

ЛАБОРАТОРНА РОБОТА №4

ФІ-12 ЗАВАЛІЙ ОЛЕКСАНДР
ВАРІАНТ №5

Завдання

Предметна область:

Навчально-методичне управління (облік площі приміщень).

Основні предметно-значущі сутності:

Приміщення, Підрозділи.

Основні предметно-значущі атрибути сутностей:

- **Приміщення:** назва або номер приміщення, вид приміщення (аудиторія, кабінет і т.п.), площа, кількість посадочних місць, підрозділ.
- **Підрозділи:** назва, вид підрозділу.

Основні вимоги до функцій системи:

- Вибрати назви або номери приміщень за підрозділами;
- Підрахувати загальну площу навчальних аудиторій по приміщеннях і в цілому по навчальному закладу;
- Підрахувати загальну кількість посадочних місць для співробітників по підрозділам.

Тригери:

1. На видалення запису з таблиці «Приміщення». Якщо для приміщення зазначено підрозділ, заборонити видалення запису.
2. Створити представлення «Аудиторії» з полями «код приміщення», «назва приміщення», «підрозділ», в яку повинні входити приміщення виду «Аудиторія». Оновлювати представлення «Аудиторії».

Процедура:

Процедура повинна повертати кількість приміщень для зазначеного підрозділу.

Завдання для лабораторної роботи

Сформулювати запити для виконання завдань з вибраного варіанта.

Реалізація завдання

Для реалізації потрібної вибірки використаємо відповідні дані, які додали в процесі виконання минулої лабораторної роботи.

```
SELECT SubdivisionType, count(Rm.SubdivisionId)[Amount of rooms of this type]
FROM Subdivision Sub, Room Rm
WHERE Sub.SubdivisionId = Rm.SubdivisionId
GROUP BY Sub.SubdivisionType, Rm.SubdivisionId
```

	SubdivisionType	Amount of rooms of this type
1	BUSINESS STORAGE	2
2	CHURCH	1
3	HOTEL_ROOM	3
4	APARTMENTS	1
5	APARTMENTS	1
6	FAMILY DWELLINGS	1
7	APARTMENTS	1
8	TAX CLASS	1
9	STORE	5
10	FAMILY DWELLINGS	1
11	APARTMENTS	3

Рис. 1: Таблиця відповідності підрозділу та кількості приміщень цього типу.

Наступна таблиця містить: тип підрозділу, назву цього підрозділу, тип кімнати відповідний до цього підрозділу, тип будівлі та дані власника цього будинку. Для отримання потрібних даних виконаємо таку команду:

```
SELECT SubdivisionType, SubdivisionName, Rm.TypeOfRoom, B1.TypeOfBuilding,
Ow.FirstName, Ow.LastName
FROM Subdivision Sub, Room Rm, Building B1, Ownerss Ow
WHERE Sub.SubdivisionId = Rm.SubdivisionId AND Rm.BiuldingId=B1.BiuldingId
AND B1.IdOwner=Ow.OwnerId
```

	SubdivisionType	SubdivisionName	TypeOfRoom	TypeOfBuilding	FirstName	LastName
1	HOTEL_ROOM	RR	Deluxe Room	RENTALS - ELEVATOR APARTMENTS	Abby	Aagaard
2	STORE	RK	Room	TWO FAMILY DWELLINGS	Abe	Aaland
3	FAMILY DWELLINGS	B3	Premier Twin Room	COOPS - WALKUP APARTMENTS	Adaline	Aanerud
4	CHURCH	M9	Club Room	CONDO NON-BUSINESS STORAGE	Abbie	Aadland
5	APARTMENTS	D5	Classic Room	RENTALS - ELEVATOR APARTMENTS	Abram	Aamodt
6	STORE	RK	Executive Room	RELIGIOUS FACILITIES	Abe	Aaland
7	APARTMENTS	R2	Double Room	CONDO NON-BUSINESS STORAGE	Abbie	Aadland
8	APARTMENTS	L9	Junior Suite	TWO FAMILY DWELLINGS	Abbey	Aaby
9	FAMILY DWELLINGS	C0	Standard Room	THREE FAMILY DWELLINGS	Abraham	Aalund
10	BUSINESS STORAGE	RS	Signature Suite	RENTALS - ELEVATOR APARTMENTS	Abram	Aamodt
11	APARTMENTS	C6	Premier Twin Room	CONDO STORE BUILDINGS	Aaron	Aaberg
12	HOTEL_ROOM	RR	Deluxe Suite	TWO FAMILY DWELLINGS	Abbey	Aaby
13	HOTEL_ROOM	RR	Club Room	CONDO NON-BUSINESS STORAGE	Abbie	Aadland
14	STORE	RK	Traditional Room	COOPS - WALKUP APARTMENTS	Adaline	Aanerud
15	BUSINESS STORAGE	RS	Regency Club	TAX CLASS 1 CONDOS	Adah	Aanderud
16	STORE	RK	Premier Twin Room	LOFT BUILDINGS	Abbie	Aadland
17	TAX CLASS	R6	Business Plan	TWO FAMILY DWELLINGS	Abbey	Aaby
18	APARTMENTS	L9	Premium Room	THREE FAMILY DWELLINGS	Abraham	Aalund
19	STORE	RK	Signature Suite	CONDO NON-BUSINESS STORAGE	Abbie	Aadland
20	APARTMENTS	L9	Premier Room	COOPS - WALKUP APARTMENTS	Adaline	Aanerud

Рис. 2: Результат.

При виникненні питань щодо полів таблиці або реалізації додав код нижче.

```
CREATE DATABASE TestProject
GO
USE TestProject

CREATE TABLE Ownerss(
    OwnerId INT IDENTITY(1, 1) PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL );

CREATE TABLE Building(
    BiuldingId INT IDENTITY(1, 1) PRIMARY KEY CLUSTERED ,
    IdOwner INT NOT NULL,
    TypeOfBuilding VARCHAR(50) NOT NULL,
    AmountOfRooms INT CHECK (AmountOfRooms BETWEEN 1 AND 120),
    AmountOfFloors INT CHECK (AmountOfFloors BETWEEN 1 AND 32),
    CONSTRAINT Error_owner_id FOREIGN KEY (IdOwner)
    REFERENCES Ownerss (OwnerId) );

CREATE TABLE Subdivision(
    SubdivisionId INT IDENTITY(1, 1) PRIMARY KEY,
    SubdivisionName VARCHAR(50) UNIQUE NOT NULL,
    SubdivisionType VARCHAR(50) NOT NULL );

CREATE TABLE TypeOfSeats(
    SeatsId INT IDENTITY(1, 1) PRIMARY KEY,
    TypeOfSeats VARCHAR(50) UNIQUE NOT NULL );

CREATE TABLE Room(
    RoomId INT IDENTITY(1, 1) PRIMARY KEY,
    TypeOfRoom VARCHAR(50) NOT NULL,
    Area INT CHECK (Area BETWEEN 10 AND 70),
    AmountOfSeats INT CHECK (AmountOfSeats BETWEEN 0 AND 50),
    RoomNumber INT CHECK (RoomNumber BETWEEN 1 AND 80),
    SubdivisionId INT,
    BiuldingId INT NOT NULL,
    Storey INT CHECK (Storey BETWEEN 1 AND 32)
    CONSTRAINT Error_subdivision_id FOREIGN KEY (SubdivisionId)
    REFERENCES Subdivision (SubdivisionId) ON DELETE CASCADE,
    CONSTRAINT Error_biulding_id FOREIGN KEY (BiuldingId)
    REFERENCES Building (BiuldingId) ON DELETE CASCADE );

CREATE TABLE RoomSeats(
    RoomId INT FOREIGN KEY REFERENCES Room(RoomId),
    SeatsId INT FOREIGN KEY REFERENCES TypeOfSeats(SeatsId),
    PRIMARY KEY(RoomId, SeatsId) );
```