МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

КАФЕДРА СИСТЕМОТЕХНІКИ

Звіт

З практичної роботи №5

На тему: «Створення процедур і функцій 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 збережених процедур (Stored Procedures) і функцій (Stored Functions) серверної частини високонавантаженої інформаційної системи;

– набуття практичних навичок з розробки SQL-запитів на вибірку й модифікацію даних, що використовуються в збережених процедурах і функціях, для забезпечення основних бізнес-процесів високонавантаженої інформаційної системи;

– набуття практичних навичок з розробки та використання курсорів (Cursors) у збережених процедурах;

– формування необхідних практичних умінь для аналізу плану виконання SQL-запитів за допомогою оператора EXPLAIN;

– формування необхідних практичних умінь для створення збережених процедур і функцій, з урахуванням особливостей роботи високонавантаженої інформаційної системи зберігання даних.

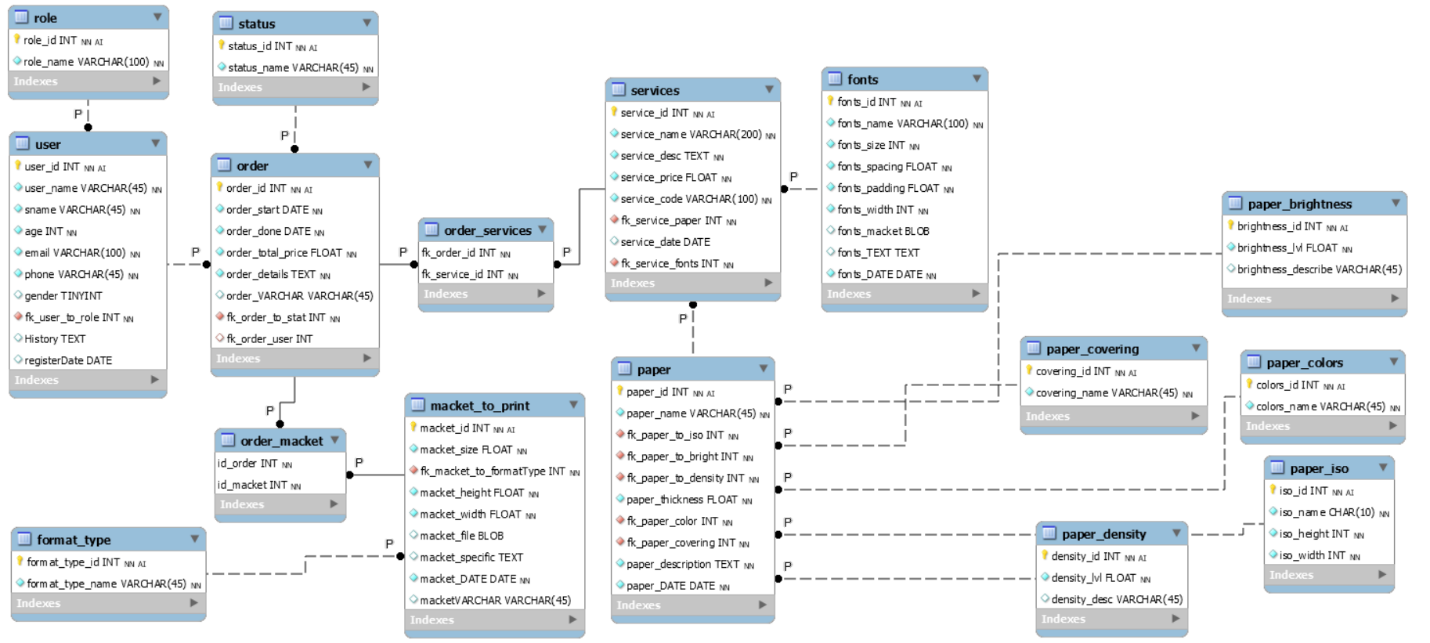
**Тема індивідуального завдання:** Інформаційна система «Надання послуг типографії»

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

Таблиця 5.1. – Порівняльний аналіз можливостей процедур та функцій

|  |  |  |  |
| --- | --- | --- | --- |
| № | Аналізовані параметри | Процедура | Функція |
| 1. | Синтаксис. Повернення результату. | Запит SELECT, SET @var – призначення значень глобальним змінним, RETURN. | RETURNS{STRING|INTEGER|REAL|DECIMAL} |
| 2. | Синтаксис. Формальні й фактичні параметри. | IN, OUT, INOUT | IN |
| 3. | Синтаксис. Виклик. | CALL sp\_name([parameter[,...]]) CALL sp\_name[()] | SELECT col\_1, col\_2, fuction\_name([param] ) FROM database.table; |
| 4. | Результат, що повертається (результуюча множина, одиночне значення тощо) | результуюча множина | одиночне значення |
| 5. | Можливі операції (операції над даними, створення набору даних тощо) | умовні оператори, цикли, курсори, підзапити, агрегатні функції | арифметичні операції, аг |
| 6. | Підтримують наступні операції (зазначити: транзакції, SQL інструкції SELECT, UPDATE, CURSOR тощо) | SELECT, CREATE, UPDATE, DELETE, INSERT, CASE, IF-ELSE, CASE |  |
| 7. | Взаємний виклик (процедура-функція, функція-процедура) | ТАК | НІ |
| 8. | Статус програмного об’єкта (глобальний – для всіх БД,локальний – тільки для однієї БД тощо) | локальний | локальний |
| 9. | Порядок виконання сервером MySQL. | компілюються один раз під час виклику на стороні сервера | компілюються постійно під час виклику |
| 10. | Фізичне зберігання (шлях, імена файлів та їх розширення) | в БД існує таблиця proc. Файли зберігаються як псевдокод за шляхом <datadir> /mysql/proc.MYI и <datadir> /mysql/proc.MYD. |  |

Процедура вызыв из триггера

Таблиця 5.2 – Порівняльний аналіз можливостей процедур і тригерів

|  |  |  |  |
| --- | --- | --- | --- |
| № | Аналізовані параметри | Процедура | Тригер |
| 1. | Синтаксис. Повернення результату. |  | RETURNS{STRING|INTEGER|REAL|DECIMAL} |
| 2. | Синтаксис. Формальні й фактичні параметри. | IN, OUT, INOUT | IN |
| 3. | Синтаксис. Виклик. | CALL sp\_name([parameter[,...]])CALL sp\_name[()] | CALL |
| 4. | Результат, що повертається (результуюча множина, одиночне значення тощо) | результуюча множина |  |
| 5. | Можливі операції (операції над даними, створення набору даних тощо) |  |  |
| 6. | Підтримують наступні операції (зазначити: транзакції, SQL інструкції SELECT, UPDATE, CURSOR тощо) | + |  |
| 7. | Взаємний виклик (процедура-тригер, тригер-процедура) | + | + |
| 8. | Статус програмного об’єкта (глобальний – для всіх БД, локальний – тільки для однієї БД тощо) |  |  |
| 9. | Порядок виконання сервером MySQL. |  |  |
| 10. | Фізичне зберігання (шлях, імена файлів та їх розширення) |  |  |

Таблиця 5.3 – Переваги використання процедур (функцій)

|  |  |  |  |
| --- | --- | --- | --- |
| № | Аналізовані параметри | Процедура | SQL - запит |
| 1 | Принцип виконання SQL-коду в СУБД MySQL | Компілюється один раз, викликається з кешу. Виконують функції бізнес-логіки | Компілюється при кожному виклику. Виконують функції розрахунків |
| 2 | Вплив на апаратні ресурси сервера | Збільшує (більша частина виконується на стороні сервера) |  |
| 3 | Вплив на мережний трафік | Скорочує |  |
| 4 | Можливість, що-небудь автоматизувати | Так |  |
| 5 | SQL-синтаксис (обмеження) | Міграція на іншу СУБД може призвести до проблем |  |
| 6 | Принцип зберігання SQL-коду |  |  |

Завдання 5.1.

– таблицю з переліком 6–9 функцій та 6-9 процедур для бізнес-функцій високонавантаженої системи.

Таблиця 5.4 – Процедури та функції високонавантаженої системи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ім’я | Призначення процедури (функції) | Взаємозв’язок | Використовувані таблиці |
| 1 | buisness\_change\_status | Виконує зміну статусу замовлення на наступне | не використовує | inno.order |
| 2 | report\_user | Створює звіт про користувачів | не використовує | inno.user, inno.order |
| 3 | p\_get\_format\_type | Виводить інформацію про макети та їх формати | використовує функцію get\_format\_type | inno.macket\_to\_print |
| 4 | get\_format\_type | Повертає значення формату макету | викликається процедурою p\_get\_format\_type | inno.format\_type |
| 5 | p\_get\_order\_by\_status | Повертає всі замовлення за назвою заданого статусу | використовує функції get\_statusid\_by\_name,  fullname | inno.status, inno.order, inno.user |
| 6 | get\_statusid\_by\_name | Повертає номер статусу за заданим ім’ям | викликається процедурою p\_get\_order\_by\_status | inno.status |
| 7 | fullname | Повертає конкатеноване значення | викликається процедурою p\_get\_order\_by\_status |  |
| 8 | p\_get\_service\_by\_pname\_fname | Отримати назву послуги, що використовує задані типи паперу, шрифту | викликає функціїї get\_paperid\_by\_name, get\_fontsid\_by\_name | inno.services, inno.paper, inno.fonts |
| 9 | registered\_users | Отримати користувачів разом з їх статусом | використовує функцію fullname | inno.user, inno.role |

– створення вісім (8) збережених функцій, що використовують зв’язані базові таблиці БД (за допомогою тільки інструкції «WHERE», тільки інструкції «INER JOIN»,вкладеного запиту) і функції «CONCAT()», «GROUP\_CONCAT()»;

WHERE

1. Вивести всі замовлення та інформацію про користувача:

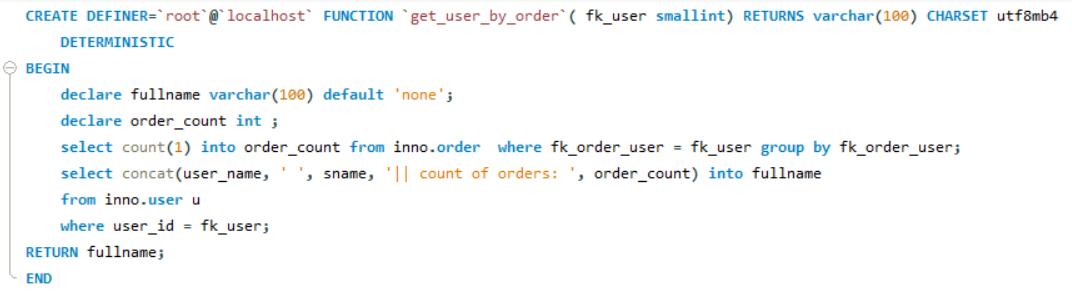


Рисунок 1. Створення функції

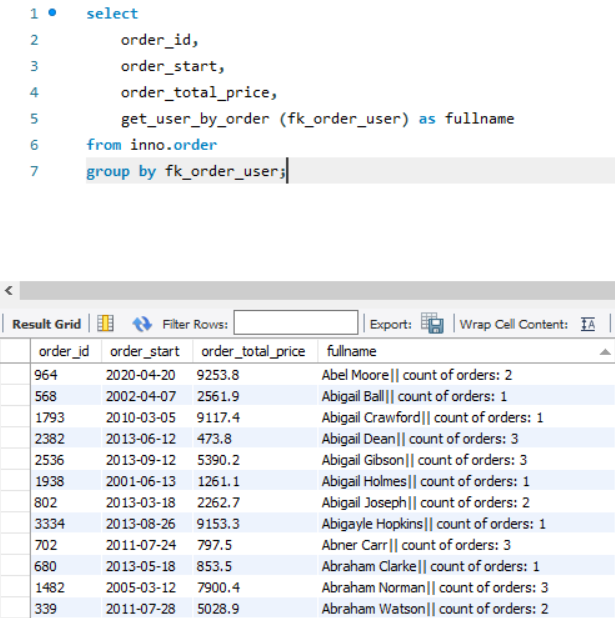


Рисунок 2. Виконання функції

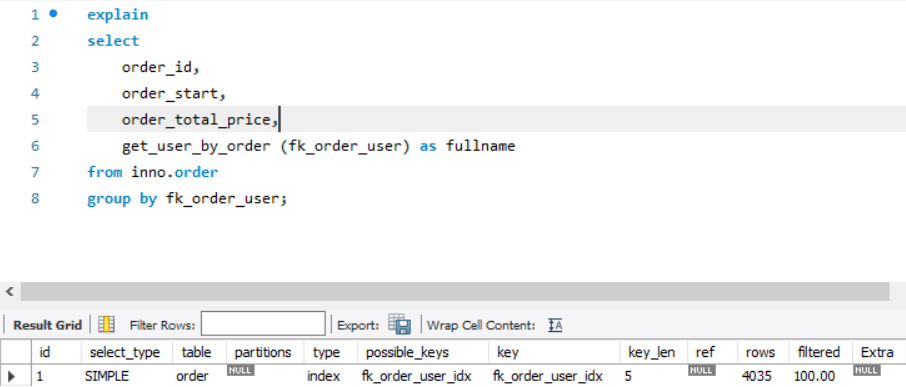


Рисунок 3. Аналіз запиту за допомогою EXPLAIN

Запит не може бути покращеним через необхідність виконання full table scan для виводу всіх значень.

JOIN

1. Вивести всі замовлення та інформацію про користувача:

– створення трьох (3) збережених процедур, які використовують (викликають) збережені функції;

1. Вивести інформацію про макети та їх формати

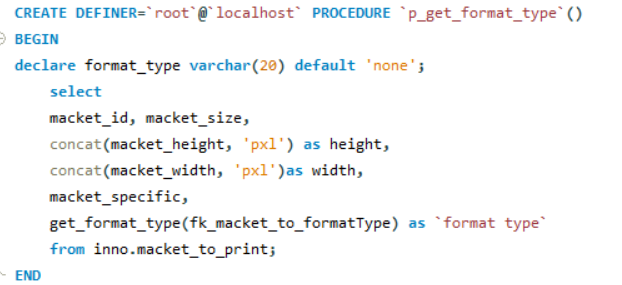


Рисунок 5.10 – Код процедури

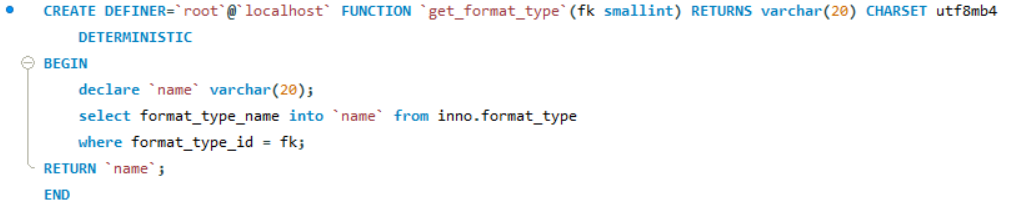


Рисунок 5.11 – Код функції, що викликаний процедурою

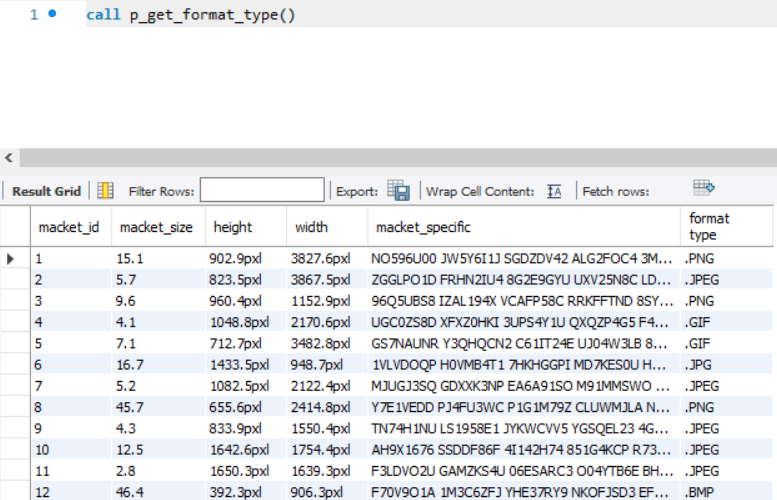


Рисунок 5.12 – Виклик процедури

2. Повернути всі замовлення за назвою статусу

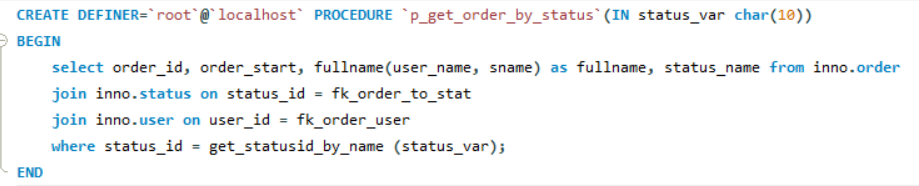


Рисунок 5.13 – Код процедури

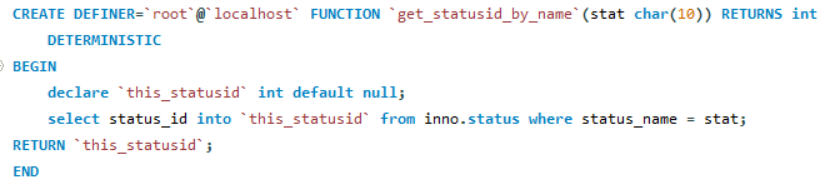


Рисунок 5.14 – Код функції, що викликається процедурою



Завдання 5.15 – Код функції, що викликається процедурою

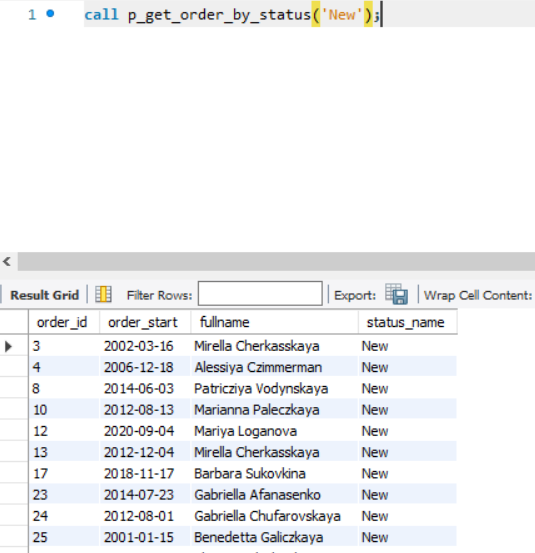


Рисунок 5.16 – Результат виклику процедури

3. Вивести назву послуги за заданим типом паперу та шрифту

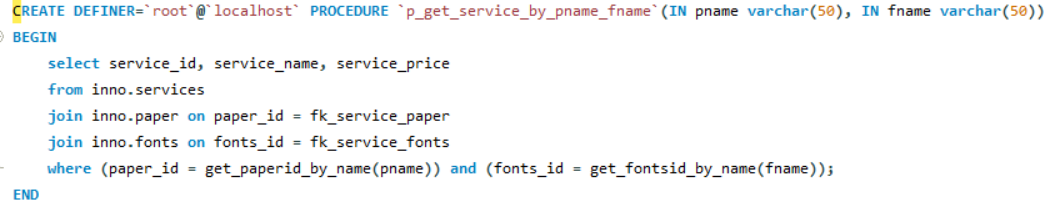


Рисунок 5.17 – Код збереженої процедури

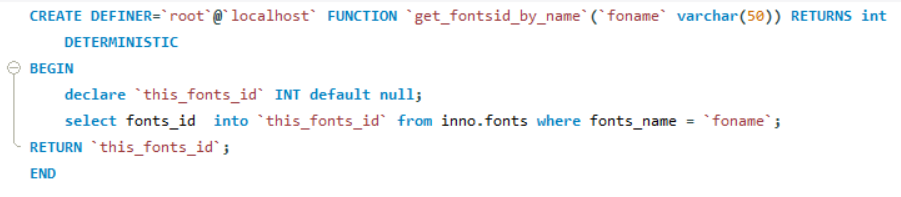


Рисунок 5.18 – Код функції що викликається

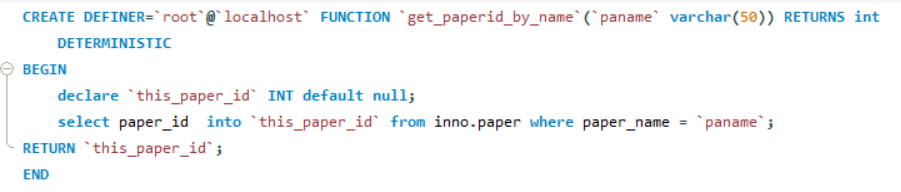


Рисунок 5.20 – Код функції що викликається

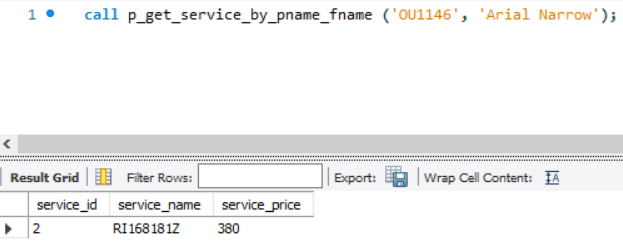


Рисунок 5.21 – Результат виконання виклику процедури

– створення трьох (3) збережених процедур, що використовують зв’язані базові таблиці БД (за допомогою тільки інструкції «WHERE», тільки інструкції «INER JOIN», вкладеного запиту) і функції «CONCAT()», «GROUP\_CONCAT()»;

WHILE

1. Вивести користувачів та їх ролі

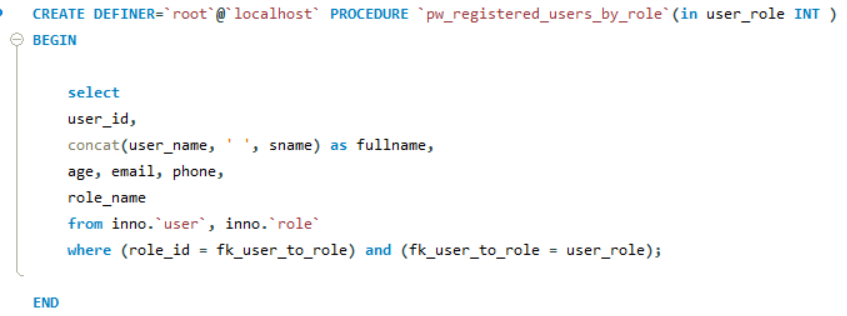


Рисунок Т – Створення процедури

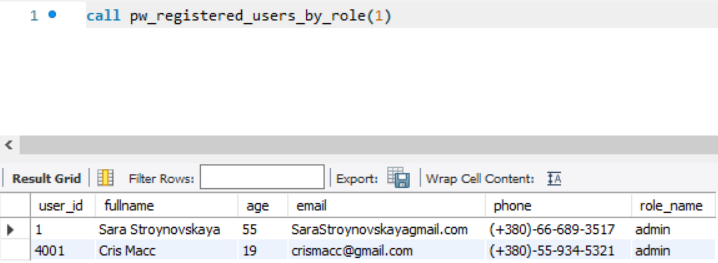


Рисунок Т – Результат виклику функції

2. Вивести всі завершені замовлення

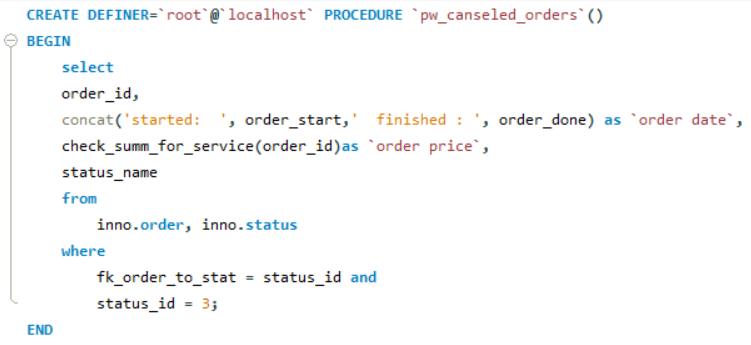


Рисунок Т – Створення процедури

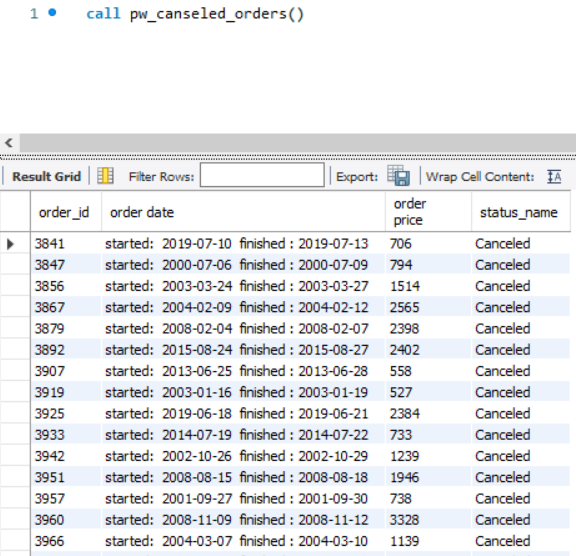


Рисунок Т – Виклик процедури

3. Вивести всі послуги, що включають папір чорного кольору. Надати інформацію щодо розміру паперу та його формат.



Рисунок Т – створення процедури

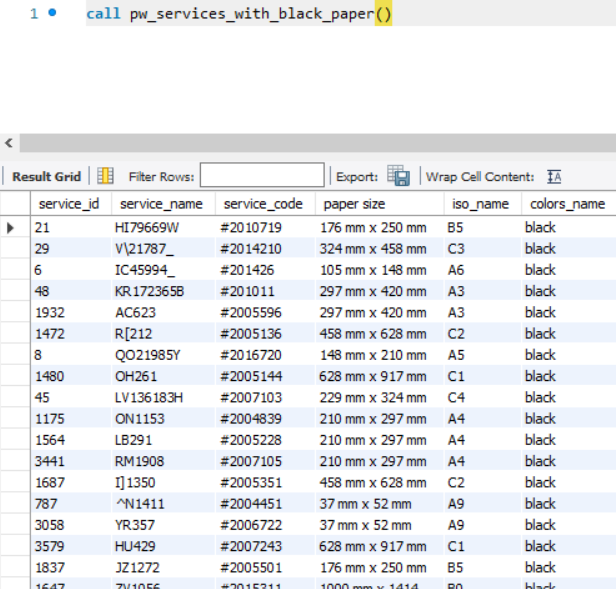


Рисунок Т – Виклик процедури

JOIN

1. Вивести користувачів та їх ролі

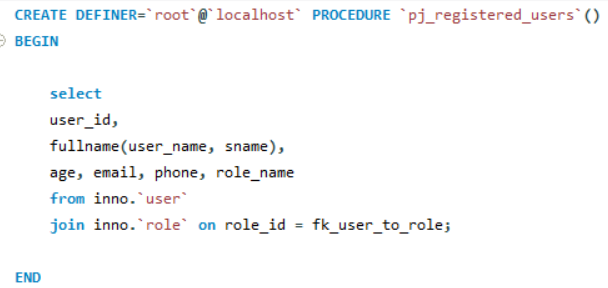


Рисунок Т – Створення процедури

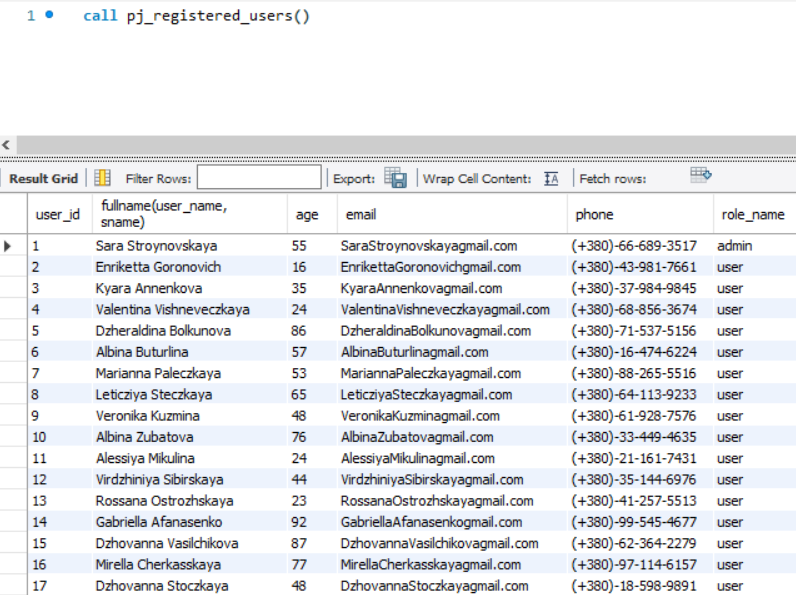


Рисунок Т – Виклик процедури

2. Вивести 10 найбільш великих файлів на друк, наданих користувачами,

відсортованих за розміром з наданням формату файлу.

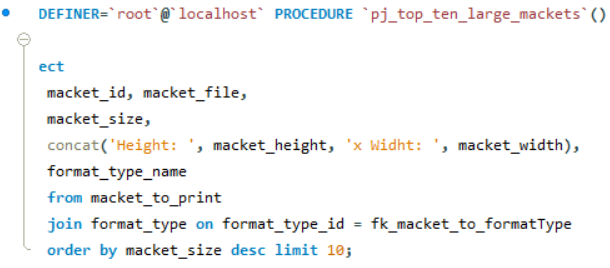


Рисунок Т – Створення процедури

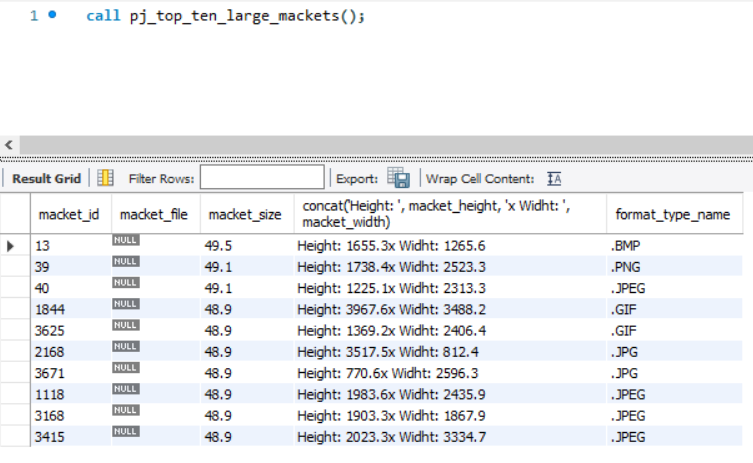


Рисунок Т – Виклик процедури

– створення трьох (3) збережених процедур, що використовують оператори «IF...THEN...ELSEIF», «CASE v.1», «CASE v.2»;

1. Визначити розмір знижки для користувачів системи. Якщо зареєстрований 10 років потому – 10% знижки, якщо 5 років – 5%. Зареєстровані менш ніж 5 років тому – не отримують знижку

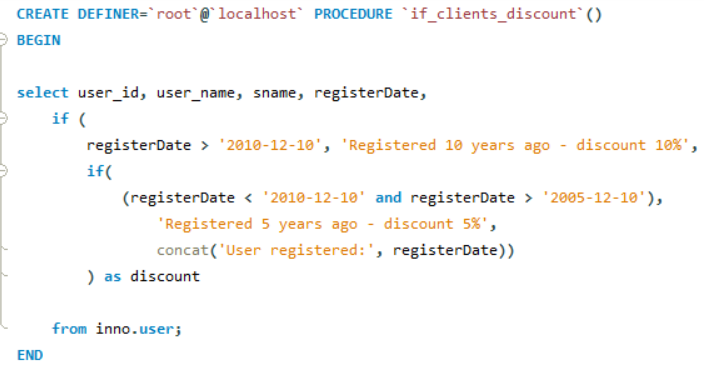


Рисунок Т – Створення процедури

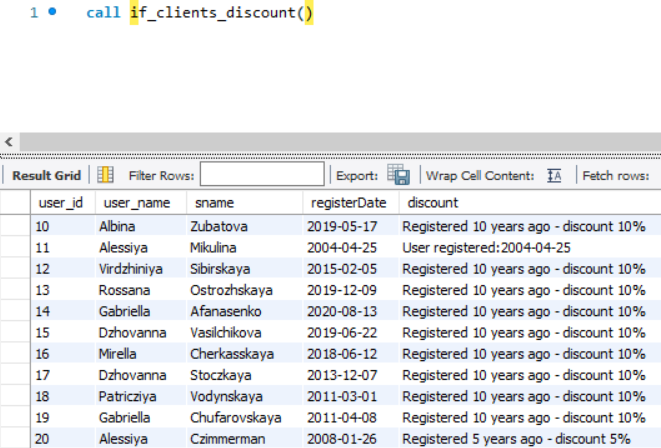


Рисунок Т – Використання процедури

2) Відобразити в замовленнях користувачів, які видалили профіль.

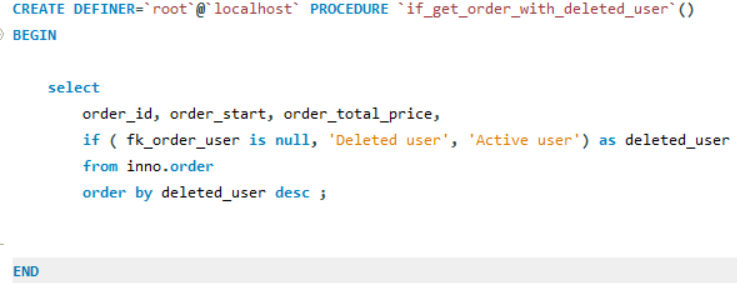


Рисунок Т – Створення процедури

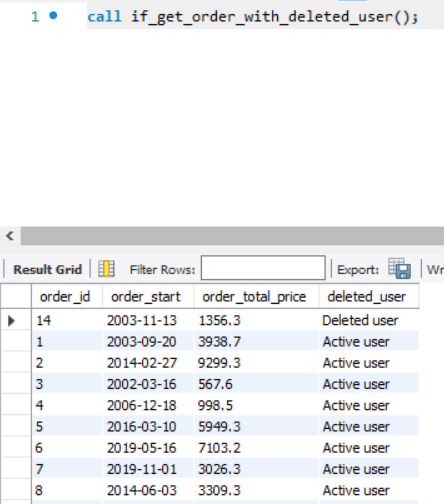


Рисунок Т – Використання процедури

3) Відобразити дані про папір та його тип покриття (обмежено до 5)



Рисунок Т – створення процедури

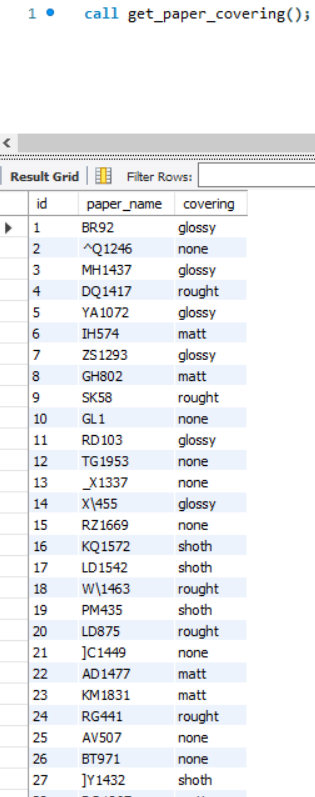


Рисунок Т – Результат виклику процедури

– створення однієї (1) збереженої процедури, яка використовує (створює й видаляє) тимчасову таблицю;

Призначення створеної процедури: дана процедура автоматично розраховує значення кінцевої ціни замовлення, виходячи з того, скільки послуг обрав користувач, враховуючи їх ціну.

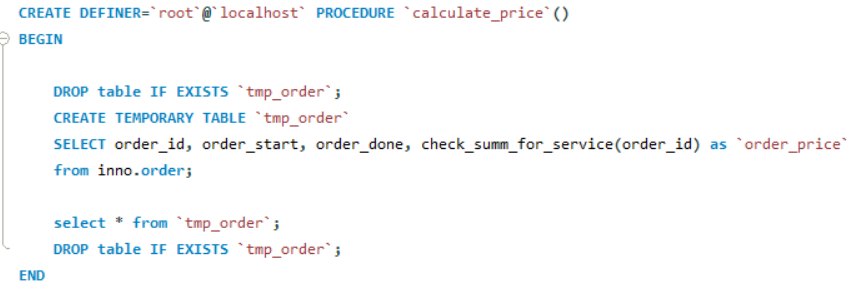


Рисунок Т – створення процедури

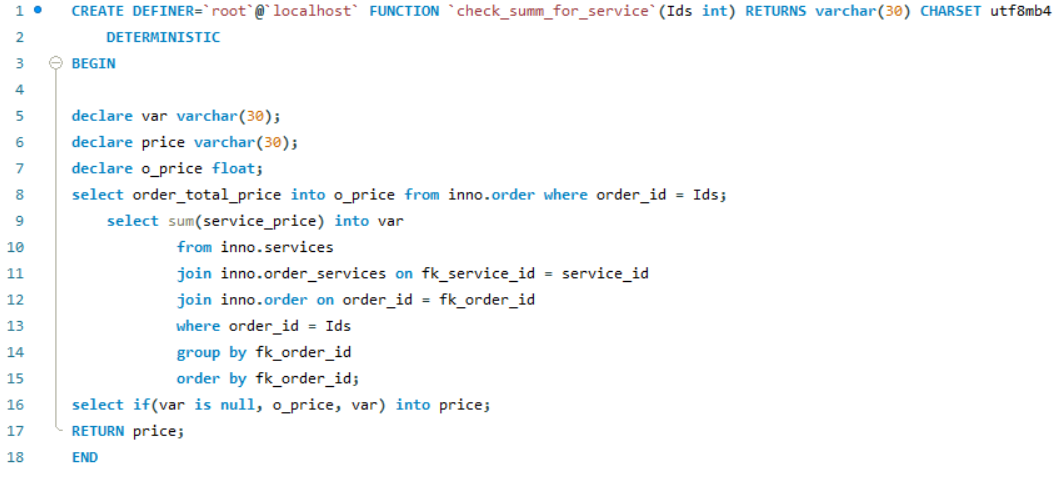


Рисунок Т – створення функції для розрахунку ціни

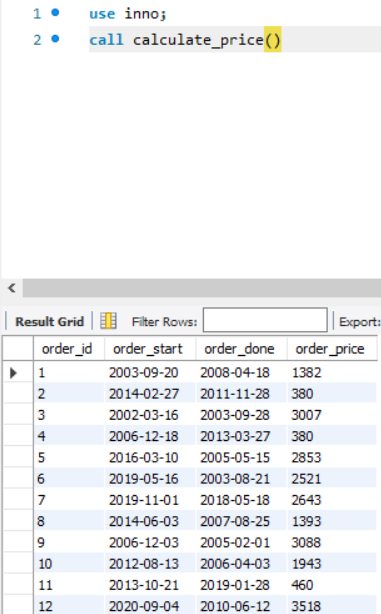


Рисунок Т – Результат виклику

– створення трьох варіантів однієї збереженої процедури, яка використовує тимчасову таблицю й курсор. Кожен варіант має реалізувати один з операторів циклу:1) «WHILE», 2) «LOOP», 3) «REPEAT».

1. LOOP



Рисунок Т – Створення процедури з LOOP

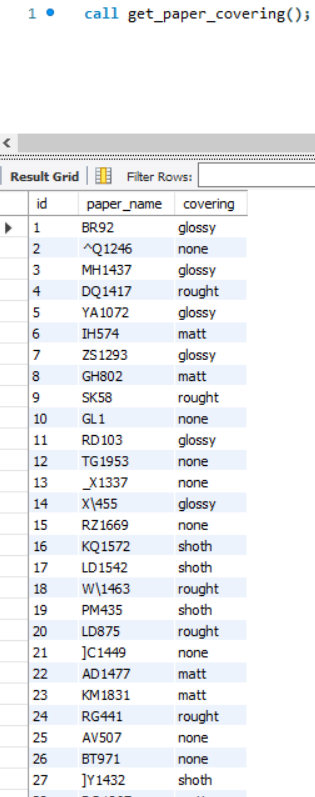


Рисунок Т – Результат виклику процедури

2. WHILE



Рисунок Т – Створення процедури

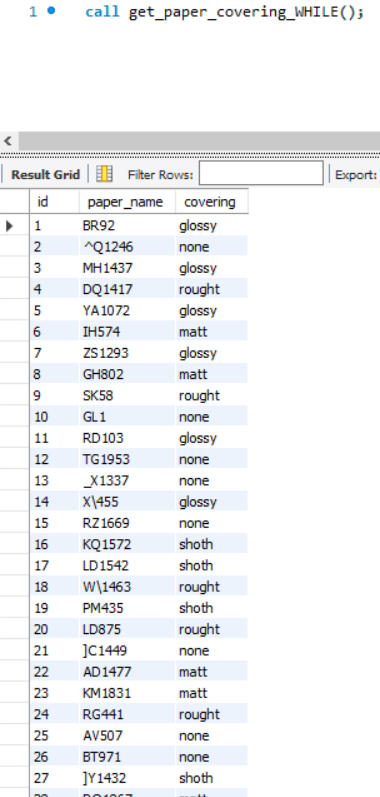


Рисунок Т – Виклик процедури

3 REPEAT



Рисунок Т – Створення процедури

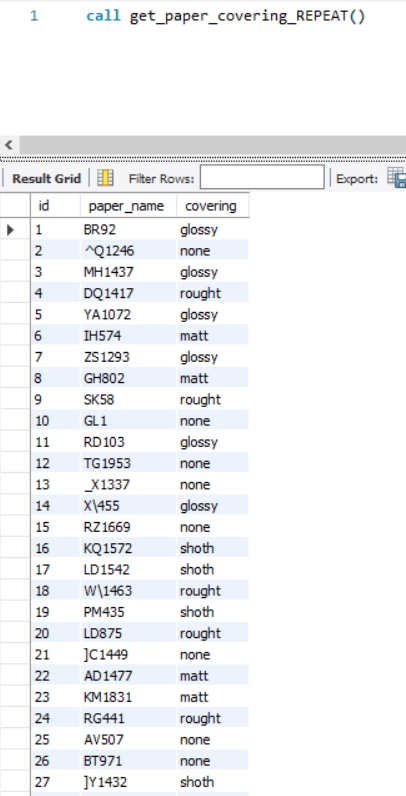


Рисунок Т – Виклик процедури

1. Зміна статусу замовлення на наступний:

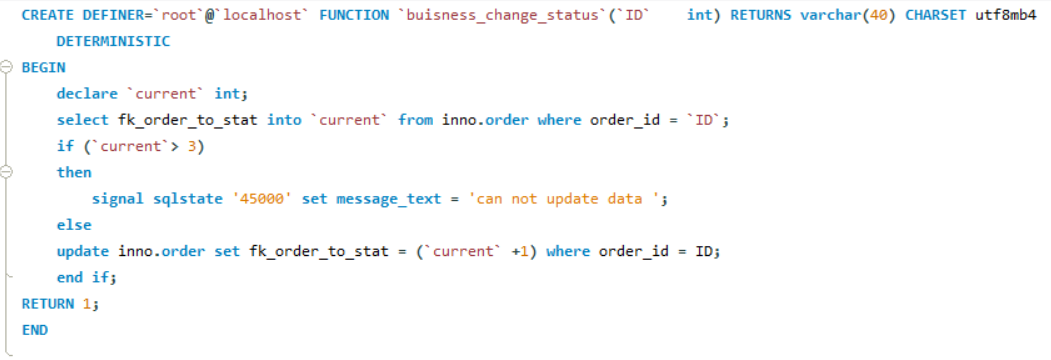


Рисунок 5.1 – Код збереженої функції

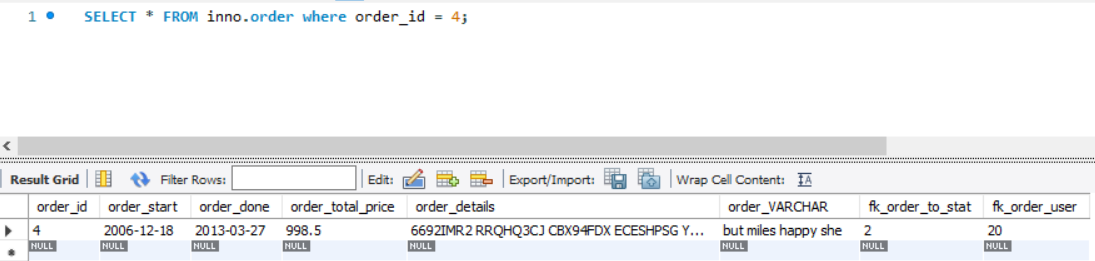


Рисунок 5.2 – Вибірка даних до виконання

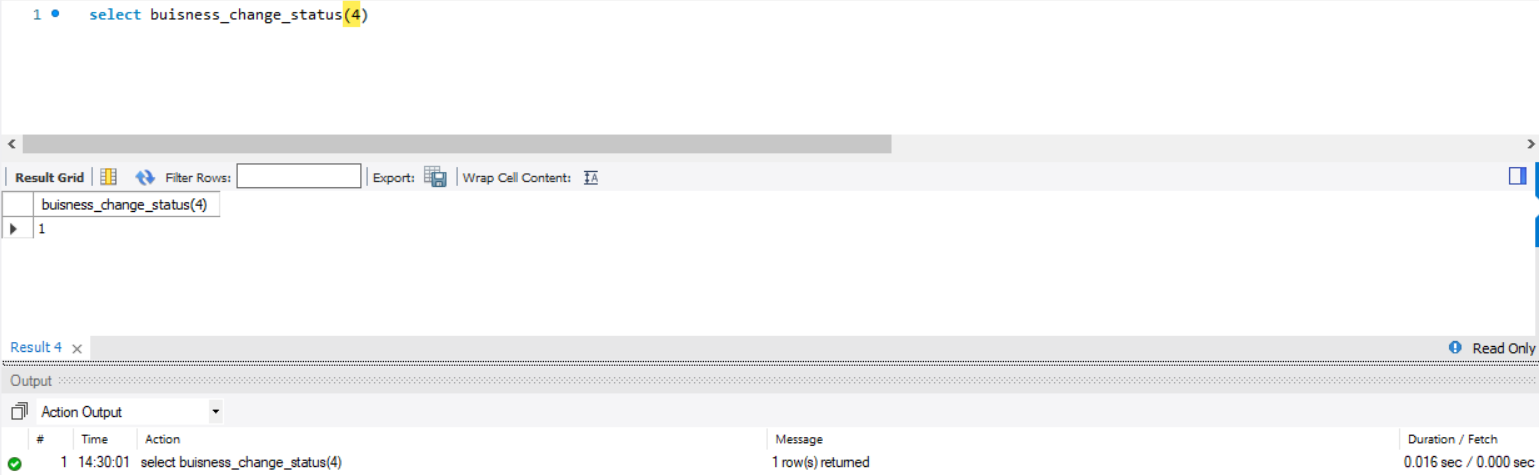


Рисунок 5.3 – Виклик функції

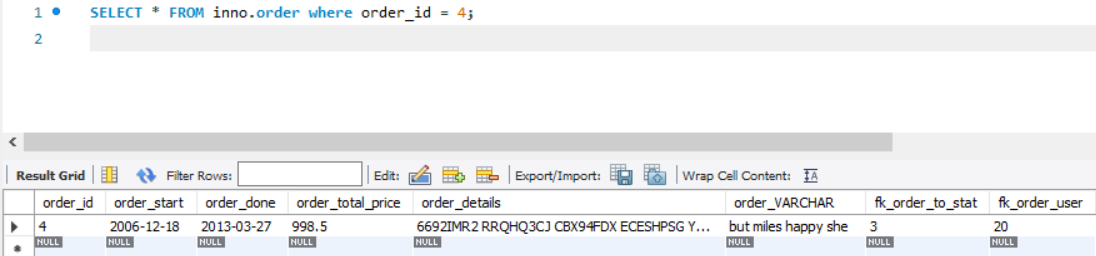


Рисунок 5.4 – Результат виконання

Обробка помилок:

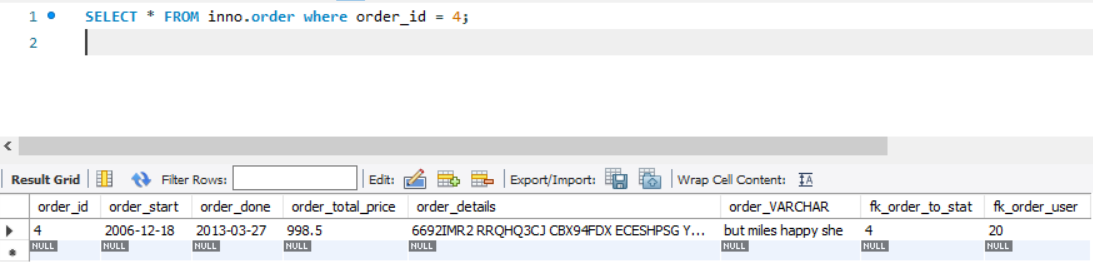


Рисунок 5.5 – Вибірка до виклику функції

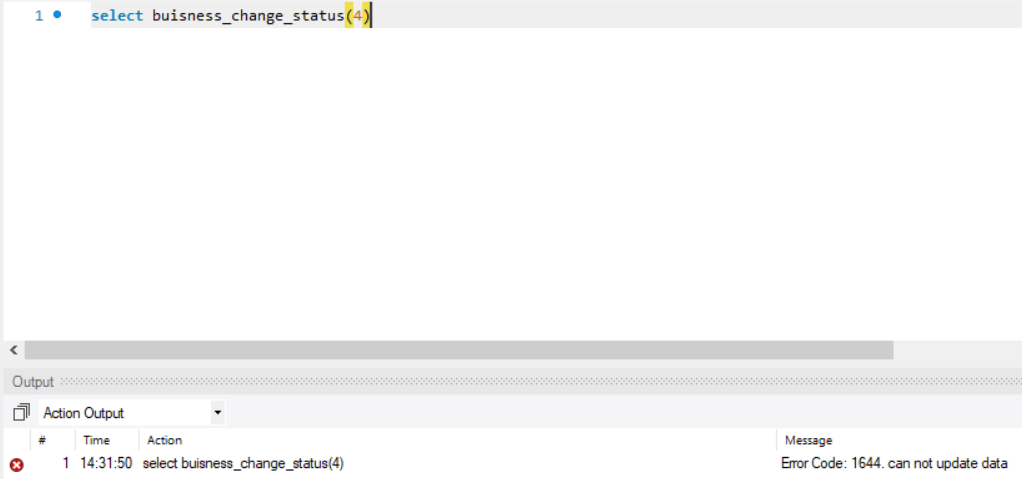


Рисунок 5.6 – Обробка помилки системою

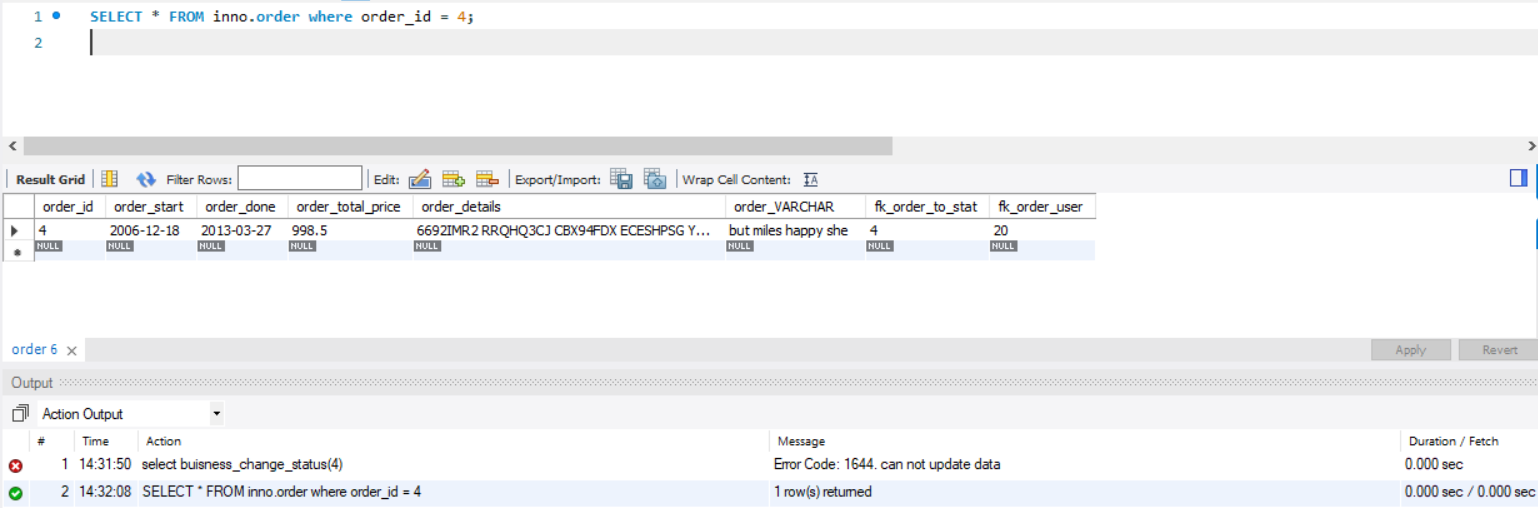


Рисунок 5.7 – Вибірка після помилки

2. Створення звіту по користувачам

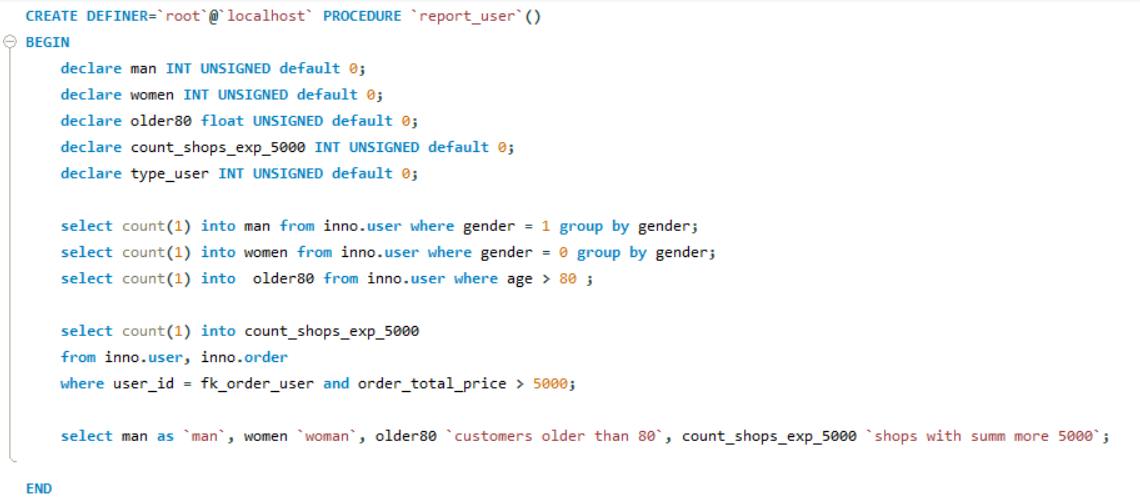


Рисунок 5.8 – Код збереженої процедури

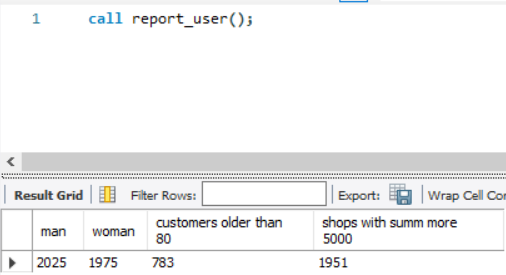


Рисунок 5.9 – Виклик функції

Висновок:

Набуто практичних навичок з розробки тригерів для підтримки цілісності зв’язків, модифікації даних і забезпечення основних бізнес-процесів високонавантаженої інформаційної системи. Сформовано необхідні практичні уміння для аналізу плану виконання SQL-запитів за допомогою оператора EXPLAIN. Сформовано необхідні практичні уміння для створення тригерів, з урахуванням особливостей реалізації логіки роботи інтерфейсу високонавантаженої інформаційної системи зберігання даних.