**Методичні рекомендації**

**до виконання лабораторної роботи №№ 10-11**

Багатотабличні запити. Природні, внутрішні й перехресні об'єднання.

Мета: Здобути практичні навички використання об'єднань при реалізації запитів до декількох таблиць бази даних.

!!! Увага!!!

При виконанні лабораторної роботи необхідно реалізувати 3 запити:

1 Об'єднання двох таблиць

2 Об'єднання декількох таблиць (3-4 таблиці)

3 Самооб'єднання таблиць

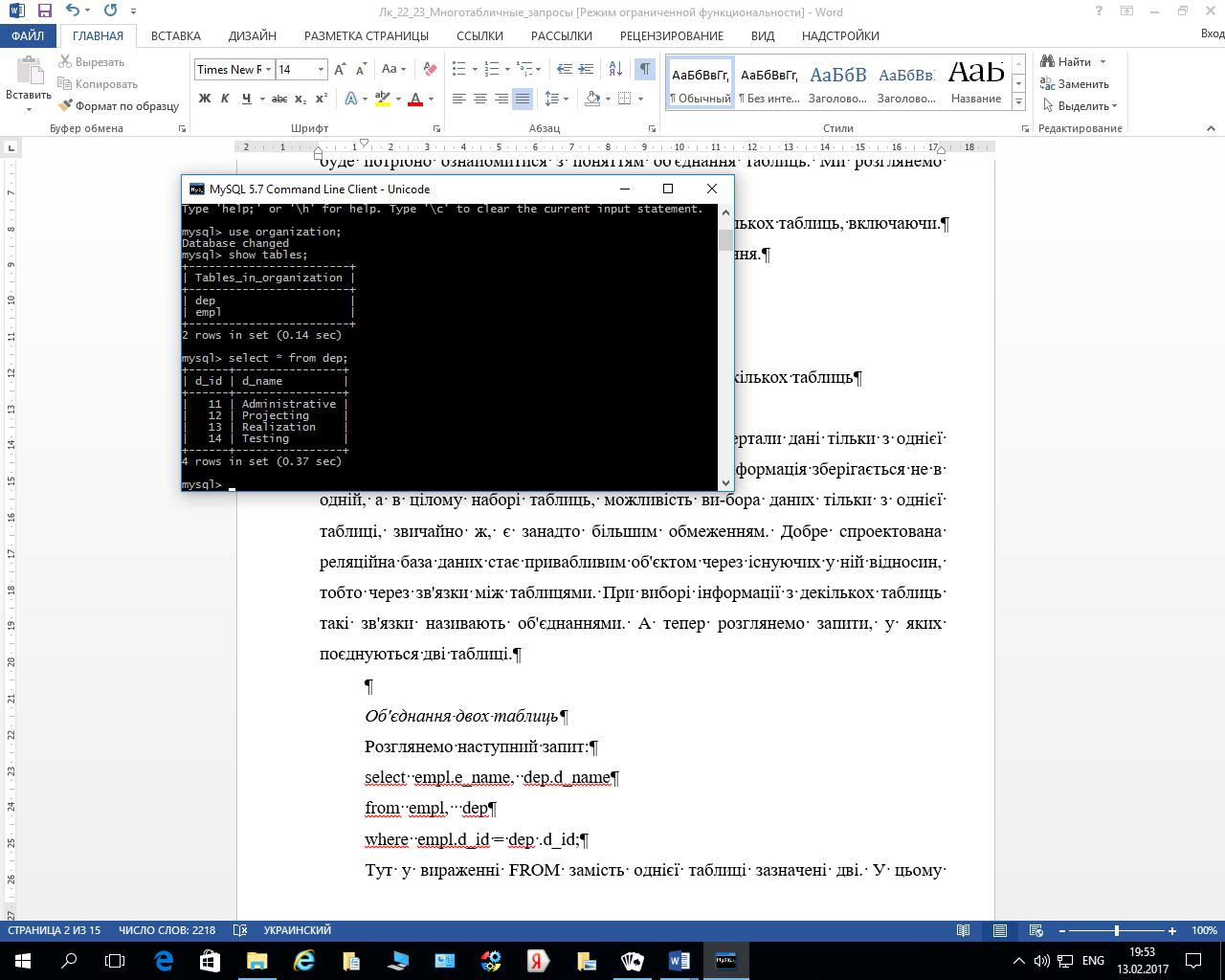
ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Запити, розглянуті в попередній лабораторній роботі, дозволяли отримати дані тільки з однієї таблиці. В умовах нормалізованої бази даних, коли інформація зберігається не в одній, а в цілому наборі таблиць, можливість ви-бора даних тільки з однієї таблиці, звичайно ж, є занадто великим обмеженням. Добре спроєктована реляційна база даних стає привабливим об'єктом через існуювання у ній відносин, тобто через зв'язки між таблицями. При виборі інформації з декількох таблиць такі зв'язки називають о***б'єднаннями***. А тепер розглянемо запит, у якому поєднуються дві таблиці.

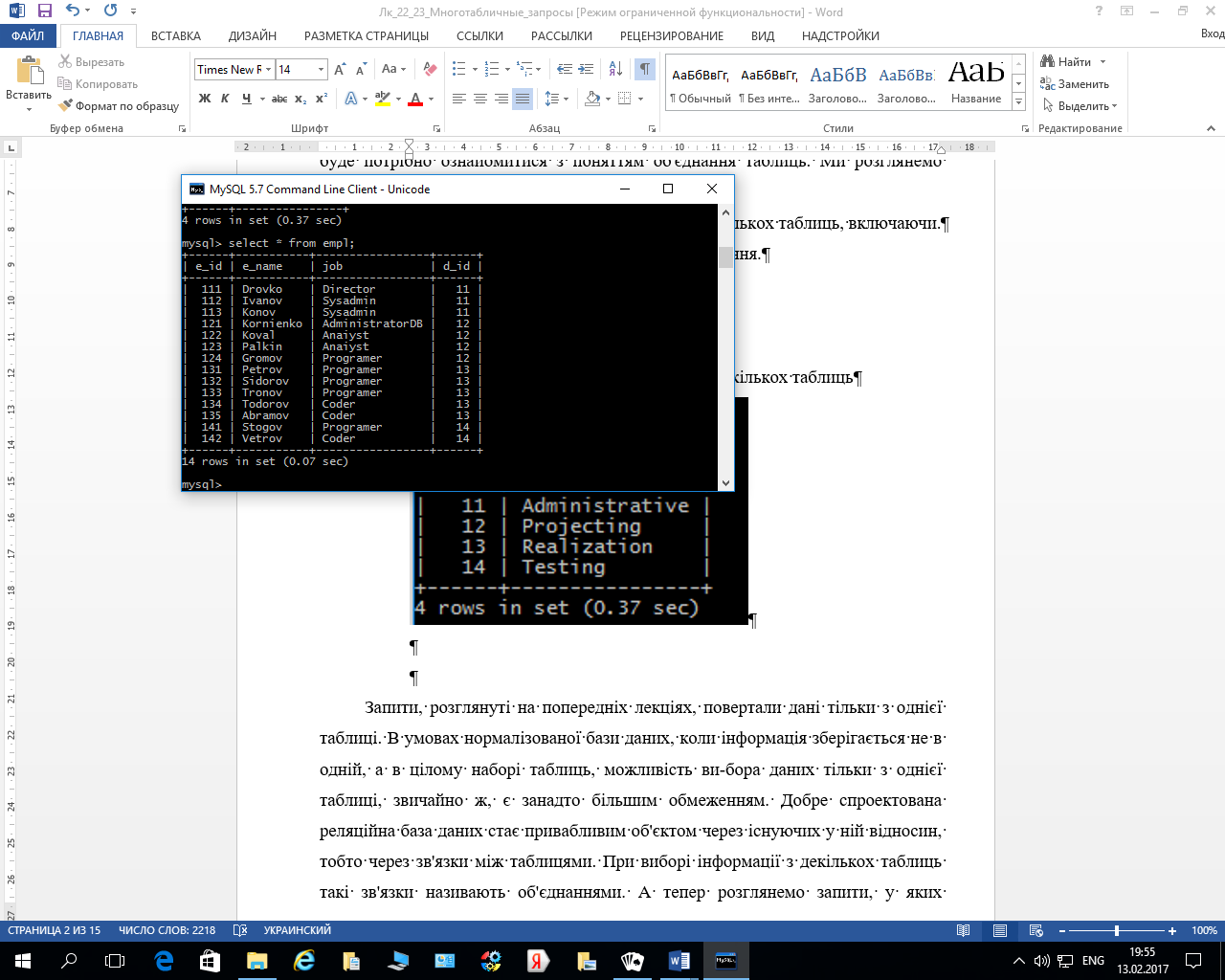
1. **Об'єднання двох таблиць**

Для прикладу використовуємо 2 таблиці:

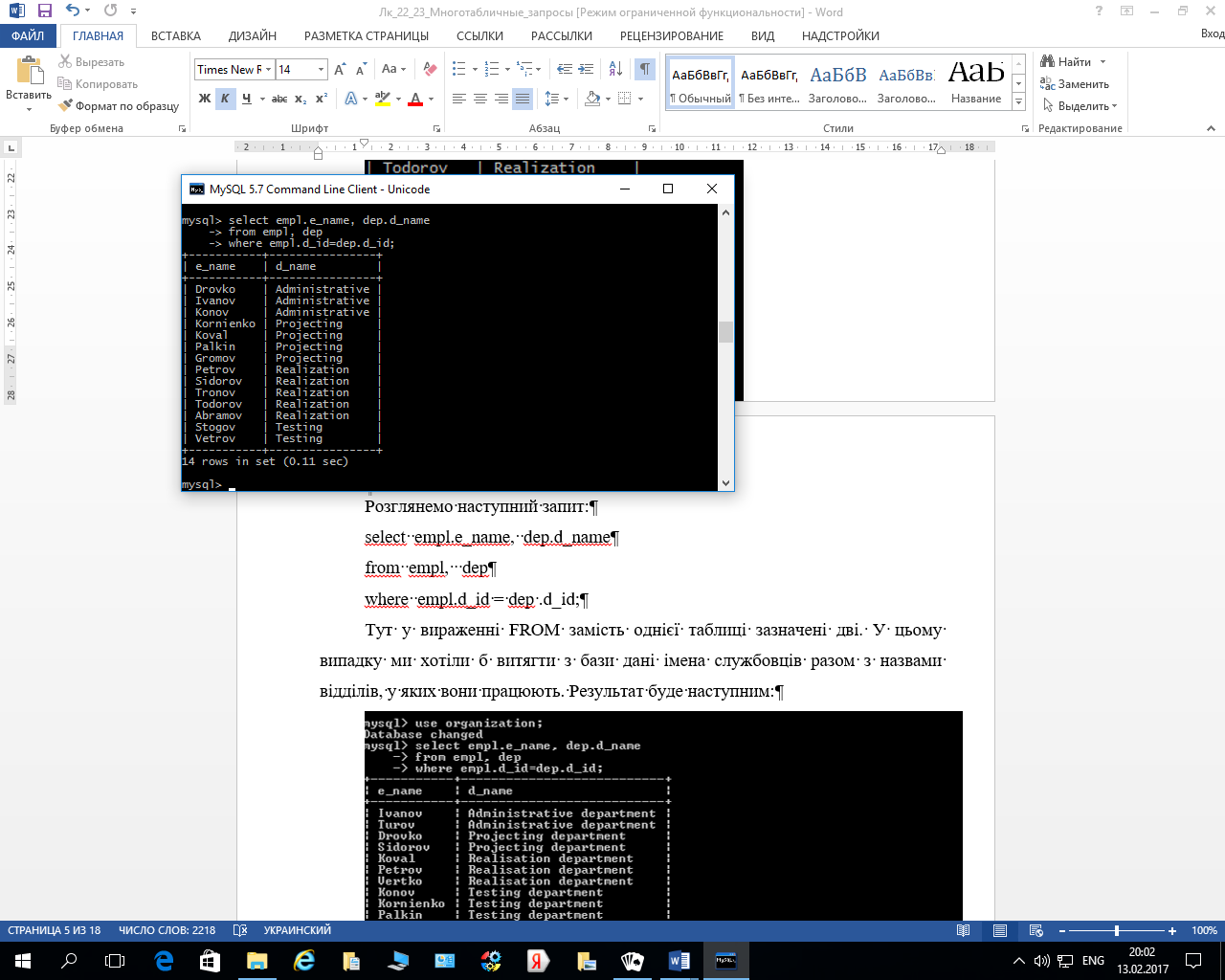
1 Відділ(Номер відділу, Назва відділу)



2 **Співробітник**(Табельный\_номер, ПІБ\_співробітника, Посада, Номер\_відділу)



Реалізуємо запит 1: отримати прізвища співробітників разом з назвами відділів, у яких вони працюють.



Як вийшов такий результат? Спочатку ми вибрали стовпці, що належать двом різним таблицям. (Як бачите, ми застосували нотацію, що використає крапку-роздільник, щоб розрізнити поле e\_name з таблиці empl і поле d\_name з таблиці dep). Для цього треба було включити обидві таблиці у опцію FROM.

Найцікавішим у цьому запиті є опція WHERE. Набір умов, використовуваних при створенні зв'язку для об'єднання таблиць, називають *умовою об'єднання*. У цьому випадку ми використали умову

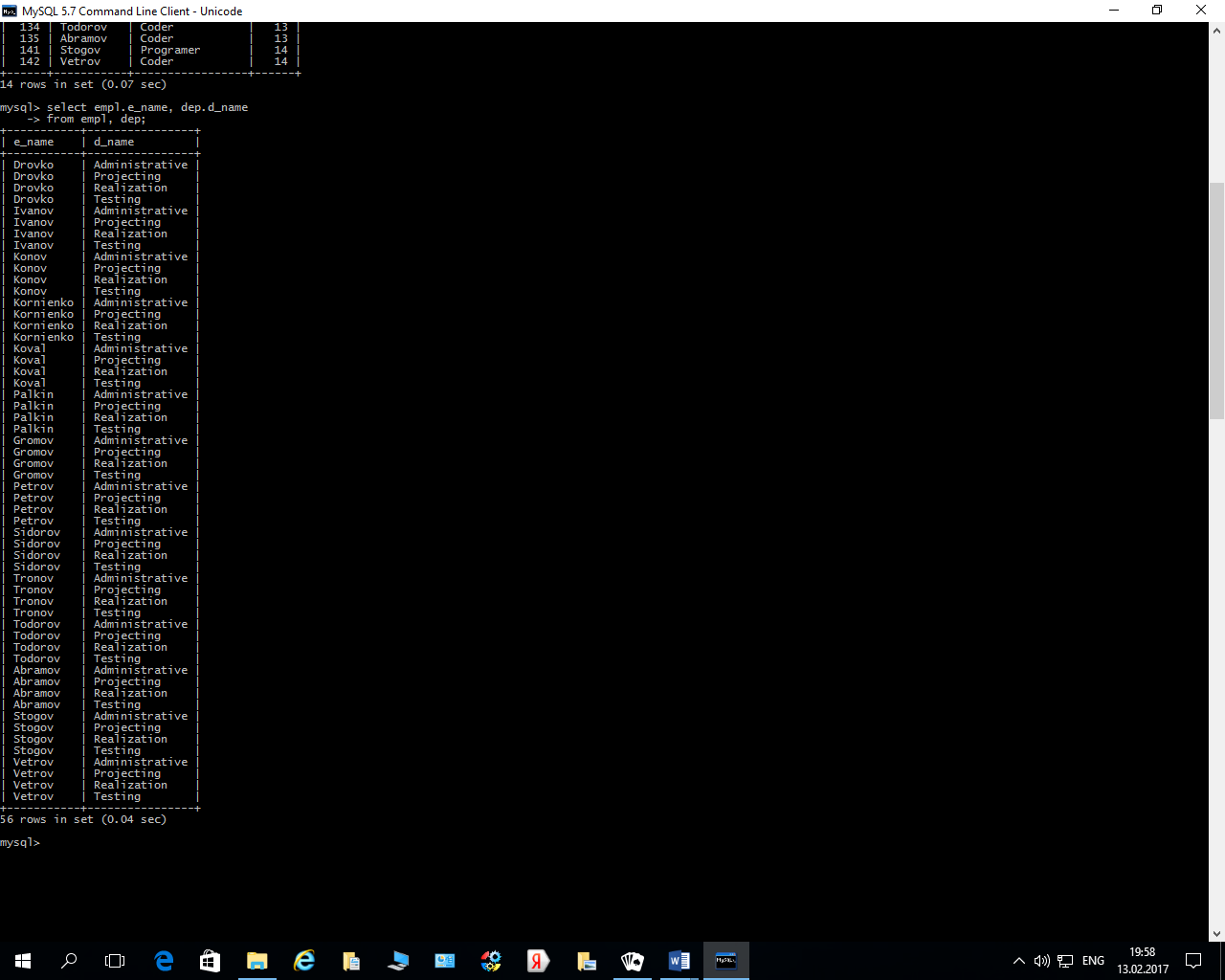
empl. d\_іd = dep . d\_іd,

що зв'язує таблиці по зовнішніх ключах

Якщо виконати подібний запит без опції WHERE,

select empl.e\_name, dep.d\_name from empl, dep;

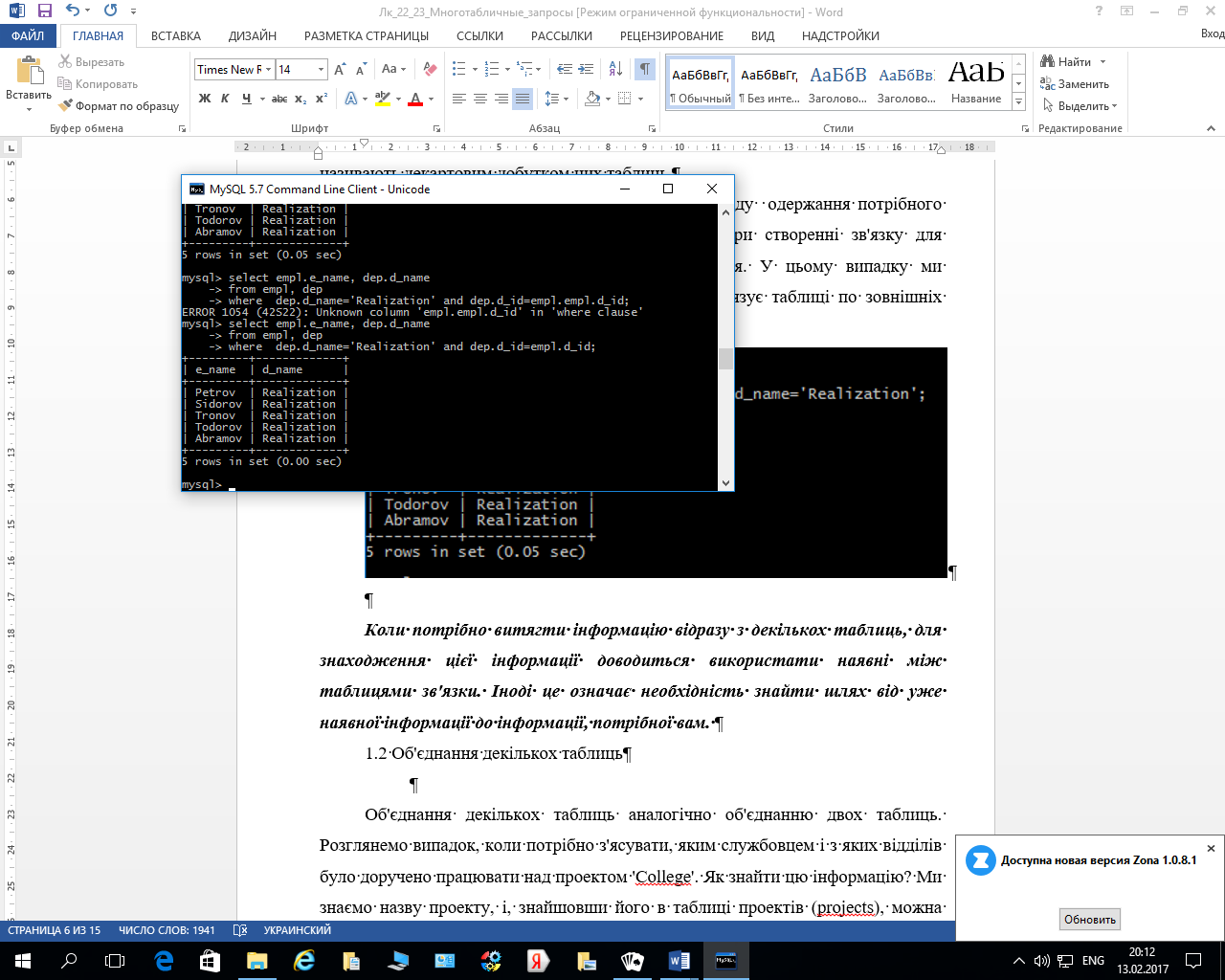
ми одержимо наступний результат:



Ми отримали всі можливі комбінації прізвищ співробітників і назв відділів. Однак це не дає ніякої можливості з'ясувати, які рядки несуть правильну інформацію, а які - помилкову. Результуюча множина кортежів, що містить всі можливі комбінації даних із двох таблиць, називають **декартовим добутком** цих таблиць.

Отже, опція WHERE є важливою з погляду отримання потрібного нам результату.

Запит 2: Отримати прізвища співробітників, які працюють в відділі реалізації:



1. **Об'єднання декількох таблиць**

Коли потрібно отримати інформацію відразу з декількох таблиць, доводиться використтовувати наявні між таблицями зв'язки. Іноді це означає необхідність **знайти шлях від уже наявної інформації до інформації, потрібної вам.**

Розглянемо випадок, коли потрібно з'ясувати, ***яким співробітникам і з яких відділів було доручено працювати над проєктом 'College'.*** Як знайти цю інформацію? Ми знаємо назву проєкту, і, знайшовши його в таблиці проєктів (projects), можна з'ясувати його кодовий номер (pr\_іd). Це можна використати для того, щоб знайти відповідні рядки в таблиці proj\_emp і побачити, які службовці працювали над даним проектом - ми одержимо кодові номери службовців (e\_іd), а по таблиці службовців (empl) одержати прізвища службовців і з'ясувати номера відділів, у яких ці службовці працюють. Із цією інформацією ми можемо звернутися до таблиці відділів (dep) і знайти назву відповідного відділу!

Маючи план у вигляді зазначеного шляху через чотири таблиці, ми повинні створити запит, що відбиває нашу логіку. Такий запит може виглядати в такий спосіб:

select dep .d\_name, empl.e\_name

from projects , proj\_emp, empl, dep

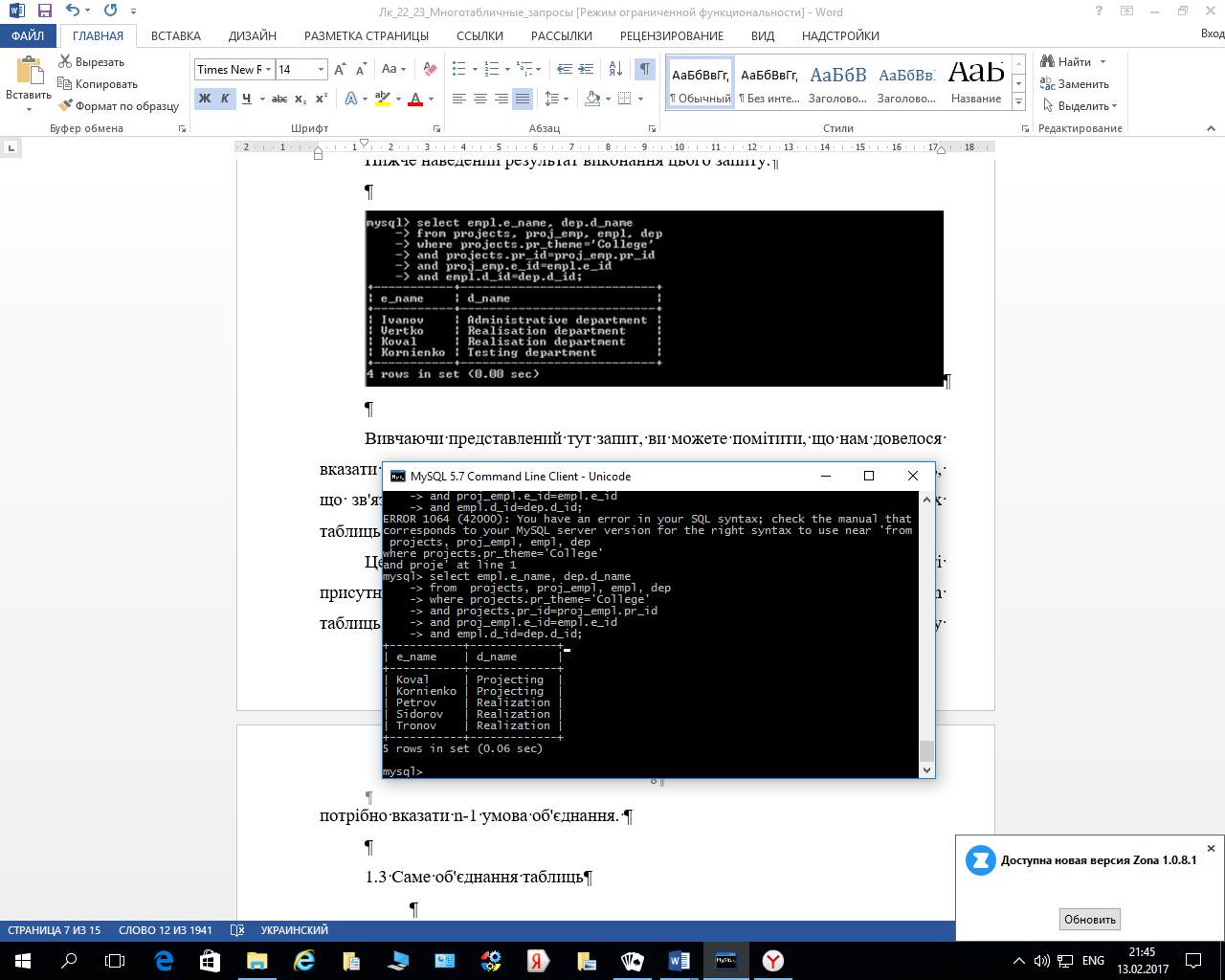
where projects .pr\_name='College'

and projects .pr\_іd = proj\_emp .pr\_іd

and proj\_emp .e\_іd = empl. e\_іd

and empl. d\_іd = dep .d\_іd;

Нижче наведений результат виконання цього запиту.

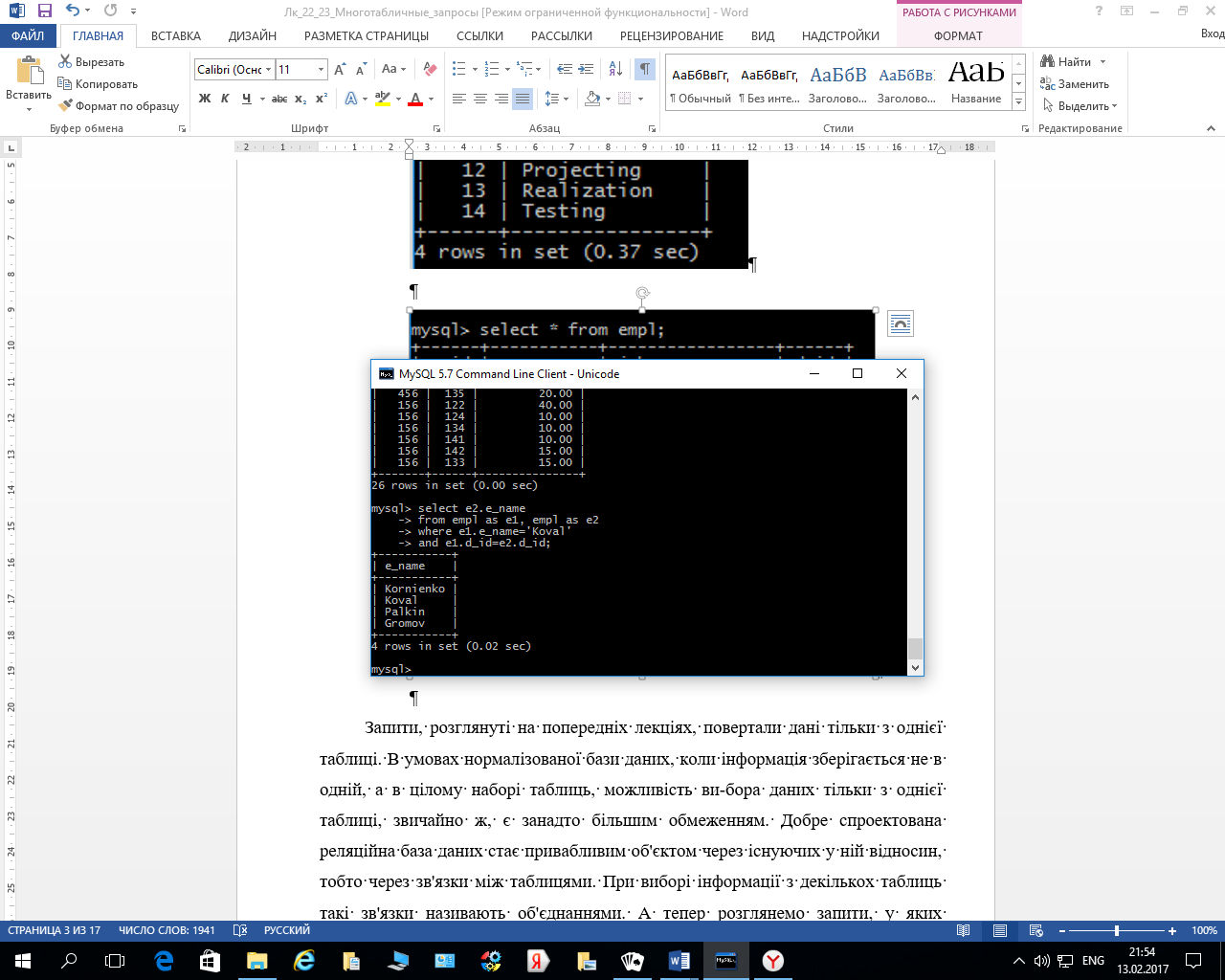


Вивчаючи представлений тут запит, ви можете помітити, що нам довелося вказати спочатку всі таблиці спланованого нами шляху, а потім умови об'єднань, що зв'язують ці таблиці одну з іншої. Як бачите, для зв'язування чотирьох таблиць нам знадобилися три умови об'єднання.

Це спостереження можна використати для перевірки того, що в запиті присутні всі умови, необхідні для об'єднання. Якщо потрібно об'єднати n таблиць, те, як правило, кожна умова буде зв'язувати пари таблиць, тому потрібно вказати n-1 умова об'єднання.

1. **Самооб'єднання таблиць**

Аналогічно об'єднанню таблиці з іншою таблицею, можна об'єднати таблицю саму із собою. Але навіщо це робити? Іноді вас можуть цікавити зв'язки між рядками однієї й тієї ж таблиці. ***Припустимо, що ви хочете з'ясувати прізвища службовців відділу, у якому працює Koval.*** Для цього необхідно знайти в таблиці empl номер відділу (d\_іd), у якому працює Konov, а потім подивитися в таблиці empl, хто ще працює в тім же відділі.



У цьому запиті для таблиці empl ми визначили два різних *псевдоніми.* По суті ми повідомили системі MySQL, що хотіли б мати дві окремих таблиці, el й е2, які повинні містити ті самі дані. Після цього ми поєднуємо їх так само, як будь-які інші таблиці. Тепер спочатку знаходимо рядок 'Koval' у таблиці el (el .e\_name='Konov'), а потім у таблиці е2 шукаємо рядки з тим же значенням номера відділу, що й в Koval (el .d\_іd = e2 .d\_іd). Спочатку це може здаватися незвичним, але в роботі з декількома таблицями в запитах ідея самооб'єднання таблиць не повинна викликати серйозних труднощів.

Дуже просто додати умову, що виключить Koval з результуючого списку:

select e2.name

from empl el, empl e2

where el.e\_name = 'Koval'

and el.d\_іd = e2 .d\_іd

and e2.e\_name != 'Koval';

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1. Реалізувати за допомогою оператора SELECT необхідно реалізувати 3 запити:

1 Об'єднання двох таблиць

2 Об'єднання декількох таблиць (3-4 таблиці)

3 Самооб'єднання таблиць

запитів до таблиць бази даних згідно Вашого варіанта.

ІІ. Оформити звіт по виконанню лабораторної роботи.

Звіт повинен включати наступні пункти для кожного запиту:

1. Формулювання запиту.

2. Виведення вмісту таблиць, для яких сформульовано запит.

3. Оператор SELECT до відповідного запиту.

4. Копія вікна виконання оператора SELECT.