# Упражнение: Моделиране на бази данни

## База данни People

TODO: да се добави предисловие на поредицата задачи върху първата база данни, примерно нещо такова:

В настоящата задача ще създадем база данни “People” с няколко таблици в нея, които ще свържем с foreign key constraint, ще добавяме записи и ще дефинираме constraints.

### Създайте база данни People

**Създайте база данни** симе“**People**”.

Внимание: в предното упражнение базата данни се казва “Minions”. За да нама конфликт с имената, съм дал име “People” на базата данни в това упражнение.

TODO: Това е първата задача, трябва да е разписана по-подробно. Добави screenshots.

Ако се прави с SQL заявка (не чрез цъкане с мишката), да се даде screenshot на заявката + резултата в SSMS.

### Създайте таблици People и Towns

В новосъздадената база данни **People** добавете таблица **People (Id, Name, Age)**. След това добавете нова таблица **Towns (Id, Name)**. Задайте **Id** колоните на двете таблици да бъдат **първичен ключ** като **ограничение**.

TODO: Това е втората задача, тряба да е разписана по-подробно. Добави screenshots.

Ако се прави с SQL заявка (не чрез цъкане с мишката), да се даде screenshot на заявката + резултата в SSMS.

Сменил съм името: Minions 🡪 People

### Свържете таблиците People и Towns

Променете структурата на таблицата People, за да имате **нова колона TownId**, която ще бъде от същия тип като колоната **Id** на **таблицата Towns**. Добавете **ново ограничение**, което прави **TownId външен ключ** и реферира към колоната **Id** на таблицата **Towns**.

TODO: shreenshots

Ако се прави с SQL заявка (не чрез цъкане с мишката), да се даде screenshot на заявката + резултата в SSMS.

### Вмъкнете записи в двете таблици

**Попълнете и двете таблици** с примерни записи, дадени в таблицата по-долу.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **People** | | | |  | **Towns** | |
| **Id** | **Name** | **Age** | **TownId** |  | **Id** | **Name** |
| 1 | Kevin | 22 | 1 |  | 1 | Sofia |
| 2 | Bob | 15 | 3 |  | 2 | Plovdiv |
| 3 | Steward | NULL | 2 |  | 3 | Varna |

**Използвайте само SQL заявки**. Въведете идентификатора ръчно (не използвайте identity).

TODO: shreenshots

Ако се прави с SQL заявка (не чрез цъкане с мишката), да се даде screenshot на заявката + резултата в SSMS.

### Създайте таблица Users

Използвайки **SQL заявка**, създайте таблица **Users** с колони:

* **Id** – уникален номер за всеки потребител. Няма да има **повече от 263-1 потребители.** (Автоматично нараства)
* **Username** – уникалният идентификатор на потребителя ще бъде **не повече от 30 знака** (**не Unicode**) (Задължително)
* **Password** – паролата няма да бъде **по-дълга от 26 символа** (**не Unicode**)(Задължително)
* **ProfilePicture** – изображение с размер **до 900 KB.**
* **LastLoginTime**
* **IsDeleted** – показва дали потребителят е изтрил своя профил. Възможните състояния са **true** или **false**.

Направете колоната **Id** първичен ключ. Попълнете таблицата с точно **5 записа**.

TODO: shreenshots, които служат за подсказка как да се реши задачата

### Сменете първичния ключ

Използвайки **SQL заявки**, модифицирайте таблицата **Users** от предишната задача. Първо **премахнете текущия първичен ключ**, след което създайте **нов първичен ключ**, който ще бъде **комбинацията** от полета **Id** и **Username**.

TODO: shreenshots, които служат за подсказка как да се реши задачата

### Добавете ограничение за проверка

Използвайки **SQL заявки**, модифицирайте таблицата **Users**. Добавете **ограничение за проверка**, за да се уверите, че стойностите в полето за парола са **с дължина поне 5 символа**.

TODO: shreenshots, които служат за подсказка как да се реши задачата

### Сложете подразбираща се стойност

Използвайки **SQL заявки**, модифицирайте таблицата **Users**. Направете **подразбиращата се стойност** (default value) за полето **LastLoginTime** да бъде **текущото време** (current data + time).

TODO: shreenshots, които служат за подсказка как да се реши задачата

### Задайте уникално поле

Използвайки **SQL заявки**, модифицирайте таблицата **Users**. Премахнете полето **Username** от първичния ключ, така че само полето **Id** да бъде първичен ключ. Сега **добавете уникално ограничение** към полето **Username**, за да сте сигурни, че стойностите там са **дълги поне 3 символа**.

TODO: shreenshots, които служат за подсказка как да се реши задачата

### Свържете таблиците People и Users

Свържете таблиците People и Users, така че всеки човек да може да има незадължителен потребител (username), с който е свързан. Дефинирайте подходящи ограничения, така че връзката **People – Users** да е с множественост **1 към 0 или 1**.База данни Movies

С помощта на **SQL заявки** създайте база данни **Movies** със следните обекти:

* **Directors** (Id, DirectorName, Notes)
* **Genres** (Id, GenreName, Notes)
* **Categories** (Id, CategoryName, Notes)
* **Movies** (Id, Title, DirectorId, CopyrightYear, Length, GenreId, CategoryId, Rating, Notes)

Задайте **най-подходящите типове данни** за всяка колона. **Задайте първичен ключ** за всяка таблица. Попълнете всяка таблица с точно **5 записа**. Уверете се, че колоните, които присъстват в 2 таблици, са от **един и същи тип данни**. Помислете кои полета винаги са **задължителни** и кои са **незадължителни**. Свържете таблицата Movies с таблиците Directors, Genres и Categories чрез **foreign key constraint**.

## База данни CarRental

С помощта на **SQL заявки** създайте база данни **CarRental** със следните обекти:

* **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)

**Свържете таблиците**, които изглежда, че трябва да се реферират една друга. Вмъкнете примерни данни за всяка от таблиците.

Проверете **дали между таблиците връзките работят правилно** чрез опит за вкарване на невалидни референции към несъществуващ запис в свързана таблица (трябва да получите грешка).

## База данни Hotel

С помощта на **SQL заявки** създайте база данни **Hotel** със следните таблици в нея:

* **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)

**Свържете таблиците**, които изглежда, че трябва да се реферират една друга.

Забележка: за тази задача може да си помогнете с **ChatGPT** или подобен генеративен AI чатбот.

Проверете **дали връзките между таблиците работят правилно** чрез опит за вкарване на невалидни референции към несъществуващ запис в свързана таблица (трябва да получите грешка).