Module 1

- Database
  - Database Files
  - Types of Databases
- Data Types
- Data Integrity
  - Entity Integrity
  - Referential Integrity
  - Domain Integrity
- User-defined Integrity
- Create/Alter/Drop

Practice Exercise

Practice 1

Practice the same with SSMS GUI and writing queries as well.

Practice 2

Write a SQL statement to create a table named countries including columns CountryId, CountryName and RegionId and make sure that no countries except Italy, India and China will be entered in the table. and combination of columns CountryId and RegionId will be unique.

Practice 3

Write a SQL statement to create a table named JobHistory including columns EmployeeId, StartDate, End_Eate, Job_Id and Department_Id and make sure that the value against column EndDate will be entered at the time of insertion to the format like '–/–/—-'.

Practice 4

Write a SQL statement to create a table named jobs including columns JobId, JobTitle, MinSalary and MaxSalary, and make sure that, the default value for JobTitle is blank and MinSalary is 8000 and MaxSalary is NULL will be entered automatically at the time of insertion if no value assigned for the specified columns.

Practice 5

Write a SQL statement to create a table employees including columns Employee_Id, FirstName, LastName, Email, PhoneNumber, Hire_Date, Job_Id, Salary, Commission, Manager_Id and Department_Id and make sure that, the Employee_Id column does not contain any duplicate value at the time of insertion, and the foreign key column DepartmentId, reference by the column DepartmentId of Departments table, can contain only those values which are exists in the Department table and another foreign key column JobId, referenced by the column JobId of jobs table, can contain only those values which are exists in the jobs table.

Practice 6

Alter statement Write a SQL statement to add a foreign key constraint named fk_job_id on JobId column of JobHistory table referencing to the primary key JobId of jobs table. Write a SQL statement to drop the existing foreign key fk_job_id from JobHistory table on JobId column which is referencing to the JobId of jobs table. Write a SQL statement to add a new column named location to the JobHistory table.

Practice 7

Understanding Database Files in SQL

Create a database named '**CompanyDB**' with the necessary physical and logical files. Define its primary file with a size of 10MB and a log file with a size of 5MB. Write the SQL query to achieve this and explain the purpose of each file defined.

Practice 8

Exploring Types of Databases

Write a query to create a table named '**DatabaseTypes**' that lists various categories of databases such as Relational, NoSQL, and Distributed databases with columns for **TypeID (Primary Key), Name, and Description**. Insert at least three records into the table that describe different types of databases.

Practice 9

Diving into Data Types

Create a table named '**OrderDetails**' to store order information with the following columns: **OrderID (integer), ProductName (varchar - 100), Quantity (integer), Price (decimal - 10,2), OrderDate (date)**. Write the SQL query for the table structure and explain why the selected data types are appropriate for each column.

Practice 10

Enforcing Entity Integrity

Create a table named '**Employees**' with the following columns: **EmployeeID (Primary Key), EmployeeName (varchar - 50), Department (varchar - 30), DateJoined (date)**. Write the SQL query for this and explain how the Primary Key constraint ensures entity integrity in this table.

Practice 11

Implementing Referential Integrity

Create two tables: '**Departments**' with columns **DeptID** (Primary Key) and **DeptName**, and '**Employees**' with columns **EmployeeID** (Primary

Key), **EmployeeName**, and **DeptID** (Foreign Key). Write queries to create these tables and describe how the foreign key ensures referential integrity, including handling improper data through database rules.

Practice 12

Utilizing Domain Integrity

Define a table named '**Products**' with the following structure: **ProductID (Primary Key), ProductName (varchar - 50), ProductPrice (decimal - 8,2)**. Ensure that the **ProductPrice** is never negative by adding a domain constraint. Write the SQL for the schema and explain the importance of domain integrity.

Practice 13

Creating, Altering, and Dropping Tables

Create a table named '**Customers**' with columns **CustomerID** (Primary Key), **CustomerName**, and **ContactNo**. Later, add a new column **Email** to the table. Finally, write the SQL query to drop the table. Ensure to explain the role and implications of each operation: creation, alteration, and deletion.

Assignment Exercise

Assignment 1

You have been hired to create a relational database to support a car sales business. You need to store information on the business's employees, inventory, and completed sales. You also need to account for the fact that each salesperson receives a different percentage of their sales in commission. What tables and columns would you create in your relational database, and how would you link the tables?

Assignment 2

Real-World Assignment: Library Management System

A public library wants to manage its database efficiently to support the borrowing system. Your task is to design and implement the required database in SQL using the following structure:

Tables to be created:

Books: Contains the following columns: **BookID (Primary Key), Title (varchar - 100), Author (varchar - 50), Genre (varchar - 30), PublishDate (date), CopiesAvailable (integer)**.

Members: Contains the following columns: **MemberID (Primary Key), FullName (varchar - 50), Address (varchar - 150), PhoneNumber (varchar - 15), MembershipDate (date)**.

BorrowingRecords: Contains the following columns: **RecordID (Primary Key), MemberID (Foreign Key referencing Members.MemberID), BookID**

(Foreign Key referencing Books.BookID), BorrowDate (date), DueDate (date), ReturnDate (date - nullable).

Tasks to Perform:

Create the mentioned tables and include constraints to ensure **entity, referential, and domain integrity**. For example, ensure the **DueDate** is always after the **BorrowDate**.

Populate the tables with example records. Include at least 6 books, 5 members, and 10 borrowing records.

Write a query to fetch the list of books that are currently available for borrowing (where no copies are issued).

Create a view that displays the details of all books borrowed by any specific member by providing their MemberID.

Write a query to update the **ReturnDate** of a specific record when a book is returned. Ensure this updates the count of **CopiesAvailable** in the **Books** table accordingly.

Create a view to list members who currently have overdue books (where **DueDate** is exceeded and **ReturnDate** is null).

Create a stored procedure to add new borrowing records for a specific member and book. Ensure the borrowing process decrements the **CopiesAvailable** count in the database for the specific book.

Business Logic:

Ensure that no borrowing record can be created if the member already has 3 overdue books.

Ensure no borrowing record for a book is created if no copies are available.

The library operates only during working hours (9 AM to 6 PM) on working days, so all borrow records should include only such dates.

Provide a record-keeping mechanism (triggers) that logs all updates made to the **BorrowingRecords** table into a separate table named **AuditLogs** with fields like **LogID, ActionType, RecordID, ActionDate**.

Supported Files

sql naming_conventions.txt

Introduction to Relational Databases

Introduction to Select Statement

Filtering Results with WHERE Statements

Utilizing Joins

Executing Sub queries and Unions

Aggregating Data

Advanced Data Aggregations

Built in Functions

Query Optimization

Modifying Data

Advanced Data Modification

Stored Procedure

Transaction

Error handling

Designing Tables

triggers