

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

Звіти до комп'ютерних практикумів з кредитного модуля “Бази даних”

Прийняв

доцент кафедри АСОІУ

Ліщук К.І.

29.12.2020р

Виконав

Студент групи ІТ-02

Макаров І.С.

Київ – 2020

Комп'ютерний практикум № 2

Створення бази даних

Мета:

- Створення бази даних шляхом визначення схеми БД та заповнення її тестовими даними
- Навчитися проектувати бази даних, вводити і редагувати структуру таблиць та дані в таблицях
- Вивчити команди SQL для роботи з таблицями (створення, зміни та видалення таблиць)
- Вивчити використовувані в SQL засоби для підтримки цілісності даних та їх практичне застосування

Постановка задачі комп'ютерного практикуму № 2:

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

1. Створити схему БД згідно з розробленою в роботі №1 ER-моделлю
2. Розробити SQL-скрипти для:
 - створення таблиць в БД засобами мови SQL (для 5-6 таблиць). Передбачити необхідність наявності обмежень для підтримки цілісності даних;
 - зміни структури таблиць та даних в таблицях засобами мови SQL (до 20 різних за суттю запитів для 5-6 таблиць);
 - видалення окремих елементів або самих таблиць засобами мови SQL (до 10 різних за суттю команд); встановлення зв'язків між таблицями засобами мови SQL.
3. Згенерувати схему даних засобами СУБД
4. Імпортувати дані в створену БД з використанням засобів СУБД

РЕЗУЛЬТАТЫ

Такс, работу вроде сделал, и вот краткое пояснение. Я написал 6 скриптов, ссылка на которые вот: <https://github.com/kinfi4/Ministry-of-Defence-database-scripts>

Скрипты создания, заполнения таблиц там нечего описывать, вот с ограничениями там более интересно. Напомню, что в моем варианте есть два ограничения: поле `graduation_year`, в таблице характеристик может иметь только человек со званием не ниже лейтенанта. Второе ограничение похожее, оно гласит что командиром армии, дивизии, отделения или роты так же может быть только человек со званием лейтенант или выше. Взводом и отрядом может кто угодно управлять.

Итак, как же я создал эти ограничения. По неопытности и наивности я вообще думал что можно будет как то сделать это с помощью старых добрых `check constraints`. Но как оказалось в них не может быть подзапросов, я нашел даже где то большую статью где рассказывалось, почему современные СУБД имеют такое ограничение, но врожденная лень, а так же дедлайны на работе избавили меня от желания разбираться в первопричинах сего ограничения, и я просто принял тот факт что придется думать что то еще.

Ну довольно быстро в мою светлую голову пришла идея что для такой задачи может подойти другой механизм присущий всем системам управления базами данных - триггеры. Триггер это грубо говоря `sql` код, что может быть выполнен, до, после или даже вместо операции над нашей таблицей.

Итак, в файлах [*add_constraints__characteristics.sql*](#) и [*add_constraints__characteristics.sql*](#) вы можете найти код моих триггеров, что запускаются `FOR EACH ROW`, на операции `UPDATE OR INSERT`, в нужных таблицы, и в случае, если данная операция не валидна, триггер кинет исключение с текстом его причины, вызвав функцию `f_raise`. `f_raise` объявлена в [*add_constraints__characteristics.sql*](#), если вам интересно.

Такс, ну и остались запросы на изменение таблиц, и изменение данных в этим самых таблицах. Хех, ну вообще довольно занудное задание, я туда было вписал пару запросов что действительно мнегодились, когда я при создании таблицы что то не так сделал. Ну там, забыл `foreign key constraint` написать, или с типом данных прогадал чутка. Из остальных... Хм, ну там есть пара запрос на `UPDATE` что хоть что то стоят, а так, в основном там фигня в 3 строчки. А вот задалииии бы вы сделать не 20 запросов, а скажем 4 но интересных, можно было бы что то придумать.

Ну и с запросы на `DELETE`, `DROP`... Я решил создать просто отдельные две таблички, и на них показать запросы, чтоб свое не портить.

Такие дела....

Скрины:

Data Output		Explain	Messages	Notifications
	name character varying (40)	rank_name character varying (30)		
1	Нерей	Рядовой		

```
ERROR: Cant perfort this operation, because Hereй cant have graducation year
CONTEXT: PL/pgSQL function f_raise(text) line 3 at RAISE
SQL statement "SELECT f_raise(errorMessage)"
PL/pgSQL function checkmilitarymancanhavegraducationyear() line 14 at SQL statement
SQL state: P0001
```

```
ERROR: Cant perfort this operation, because Hereй cant command this unit
CONTEXT: PL/pgSQL function f_raise(text) line 3 at RAISE
SQL statement "SELECT f_raise(errorMessage)"
PL/pgSQL function checkmilitarymancanbecommander() line 14 at SQL statement
SQL state: P0001
```

Такс, ну и схема, тут прям не оч видно что к чему, так что думаю прсто на защите вам покажу ее лучше.

