МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА СИСТЕМОТЕХНІКИ

Звіт

З практичної роботи №6

На тему: «Cтворення й використання уявлень для високонавантажених баз даних на платформі СУБД MySQL»

з дисципліни «Проектування високонавантажених систем зберігання даних»

|  |  |
| --- | --- |
| Виконав:  ст. гр. ІТКНу-19-2  Марковець Н.С. | Перевірив викладач:  Коваленко А.І. |
|  |  |

Харків 2020

**Мета:**

– набуття практичних навичок зі створення тимчасових таблиць, що використовуються як джерело даних SQL-запитів на вибірку й модифікацію даних;

– набуття практичних навичок з розробки уявлень (VIEW), що використовуються як джерело даних SQL-запитів на вибірку й модифікацію даних;

– набуття практичних навичок з розробки SQL-запитів на вибірку й модифікації даних з використанням уявлень (VIEW) і тимчасових таблиць для забезпечення основних бізнес-процесів високонавантаженої інформаційної системи;

– формування необхідних практичних умінь для аналізу плану виконання SQL-запитів до уявлень (VIEW) за допомогою оператора EXPLAIN;

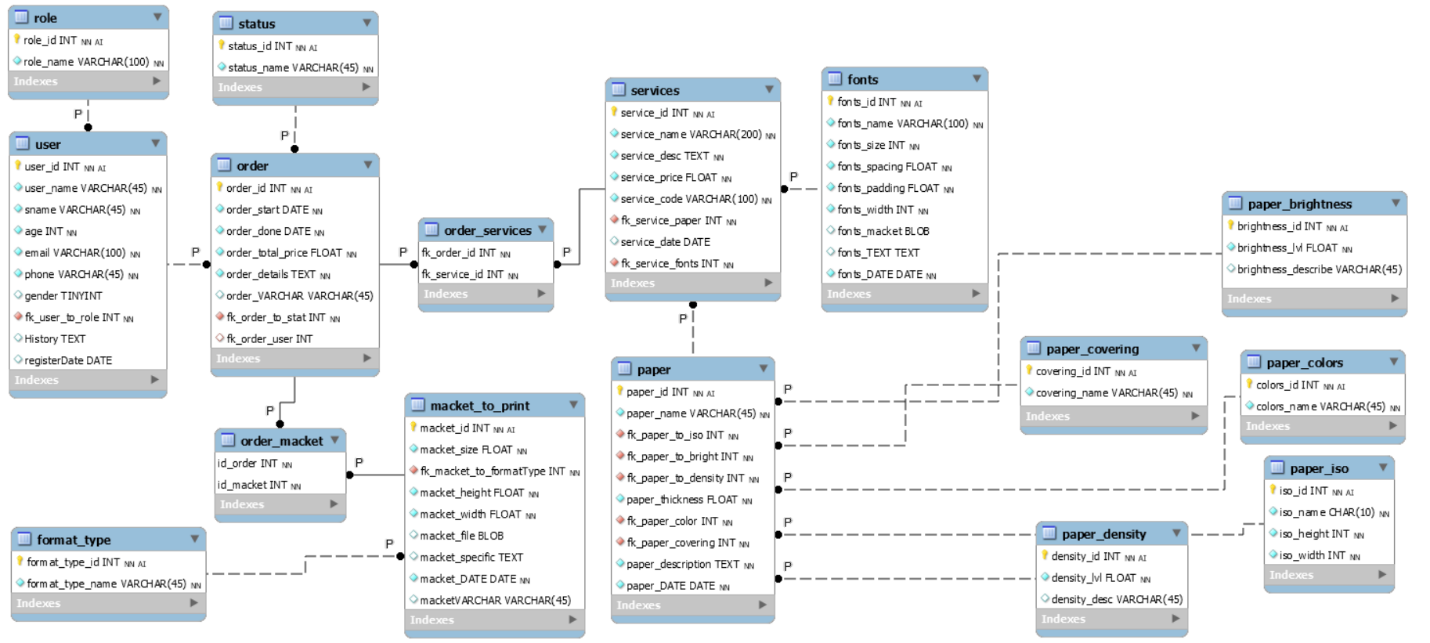
– формування необхідних практичних умінь для створення уявлень (VIEW), з урахуванням особливостей роботи високонавантажених інформаційних систем зберігання даних.**Тема індивідуального завдання:** Інформаційна система «Надання послуг типографії»

Рисунок 1. Схема фізичної моделі даних типу InnoDb у нотації IDEF1X

Завдання 6.1

– код SQL-запиту, що використовується для створення тимчасової таблиці;

– код SQL-запитів до тимчасової таблиці з операторами SELECT, UPDATE, DELETE і INSERT.

– скріншоти результатів виконання SQL-запитів до тимчасової таблиці для двох іменованих підключень (сесій) «name1» і «name2»;

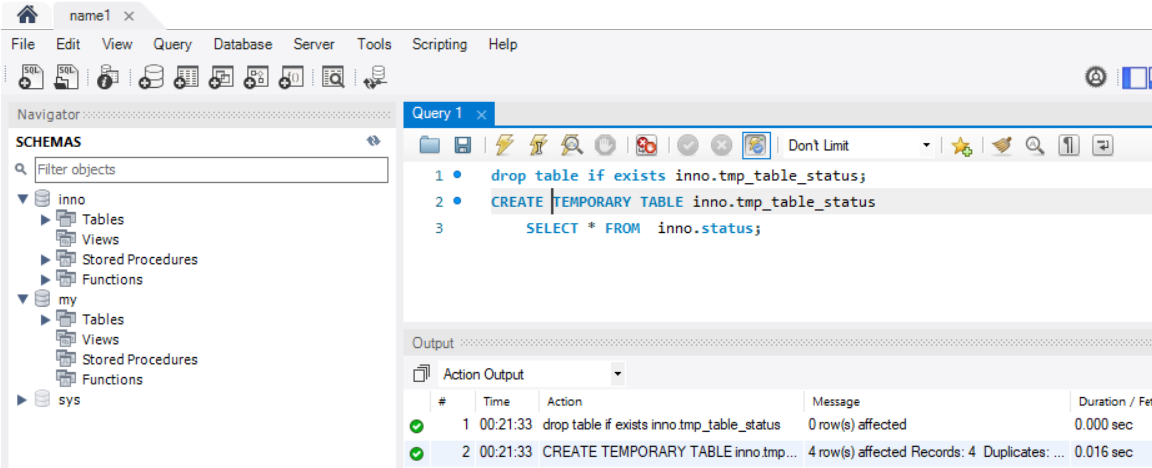


Рисунок 6.1 – Створення тимчасової таблиці на основі таблиці inno.status

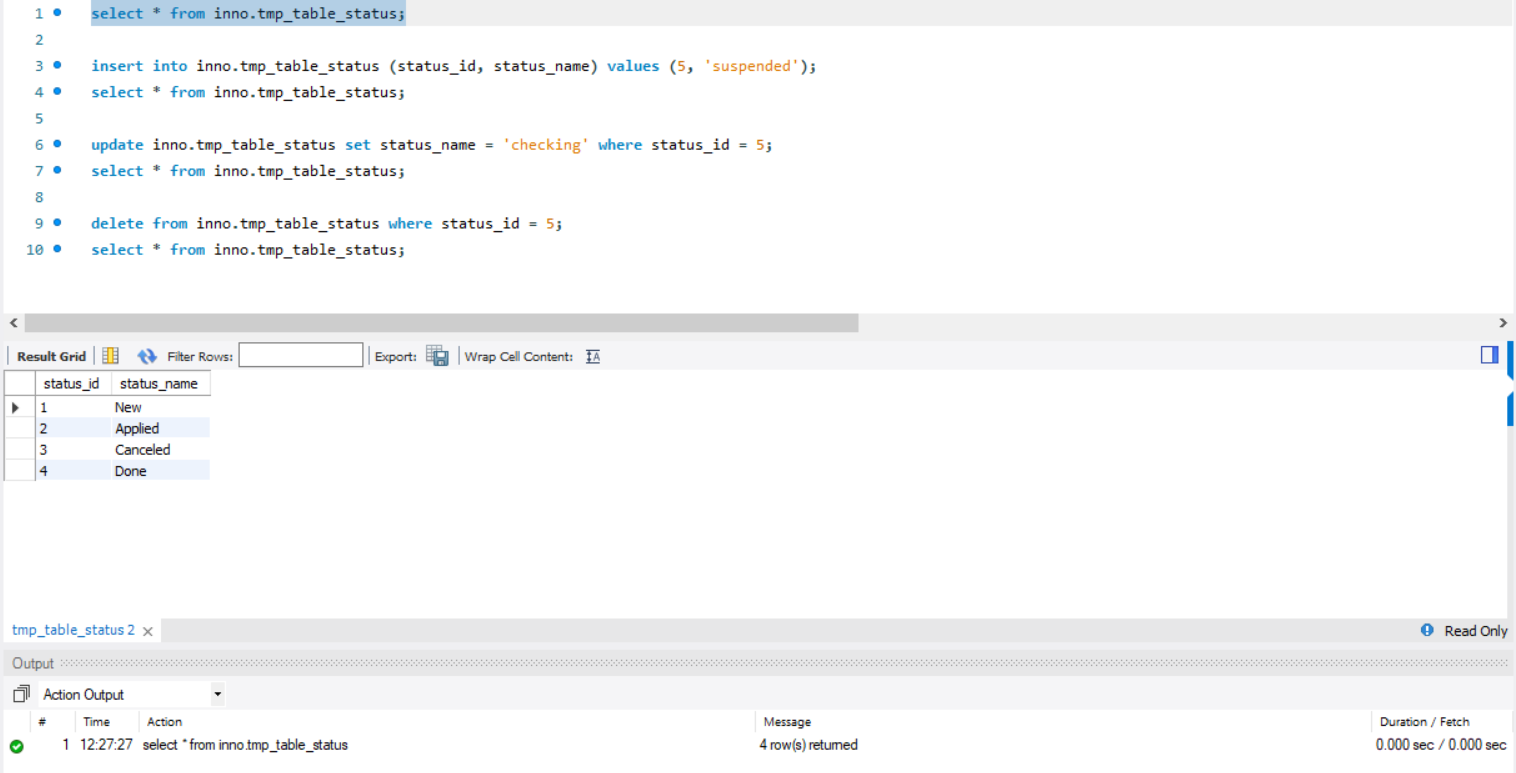


Рисунок 6.2 – Запит SELECT до тимчасової таблиці

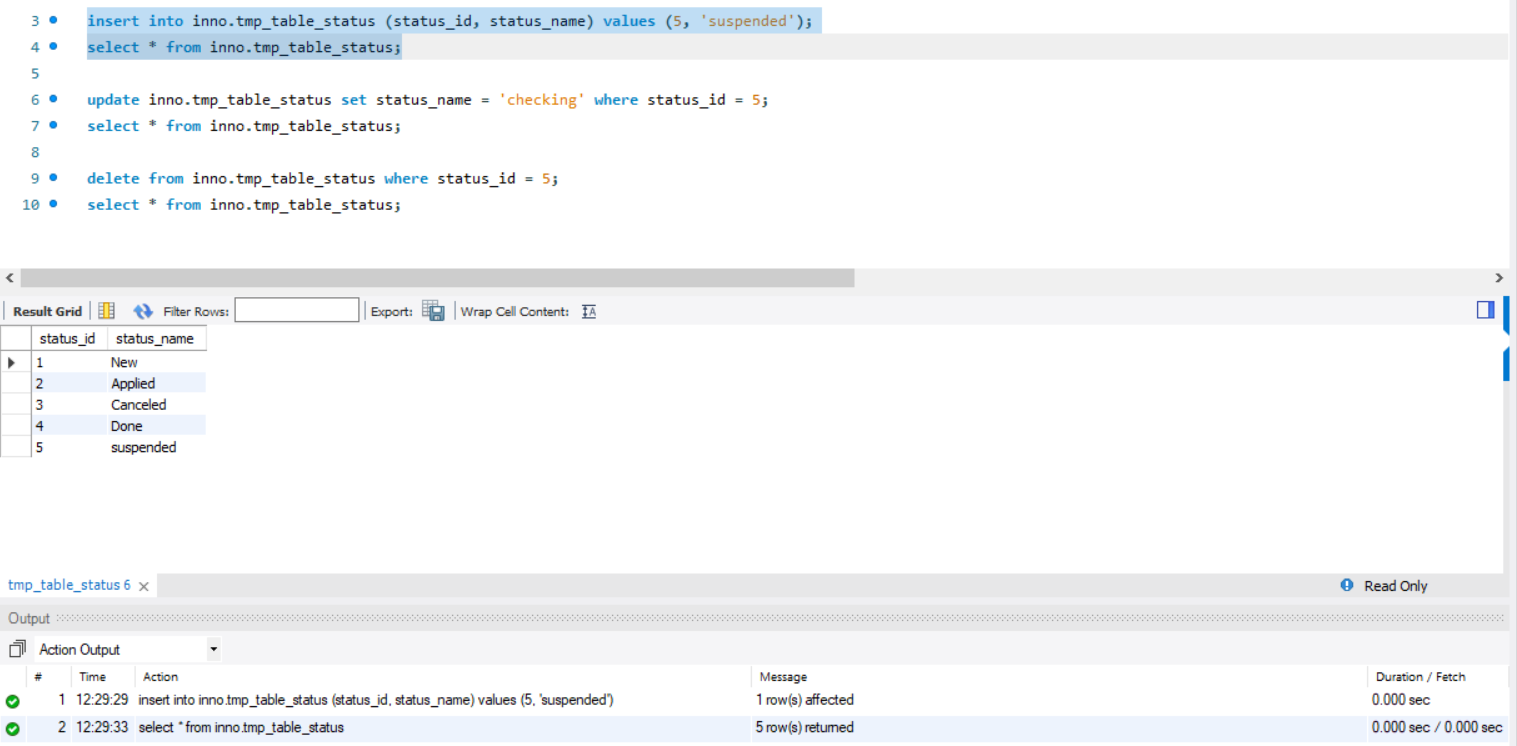


Рисунок 6.3 – Запит INSERT до тимчасової таблиці

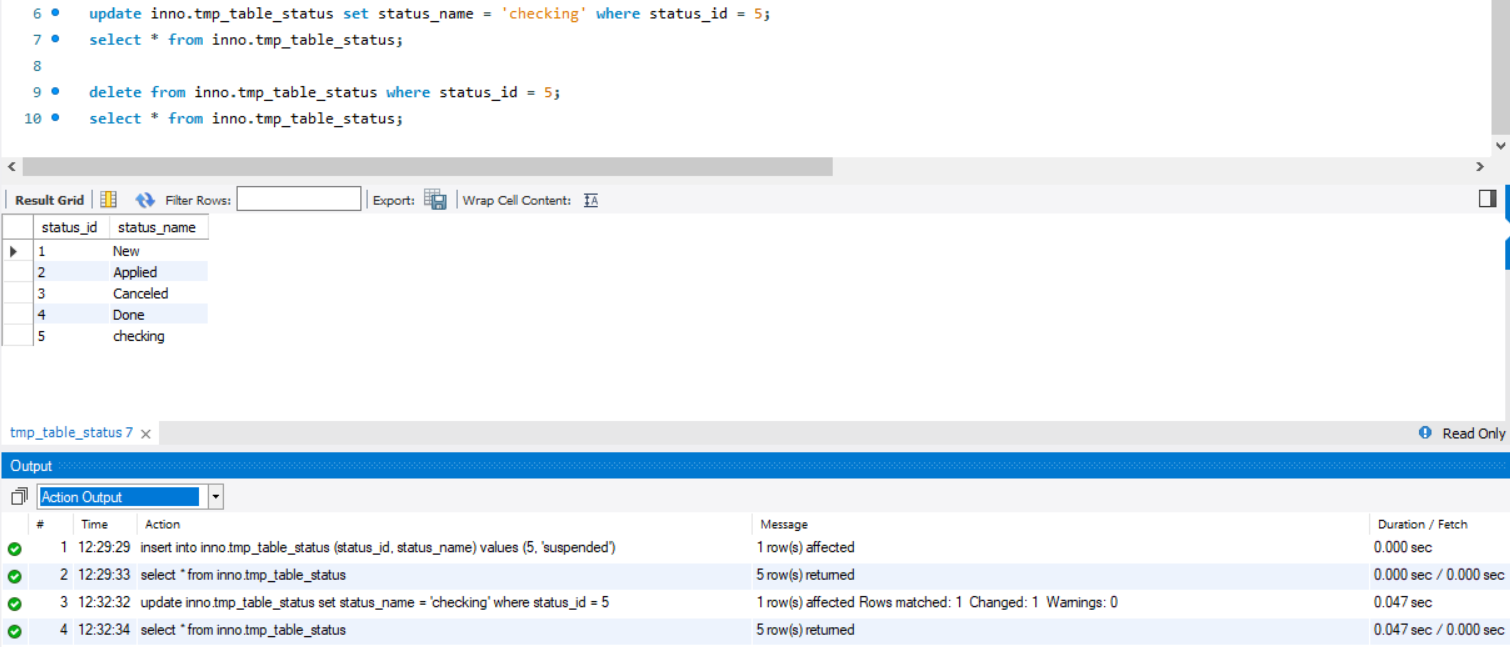


Рисунок 6.4 – Запит UPDATE до тимчасової таблиці

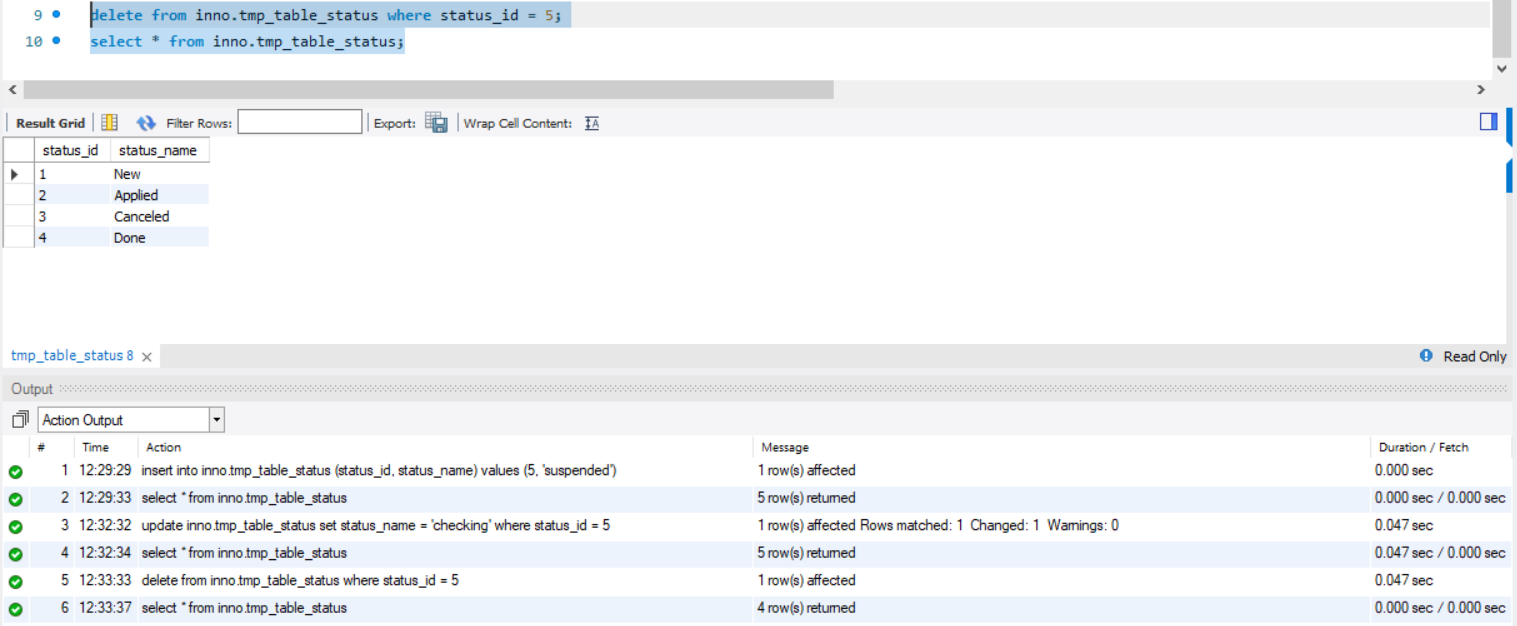


Рисунок 6.5 – Запит DELETE до тимчасової таблиці

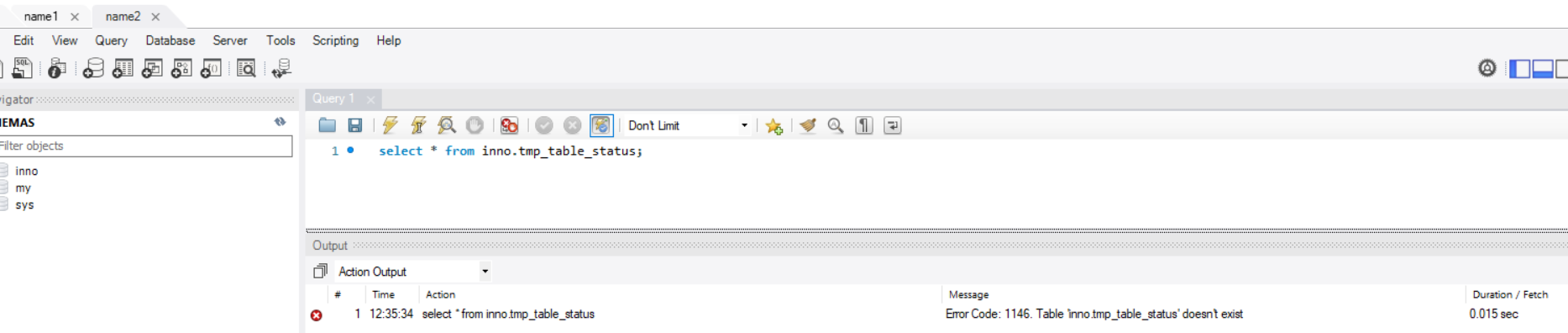


Рисунок 6.6 – Результат виконання запиту з іменованого підключення name2

Завдання 6.2

|  |  |  |
| --- | --- | --- |
|  |  |  |

Завдання 6.3

– код SQL-запиту CREATE VIEW, який створює базове уявлення VIEW 1 і використовує алгоритм MERGE;

– код 5-6 SQL-запитів (5 – «5 (90-95)», 6 – «5+ (>95)») на вибірку даних з базового уявлення VIEW 1 (на основі алгоритму MERGE). Скріншоти результатів виконання SQL-запитів до уявлення VIEW 1;

– код SQL-запиту CREATE VIEW, який створює уявлення VIEW 2 на основі базового уявлення VIEW 1 і використовує алгоритм MERGE;

– код 5-6 SQL-запитів (5 – «5 (90-95)», 6 – «5+ (>95)») на вибірку даних з уявлення VIEW 2 (на основі алгоритму MERGE). Скріншоти результатів виконання SQL-запитів до уявлення VIEW 2;

– код SQL-запиту CREATE VIEW, який створює базове уявлення VIEW 3 і використовує алгоритм TEMPTABLE;

– код 5-6 SQL-запитів (5 – «5 (90-95)», 6 – «5+ (>95)») на вибірку даних з базового уявлення VIEW 3 (на основі алгоритму TEMPTABLE). Вибірка має містити поля згрупованих даних (GROUP BY) і функцій агрегації. Скріншот результатів виконання SQL-запитів до уявлення VIEW 3;

– код SQL-запиту CREATE VIEW, який створює уявлення VIEW 4 на основі базового уявлення VIEW 3 й використовує алгоритм TEMPTABLE;

– код 5-6 SQL-запитів (5 – «5 (90-95)», 6 – «5+ (>95)») на вибірку даних з уявлення VIEW 4 (на основі алгоритму TEMPTABLE). Скріншот результатів виконання SQL-запитів до уявлення VIEW 4;

– скріншоти Explain-таблиць, отриманих під час виконання SQL-запитів до уявлень, що використовують алгоритми MERGE і TEMPTABLE. Оцінка планувиконання кожного SQL-запиту з висновком «неможливо оптимізувати» або «вимагає оптимізації»;

**6.3.1** Код SQL-запиту CREATE VIEW, який створює базове уявлення VIEW 1 і використовує алгоритм MERGE. Представлення створене для відображення замовлень, розрахунку кінцевої ціни, виведення контактних даних користувача.

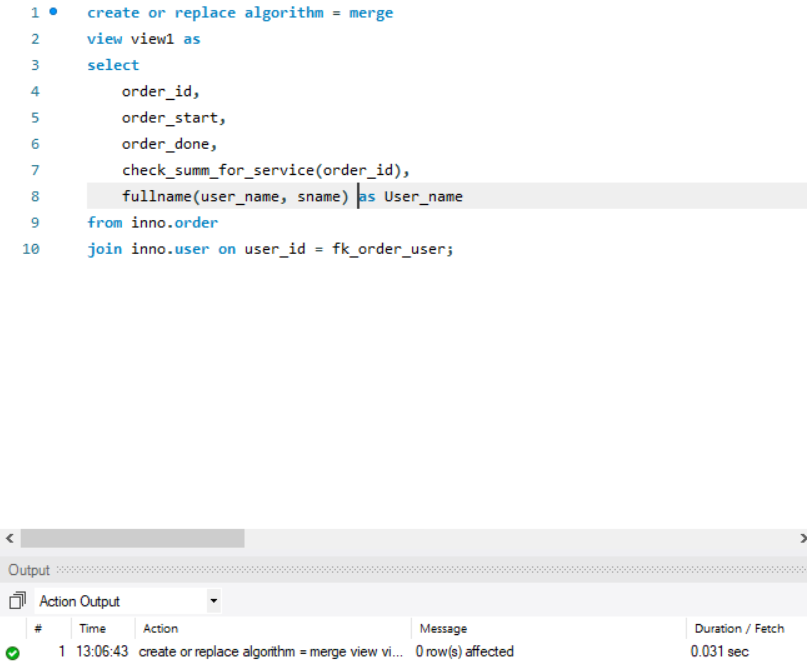


Рисунок 6.7 – Створення представлення view1

**6.3.2** – код 5-6 SQL-запитів (5 – «5 (90-95)», 6 – «5+ (>95)») на вибірку даних з базового уявлення VIEW 1 (на основі алгоритму MERGE). Скріншоти результатів виконання SQL-запитів до уявлення VIEW 1;

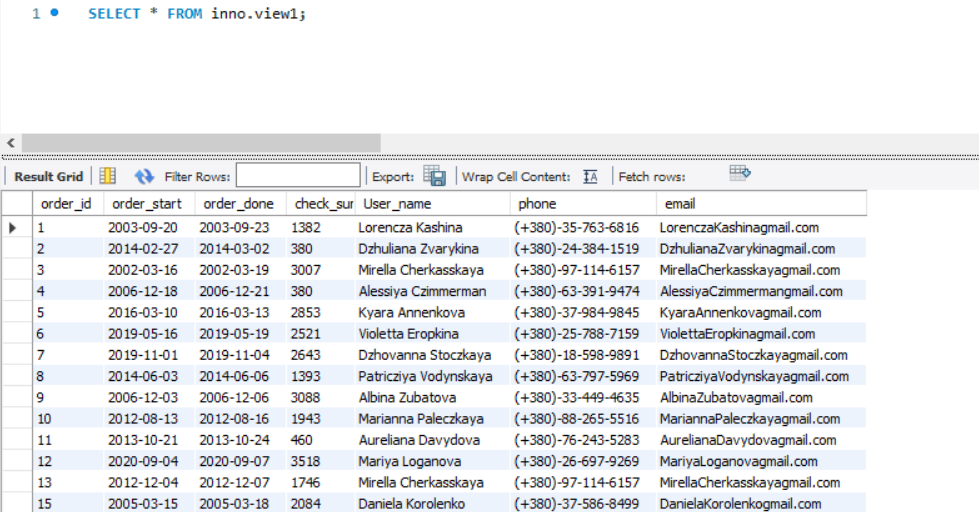


Рисунок 6.8 – Запит 1 (на вибірку всіх даних)

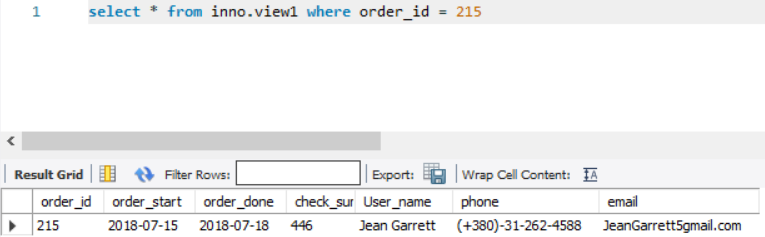


Рисунок 6.9 – Запит 2 (на вибірку даних за номером замовлення)

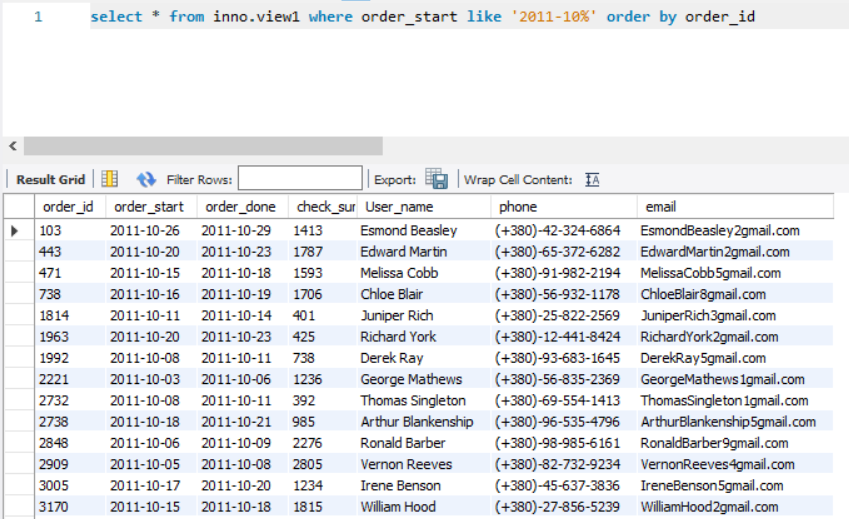


Рисунок 6.10 – Запит 3 (на вибірку даних з датою початку 2011-10)

Для наступного запиту додаємо псевдонім для функції розрахунку ціни замовлення, для зручного звернення то полів.

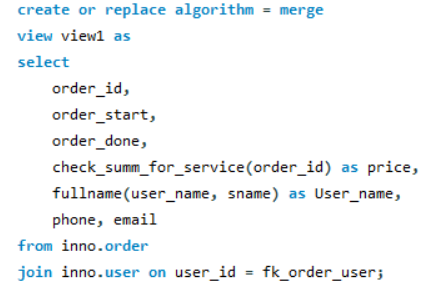


Рисунок 6.10 – Додавання псевдоніму price

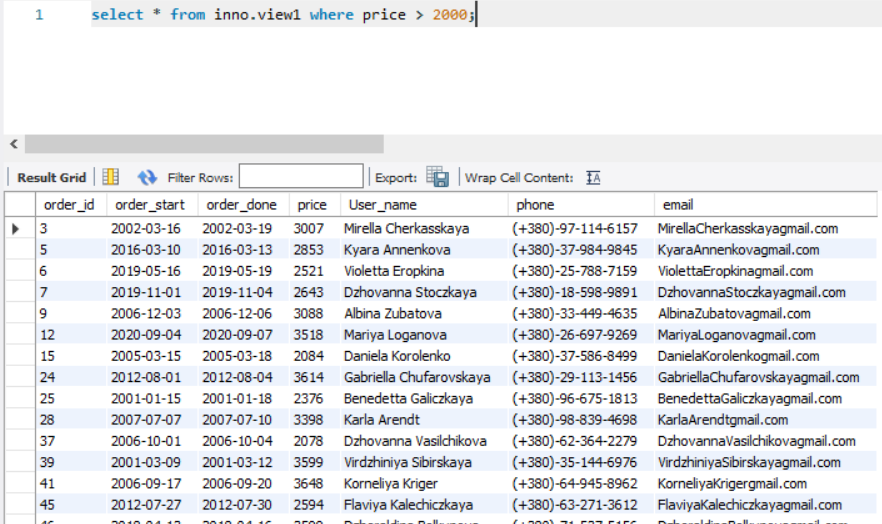


Рисунок 6.11 – Запит 4 (на вибірку замовлень з ціною більшою за 2000)

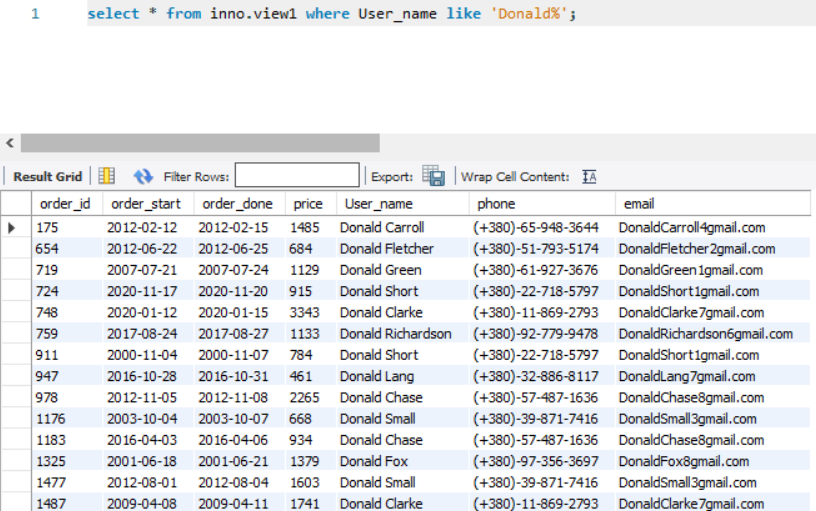


Рисунок 6.12 – Запит 5 (Всі замовлення з ім’ям користувача Donald)

Для наступної вибірки даних, змінимо дані в таблиці, на основі якої побудовано view1.

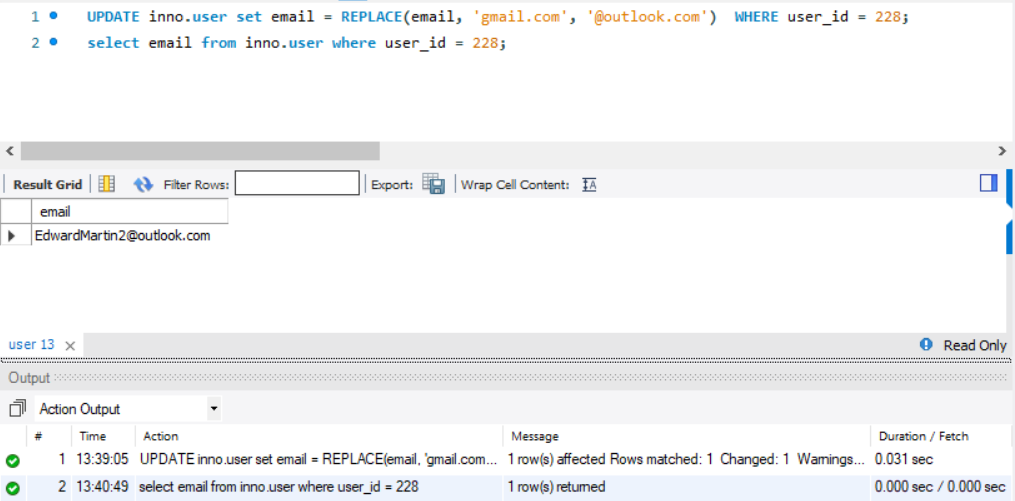


Рисунок 6.13 – Заміна даних в батьківській таблиці

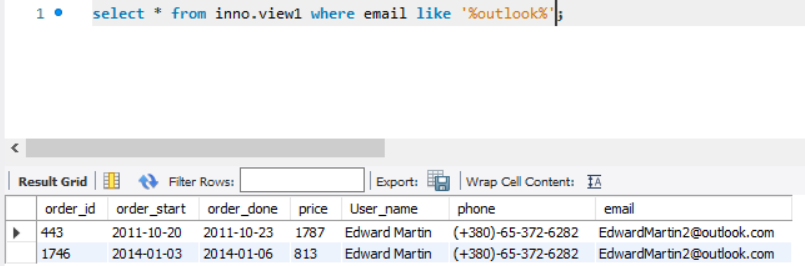


Рисунок 6.14 – Запит 6 (вивести замовлення у яких користувач має пошту з доменом outlook)

**6.3.3** Код SQL-запиту CREATE VIEW, який створює уявлення VIEW 2 на основі базового уявлення VIEW 1 і використовує алгоритм MERGE;

Висновок:

Набуто практичних навичок з розробки збережених процедур та функцій, що забезпечують бізнес-функції високонавантаженої системи зберігання даних. Опрацьовано всі типи операторів циклу, створено 20 збережених процедур, 16 функцій. Створені процедури та функції, що приймають та не приймають параметри. Досліджено вивід даних з процедури за допомогою глобальних змінних зі знаком @. Виконано створення 4 тимчасових таблиць з їх видаленням. Тимчасові таблиці побудовані як власноруч так і на основі інших таблиць (процедура calculate\_price). Засвоєно роботу з курсорами, що використовується в кожній процедурі з циклом.