

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

Problem 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**.

Problem 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**).

Problem 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**).

Problem 4. Направете записи в двете таблици

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

minions				towns	
id	name	age	town_id	id	name
1	Kevin	22	1	1	Sofia
2	Bob	15	3	2	Plovdiv
3	Steward	NULL	2	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**).

Problem 5. 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**).

Problem 6. 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**).

Problem 7. 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 празни стойности)

Направете ИД първичен ключ. Попълните 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).

Problem 8. 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**.

Make **id** primary key. Populate the table with **5 records**. Submit your **CREATE** and **INSERT** statements. Submit your **CREATE** and **INSERT** statements as **Run queries & check DB**.

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

Problem 9. 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**. . Изпратете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (**Run queries & check DB**).

Problem 10. 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. Изпратете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (**Run queries & check DB**).

Problem 11. 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**. . Изпратете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (**Run queries & check DB**).

Problem 12. 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.** . Изпратете заявката си до системата за проверка като стартирате заявките и проверите базата от данни (**Run queries & check DB**).

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. Задай 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 проверите базата от данни (**Run queries & check DB**).

БД – Хотел

С помощта на SQL заявките създаде 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. Задай 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 проверите базата от данни (**Run queries & check DB**).

Problem 13. 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 проверите базата от данни (**Run queries & check DB**).

Problem 14. 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**.

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "Обучение за ИТ кариера" на МОН за подготовка по професия "Приложен програмист".



- Курсът е базиран на учебно съдържание и методика, предоставени от фондация "Софтуерен университет" и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

