TOC for MS SQL Server

Day 1

- Introduction of SQL
    - DBMS & RDBMS
    - MS SQL Server DB Architecture
    - Database files
- Overview of SQL Database Objects
- System Databases
- SQL Commands- DDL, DML, DCL, TCL, DQL
- Database Constraints
- Creating Database, Schemas & Tables
- Backup and Restore Database

Day 2

- Inserting, Updating and Deleting Data
- Identity Column
- Filtering Data – Where Clause
    - Conditional Operators
    - Relational Operators
    - IN / NOT IN Operator
    - Between AND Operator
    - Like Operator
    - IS NULL / IS NOT NULL Operator
- Handling Null Values
- Column Alias
- Order By Clause
- Built-In Functions
    - Number / Math Functions
    - String Functions
    - Date / Time Functions

Day 3

- Built-In Functions
    - Analytical Functions
    - Aggregate Functions
- Group By Clause
- OVER Clause and PARTITION BY clause
- Top N Analysis using Ranking functions
- Derived Tables
- Common Table Expressions (CTE)
- Temporary Tables
- Sequences

Day 4

- SET Operations
    - UNION, UNION ALL, INTERSECT, EXCEPT

- Joins & it's Types
    - CROSS JOIN
    - INNER JOIN
    - OUTER JOIN
    - SELF JOIN
- Subqueries & it's types
    - Single Row Subquery
    - Multi Row Subquery
    - ANY, ALL Operators
    - Multi Column Subquery
    - Corelated Subquery

Day 5

- Synonyms
- Views
- Procedural Blocks
    - Variables
    - Print Command
    - Decision Construct – IF ELSE
    - Iterations & Loops
- Stored Procedures
    - Advantages
    - Input & Output Parameters

Day 6

- User Defined Functions
    - Scalar Valued Functions
    - Table Valued Functions
- Procedures vs Functions
- Triggers
    - DDL Triggers – DB Level, System Level
    - DML Triggers – After, Instead of

Day 7

- Cursors
- Exception Handling
    - System Exceptions
    - User Defined Exceptions
- Transaction Management
    - ACID properties
    - Isolation Locks
    - Challenges while working with multiple transactions and Shared resource

Day 8

- Pivot / Unpivot
- Indexes
    - Clustered
    - Non-Clustered
- Best practices on Performance tuning
- Tools for Query Optimization
- SQL Coding Guidelines

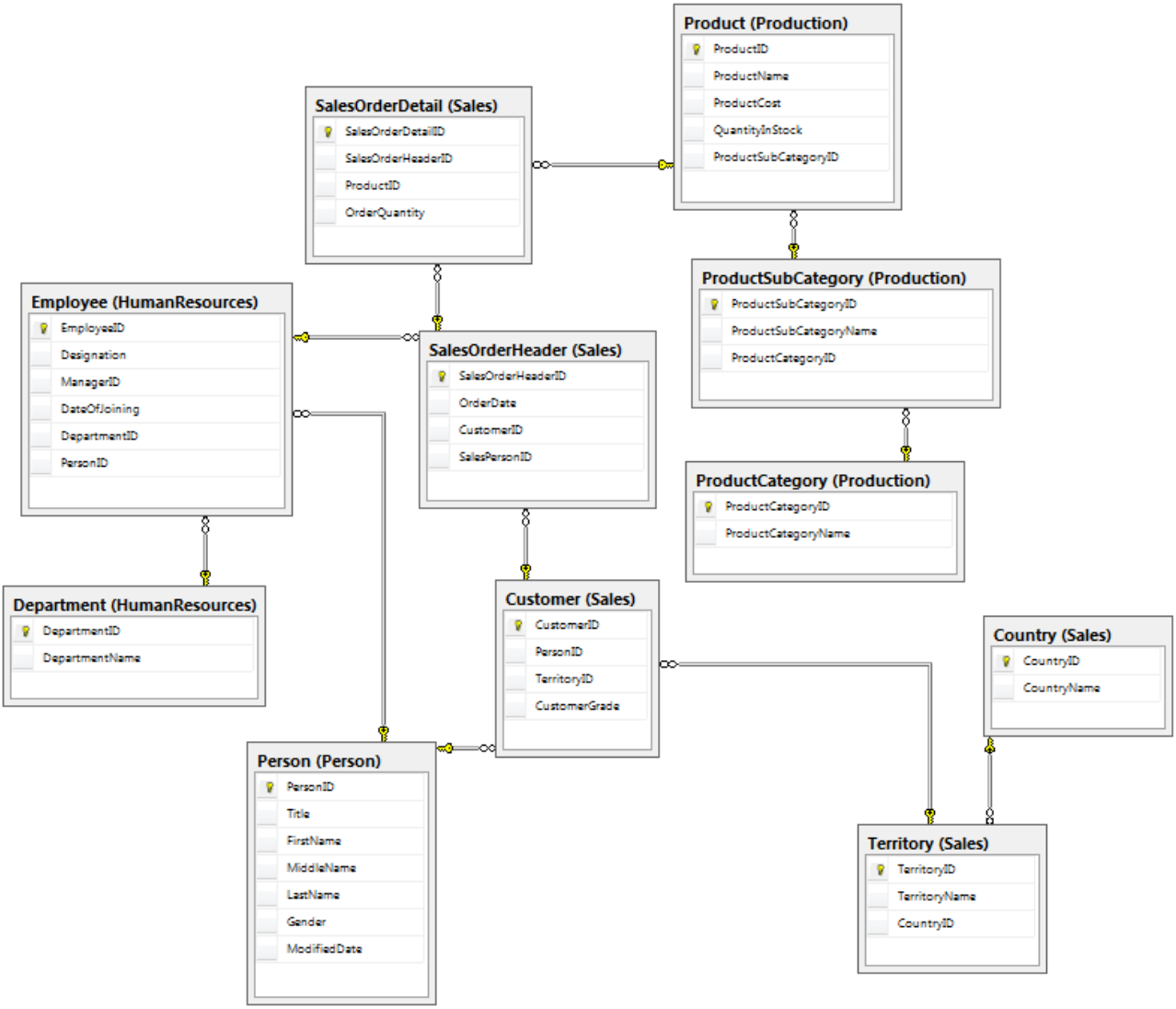
Day 9

- Normalization
    - 1 NF
    - 2 NF
    - 3 NF
- Revision & Summary

Assignments:

Day 1

1. Create Physical Data Model as per the diagram below. Identify and apply correct datatypes and constraints for the columns of each table



1. Restore AdventureWorks Database
   Step 1: Download: https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms – Download AdventureWorks2019.bak
   Step 2: Restore the database from backup file - https://www.youtube.com/watch?v=mRI8vpn-tyk

Day 2

1. Write a query that displays all the rows from the Person.Person table where the rows were modified after December 29, 2000. Display the business entity ID number, the name columns, and the modified date.
2. Rewrite the query from question 1 so that it displays the rows that were not modified during December 2000.
3. Write a query that displays the product ID and name for each product from the Production.Product table with the name starting with Chain.

4. Write a query that displays the business entity ID number, first name, middle name, and last name from the Person.Person table for only those rows that have E or B stored in the middle name column.
5. Write a query displaying the order ID, order date, and total due from the Sales.SalesOrderHeader table. Retrieve only those rows where the order was placed during the month of September 2001 and the total due exceeded $1,000.
6. Write a query displaying the sales orders where the total due exceeds $1,000. Retrieve only those rows where the salesperson ID is 279 or the territory ID is 6.
7. Write a query displaying the ProductID, Name, and Color columns from rows in the Production.Product table. Display only those rows in which the color is not blue.
8. Write a query that returns the business entity ID and name columns from the Person.Person table. Sort the results by LastName, FirstName, and MiddleName.
9. Write a query that displays in the "AddressLine1 (City PostalCode)" format from the Person.Address table.
10. Write a query using the Production.Product table displaying the product ID, color, and name columns. If the color column contains a NULL value, replace the color with No Color.
11. Modify the query written in question 2 so that the description of the product is displayed in the "Name: Color" format. Make sure that all rows display a value even if the Color value is missing.
12. Write a query using the Sales.SpecialOffer table. Display the difference between the MinQty and MaxQty columns along with the SpecialOfferID and Description columns.
13. Write a query using the Sales.SpecialOffer table that multiplies the MaxQty column by the DiscountPCT column. If the MaxQty value is null, replace it with the value 10. Include the SpecialOfferID and Description columns in the results.
14. Write a query that displays the first 10 characters of the AddressLine1 column in the Person.Address table.
15. Write a query that calculates the number of days between the date an order was placed and the date that it was shipped using the Sales.SalesOrderHeader table. Include the SalesOrderID, OrderDate, and ShipDate columns.
16. Write a query that displays only the date, not the time, for the order date and ship date in the Sales.SalesOrderHeader table. (Use any of the styles that return only date)
17. Write a query that adds six months to each order date in the Sales.SalesOrderHeader table. Include the SalesOrderID and OrderDate columns.
18. Write a query that displays the year of each order date and the numeric month of each order date in separate columns in the results. Include the SalesOrderID and OrderDate columns.
19. Write a statement that generates a random number between 1 and 10 each time it is run.
20. Write a query using the Sales.SalesOrderHeader table to display the orders placed during 2001 by using a function. Include the SalesOrderID and OrderDate columns in the results.
21. Write a query using the Sales.SalesOrderHeader table listing the sales in order of the month the order was placed and then the year the order was placed. Include the SalesOrderID and OrderDate columns in the results.

Day 3

1. Write a query to determine the number of customers in the Sales.Customer table.
2. Write a query using the Production.Product table that displays the minimum, maximum, and average ListPrice.
3. Write a query that shows the total number of items ordered for each product. Use the Sales.SalesOrderDetail table to write the query.
4. Write a query using the Sales.SalesOrderDetail table that displays a count of the detail lines for each SalesOrderID.
5. Write a query using the Production.Product table that lists a count of the products in each product line.
6. Write a query that displays the count of orders placed by year for each customer using the Sales.SalesOrderHeader table.
7. Write a query that creates a sum of the LineTotal in the Sales.SalesOrderDetail table grouped by the SalesOrderID. Include only those rows where the sum exceeds 1,000.
8. Write a query that groups the products by ProductModelID along with a count. Display the rows that have a count that equals 1.
9. Write a query using the Sales.SalesOrderHeader, Sales.SalesOrderDetail, and Production.Product tables to display the total sum of products by ProductID and OrderDate.
10. Display the 3rd joined employee.
11. Display the customer who has placed 2nd highest orders
12. Display top 25% of costliest products in every subcategory
13. Create a sequence to be used in two different temporary tables (Note: create temporary tables if required)

Day 4

1. The HumanResources.Employee table does not contain the employee names. Join that table to the Person.Person table on the BusinessEntityID column. Display the job title, birth date, first name, and last name.
2. The customer names also appear in the Person.Person table. Join the Sales.Customer table to the Person.Person table. The BusinessEntityID column in the Person.Person table matches the PersonID column in the Sales.Customer table. Display the CustomerID, StoreID, and TerritoryID columns along with the name columns.
3. Write a query that joins the Sales.SalesOrderHeader table to the Sales. SalesPerson table. Join the BusinessEntityID column from the Sales.SalesPerson table to the SalesPersonID column in the Sales.SalesOrderHeader table. Display the SalesOrderID along with the SalesQuota and Bonus.
4. The catalog description for each product is stored in the Production.ProductModel table. Display the columns that describe the product from the Production.Product table, such as the color and size along with the catalog description for each product.
5. Write a query that displays the names of the customers along with the product names that they have purchased. Hint: Five tables will be required to write this query!
6. Write a query that displays all the products along with the SalesOrderID even if an order has never been placed for that product. Join to the Sales.SalesOrderDetail table using the ProductID column.

7. The Sales.SalesOrderHeader table contains foreign keys to the Sales.CurrencyRate and Purchasing.ShipMethod tables. Write a query joining all three tables, making sure it contains all rows from Sales.SalesOrderHeader. Include the CurrencyRateID, AverageRate, SalesOrderID, and ShipBase columns.
8. Get all the order details to gererate a report that displays, OrderID, OrderNumber, OrderDate, Shipping Date and the product names, subcategory and category which are the part of that order and include the name of customer who has placed the order as well as the name of territory and country from where order has been placed [Hint: Identify the correct set of related tables]
9. Get the Youngest Employee
10. Create a temp. table and copy the data form Production.Product table (only red colored products) in the temp. table [Hint: use subquery]

Day 5

1. Write a procedure to insert data in Patient and PatientAddress tables using input parameters [PatientID is an Identity column in Patient table and PatientAddress table is related to Patient table]
2. An input string representing passenger data comes in a below format to a procedure. Extract the data from the string and store in a temporary table. String format: "[P9001,John Roy,Male,12-Jan-2009]"
3. Modify the Day 5 - Lab 2 to validate the below
    1. No duplicate entry for a passenger must be attempted to insert in table
    2. The Age of the passenger must be between 6 to 90

Day 6

1. Create a Function to check and print all the prime numbers between a range defined by user.
2. Create a Function that takes CategoryName as a parameter and gets the products associated with that category.
3. Create a trigger to implement banking scenario. Whenever the bank balance is updated for any bank account a transaction is captured in a Bank Transaction table. The Bank Transaction table acts as a source to generate bank statements. Hint: Need to create two tables (BankAccounts with some rows and BankTransactions with no records) and a trigger for Update on Balance column of BankAccounts table.
   Table: BankAccounts
   Columns:
    - AccountID (Numeric- Autogenerated)
    - CustomerName
    - AccountType (Current / Saving)
    - Balance (>0)
    - Modified Date

   Table: BankTransactions
   Columns
    - TransactionID (Numeric- Autogenerated)

- AccountID (FK)
- TransactionDate
- TransactionType (Debit / Credit)
- TransactionAmount

Day 7

1. Create a procedure that takes a string parameter. The input string may be a string or a numeric or NULL value. Convert the string to Integer. If it cannot be converted write an exception handling section to handle the appropriate error. If the string is converted to integer print Hello the input integer number of times
2. Create a temp table to represent employees. Design a user defined exception to handle the salary input less than 10000.
3. Write a cursor to fetch top 10 costliest products
4. Document your understanding and possible solutions to Deadlock concept [Hint: You may explore online]

Day 8

1. Document your understanding on Query plan
2. Document your understanding on different types of Scans while reading from a table
3. Consider the below SalesPerson table

Name	Year	Sales
Sam	2010	20000
John	2010	30000
John	2010	15000
Sam	2011	70000
John	2011	89000
Sam	2011	12000

Calculate the overall sales for both the salespersons corresponding to the year values as shown below (Pivot)

Year	Sam	John
2010	20000	45000
2011	82000	89000

4. Document some Good and Bad practices while working with the below DB objects / commands
   a. Tables
   b. Joins
   c. Indexes

Day 9

Miscellaneous labs (Explore online and create a PPT presentation)

1. What is Offset and Fetch command in SELECT query? Explain with examples
2. What is rollup and cube? Explain with examples
3. How to work with XML data in MS SQL?
4. How to work with JSON data in MS SQL?