--Exercises: Database Introduction

--1. Create Database

--You now know how to create databases using the GUI of the SSMS. Now it's time to create it using SQL queries. In that task (and the several following it) you will be required to create the database from the previous exercise using only SQL queries. First, just create new database named Minions.

--2. Create Tables

--In the newly created database Minions add table Minions (Id, Name, Age). Then add a new table Towns (Id, Name). Set Id columns of both tables to be primary key as constraint.

--3. Alter Minions Table

--Change the structure of the Minions table to have a new column TownId that would be of the same type as the Id column in Towns table. Add a new constraint that makes TownId foreign key and references to Id column in Towns table.

--4. Insert Records in Both Tables

--Use only SQL queries. Insert the Id manually (don't use identity).

--5. Truncate Table Minions

--Delete all the data from the Minions table using SQL query.

--6. Drop All Tables

--Delete all tables from the Minions database using SQL query.

--7. Create Table People

--Make the Id a primary key. Populate the table with only 5 records. Submit your CREATE and INSERT statements as Run queries & check DB.

--8. Create Table Users

--Make the Id a primary key. Populate the table with exactly 5 records. Submit your CREATE and INSERT statements as Run queries & check DB.

--9. Change Primary Key

--Using SQL queries modify table Users from the previous task. First remove the current primary key and then create a new primary key that would be a combination of fields Id and Username.

--10. Add Check Constraint

--Using SQL queries modify table Users. Add check constraint to ensure that the values in the Password field are at least 5 symbols long.

--11. Set Default Value of a Field

--Using SQL queries modify table Users. Make the default value of LastLoginTime field to be the current time.

--12. Set Unique Field

--Using SQL queries modify table Users. Remove Username field from the primary key so only the field Id would be primary key. Now add unique constraint to the Username field to ensure that the values there are at least 3 symbols long.

--13. Movies Database

--Using SQL queries create Movies database with the following entities:

--Directors (Id, DirectorName, Notes)

--Genres (Id, GenreName, Notes)

--Categories (Id, CategoryName, Notes)

--Movies (Id, Title, DirectorId, CopyrightYear, Length, GenreId, CategoryId, Rating, Notes)

--Set the most appropriate data types for each column. Set a primary key to each table. Populate each table with exactly 5 records. Make sure the columns that are present in 2 tables would be of the same data type. Consider which fields are always required and which are optional. Submit your CREATE TABLE and INSERT statements as Run queries & check DB.

--14. Car Rental Database

--Using SQL queries create CarRental database with the following entities:

--Categories (Id, CategoryName, DailyRate, WeeklyRate, MonthlyRate, WeekendRate)

--Cars (Id, PlateNumber, Manufacturer, Model, CarYear, CategoryId, Doors, Picture, Condition, Available)

--Employees (Id, FirstName, LastName, Title, Notes)

--Customers (Id, DriverLicenceNumber, FullName, Address, City, ZIPCode, Notes)

--RentalOrders (Id, EmployeeId, CustomerId, CarId, TankLevel, KilometrageStart, KilometrageEnd, TotalKilometrage, StartDate, EndDate, TotalDays, RateApplied, TaxRate, OrderStatus, Notes)

--Set the most appropriate data types for each column. Set a primary key to each table. Populate each table with only 3 records. Make sure the columns that are present in 2 tables would be of the same data type. Consider which fields are always required and which are optional. Submit your CREATE TABLE and INSERT statements as Run queries & check DB.

--15. Hotel Database

--Using SQL queries create Hotel database with the following entities:

--Employees (Id, FirstName, LastName, Title, Notes)

--Customers (AccountNumber, FirstName, LastName, PhoneNumber, EmergencyName, EmergencyNumber, Notes)

--RoomStatus (RoomStatus, Notes)

--RoomTypes (RoomType, Notes)

--BedTypes (BedType, Notes)

--Rooms (RoomNumber, RoomType, BedType, Rate, RoomStatus, Notes)

--Payments (Id, EmployeeId, PaymentDate, AccountNumber, FirstDateOccupied, LastDateOccupied, TotalDays, AmountCharged, TaxRate, TaxAmount, PaymentTotal, Notes)

--Occupancies (Id, EmployeeId, DateOccupied, AccountNumber, RoomNumber, RateApplied, PhoneCharge, Notes)

--Set the most appropriate data types for each column. Set a primary key to each table. Populate each table with only 3 records. Make sure the columns that are present in 2 tables would be of the same data type. Consider which fields are always required and which are optional. Submit your CREATE TABLE and INSERT statements as Run queries & check DB.

--16. Create SoftUni Database

--Now create bigger database called SoftUni. You will use the same database in the future tasks. It should hold information about

--Towns (Id, Name)

--Addresses (Id, AddressText, TownId)

--Departments (Id, Name)

--Employees (Id, FirstName, MiddleName, LastName, JobTitle, DepartmentId, HireDate, Salary, AddressId)

--The Id columns are auto incremented, starting from 1 and increased by 1 (1, 2, 3, 4…). Make sure you use appropriate data types for each column. Add primary and foreign keys as constraints for each table. Use only SQL queries. Consider which fields are always required and which are optional.

--17. Backup Database

--Backup the database SoftUni from the previous task into a file named "softuni-backup.bak". Delete your database from SQL Server Management Studio. Then restore the database from the created backup.

--Hint: https://support.microsoft.com/en-gb/help/2019698/how-to-schedule-and-automate-backups-of-sql-server-databases-in-sql-se

--18. Basic Insert

--Use the SoftUni database and insert some data using SQL queries.

--Towns: Sofia, Plovdiv, Varna, Burgas

--Departments: Engineering, Sales, Marketing, Software Development, Quality Assurance

--• Employees:

--Name Job Title Department Hire Date Salary

--Ivan Ivanov Ivanov .NET Developer Software Development 01/02/2013 3500.00

--Petar Petrov Petrov Senior Engineer Engineering 02/03/2004 4000.00

--Maria Petrova Ivanova Intern Quality Assurance 28/08/2016 525.25

--Georgi Teziev Ivanov CEO Sales 09/12/2007 3000.00

--Peter Pan Pan Intern Marketing 28/08/2016 599.88

--19. Basic Select All Fields

--Use the SoftUni database and first select all records from the Towns, then from Departments and finally from Employees table. Use SQL queries and submit them to Judge at once. Submit your query statements as Prepare DB & Run queries.

--20. Basic Select All Fields and Order Them

--Modify the queries from the previous problem by sorting:

--• Towns - alphabetically by name

--• Departments - alphabetically by name

--• Employees - descending by salary

--Submit your query statements as Prepare DB & Run queries.

--21. Basic Select Some Fields

--Modify the queries from the previous problem to show only some of the columns. For table:

--• Towns – Name

--• Departments – Name

--• Employees – FirstName, LastName, JobTitle, Salary

--Keep the ordering from the previous problem. Submit your query statements as Prepare DB & Run queries.

--22. Increase Employees Salary

--Use SoftUni database and increase the salary of all employees by 10%. Then show only Salary column for all the records in the Employees table. Submit your query statements as Prepare DB & Run queries.

--23. Decrease Tax Rate

--Use Hotel database and decrease tax rate by 3% to all payments. Then select only TaxRate column from the Payments table. Submit your query statements as Prepare DB & Run queries.

--24. Delete All Records

--Use Hotel database and delete all records from the Occupancies table. Use SQL query. Submit your query statements as Run skeleton, run queries & check DB.