Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут прикладної математики та фундаментальних наук   
Кафедра прикладної математики

Звіт

про виконання лабораторної роботи №3

“Підпрограми СУБД Oracle”

з курсу “Організація баз даних і знань”

Виконала:

Студентка групи ПМ-22

Кудрявцева О.М

Перевірив

Пабирівський В.В.

Львів-2024р.

**Завдання:**

Завдання лабораторної роботи виконувати в призначеній предметній області. При написанні процедур повинні бути використані лише базові таблиці відповідно до створеної у ЛР №1 схеми БД, тобто результати роботи процедур повинні записуватися у відповідні поля існуючих таблиць БД та не можна створювати жодних тимчасових таблиць. Змістовне призначення процедур є наступним:

Написати процедуру визначення рейтингу кожного лікаря на основі власного алгоритму, який враховував би кількість хворих, тривалість лікування, ефективність лікування, кількість протипоказань, тощо. Написати процедуру, яка б викликала першу процедуру для всіх лікарів.

У звіті має бути наведено:

Для кожної процедури / функції:

1) Призначення процедури / функції.

2) Опис алгоритму роботи (концепція роботи та покроковий опис, можливо блок-схема).

3) Опис вхідних / вихідних параметрів (їх призначення, типи, варіанти застосувань).

4) Опис внутрішніх ідентифікаторів (їх призначення, типи, застосування).

5) Результати виконання.

**Хід роботи:**

**SQL-запити для створення процедури, що визначає рейтинг певного лікаря:**

CRETE PROCEDURE DoctorRating

AS

BEGIN

DECLARE @TotalPatientsCount float;

DECLARE @TotalTreatmentDuration float;

DECLARE @TotalContradictionsCount float;

DECLARE @TotalTreatmentScale float;

DECLARE @Rating float;

SELECT @TotalPatientsCount =

CASE

WHEN COUNT(DISTINCT H.PatientID) > 5 THEN 5

ELSE COUNT(DISTINCT H.PatientID)

END

FROM HistoryOfIllness H

WHERE H.DoctorID = @DoctorID;

SELECT @TotalTreatmentDuration =

CASE

WHEN SUM(CASE WHEN DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) > 50 THEN 50 ELSE DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) END) > 50 THEN 5

ELSE (SUM(CASE WHEN DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) > 50 THEN 50 ELSE DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) END) / 10.0)

END

FROM HistoryOfIllness H

WHERE H.DoctorID = @DoctorID;

SELECT @TotalContradictionsCount =

CASE

WHEN COUNT(CASE WHEN P.ContraindicatedMedications IS NOT NULL THEN 1 ELSE NULL END) > 5 THEN 5

ELSE COUNT(CASE WHEN P.ContraindicatedMedications IS NOT NULL THEN 1 ELSE NULL END)

END

FROM HistoryOfIllness H

INNER JOIN Patient P ON H.PatientID = P.ID

WHERE H.DoctorID = @DoctorID;

SELECT @TotalTreatmentScale = SUM(COALESCE(H.TreatmentScale, 0)) / NULLIF(COUNT(DISTINCT H.PatientID), 0) / 20.0

FROM HistoryOfIllness H

WHERE H.DoctorID = @DoctorID;

SET @Rating = (((@TotalPatientsCount)\*0.25) + ((@TotalTreatmentScale)\*0.25) + (@TotalTreatmentDuration)\*0.25) + ((@TotalContradictionsCount)\*0.25);

UPDATE Doctor SET Rating = @Rating WHERE ID = @DoctorID;

END

Створила процедуру DoctorRating та призначила тимчасові значення за допомогою DECLARE.

Визначила кількість пацієнта за допомогою:

SELECT @TotalPatientsCount =

CASE

WHEN COUNT(DISTINCT H.PatientID) > 5 THEN 5

ELSE COUNT(DISTINCT H.PatientID)

END

FROM HistoryOfIllness H

WHERE H.DoctorID = @DoctorID;

Цей запит рахує кількість пацієнтів деякого лікаря і якщо кількість перебільшує 5, то значення залишається 5. Якщо ж лікар не має пацієнтів, то його рейтинг матиме значення NULL.

Тривалість лікування визначала за наступною логікою: якщо пацієнти деякого лікаря в сумі хворіли більше 50 днів, то оцінка буде 5. Якщо ж менше 50 днів то значення що вийшло ми ділимо на 10. Якщо пацієнт не закінчив лікування, то за допомогою COALESCE, якщо дата закінчення лікування NULL процедура вибирає поточну дату.

SELECT @TotalTreatmentDuration =

CASE

WHEN SUM(CASE WHEN DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) > 50 THEN 50 ELSE DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) END) > 50 THEN 5

ELSE (SUM(CASE WHEN DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) > 50 THEN 50 ELSE DATEDIFF(day, H.StartDate, COALESCE(H.EndDate, GETDATE())) END) / 10)

END

FROM HistoryOfIllness H

WHERE H.DoctorID = @DoctorID;

Потім визначила кількість протипоказань за такою самою логікою як кількість пацієнтів:  
SELECT @TotalContradictionsCount = COUNT(DISTINCT H.PatientID) / 2

FROM HistoryOfIllness H

INNER JOIN PrescribedMedication PM ON H.HistoryID = PM.HistoryID

INNER JOIN Medicines M ON PM.NameOfMedicines = M.NameOfMedicines

INNER JOIN Patient P ON M.CompoundOfMedicines = P.ContraindicatedMedications

WHERE H.DoctorID = @DoctorID;

Визначила середню оцінку ефективності лікування:

SELECT @TotalTreatmentScale = SUM(COALESCE(H.TreatmentScale, 0)) / NULLIF(COUNT(DISTINCT H.PatientID), 0) / 20

FROM HistoryOfIllness H

INNER JOIN Doctor D ON H.DoctorID = D.ID

INNER JOIN Patient P ON H.PatientID = P.ID

WHERE D.ID = @DoctorID;

Далі за певною формулою визначила рейтинг використовуючи кількість пацієнтів, кількість протипоказань, середню оцінку ефективності лікування та тривалість лікування:

SET @Rating = (((@TotalPatientsCount)\*0.25) + ((@TotalTreatmentScale)\*0.25) + (@TotalTreatmentDuration)\*0.25) + ((@TotalContradictionsCount)\*0.25);

Та оновила таблицю Doctor вставивши значення, що вийшло в поле Rating:

UPDATE Doctor SET Rating = @Rating WHERE ID = @DoctorID;

**Процедура, що викликає першу процедуру для всіх лікарів:**

CREATE PROCEDURE ExecuteDoctorRating

AS

BEGIN

DECLARE @DoctorID INT;

DECLARE doctor\_cursor CURSOR FOR

SELECT ID FROM Doctor;

OPEN doctor\_cursor;

FETCH NEXT FROM doctor\_cursor INTO @DoctorID;

WHILE @@FETCH\_STATUS = 0

BEGIN

EXEC [dbo].[DoctorRating] @DoctorID;

FETCH NEXT FROM doctor\_cursor INTO @DoctorID;

END

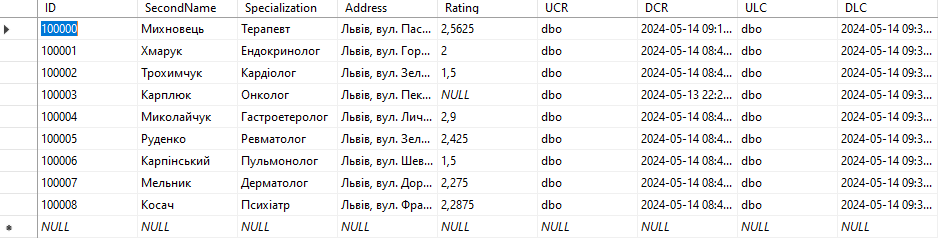
CLOSE doctor\_cursor;

DEALLOCATE doctor\_cursor;

END

Де за допомогою курсора процедура проходить по всіх лікарях.

**Результати виконання процедури:**

В таблицю Doctor записані всі рейтинги лікарів:  


**Посилання на репозиторій GitHub:**

<https://github.com/Oleksandra-Kudriavtseva/DBSQLKudriavtseva.git>

**Висновок:**

Написала процедуру для таблиць, створених при виконанні лабораторної роботи №1 для визначення рейтингу лікаря та створила процедуру, що викликає першу процедуру для всіх лікарів.