# Упражнение: SQL INSERT, UPDATE, DELETE

Тествайте решенията си в **Judge системата**: [https://judge.softuni.org/Contests/Practice/Index/4605#0](https://judge.softuni.org/Contests/Practice/Index/4605" \l "0)

## Създайте база данни Minions

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

## A screenshot of a computer Description automatically generated

## Създайте таблици Minions и Towns

В новосъздадената база данни Minions добавете таблица **Minions (Id, Name, Age)**. Задайте **Id** да бъде **първичен ключ**.

A screenshot of a computer

Description automatically generated

След това добавете нова таблица **Towns (Id, Name)**. Задайте **Id** на колоната на втората таблица да бъде **първичен ключ**.

A blue and white box with black text

Description automatically generated

## Свържете двете таблици

Променете **структурата** на таблицата **Minions**, за да имате **нова колона TownId**, която ще бъде от същия тип като колоната **Id** на **таблицата Towns**.

Добавете **ново ограничение** (foreign key constraint), което прави **TownId външен ключ** и реферира към колоната **Id** на таблицата **Towns**.

A screenshot of a computer

Description automatically generated

## Вмъкнете записи в двете таблици

Напишете **поредица** от **SQL** **заявки**, която **вмъква** примерните **записи**, дадени в **таблицата** **по-долу**. След това изберете всички записи от таблицата **Minions**, след което всички записи от таблицата **Towns**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Minions** | | | |  | **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 |

Поради външния ключ ще се наложи **първо да създадете градовете**, a след това **минионите**, защото те **реферират** вече **съществуващи** **градове** в колоната TownId.

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

Изберете **всички записи** от таблицата **Minions**, след което направете същото с таблицата **Towns**.

## Променете името и възрастта на миньон

Напишете **поредица** от **SQL** **заявки**, която променя **името** и **възрастта** на **миньона** с **Id** равно на **1**. Новото име трябва да е "**Ivan**", а годините му да са **15**. След това да се изберат **всички записи** от таблицата **Minions**.

### Пример

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Age** | **TownId** |
| 1 | Ivan | 15 | 1 |
| 2 | Bob | 15 | 3 |
| 3 | Steward | NULL | 2 |

## Увеличете годините на миньон

Напишете **поредица** от **SQL** **заявки**, която увеличава с **1** годините на всеки **миньон**, чието **Id** е **различно** от **3** и избира **всчики записи** от таблицата **Minions**.

### Пример

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Age** | **TownId** |
| 1 | Ivan | 16 | 1 |
| 2 | Bob | 16 | 3 |
| 3 | Steward | NULL | 2 |

## Изтрийте миньон с ID

Напишете **поредица** от **SQL** **заявки**, която **премахва миньона**, чието **Id** е равно на **2** и избира **всчики записи** от таблицата **Minions**.

### Пример

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Age** | **TownId** |
| 1 | Ivan | 15 | 1 |
| 3 | Steward | NULL | 2 |

## Изтрийте данните от таблицата Minions

Напишете **поредица** от **SQL** **заявки**, която **изтрива всички данни** от таблицата **Minions** и избира **бройката** на **записите** в таблицата **Minions**.

#### Подсказка

Изпълнете **следната заявка**, за да извлечете **бройката** на **записите** в таблицата **Minions**:

|  |
| --- |
| **SELECT COUNT(\*) AS [Count] FROM Minions** |

### Пример

|  |
| --- |
| **Count** |
| 0 |

## Изтрийте всички таблици

**Изтрийте всички таблици** от базата данни на **Minions** с помощта на **SQL заявка**. **Добавете** следния код **след** **заявката** **си** и проверете дали **след изпълнението му**, ще се изведат **очакваните съобщения**, за **изтрити таблици**. Качете **всичко** в **Judge**.

|  |
| --- |
| IF OBJECT\_ID('Minions') IS NULL  PRINT 'Table Minions deleted.';  ELSE  PRINT 'Table Minions exists.';  IF OBJECT\_ID('Towns') IS NULL  PRINT 'Table Towns deleted.';  ELSE  PRINT 'Table Towns exists.'; |

## База данни Hotel

Създайте база данни **Hotel** със следните **таблици**:

* **Employees** (Id, FirstName, LastName, Title, Notes)
* **Customers** (Id, FirstName, LastName, PhoneNumber, EmergencyName, EmergencyNumber, Notes)
* **RoomStatus** (Id, RoomStatus, Notes)
* **RoomTypes** (Id, RoomType, Notes)
* **BedTypes** (Id, 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)

Задайте **най-подходящите типове данни** за всяка колона. **Задайте първичен ключ** за всяка таблица. 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.

## Попълнете таблиците с примерни данни

Попълнете **всяка една** от таблиците с **5** реда **примерни данни**, с които да работите. Може да си помогнете с **ChatGPT** или подобен генеративен **чатбот**, например ето така:

A screenshot of a computer

Description automatically generated

Примерно запитване към **ChatGPT**: <https://chat.openai.com/share/e7678a91-e807-4f61-8b9c-ddfb4f215996>

|  |
| --- |
| **SELECT COUNT(\*) FROM Employees**  **SELECT COUNT(\*) FROM Customers**  **SELECT COUNT(\*) FROM RoomStatus**  **SELECT COUNT(\*) FROM BedTypes**  **SELECT COUNT(\*) FROM Rooms**  **SELECT COUNT(\*) FROM Payments**  **SELECT COUNT(\*) FROM Occupancies** |

## Намалете данъчната ставка

Използвайки базата данни **Hotel**, напишете **поредица** от SQL заявки, която да селектира **колоната** **TaxRate** от таблицата **Payments**. След това **намалете данъчната ставка с 3%** за всички **плащания** чрез **SQL UPDATE** команда. Накрая селектирайте отново **колоната** **TaxRate** от таблицата **Payments**. Качете **трите** заявки в **Judge**.

## Изтрийте всички записи

Използвайте базата данни **Hotel** и. Използвайте SQL заявка. Проверете след това дали **данните** са **изтрити**.

Използвайки базата данни **Hotel**, напишете **поредица** от SQL заявки, която избира **бройката** на **всички** **записи** от таблицата **Occupancies**. След това **изтрийте всички записи** от таблицата **Occupancies**. Накрая **отново** вземете **бройката** на **записите** от таблицата **Occupancies**. Качете **трите** заявки в **Judge**.

#### Подсказка

За да намерите **бройката** на **записите** в таблицата **Occupancies** използвайте следната **заявка**:

|  |
| --- |
| **SELECT COUNT(\*) FROM Occupancies** |