** table is a unit of measures lookup table, it contains 38 different measures related to volume, length, temperature, and others.

2. Manufacturing:

The **BillOfMaterials** table contains the Items required to make bicycles and bicycle subassemblies. It identifies the hierarchical relationship between a parent product and its components, it includes these main columns:

- **ProductAssemblyID:** Parent product identification number. Foreign key to Product.ProductID.
- **ComponentID:** Component identification number. Foreign key to Product.ProductID.
- **BOMLevel:** Indicates the depth the component is from its parent (AssemblyID).
- **PerAssemblyQty:** Quantity of the component needed to create the assembly.

Some products such as "Road-150 Red Bicycle, 52" consist of 18 different components.

The **ProductListPriceHistory** table contains the changes in the list price of a product over time, it contains the start and end date for the specific list price and the list price for each period of time. Also, the end date may be null which means the given list price is the current price.

The **ProductCostHistory** table contains the Changes in the cost of a product over time, it contains the start and end date for the specific cost and the StandardCost for each period of time. Also, the end date may be null which means the given cost is the current cost.

The **WorkOrder** table contains the manufacturing work orders, it includes these main columns:

- **WorkOrderID:** Primary key for WorkOrder records (Identity).
- **ProductID:** Product identification number. Foreign key to Product.ProductID.
- **OrderQty:** Product quantity to build.
- **ScrappedQty:** Quantity that failed inspection.
- **Start/End Date:** Work order starts and end date.
- **DueDate:** Work order due date.
- **ScrapReasonID:** Reason for inspection failure.

The **ScrapReason** table is the manufacturing failure reasons lookup table, it contains 16 different reasons, for example: Wheel misaligned, and Drill size too large.

The **TransactionHistory** table contains the records of each purchase order, sales order, or work order transaction year to date, it contains these main columns:

- **TransactionID:** Primary key for TransactionHistory records (Identity).
- **ProductID:** Product identification number. Foreign key to Product.ProductID.
- **ReferenceOrderID:** Purchase order, sales order, or work order identification number (associated with SalesOrderHeader. SalesOrderID and PurchaseOrderHeader. PurchaseOrderID).
- **TransactionDate:** Date and time of the transaction.
- **TransactionType:** W = WorkOrder, S = SalesOrder, P = PurchaseOrder.
- **Quantity:** Product quantity.
- **ActualCost:** Product cost.

The **TransactionHistoryArchive** table contains the transactions for previous years.

TransactionType	Year	NumOfTransactions
PurchaseOrder	2011	68
SalesOrder	2011	5716
WorkOrder	2011	8904
PurchaseOrder	2012	576
SalesOrder	2012	21689
WorkOrder	2012	18693
PurchaseOrder	2013	2914
SalesOrder	2013	56573
WorkOrder	2013	28905
PurchaseOrder	2014	5230
SalesOrder	2014	37339
WorkOrder	2014	16089

The **WorkOrderRouting** table contains the work order details., it contains these main columns:

- **WorkOrderID:** Primary key. Foreign key to WorkOrder.WorkOrderID.
- **ProductID:** Primary key. Foreign key to Product.ProductID.
- **OperationSequence:** Primary key. Indicates the manufacturing process sequence.
- **LocationID:** Manufacturing location where the part is processed. Foreign key to Location.LocationID.
- **Scheduled Start/End Date:** Planned manufacturing start and end dates.
- **Actual Start/End Date:** Actual start and end dates.
- **PlannedCost:** Estimated manufacturing cost.
- **ActualCost:** Actual manufacturing cost.

These columns can be used in further analysis to determine the reason behind the manufacturing delay.

3. **Inventory:**

The **ProductInventory** table contains the product inventory information, it includes these main columns:

- **ProductID:** Product identification number. Foreign key to Product.ProductID.
- **LocationID:** Inventory location identification number. Foreign key to Location.LocationID.
- **Shelf:** Storage compartment within an inventory location.
- **Bin:** Storage container on a shelf in an inventory location.
- **Quantity:** Quantity of products in the inventory location.

The **Location** table contains the product inventory and manufacturing locations, it includes these main columns:

- **LocationID:** Primary key for Location records (Identity).
- **Name:** Location description.
- **CostRate:** Standard hourly cost of the manufacturing location.
- **Availability:** Work capacity (in hours) of the manufacturing location.

It contains 7 different inventories such as "Metal Storage" and 7 different manufacturing locations such as "Frame Welding".

Business logic:

The Production schema of Adventure Works Cycles encompasses the entire production process, which involves meticulous management of manufacturing procedures, inventory control, and the production of various products. The company's business operations revolve around two distinct types of products: purchased items and those manufactured in-house. Each product and component within the schema hold essential details regarding its characteristics, including class, style, category, price, and other relevant attributes. Furthermore, all products are categorized into predefined categories and sub-categories for streamlined organization. Additionally, comprehensive maintenance documents and assembly instructions, accompanied by detailed diagrams, are associated with each product. Moreover, the schema includes essential elements such as product models, product catalogs, and product photos, all of which are available in multiple languages to cater to a diverse customer base. In addition, Adventure Works Cycles' website provides a platform for customers to rate products and share their reviews. The schema also captures critical information related to manufacturing work orders, including order quantity, scrapped quantity, and scrapping reasons for each order. The logging of all orders within the database ensures comprehensive monitoring and analysis. Lastly, the schema houses pertinent data about product inventory and various manufacturing locations, enabling efficient management and control of these crucial aspects of the business.
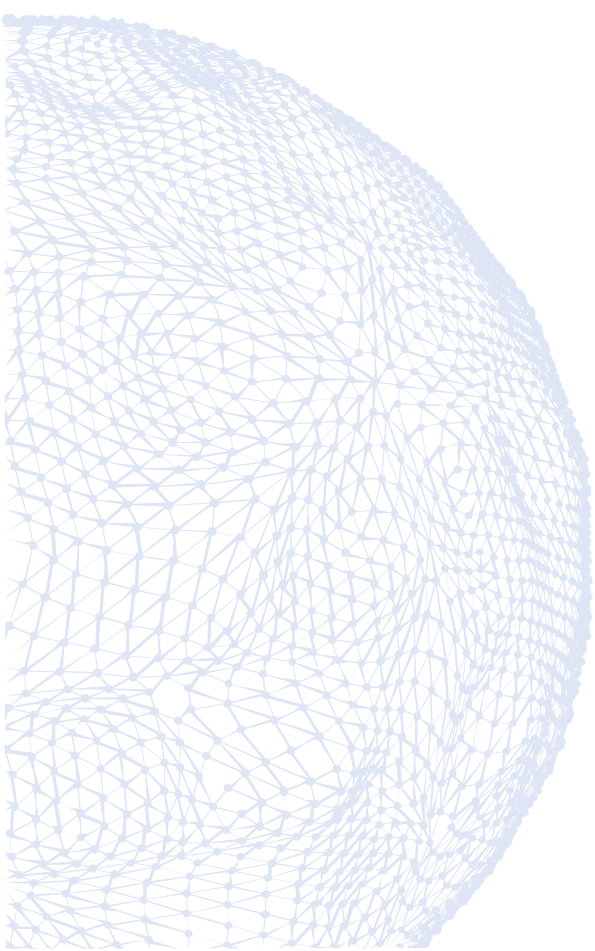
- **Sales schema:**
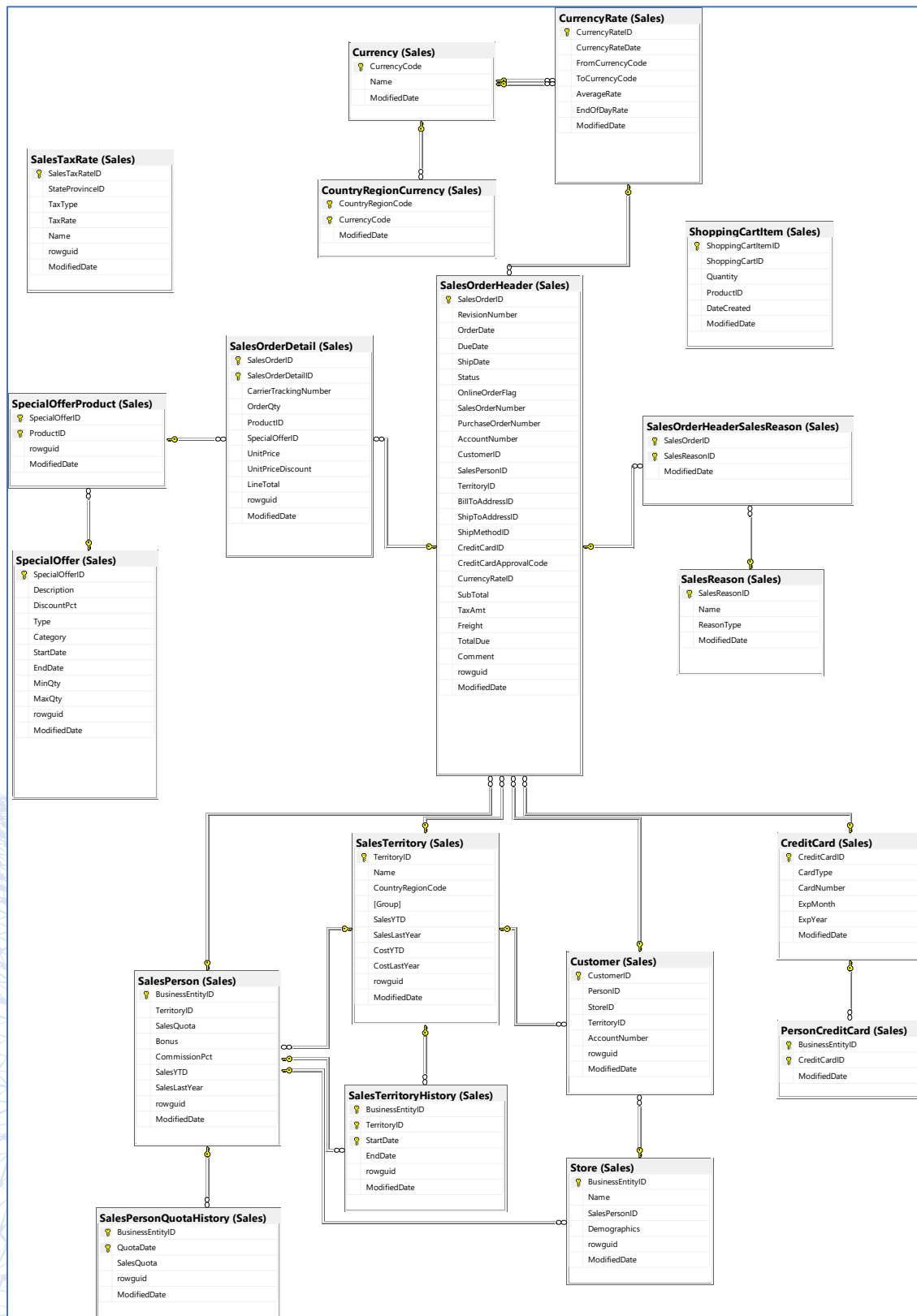
Sales Schema Entity Relationship (ER) Diagram

Tables:

The **Store** table contains customers (resellers) of Adventure Works products, it includes these main columns:

- **BusinessEntityID:** Primary key. Foreign key to BusinessEntity.BusinessEntityID.
- **Name:** Name of the store.
- **SalesPersonID:** ID of the salesperson assigned to the customer. Foreign key to SalesPerson.BusinessEntityID.
- **Demographics:** Demographic information about the store such as the number of employees, annual sales, and store type in XML format.

The **Customer** table contains the customer information, it includes these main columns:

- **CustomerID:** Primary key (Identity).
- **PersonID:** Foreign key to Person.BusinessEntityID.
- **StoreID:** Foreign key to Store.BusinessEntityID.
- **TerritoryID:** ID of the territory in which the customer is located. Foreign key to SalesTerritory.TerritoryID.

The **SalesPerson** table contains the sales representative's current information, it includes these main columns:

- **BusinessEntityID:** Primary key for SalesPerson records. Foreign key to Employee.BusinessEntityID.
- **TerritoryID:** Territory currently assigned to. Foreign key to SalesTerritory.TerritoryID.
- **SalesQuota:** Projected yearly sales.
- **Bonus:** Bonus due if the quota is met.
- **CommissionPct:** Commision percent received per sale.
- **SalesYTD:** Sales total year to date.
- **SalesLastYear:** Sales total of the previous year.

The **SalesPersonQuotaHistory** table contains sales performance tracking, it includes these main columns:

- **BusinessEntityID:** Salesperson identification number. Foreign key to SalesPerson.BusinessEntityID.
- **QuotaDate:** Sales quota date.
- **SalesQuota:** Sales quota amount.

The **SalesTerritory** table is a sales territory lookup table, it includes these main columns:

- **TerritoryID:** Primary key for SalesTerritory records.
- **Name:** Sales territory description.
- **CountryRegionCode:** ISO standard country or region code. Foreign key to CountryRegion.CountryRegionCode.
- **Group:** Geographic area to which the sales territory belongs (for example, North America, and Europe).
- **SalesYTD:** Sales in the territory year to date.
- **CostYTD:** Business costs in the territory year to date.

TerritoryName	NumOfSalesPersons	NumOfCustomers
Australia	1	3665
Canada	2	1791
Central – US	1	132
France	1	1884
Germany	1	1852
Northeast – US	1	113
Northwest – US	3	3520
Southeast – US	1	176
Southwest – US	2	4696
United Kingdom	1	1991

The **SalesTerritoryHistory** table contains the sales representative transfers to other sales territories, it includes these main columns:

- **BusinessEntityID:** Primary key. The sales representative. Foreign key to SalesPerson.BusinessEntityID.
- **TerritoryID:** Primary key. Territory identification number. Foreign key to SalesTerritory. TerritoryID.
- **StartDate:** Primary key. Date the sales representative started work in the territory.
- **EndDate:** Date the sales representative left work in the territory (may be null, which means that the sales representative still works in this territory).

The **CreditCard** table contains the customer's credit card information, it includes these main columns:

- **CreditCardID:** Primary key for CreditCard records.
- **CardType:** Credit card name, which contains 4 different types, for example, SuperiorCard, and Vista.
- **CardNumber:** Credit card number.
- **Exp Month/Year:** Credit card expiration month and year.

The **PersonCreditCard** table is a cross-reference table mapping people to their credit card information in the CreditCard table, where the people may have multiple credit cards.

The **SalesTaxRate** table is the tax rate lookup table, it contains these main columns:

- **SalesTaxRateID:** Primary key for SalesTaxRate records.
- **StateProvinceID:** State, province, or country/region the sales tax applies. Foreign key to StateProvince.StateProvinceID.
- **TaxType:** 1 = Tax applied to retail transactions, 2 = Tax applied to wholesale transactions, 3 = Tax applied to all sales (retail and wholesale) transactions.
- **TaxRate:** Tax rate amount.
- **Name:** Tax rate description.

The **ShoppingCartItem** table contains online customer orders until the order is submitted or canceled, it contains these main columns:

- **ShoppingCartItemID:** Primary key for ShoppingCartItem records.
- **ShoppingCartID:** Shopping cart identification number (created from the web session).
- **Quantity:** Product quantity ordered.
- **ProductID:** Product ordered. Foreign key to Product.ProductID.

The SalesOrder follows the Header/Detail model, which is a form of a one-to-many relationship, with the header table containing primary key fields that uniquely identify data in the detail table.

The **SalesOrderHeader** table contains general sales order information, it includes these main columns:

- **SalesOrderID:** Sales order primary key (Identity).
- **OrderDate:** Dates the sales order was created.
- **DueDate:** The date the order is due to the customer.
- **ShipDate:** The date the order was shipped to the customer.
- **Status:** Order current status. 1 = In process; 2 = Approved; 3 = Backordered; 4 = Rejected; 5 = Shipped; 6 = Cancelled.

- **OnlineOrderFlag:** 0 = Order placed by the salesperson. 1 = Order placed online by the customer.
- **PurchaseOrderNumber:** Customer purchase order number reference, it equals null if the OnlineOrderFlag = 1.
- **CustomerID:** Customer identification number. Foreign key to Customer.CustomerD.
- **SalesPersonID:** Salesperson who created the sales order. Foreign key to SalesPerson.BusinessEntityID. Equals null if the OnlineOrderFlag = 1.
- **TerritoryID:** Territory in which the sale was made. Foreign key to SalesTerritory.TerritoryID.
- **BillToAddressID:** Customer billing address. Foreign key to Address.AddressID.
- **ShipToAddressID:** Customer shipping address. Foreign key to Address.AddressID.
- **ShipMethodID:** Shipping method. Foreign key to ShipMethod.ShipMethodID.
- **CreditCardID:** Credit card identification number. Foreign key to CreditCard.CreditCardID.
- **CreditCardApprovalCode:** The approval code provided by the credit card company; may be null.
- **CurrencyRateID:** Currency exchange rate used. Foreign key to CurrencyRate.CurrencyRateID.
- **TaxAmt:** Tax amount.
- **Freight:** Shipping cost.
- **Comment:** Sales representative comments.

The **CurrencyRate** table contains the currency exchange rates, it includes these main columns:

- **CurrencyRateID:** Primary key for CurrencyRate records.
- **CurrencyRateDate:** Date and time the exchange rate was obtained.
- **FromCurrencyCode:** Exchange rate was converted from this currency code.
- **ToCurrencyCode:** Exchange rate was converted to this currency code.
- **AverageRate:** Average exchange rate for the day.
- **EndOfDayRate:** Final exchange rate for the day.

The **Currency** table is a lookup table containing standard ISO currencies and currency names.

The **CountryRegionCurrency** table is a Cross-reference table mapping ISO currency codes to a country or region, where the country may have multiple currencies and the currency may belong to multiple countries, it contains these main columns:

- **CountryRegionCode:** ISO code for countries and regions. Foreign key to CountryRegion.CountryRegionCode.
- **CurrencyCode:** ISO standard currency code. Foreign key to Currency.CurrencyCode.

The **SalesReason** table is a lookup table of customer purchase reasons, it contains these main columns:

- **SalesReasonID:** Primary key for SalesReason records.
- **Name:** Sales reason description.
- **ReasonType:** Category the sales reason belongs to.

There are 10 different sales reasons, for example Manufacturer, and Magazine Advertisement, and 3 reason types: Marketing, Other, and, Promotion.

The **SalesOrderHeaderSalesReason** table is a cross-reference table mapping sales orders to sales reason codes, where the sales order may have multiple sales reasons.

The **SalesOrderDetail** table contains individual products associated with a specific sales order, it includes these columns:

- **SalesOrderID:** Primary key. Foreign key to SalesOrderHeader.SalesOrderID.
- **SalesOrderDetailID:** Primary key. One incremental unique number per product sold.
- **CarrierTrackingNumber:** Shipment tracking number supplied by the shipper (It may be null).
- **OrderQty:** Quantity ordered per product.
- **ProductID:** Product sold to a customer. Foreign key to Product.ProductID.
- **SpecialOfferID:** Promotional code. Foreign key to special offers.SpecialOfferID.
- **UnitPrice:** The selling price of a single product.
- **UnitPriceDiscount:** Discount amount.

The **SpecialOfferProduct** table is a cross-reference table mapping products to special offer discounts, where the product may have multiple special offers and the special offer may belong to a single product.

The **SpecialOffer** table is the sale discounts lookup table, it contains these main columns:

- **SpecialOfferID:** Primary key for SpecialOffer records.
- **Description:** Discount description.
- **DiscountPct:** Discount percentage.
- **Type:** Discount type category, there are 6 different types, for example, Seasonal Discount.
- **Category:** Group the discount applies to such as Reseller or Customer.
- **Start/End Date:** Discount start and end dates.