Лабораторна робота №5

ФІ91, Корешков Михайло

**Завдання**: розібратись з підзапитами. Написати декілька підзапитів для свого варіанта.

# 1. Підготовка

Для цієї дабораторної роботи я додав нове чисто числове поле – Abonents.CallsByMonth. Воно містить інформацію про кількість дзвінків абонента за поточний місяць (чи щось таке – не важливо)

Я не хотів додавати ці числа вручну, тому також зробив автоматичне заповнення цього поля в нових записах випадковим числом від 0 до 1000.

## 1.1 Перша спроба

Спочатку написав просто default rand() \* 1000 , але в такому випадку воно обирало єдине значення для всього інсерту (всі додані записи мали однакове значення в випадковому полі)

Далі додав тригер на додавання запису:

------

create trigger dbo.RandomizeAbonents

on dbo.Abonents

after INSERT

as

update dbo.Abonents

set CallsPerMonth = rand() \* 1000

where Abonents.AbonentID in (select AbonentID from inserted);

go

------

Працювало буквально так само – випадкове число визначалося один раз.

## 1.2 Друга спроба

Вирішив не заглиблюватись та піти іншим шляхом - використати NEWID(), що повертає кожен раз гарантовано інше значення.

------

create trigger dbo.RandomizeAbonents

on dbo.Abonents

after INSERT

as

update dbo.Abonents

set CallsPerMonth = rand(abs(checksum(NEWID()))) \* 1000

where Abonents.AbonentID in (select AbonentID from inserted);

go

------

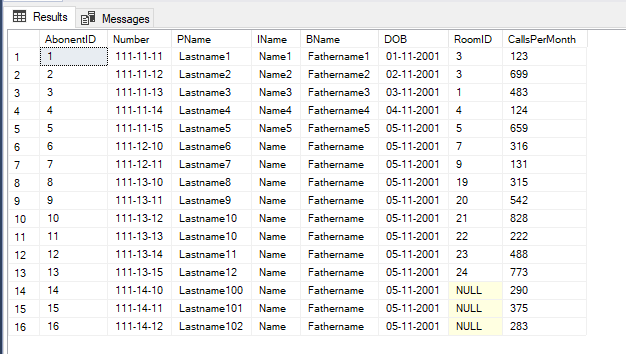
Запрацювало бездоганно, але виникла підозра, що та сама конструкція для отримання випадкового числа буде працювати й в першому варіанті

## 1.3 Остаточний варіант

Видалив тригер. Повернув дефолт, але у наступному вигляді

CallsPerMonth int default rand(abs(checksum(NEWID()))) \* 1000

**Результат вставки**:



## 1.4. Допоміжне представлення

В запитах часто потрібно буде дивитись на підрозділ абонента. Тому створив отаке

create or alter view AbonentsDivisions

as

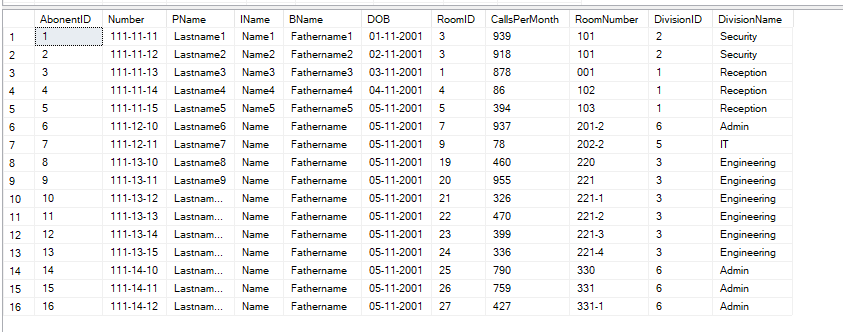
select Abonents.\*, Rooms.RoomNumber, Divisions.DivisionID, Divisions.DivisionName from Abonents

join Rooms on Rooms.RoomID = Abonents.RoomID

join Divisions on Divisions.DivisionID = Rooms.DivisionID;

go

**Вивід**:



# 2. Запити

Спочтаку отримаємо середні значення по підрозділам. Для цього потрібно group by

create or alter view AvgCallsByDivisions

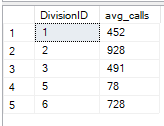
as

select DivisionID, AVG(CallsPerMonth) as avg\_calls from AbonentsDivisions group by DivisionID;

go

select \* from AvgCallsByDivisions;

**Вивід:**



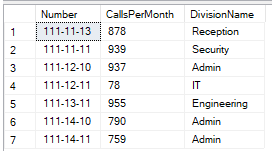
## 2.1. Вивід списку атрибутів відношення, кількісні оцінки якого вище середнього [по підрозділу]

select Number, CallsPerMonth, DivisionName from AbonentsDivisions

where

CallsPerMonth >= (select avg\_calls from AvgCallsByDivisions where AvgCallsByDivisions.DivisionID = AbonentsDivisions.DivisionID)

**Вивід:**



## 2.2. Вивід списку атрибутів відношення, кількісні оцінки якого максимальні

select Number, CallsPerMonth, DivisionName from AbonentsDivisions as b

where

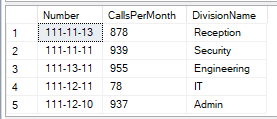
b.CallsPerMonth = (

select MAX(a.CallsPerMonth) from AbonentsDivisions as a

where a.DivisionID = b.DivisionID

);

**Вивід:**



## 2.3. Вивід списку атрибутів відношення, яке відповідає деякій умові з підзапиту, наприклад, вивести імена студентів відмінників і назви предметів, по яким вони здавали екзамен.

В моєму випадку шукатиму:

* підрозділи із середньою кількістю дзвінків більше середнього по всім абонентам
* всіх абонентів цього підрозділу

select AvgCallsByDivisions.DivisionID, AvgCallsByDivisions.avg\_calls from AvgCallsByDivisions

where avg\_calls >= (select AVG(CallsPerMonth) from AbonentsDivisions)

select a.Number, a.CallsPerMonth, a.DivisionName, a.DivisionID from AbonentsDivisions as a

where a.DivisionID in (

select AvgCallsByDivisions.DivisionID from AvgCallsByDivisions

where avg\_calls >= (select AVG(CallsPerMonth) from AbonentsDivisions)

)

**Вивід:**

