**Лабораторна робота 11.**

**Розробка та використання збережених процедур.**

**Мета роботи:** навчитись створювати та виконувати збережені процедури, передавати вхідні та обробляти вихідні дані.

**Задачі:**

* Поняття збережених процедур та загальний синтаксис створення та виконання збереженої процедури
* Повернення із збереженої процедури набору рядків
* Використання вхідних параметрів
* Використання вихідних параметрів та інші види повернення даних
* Виконання індивідуального завдання згідно варіанту

1. Нерідко операція з даними представляє набір інструкцій, які необхідно виконати в певній послідовності. Наприклад виконується деяка перевірка, в результаті якої видається інструкція на виконання однієї чи іншої операції. Або при зміні одних даних, також мають бути змінені і інші. Або перед записом значення мають бути обчислені за складним алгоритмом на основі вхідних даних. На цей випадок в T-SQL передбачено об’єкт, що має назву «збережена процедура» («stored procedure»). Даний об’єкт дозволяє зберегти на сервері набір інструкцій мовою T-SQL, що буде мати унікальне ім’я та виконуватись у наведеній послідовності кожного разу при зверненні до нього.

Використовуючи механізм збережених процедур, ми розміщуємо логіку нашого застосунку на сервері, внаслідок чого: спрощується обслуговування коду, так як зміни у алгоритмі роботи можна виконувати не торкаючись клієнтський застосунків; знижується навантаження на мережу, оскільки окремі команди не обмінюються даними по мережі, весь пакет виконується на сервері, після чого остаточний кінцевий результат передається на клієнта; та застосовується концепція модульного проектування, внаслідок чого спрощується реалізація складних механізмів, що складаються з окремих операцій, використовуваних багаторазово в різних місцях коду.

1. Перед тим, як розглядати приклади, наведемо основні частини загального синтаксису створення та виконання збереженої процедури. Процедура створюється командою create procedure (рис.1)

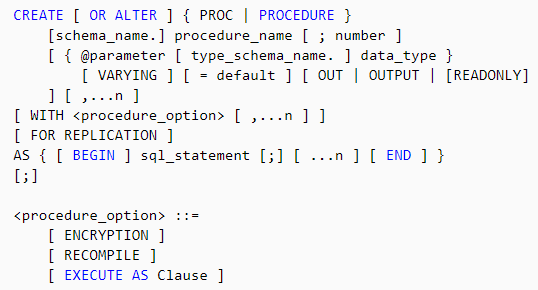


Рис.1. Синтаксис команди create procedure

Отже, при створенні процедури ми вказуємо ім’я (можна із ім’ям схеми та індексом) та наводимо перелік вхідних та вихідних параметрів із типами даних та значеннями за замовченням. Також можуть бути вказані додаткові опції для шифрування, перекомпіляції при кожному запуску (наприклад для тестування продуктивності) і запуску від імені (та з правами) іншого користувача. Далі слідує безпосередньо тіло процедури.

Тепер розберемо загальний синтаксис команди execute, що відправляє процедуру на виконання (рис.2)

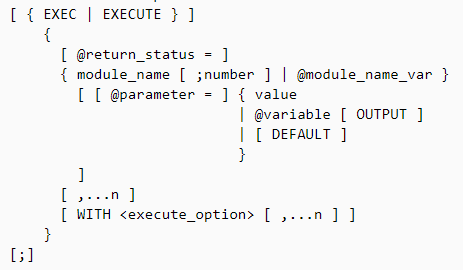


Рис.2. Синтаксис команди execute

При виконанні вказуємо ім’я процедури, та її параметри у разі необхідності. Також можемо у разі потреби обробляти код завершення.

Перед тим, як почати розглядати приклади, зафіксуємо ще одну річ. Збережена процедура може не повертати взагалі нічого (якщо вона виконує лише операції зміни даних), або повертати стандартний набір рядків, код завершення чи вихідні параметри (є ще і інші варіанти повернення значень, але їх набагато складніше обробити з боку клієнтського застосунку)

1. Тепер почнемо розглядати дані команди на прикладах, поступово нарощуючи їх складність. Для початку створимо процедуру, що не має жодних вхідних чи вихідний параметрів та повертає у вигляді набору рядків список групи 401 (рис.3)

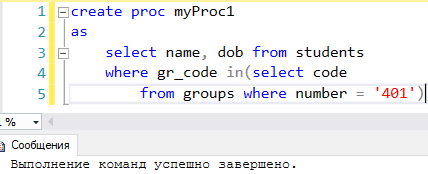


Рис.3. Процедура без параметрів

Виконаємо створену процедуру (рис.4)

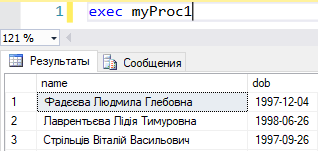


Рис.4. Виконання процедури без параметрів

Змінимо процедуру, додавши вхідний параметр номер групи, з якої ми хочемо вибирати студентів. При зміні ключове слово create змінюється на alter (рис.5)

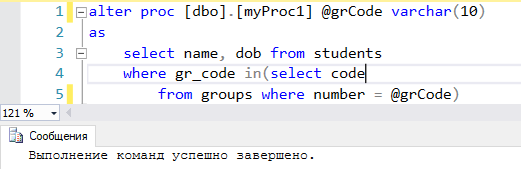


Рис.5. Додавання вхідного параметру

При виконанні процедури тепер необхідно вказати значення параметру (рис.6)

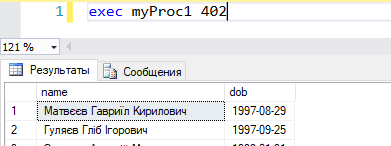


Рис.6. Виконання процедури з параметром

При вказанні у процедурі параметру йому можна вказати значення за замовченням (рис.7)

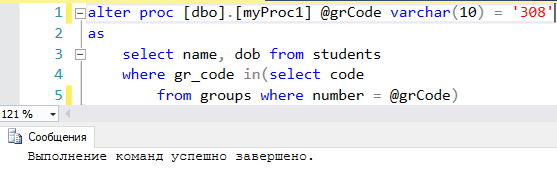


Рис.7. Вказання параметру значення за замовченням

Тепер при виконанні процедури його вказувати не обов’язково. Наступний приклад виведе студентів 308 групи (рис.8)

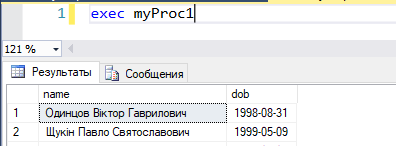


Рис.8. Виконання процедури із значенням параметру за замовченням

Додамо до нашої процедури ще декілька вхідних параметрів із значеннями за замовченням (рис.9)

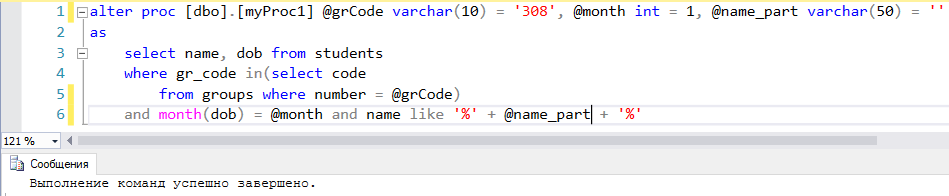


Рис.9. Процедура із трьома вхідними параметрами

Розглянемо приклад виконання даної процедури, у якому ми передамо тільки 2-й та 3-й параметри, залишивши перший за замовченням. Для цього скористаємось можливістю явного вказання імені параметру (рис.10)

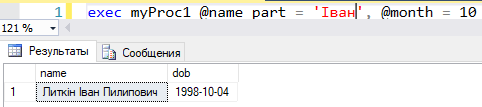


Рис.10. Передача параметрів за іменами

Для передачі параметрів також можна використовувати змінні, проілюструємо це на прикладі, що за допомогою процедури (рис.9) виберає студентів 401 групи із іменем Юлія та датою народження у травні місяці (рис.11).

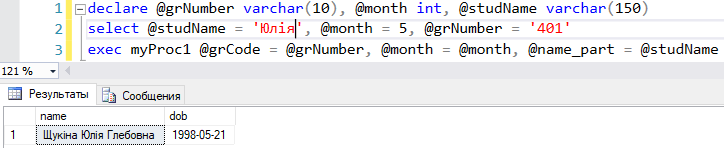


Рис.11. Передача параметрів через змінні

1. Повертатись значення із процедури можуть також у вигляді вихідних параметрів. Для цього необхідно при вказанні імені та типу параметру вказати ключове слово output. Реалізуємо процедуру, схожу на попередню, але вона буде повертати не список студентів у вигляді стандартного набору рядків, а просто кількість таких студентів (рис.12)

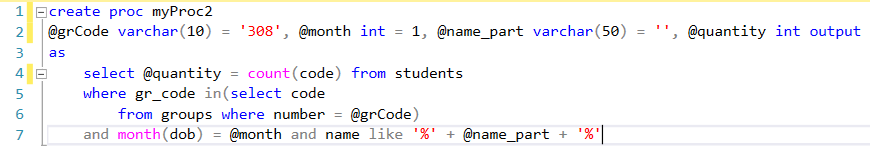


Рис.12. Процедура із вихідним параметром

При виконанні такої процедури нам уже не обійтися без використання змінних, оскільки вхідні параметри можуть бути проініціалізовані безпосередньо значеннями, а вихідний параметр має бути передано до змінної. Також обов’язково при передачі вихідного параметру треба вказати ключове слово output. Наведемо приклад виконання попередньої процедури, передавши частину вхідних параметрів у вигляді значень, а частину із використанням змінних, та обробивши вихідний параметр (рис.13)

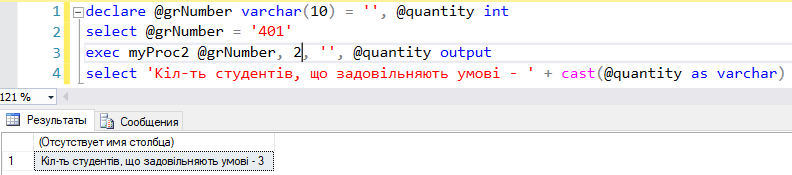


Рис.13. Виконання процедури із вихідним параметром

Якщо при визові процедури на вказати для вихідного параметру опцію output, у ній залишиться значення, що було записане перед цим (рис.14). Бачимо, що код повертає «кількість студентів - 0», незважаючи на те, що ми не змінили ніяких умов, а насправді їх 3, як і в попередньому прикладі.

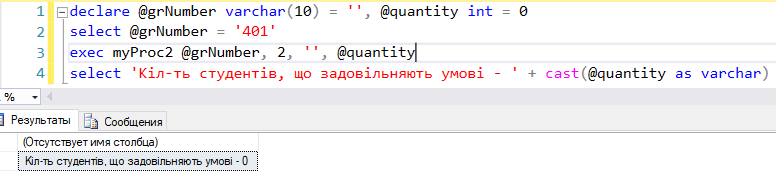


Рис.14. Помилкова обробка вихідного параметру

1. Розглянемо можливість повернення із процедури коду завершення та обробку даного коду при виклику процедури. У збереженій процедурі можливо використати команду return, що завершує виконання процедури. Дана команда має необов’язковий числовий параметр, що і повертається є якості коду завершення процедури. Якщо параметр не був вказаний явним чином, або команда return не була використана взагалі, у якості коду завершення повертається 0.

Тепер перейдемо до прикладу. Реалізуємо процедуру на основі попередньої, що також буде повертати кількість студентів у вказаній групі, із датою народження у вказаному місяці та що містять частину тексту у своєму ПІБ. Але процедура крім цього буде виконувати додатково перевірку, чи існує взагалі така студентська група, і якщо так, повертати 1, інакше 0 (рис.15)

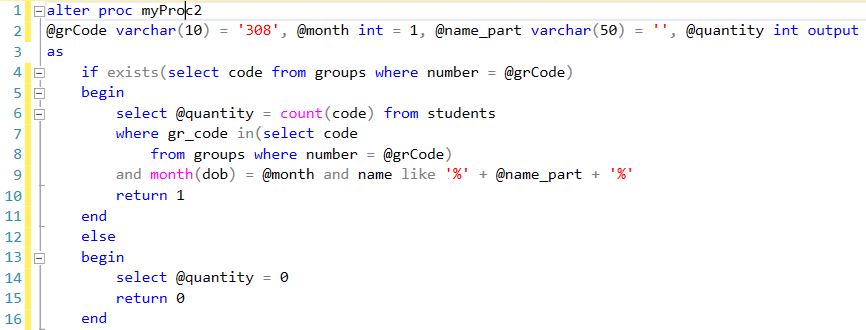


Рис.15. Процедура із кодом завершення

Наведемо код виклику даної процедури із обробкою коду завершення, який вона повертає (рис.16)

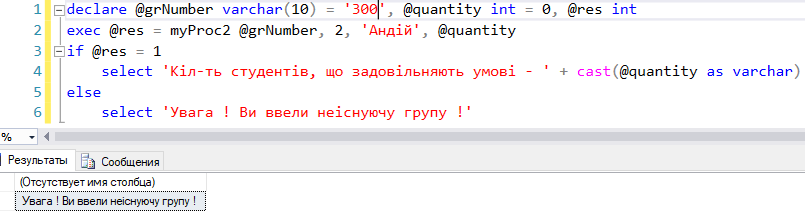


Рис.16. Обробка коду завершення збереженої процедури

1. Ще один варіант повернення даних їх процедури – це використання команд raiserror та print, при чому остання як правило використовується на етапі відлагодження коду. Реалізуємо процедуру, що видаватиме текст, переданий у вигляді вхідного параметру, у вигляді повідомлення або помилки (рис.17)

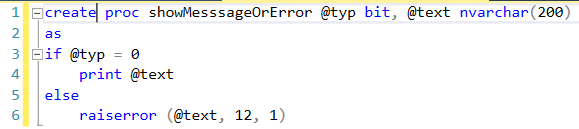


Рис.17. Приклад повернення даних через print та raiserror.

Нижче, на рис.18-19 наведемо результат роботи процедури при виведенні повідомлення та помилки відповідно.

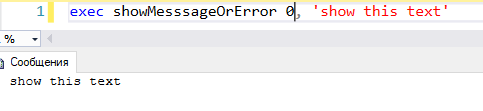


Рис.18. Виведення тексту, як повідомлення.

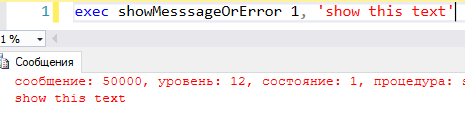


Рис.19. Виведення тексту, як помилки.

1. Наведемо ще декілька прикладів реалізації збережених процедур. Реалізуємо процедуру, що буде виконувати додавання нового викладача на кафедру. Вхідними параметрами будуть ПІБ викладача на назва кафедри. В процесі виконання виконуємо перевірку, чи є така кафедра (якщо немає, нічного не робимо, код завершення 501), та перевірку, чи є вже такий викладач (якщо є, нічного не робимо, код завершення 502). Додаємо викладача, та вихідним параметром повертаємо код викладача та кількість викладачів кафедри, що поки не прийняли жодного заліку або іспиту та код завершення 0 (рис.20)

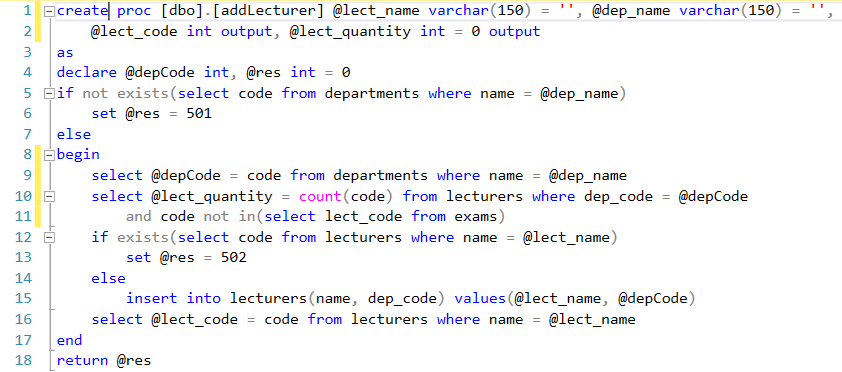
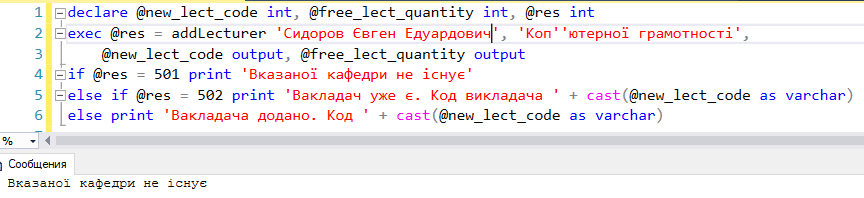
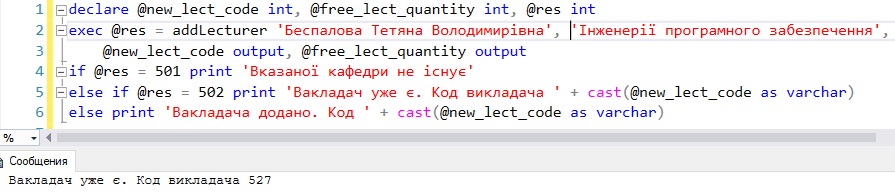


Рис.20. Процедура додавання нового викладача

Промоделюємо декілька ситуацій – додамо викладача у неіснуючу кафедру, додамо уже існуючого викладача, і нарешті додамо нового викладача (рис.21)





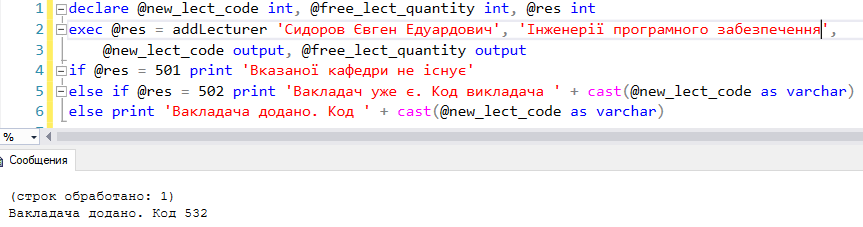


Рис.21. Виконання процедури додавання викладача із різними вхідними даними.

Розглянемо ще один приклад процедури, що використовуватиме курсори. Процедура додаватиме студентів із таблиці students\_to\_add, які ще не оброблялись, до таблиці students. Якщо кількість необроблених рядків у табличці students\_to\_add = 0, завершити процедуру із помилкою. Якщо немає студентської групи або студентів у групі більше 30, студент не додається. Після обробки рядку записати статус до поля state таблиці students\_to\_add. Спочатку наведемо вигляд таблиці students\_to\_add (рис.22)

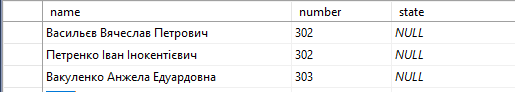
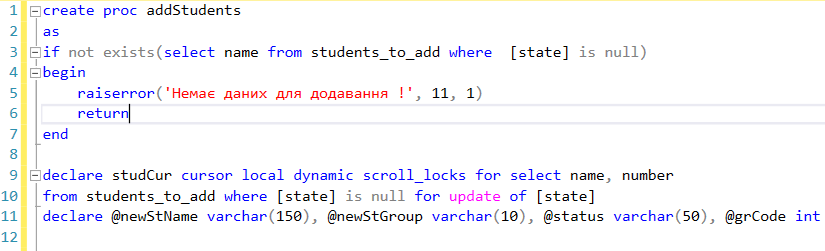
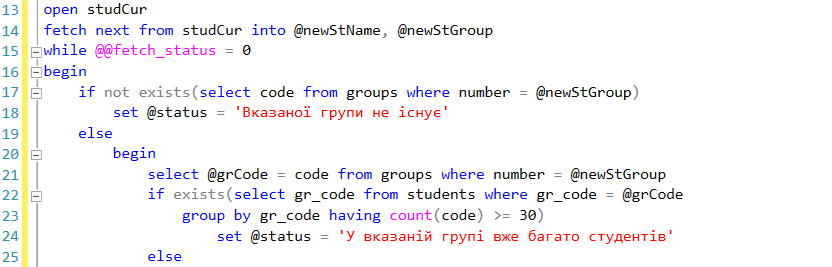


Рис.22. Таблиця із студентами для додавання

Далі наведемо код процедури (рис.23)





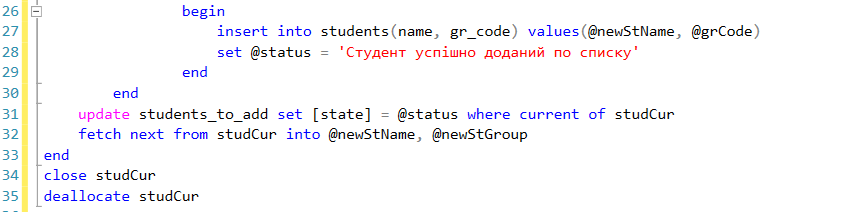


Рис.23. Процедура додавання списку студентів

Перевіримо роботу процедури. Виконаємо її та переглянемо, як змінилась таблиця students\_to\_add (рис.24)

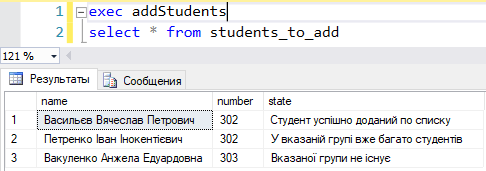


Рис.24. Таблиця студентів після відпрацювання процедури

1. Кінець роботи

**Завдання для індивідуального виконання.**

Згідно свого варіанту розробити збережену процедуру із вхідними та вихідними параметрами. Попередньо створити таблицю із вхідними даними та заповнити, що ілюструватимуть різні сценарії виконання процедури). Також використати курсори, код завершення та повернення помилки. Після обробки рядку записати статус операції до поля «статус рядку» вхідної таблиці.

* + Вхідний параметр 1: обробляємо всю вхідну таблицю або 1 рядок
  + Вхідний параметр 2: Код рядку, якщо обробляємо 1 рядок
  + Вхідний параметр 3: Обробляємо тільки необроблені або всі рядки вхідної таблиці
  + Вихідний параметр: кількість оброблених рядків
  + Код завершення: кількість доданих рядків

У якості базових взяти 2 останні приклади реалізації процедур в лабораторній роботі.

! УВАГА. Викладач має право зниження оцінки за використання «прототипу» збереженої процедури іншого варіанту.

**Варіанти для виконання індивідуальних завдань.**

1. БД Автостоянка. Процедура додавання автомобілів із вхідної таблиці (Код рядку таблиці, Номер авто, Прізвище власника, Назва стоянки, статус рядку). Не додаємо рядок, якщо немає такого власника, автостоянки, або вже є такий номер машини.
2. БД Облік кадрів. Процедура додавання співробітників із вхідної таблиці (Код рядку таблиці, Прізвище співробітника, Назва посади, Назва відділу, статус рядку). Не додаємо рядок, якщо немає такого відділу, посади, або вже є такий співробітник.
3. БД Облік книжок. Процедура додавання книжок із вхідної таблиці (Код рядку таблиці, Назва книжки, Назва видавництва, Назва типу книги, статус рядку). Не додаємо рядок, якщо немає такого видавництва, типу книги, або вже є така книга.
4. БД Лікарські рецепти. Процедура додавання рецептів із вхідної таблиці (Код рядку таблиці, Дата виписки, Прізвище лікаря, Назва лікарського засобу, статус рядку). Не додаємо рядок, якщо немає такого лікаря, лікарського засобу, або вже є такий рецепт.
5. БД Нерухомість. Процедура додавання квартир із вхідної таблиці (Код рядку таблиці, номер квартири, адреса будинку, прізвище власнику, статус рядку). Не додаємо рядок, якщо немає такої адреси, власнику, або вже є така квартира.
6. БД Облік комунального транспорту. Процедура додавання транспортних засобів із вхідної таблиці (Код рядку таблиці, номер або номерний знак транспортного засобу, назва типу, номер маршруту, статус рядку). Не додаємо рядок, якщо немає такого типу, маршруту, або вже є такий транспортний засіб.
7. БД Кінопрокат. Процедура додавання нових сеансів із вхідної таблиці (Код рядку таблиці, дата та час сеансу, назва кінотеатру, назва кінофільму, статус рядку). Не додаємо рядок, якщо немає такого кінотеатру, кінофільму, або вже є такий сеанс.
8. БД Ремонт телефонів. Процедура додавання нових пристроїв із вхідної таблиці (Код рядку таблиці, серійний номер пристрою, назва бренду, назва моделі, статус рядку). Не додаємо рядок, якщо немає такої моделі у вказаного бренду, або вже є такий пристрій.
9. БД Облік тварин ветеринарної клініки. Процедура додавання нових тварин із вхідної таблиці (Код рядку таблиці, прізвисько тварини, назва типу, прізвище власнику, статус рядку). Не додаємо рядок, якщо немає такого власнику, типу тварини, або вже є така тварина.
10. БД Розклад занять. Процедура додавання нових занять із вхідної таблиці (Код рядку таблиці, день тижня, номер пари, номер групи, назва дисципліни, статус рядку). Не додаємо рядок, якщо немає такої групи, дисципліни, або вже є заняття на цій парі.
11. БД Облік приміщень та обладнання університету. Процедура додавання нових приміщень із вхідної таблиці (Код рядку таблиці, номер приміщення, назва корпусу, назва типу, статус рядку). Не додаємо рядок, якщо немає такого корпусу, типу приміщення, або вже є таке приміщення.
12. БД Туристичної компанії. Процедура додавання нових замослень із вхідної таблиці (Код рядку таблиці, дата замовлення, назва туру, прізвище клієнта, статус рядку). Не додаємо рядок, якщо немає такого туру, клієнта, або вже є таке замовлення.
13. БД Регістратура поліклініки. Процедура додавання нових відвідувань із вхідної таблиці (Код рядку таблиці, дата відвідування, прізвище пацієнта, прізвище лікаря, статус рядку). Не додаємо рядок, якщо немає такого пацієнта, лікаря, або вже є таке відвідування.
14. БД Оренда транспортних засобів. Процедура додавання нових даних про оренду із вхідної таблиці (Код рядку таблиці, дата оренди, номер транспортного засобу, прізвище орендаря, статус рядку). Не додаємо рядок, якщо немає такого орендаря, транспортного засобу, або вже є такий запис про оренду.
15. БД Оренди приміщень. Процедура додавання нових даних про оренду із вхідної таблиці (Код рядку таблиці, дата оренди, адреса приміщення, прізвище орендаря, статус рядку). Не додаємо рядок, якщо немає такого орендаря, адреси, або вже є такий запис про оренду.