



## Упражнения: Основни MySQL команди за работа с БД. Дефиниране на данни и типове данни

### Задача 1. Създаване на база данни

Вие сега знаете как да се създадете база от данни, използвайки GUI на HeidiSQL. Сега е време да се създаде с помощта на SQL заявки. В тази задача (и няколко след това) ще се изисква да се създаде базата данни от предишното упражнение само със SQL заявки. Първо създайте нова база данни с име Minions.

### Задача 2. Създаване на таблици

В новосъздадената база данни Minions добавете таблица minions (id, name, age). След това добавете нова таблица towns (id, name). Сложете колона ИД на двете таблици да бъде първичен ключ като ограничение. Изпратете двете заявки, една след друга, разделени със „;“ до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 3. Променяне на таблицата Minions

Променете структурата на таблицата Minions да има нова колона town\_id, която да е от същия тип като id колоната на таблицата towns. Добавете ново ограничение, което прави town\_id чужд ключ и е препратка към id колоната на таблицата towns. Изпратете заявката си създаване на таблица до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 3. Направете записи в двете таблици

Попълнете двете таблици с примерни записи, посочени в таблицата по-долу.

minions			
id	name	age	town_id
1	Kevin	22	1
2	Bob	15	3
3	Steward	NULL	2

towns	
id	name
1	Sofia
2	Plovdiv
3	Varna

Използвайте само INSERT SQL заявки. Изпратете заявката до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 4. Изтрийте всички данни от таблица Minions

Изтрийте всички данни от таблицата minions, с помощта на SQL заявка. Изпратете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 5. Премахнете всички таблици

Изтрийте всички таблици от базата данни minions, с помощта на SQL заявка. Изпратете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 6. Създайте таблицата People

С помощта на SQL заявка, създайте таблица users с колони:

- **id** – unique number for every person there will be no more than people. (Auto incremented)



- **name** – full name of the person will be **no more than 200 Unicode characters**. (Not null)
- **picture** – image with **size up to 2 MB**. (Allow nulls)
- **height** – In meters. Real number precise up to **2 digits** after floating point. (Allow nulls)
- **weight** – In kilograms. Real number precise up to **2 digits** after floating point. (Allow nulls)
- **gender** – Possible states are **m** or **f**. (Not null)
- **birthdate** – (Not null)
- **biography** – detailed biography of the person it can contain **max allowed Unicode characters**. (Allow nulls)
- **id** – уникален номер за всеки човек там ще бъде не повече от  $2^{31}-1$  души. (Автоувеличава)
- **name** – пълно име на лицето ще бъде не повече от 200 Unicode символа. (Не null)
- **picture** – изображение с размер до 2 MB. (Разрешени са празни стойности)
- **height** – в метри. Реално число с точност до 2 цифри след десетична запетая. (Разрешени са празни стойности)
- **weight** – в килограми. Реално число с точност до 2 цифри след десетичната запетая. (Разрешени са празни стойности)
- **gender** – възможни състояния са **m** или **f**. (не null)
- **birthdate** – (не null)
- **biography** – подробна биография на лицето, може да съдържа максимално позволен брой знаци в Unicode. (Разрешени са празни стойности)

Направете ИД първичен ключ. Попълнете таблицата с 5 записа. Изпратете вашите CREATE и INSERT заявки до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

## Задача 7. Направете таблицата потребители Users

С помощта на SQL заявка създайте таблица users с колони:

- **id** – unique number for every user. There will be **no more than users**. (Auto incremented)
- **username** – unique identifier of the user will be **no more than 30 characters (non Unicode)**. (Required)
- **password** – password will be **no longer than 26 characters (non Unicode)**. (Required)
- **profile\_picture** – image with **size up to 900 KB**.
- **last\_login\_time**
- **is\_deleted** – shows if the user deleted his/her profile. Possible states are **true** or **false**.
- **id** – уникално число за всеки потребител. Ще има не повече от  $2^{63}-1$  потребители. (автоувеличаване)
- **name** – уникален идентификатор на потребителя, ще бъде не повече от 30 знака (не Unicode). (Задължително)
- **password** – парола ще бъде не по-дълга от 26 символа (не Unicode). (Задължително)
- **profile\_picture** – изображение с размер до 900 KB.
- **last\_login\_time**



- **is\_deleted** - показва, ако потребителят е изтрил профила си. Възможни състояния са **true** или **false**.

Направете ИД първичен ключ. Попълнете таблицата с 5 записа. Изправете CREATE и INSERT заявки. Изправете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 8. Сменете първичния ключ

С помощта на SQL заявки променете таблицата users от предишната задача. Първо премахнете текущия първичен ключ след това създайте нов първичен ключ, който ще бъде комбинация от полетата id и username. Първоначалното име на първичния ключ на ИД е pk\_users. . Изправете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 9. Установете (настройте) стойност по подразбиране на поле

С помощта на SQL заявки променете таблицата users. Направете стойността по подразбиране на полето last\_login\_time да бъде текущото време. Изправете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 10. Настройте уникално поле

С помощта на SQL заявки променете таблицата users. Премахнете полето username от първичния ключ, така че само полето id да бъде първичен ключ. Сега добавете уникално ограничение в полето username . Първоначалното име на първичен ключ (ИД, username) е pk\_users. . Изправете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 11. База от данни Movies

С помощта на SQL заявки създайте база данни за филми със следните записи:

- **directors** (id, director\_name, notes)
- **genres** (id, genre\_name, notes)
- **categories** (id, category\_name, notes)
- **movies** (id, title, director\_id, copyright\_year, length, genre\_id, category\_id, rating, notes)

Задайте най-подходящи типове данни за всяка колона. Задайте първичен ключ за всяка таблица. Попълнете всяка таблица с 5 записа. Уверете се, че колоните, които се намират в 2 таблици ще бъдат от един и същ тип данни. Помислете кои полета са винаги задължителни и които не са задължителни. . Изправете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (Run queries & check DB).

### Задача 12. БД – Коли под наем

С помощта на SQL заявки създаване база от данни car\_rental със следните таблици:

- **categories** (id, category, daily\_rate, weekly\_rate, monthly\_rate, weekend\_rate)
- **cars** (id, plate\_number, make, model, car\_year, category\_id, doors, picture, car\_condition, available)



- **employees** (id, first\_name, last\_name, title, notes)
- **customers** (id, driver\_licence\_number, full\_name, address, city, zip\_code, notes)
- **rental\_orders** (id, employee\_id, customer\_id, car\_id, car\_condition, tank\_level, kilometrage\_start, kilometrage\_end, total\_kilometrage, start\_date, end\_date, total\_days, rate\_applied, tax\_rate, order\_status, notes)

Задайте най-подходящи типове данни за всяка колона. Задайте първичен ключ за всяка таблица. Попълнете всяка таблица с 3 записа. Уверете се, че колоните, които се намират в 2 таблици ще бъдат от един и същ тип данни. Помислете кои полета са винаги задължителни и които не са задължителни. Изпратете заявката си до системата за проверка като стартирате заявките и проверете базата от данни (Run queries & check DB).

### Задача 13. БД – Хотел

С помощта на SQL заявките създайте база от данни hotel със следните таблици:

- **employees** (id, first\_name, last\_name, title, notes)
- **customers** (account\_number, first\_name, last\_name, phone\_number, emergency\_name, emergency\_number, notes)
- **room\_status** (room\_status, notes)
- **room\_types** (room\_type, notes)
- **bed\_types** (bed\_type, notes)
- **rooms** (room\_number, room\_type, bed\_type, rate, room\_status, notes)
- **payments** (id, employee\_id, payment\_date, account\_number, first\_date\_occupied, last\_date\_occupied, total\_days, amount\_charged, tax\_rate, tax\_amount, payment\_total, notes)
- **occupancies** (id, employee\_id, date\_occupied, account\_number, room\_number, rate\_applied, phone\_charge, notes)

Задайте най-подходящи типове данни за всяка колона. Задайте първичен ключ за всяка таблица. Попълнете всяка таблица с 3 записа. Уверете се, че колоните, които се намират в 2 таблици ще бъдат от един и същ тип данни. Помислете кои полета са винаги задължителни и които не са задължителни. Изпратете заявката си до системата за проверка като стартирате заявките CREATE TABLE и INSERT и проверете базата от данни (Run queries & check DB).

### Задача 14. Създайте БД SoftUni

Сега създайте по-голяма база данни, наречена soft\_uni. Ще използвате база данни в бъдещите задачи. Тя трябва да съдържа информация за:

- **towns** (id, name)
- **addresses** (id, address\_text, town\_id)
- **departments** (id, name)
- **employees** (id, first\_name, middle\_name, last\_name, job\_title, department\_id, hire\_date, salary, address\_id)

id на колони са автоматично увеличаващи се, започвайки от 1, увеличаващи се с 1 (1, 2, 3, 4...). Уверете се, че използвате подходящи типове данни за всяка колона. Добавете първичните и чужди ключове като ограничения за всяка таблица. Използвайте само SQL заявки. Помислете кои полета са винаги задължителни и които не са задължителни. Изпратете заявката си до системата за проверка като стартирате заявките CREATE TABLE statements и проверете базата от данни (Run queries & check DB).



## Задача 15. Възстановяване на БД

С помощта на `mysqldump` команда от командния ред MySQL направете резервно копие на базата данни `soft_uni` от предишните задачи, във файл с име `"softuni-backup.sql"`. Пуснете вашата база данни от Heidi или MySQL Workbench. След това възстановете базата данни от създадения архивен файл, и използвайте командния ред на mysql.