**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп'ютерних наук та інформаційних технологій**

**Кафедра систем штучного інтелекту**

****

**Звіт до лабораторної роботи №2**

з дисципліни

“Організація Баз Даних та знань”

**Виконала**:

ст. гр. КН-210

Заремба Вікторія

**Викладач:**

Мельникова Н.І.

Львів – 2019

Лабораторна робота №3

*з курсу “ОБДЗ”*

*на тему:*

***“Модифікація структури таблиць бази даних засобами SQL”***

**Мета роботи:** Розробити SQL запити для зміни структури таблиць бази даних (включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень).

**Короткі теоретичні відомості.**

Для зміни структури таблиць використовується наступна директива мови SQL.

**Синтаксис:**

ALTER [IGNORE] TABLE ім’я\_таблиці

опис\_зміни [, опис\_зміни ...]

**опис\_зміни:**

[додаткові\_параметри ...]

| ADD [COLUMN] назва\_поля опис\_поля

[FIRST | AFTER назва\_поля ]

| ADD [COLUMN] (назва\_поля опис\_поля,...)

| ADD {INDEX|KEY} [ім’я\_обмеження] (назва\_поля,...)[тип\_обмеження]

| ADD [CONSTRAINT [ім’я\_обмеження]] PRIMARY KEY (назва\_поля,...)

[тип\_обмеження]

| ADD [CONSTRAINT [ім’я\_обмеження]]

UNIQUE [INDEX|KEY] [ім’я\_обмеження](назва\_поля,...)[тип\_обмеження]

| ADD [FULLTEXT|SPATIAL] [INDEX|KEY] [ім’я\_обмеження]

(назва\_поля,...) [тип\_обмеження]

| ADD [CONSTRAINT [ім’я\_обмеження]]

FOREIGN KEY [ім’я\_обмеження] (назва\_поля,...) опис\_зв’язку

| ALTER [COLUMN] назва\_поля {SET DEFAULT значення | DROP DEFAULT}

| CHANGE [COLUMN] стара\_назва\_поля нова\_назва\_поля опис\_поля

[FIRST|AFTER назва\_поля]

| MODIFY [COLUMN] назва\_поля опис\_поля

[FIRST | AFTER назва\_поля]

| DROP [COLUMN] назва\_поля

| DROP PRIMARY KEY

| DROP {INDEX|KEY} ім’я\_обмеження

| DROP FOREIGN KEY ім’я\_обмеження

| RENAME [TO] нова\_назва\_таблиці

| ORDER BY назва\_поля [,назва\_поля] ...

| CONVERT TO CHARACTER SET кодування [COLLATE набір\_правил]

| [DEFAULT] CHARACTER SET [=] кодування [COLLATE [=] набір\_правил]

**додаткові\_параметри:**

{ENGINE|TYPE} [=] тип\_таблиці

| AUTO\_INCREMENT [=] початкове\_значення

| AVG\_ROW\_LENGTH [=] значення

| [DEFAULT] CHARACTER SET [=] кодування

| CHECKSUM [=] {0 | 1}

| [DEFAULT] COLLATE [=] набір\_правил

| COMMENT [=] 'коментар до таблиці'

| DELAY\_KEY\_WRITE [=] {0 | 1}

| MAX\_ROWS [=] значення

| MIN\_ROWS [=] значення

| ROW\_FORMAT {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

**Аргументи:**

тип\_обмеження

Вказує тип індексу для поля: USING {BTREE | HASH | RTREE}.

IGNORE

Ігнорувати помилки дублювання значень в ключових полях, тобто залишати в таблиці

перше значення, і не вставляти дублікат. За замовчуванням, при такій помилці вся дія

над таблицею відміняється.

ALTER назва\_поля

Використовується для зміни значення за замовчуванням для поля.

MODIFY назва\_поля

Змінити тип поля.

CHANGE назва\_поля

Змінити назву і тип поля.

FIRST | AFTER

Змінити розташування поля в таблиці. За замовчуванням, нове поле додається в кінець

таблиці. FIRST додасть його на початок; AFTER – одразу після вказаного поля.

DROP назва\_поля

Видалення поля, обмеження чи індексу.

ORDER BY

Відсортувати рядки в таблиці за зростанням значень у вказаному полі (полях).

AUTO\_INCREMENT

При зміні початкового значення для приросту поля-лічильника, слід вказати значення

більше за ті, які вже містяться в цьому полі.

CONVERT TO

Задає кодування, в яке потрібно перевести таблицю.

RENAME TO

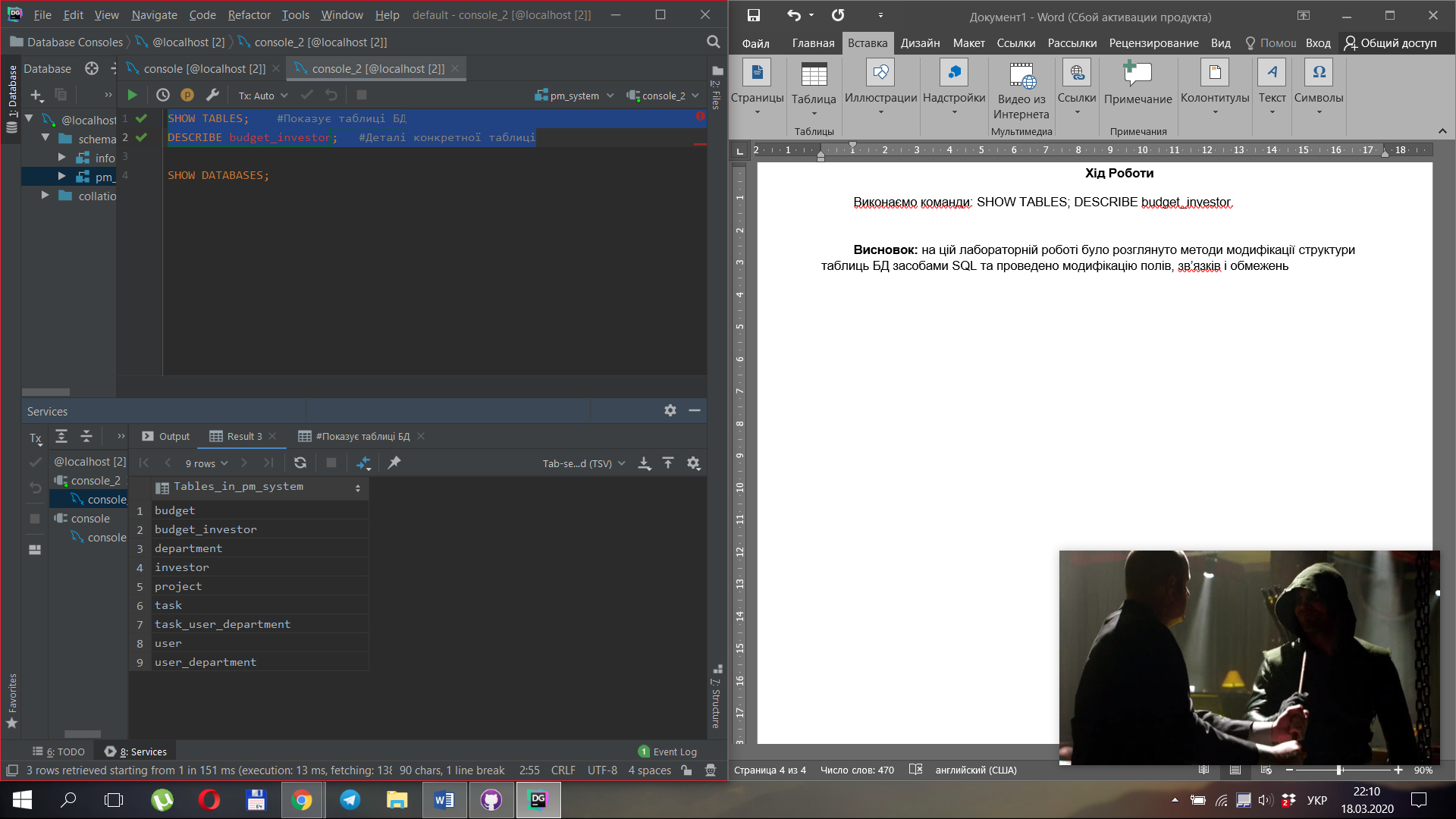
Змінити назву таблиці на вказану. Для зміни назви таблиці, або для переносу таблиці в

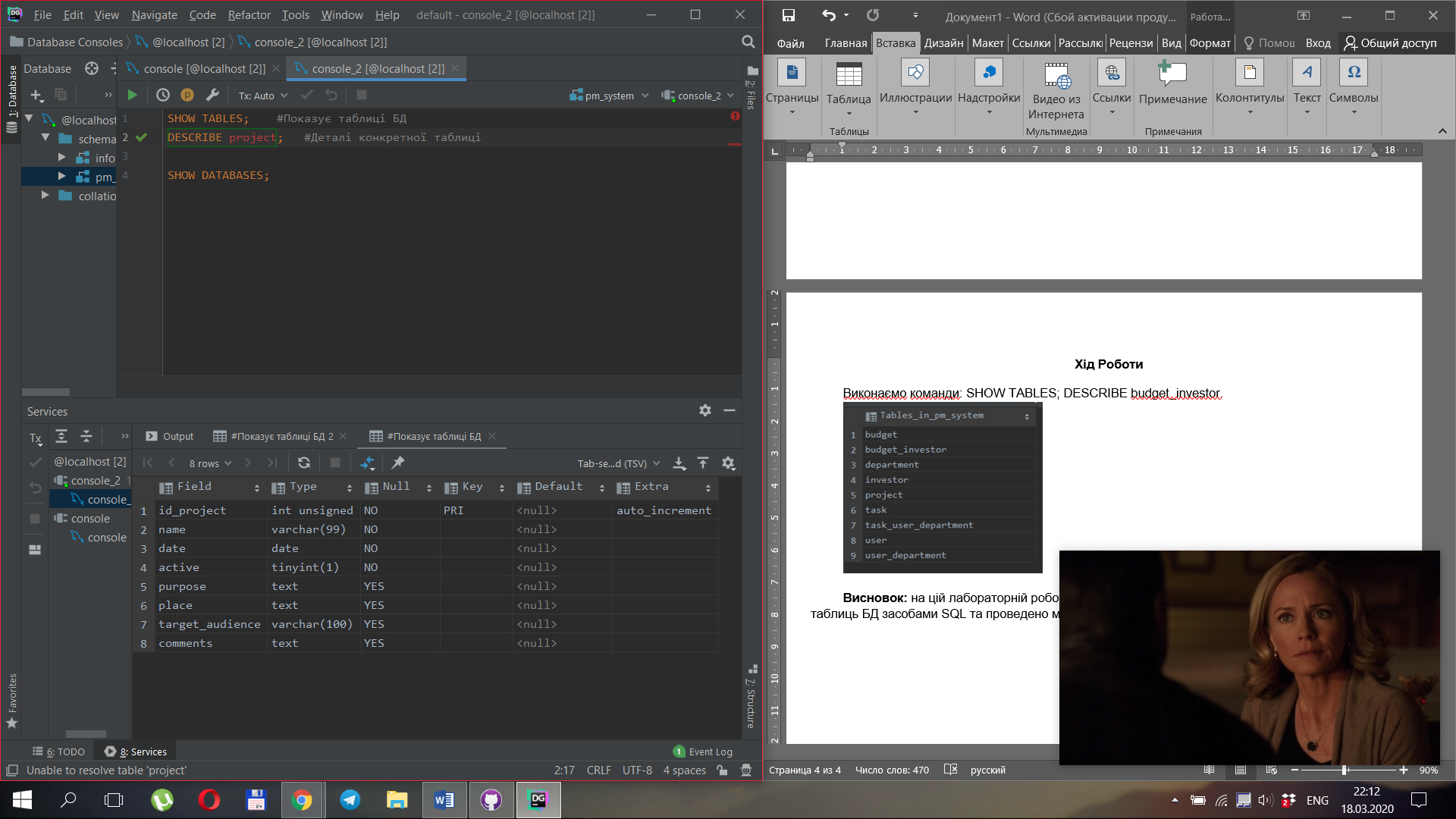
іншу базу, також можна використати директиву RENAME TABLE

ім’я\_бази\_1.назва\_таблиці\_1 TO ім’я\_бази\_2.назва\_таблиці\_2.

**Хід Роботи**

**Виконаємо команди: SHOW TABLES; DESCRIBE budget\_investor.**





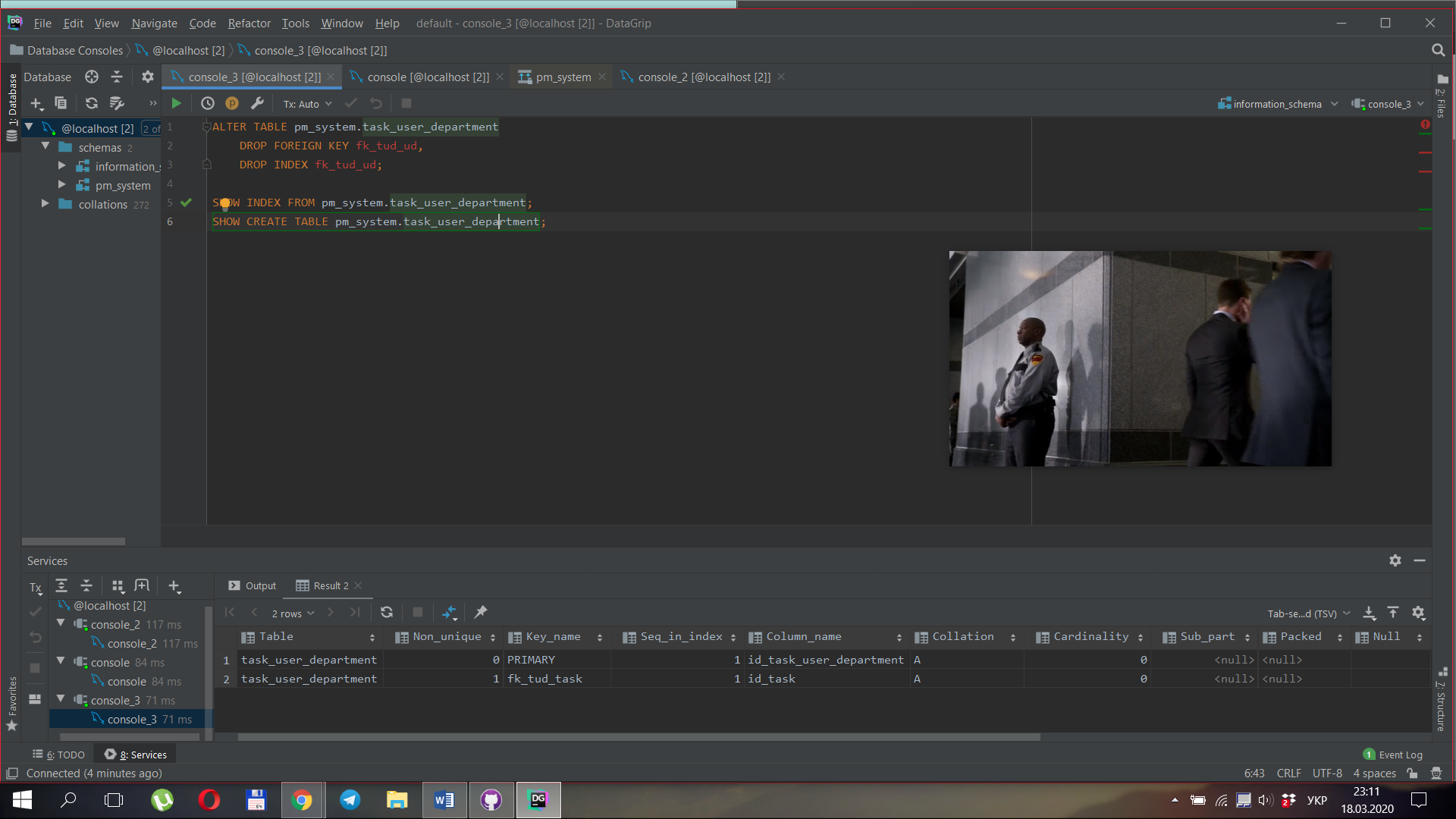
Виконаємо такі операції над схемою бази даних:

1. Вилучимо зв’язок fk\_tud\_ud
2. Вилучимо і додамо поля в таблиці task
3. Змінимо поле в таблиці User
4. Додамо поле і нове обмеження унікальності до таблиці task
5. Змінимо тип обмеження цілісності для зв’язків fk\_department\_project, fk\_task\_department, fk\_user\_user\_department, fk\_department\_user\_department, fk\_tud\_ud, fk\_tud\_task
6. **Вилучимо зв’язок fk\_tud\_ud**

Видалення зв’язку між таблицями шляхом видалення обмеження для таблиці:

ALTER TABLE pm\_system.task\_user\_department  
 DROP FOREIGN KEY fk\_tud\_ud,  
 DROP INDEX fk\_tud\_ud;

Для перевірки результату можна виконати команди:

* SHOW INDEX FROM pm\_system.task\_user\_department
* SHOW CREATE TABLE pm\_system.task\_user\_department.

CREATE TABLE `task\_user\_department` (  
 `id\_task\_user\_department` int unsigned NOT NULL AUTO\_INCREMENT,  
 `id\_user\_department` int unsigned NOT NULL,  
 `id\_task` int unsigned NOT NULL,  
 `start\_data` date NOT NULL,  
 PRIMARY KEY (`id\_task\_user\_department`),  
 KEY `fk\_tud\_task` (`id\_task`),  
 CONSTRAINT `fk\_tud\_task` FOREIGN KEY (`id\_task`) REFERENCES `task` (`id\_task`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8

(після цього повертаємо видалений зв’язок)

ALTER TABLE pm\_system.task\_user\_department  
 ADD CONSTRAINT fk\_tud\_ud FOREIGN KEY(id\_user\_department)  
 REFERENCES PM\_System.user\_department(id\_user\_department) ON UPDATE NO ACTION ON DELETE NO ACTION;

1. **Вилучимо і додамо поля в таблиці task**

Вилучимо поле “status” і додамо поле “start”

ALTER TABLE pm\_system.task  
 DROP COLUMN status,  
 ADD COLUMN start DATE;

1. **Змінимо поле в таблиці User**

Змінюємо поле phone\_number: зберігаємо розмірність, але робимо його необов’язковим

ALTER TABLE pm\_system.user  
 MODIFY COLUMN phone\_number char(12) NULL;

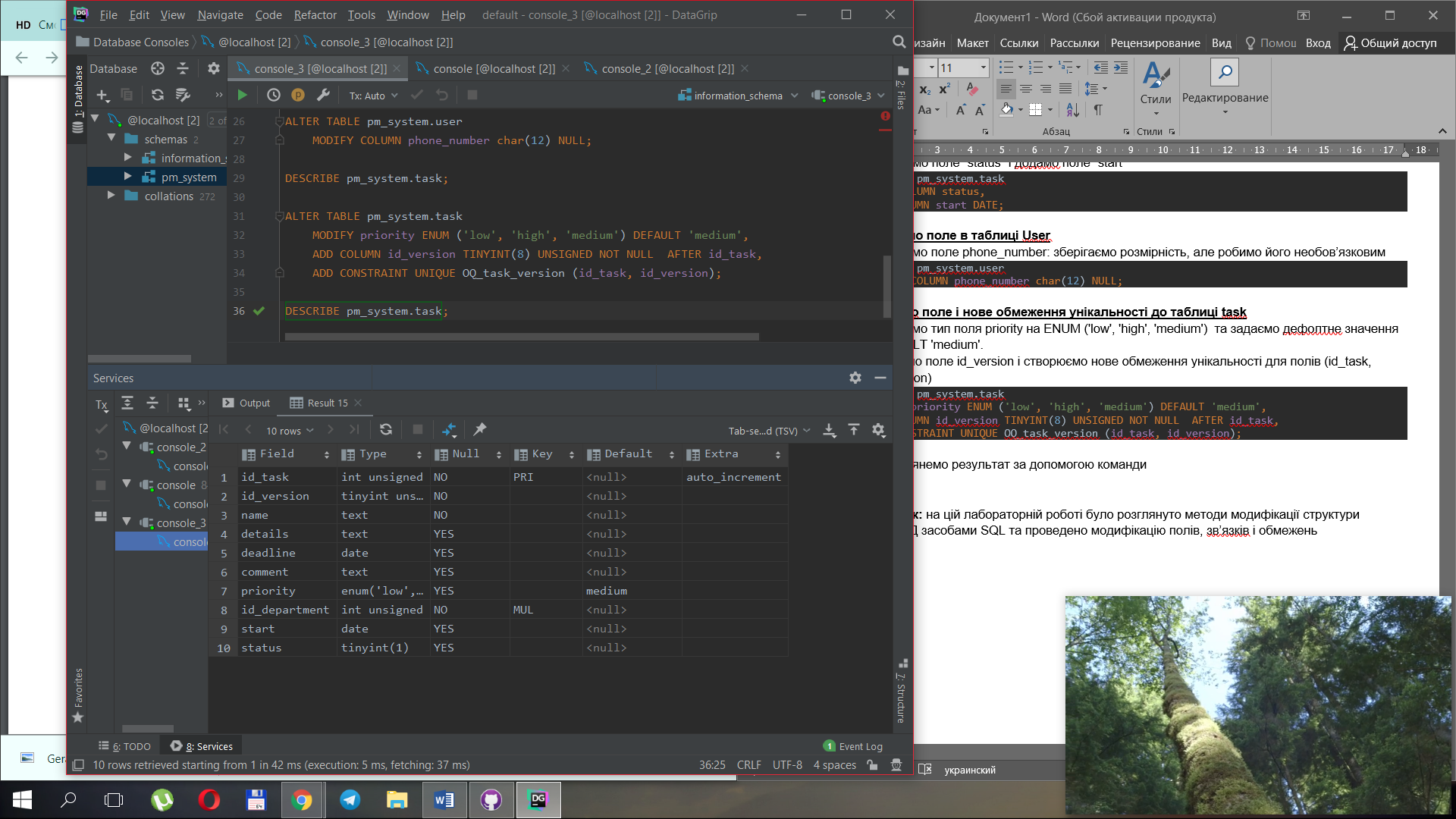
1. **Додамо поле і нове обмеження унікальності до таблиці task**

Змінюємо тип поля priority на ENUM ('low', 'high', 'medium') та задаємо дефолтне значення DEFAULT 'medium'.

Додаємо поле id\_version і створюємо нове обмеження унікальності для полів (id\_task, id\_version)

ALTER TABLE pm\_system.task  
 MODIFY priority ENUM ('low', 'high', 'medium') DEFAULT 'medium',  
 ADD COLUMN id\_version TINYINT(8) UNSIGNED NOT NULL AFTER id\_task,  
 ADD CONSTRAINT UNIQUE OQ\_task\_version (id\_task, id\_version);

Переглянемо результат за допомогою команди DESCRIBE pm\_system.task;



1. **Змінимо тип обмеження цілісності для зв’язків fk\_department\_project, fk\_task\_department, fk\_user\_user\_department, fk\_department\_user\_department, fk\_tud\_ud, fk\_tud\_task**

Змінимо обмеження цілісності для цих зв’язків, щоб при видаленніпроектів з бази видалялись всі його складові (департаменти, завдання…). Для цього потрібно видалити існуюче обмеження і створити нове.

* *fk\_department\_project*

ALTER TABLE pm\_system.department  
 DROP FOREIGN KEY fk\_department\_project;  
ALTER TABLE pm\_system.department  
 ADD CONSTRAINT fk\_department\_project FOREIGN KEY (id\_project)  
 REFERENCES pm\_system.project (id\_project) ON DELETE CASCADE;

* *fk\_task\_department*

ALTER TABLE pm\_system.task  
 DROP FOREIGN KEY fk\_task\_department;  
ALTER TABLE pm\_system.task  
 ADD CONSTRAINT fk\_task\_department FOREIGN KEY (id\_department)  
 REFERENCES pm\_system.department (id\_department) ON DELETE CASCADE;

* *fk\_user\_user\_department*

ALTER TABLE pm\_system.user\_department  
 DROP FOREIGN KEY fk\_user\_user\_department;  
ALTER TABLE pm\_system.user\_department  
 ADD CONSTRAINT fk\_user\_user\_department FOREIGN KEY(id\_user)  
 REFERENCES PM\_System.user(id\_user) ON DELETE CASCADE;

* *fk\_department\_user\_department*

ALTER TABLE pm\_system.user\_department  
 DROP FOREIGN KEY fk\_department\_user\_department;  
ALTER TABLE pm\_system.user\_department  
 ADD CONSTRAINT fk\_department\_user\_department FOREIGN KEY(id\_department)  
 REFERENCES PM\_System.department(id\_department) ON DELETE CASCADE;

* *fk\_tud\_ud*

ALTER TABLE pm\_system.task\_user\_department  
 DROP FOREIGN KEY fk\_tud\_ud;  
ALTER TABLE pm\_system.task\_user\_department  
 ADD CONSTRAINT fk\_tud\_ud FOREIGN KEY(id\_user\_department)  
 REFERENCES PM\_System.user\_department(id\_user\_department) ON DELETE CASCADE;

* *fk\_tud\_task*

ALTER TABLE pm\_system.task\_user\_department  
 DROP FOREIGN KEY fk\_tud\_task;  
ALTER TABLE pm\_system.task\_user\_department  
 ADD CONSTRAINT fk\_tud\_task FOREIGN KEY(id\_task)  
 REFERENCES PM\_System.task(id\_task) ON DELETE CASCADE;

**Висновок:** на цій лабораторній роботі було розглянуто методи модифікації структури таблиць БД засобами SQL та проведено модифікацію полів, зв’язків і обмежень