

Бази даних

Практична робота №1

Створення таблиць

Зміст

1	Теоретичні відомості	2
2	Завдання	4
3	Варіанти завдань	4
4	Вимоги до звіту	5
5	Приклад виконання.....	5

1 Теоретичні відомості

Спроектowana структура бази даних у вигляді логічної моделі надалі може бути перенесена у конкретну фізичну реалізацію тієї чи іншої реляційної СУБД. Ми надалі будемо користуватись СУБД Microsoft SQL Server.

Перед тим як почати створювати таблиці потрібно створити нову базу даних. База даних створюється за допомогою оператора мови SQL CREATE DATABASE.

Загальний синтаксис оператора CREATE DATABASE має наступний вигляд:

```
CREATE DATABASE database_name
[ CONTAINMENT = { NONE | PARTIAL } ]
[ ON
    [ PRIMARY ] <filespec> [ ,...n ]
    [ , <filegroup> [ ,...n ] ]
    [ LOG ON <filespec> [ ,...n ] ]
]
[ COLLATE collation_name ]
[ WITH <option> [ ,...n ] ]
[;]

<option> ::=
{
    FILESTREAM ( <filestream_option> [ ,...n ] )
    | DEFAULT_FULLTEXT_LANGUAGE = { lcid | language_name | language_alias }
    | DEFAULT_LANGUAGE = { lcid | language_name | language_alias }
    | NESTED_TRIGGERS = { OFF | ON }
    | TRANSFORM_NOISE_WORDS = { OFF | ON }
    | TWO_DIGIT_YEAR_CUTOFF = <two_digit_year_cutoff>
    | DB_CHAINING { OFF | ON }
    | TRUSTWORTHY { OFF | ON }
    | PERSISTENT_LOG_BUFFER=ON ( DIRECTORY_NAME='<Filepath to folder on DAX
formatted volume>' )
}

<filestream_option> ::=
{
    NON_TRANSACTED_ACCESS = { OFF | READ_ONLY | FULL }
    | DIRECTORY_NAME = 'directory_name'
}

<filespec> ::=
{
    (
        NAME = logical_file_name ,
        FILENAME = { 'os_file_name' | 'filestream_path' }
        [ , SIZE = size [ KB | MB | GB | TB ] ]
        [ , MAXSIZE = { max_size [ KB | MB | GB | TB ] | UNLIMITED } ]
        [ , FILEGROWTH = growth_increment [ KB | MB | GB | TB | % ] ]
    )
}
```

```
)
}
```

```
<filegroup> ::=
{
FILEGROUP filegroup name [ [ CONTAINS FILESTREAM ] [ DEFAULT ] | CONTAINS
MEMORY_OPTIMIZED_DATA ]
    <filespec> [ ,...n ]
}
```

Для створення бази даних нам будуть потрібні тільки ті частини синтаксису загального виразу, які позначені зеленим кольором.

Створення таблиць бази даних є першою операцією після створення бази даних. Таблиці створюються за допомогою оператора CREATE TABLE мови Transact SQL.

Загальний вигляд синтаксису досить складний, та для використання достатньо скороченого варіанту синтаксису, який наведено нижче.

```
CREATE TABLE
    { database_name.schema_name.table_name | schema_name.table_name | table_name
}
    ( { <column_definition> } [ ,...n ] )
[ ; ]
<column_definition> ::=
column_name <data_type>
    [ FILESTREAM ]
    [ COLLATE collation_name ]
    [ SPARSE ]
    [ MASKED WITH ( FUNCTION = ' mask_function ' ) ]
    [ [ CONSTRAINT constraint_name ] DEFAULT constant_expression ]
    [ IDENTITY [ ( seed,increment ) ]
    [ NOT FOR REPLICATION ]
    [ GENERATED ALWAYS AS ROW { START | END } [ HIDDEN ] ]
    [ NULL | NOT NULL ]
    [ ROWGUIDCOL ]
    [ ENCRYPTED WITH
        ( COLUMN_ENCRYPTION_KEY = key_name ,
          ENCRYPTION_TYPE = { DETERMINISTIC | RANDOMIZED } ,
          ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'
        ) ]
    [ <column_constraint> [, ...n ] ]
    [ <column_index> ]
<column_constraint> ::=
[ CONSTRAINT constraint_name ]
{
    { PRIMARY KEY | UNIQUE }
    [ CLUSTERED | NONCLUSTERED ]
    [
        WITH FILLFACTOR = fillfactor
        | WITH ( < index_option > [ , ...n ] )
    ]
    [ ON { partition_scheme_name ( partition_column_name )
        | filegroup | "default" } ]
```

```

| [ FOREIGN KEY ]
    REFERENCES [ schema_name . ] referenced_table_name [ ( ref_column ) ]
    [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
    [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
    [ NOT FOR REPLICATION ]

| CHECK [ NOT FOR REPLICATION ] ( logical_expression )
}

```

Типи даних які використовуються у якості <data_type> співпадають з одним із стандартних вбудованих типів, що були розглянуті на лекційних заняттях. (Інформацію можна взяти з відповідної лекційної презентації, що знаходиться на навчальному ресурсі Університету.)

2 Завдання

Потрібно створити скрипти мови Transact SQL для:

1. Створення бази даних.
2. Створення кожної окремої таблиці бази даних відповідно до логічної моделі бази даних.

Створені скрипти потрібно перевірити виконавши їх:

1. У середовищі Microsoft SQL Server Management Studio.
2. За допомогою програмного засобу командного рядка SQLCMD.exe.

Наступним кроком потрібно створити пакетний файл (.BAT) для створення нової бази даних та всіх таблиць бази даних за допомогою SQLCMD.exe.

3 Варіанти завдань

Завдання виконати згідно індивідуальних завдань отриманих кожним окремим студентом на початку курсу «Бази даних».

4 Вимоги до звіту

Виконані завдання потрібно оформити у вигляді окремих файлів зі скриптами на мові Transact SQL. Скрипти повинні бути оформлені згідно вимог документу «Інструкція по оформленню коду SQL». Інструкцію можна завантажити з навчального ресурсу Університету.

Виконане завдання повинно складатися з:

1. Файл зі скриптом мови Transact SQL для створення бази даних.
2. Файли зі скриптами мови Transact SQL для створення таблиць бази даних.
Скрипт створення кожної таблиці повинно розмістити у окремому файлі.
3. Пакетний файл створення пустої бази даних.

5 Приклад виконання

Далі наведено приклад, як повинен виглядати файл створення таблиці на мові Transact SQL:

```
-- CT-Students.sql
-- -----
-- Проект.....: Курс Бази даних
-- Призначення.: Скрипт призначено для створення пустої таблиці Студенти згідно
--                спроектованої логічної моделі.
-- Виконав.....: Ізварін І.В.
-- Історія змін: 1 січня 2021 -- дата створення
-- -----

-- Спочатку потрібно перевірити, чи існує таблиця з зазначеною назвою.
-- У тому випадку, якщо така таблиця вже існує її потрібно видалити
-- перед виконанням скрипту створення нової таблиці.
IF OBJECT_ID( N'Студенти', N'U' ) IS NOT NULL
BEGIN
    DROP TABLE [Студенти]
END
GO

-- Створити таблицю Студенти згідно логічної моделі
CREATE TABLE [Студенти]
(
    [Внутрішній Номер] INT NOT NULL PRIMARY KEY IDENTITY( 1, 1 ),
```

```

[Ім'я]          NVARCHAR(30) NOT NULL,
[Прізвище]      NVARCHAR(30) NOT NULL,
[По-батькові]   NVARCHAR(50) NULL,
[Народився]     DATE NOT NULL,
[Телефон]       CHAR(13) NULL,
[Адреса проживання] NVARCHAR(100) NOT NULL,
[Адреса батьків] NVARCHAR(100) NOT NULL,
[Електронна Пошта] VARCHAR(50) NULL,
[Скайп]         VARCHAR(50) NULL,
[Дата вступу]   DATE NOT NULL,
[Стать]         NCHAR(1) NOT NULL -- Ч: чоловіча, Ж: жіноча
                CHECK ( [Стать] = N'Ч' OR [Стать] = N'Ж' ),
[Група]         NVARCHAR(8) NOT NULL REFERENCES [Групи]([Код групи])
)
GO

-----
-- EOF CT-Students.sql

```