

Лабораторна робота 2

Створення бази даних у СУБД Microsoft SQL Server

Мета роботи: створити бази даних і таблиці в середовищі Microsoft SQL Server з використанням мови структурованих запитів SQL

ЗАВДАННЯ:

1. Створити базу даних за допомогою SQL-запитів
2. Заповнити таблиці комп'ютерної фірми

ЗМІСТ ЗВІТУ:

Найменування і мета роботи
База даних комп'ютерної фірми

ПОРЯДОК ВИКОНАННЯ

1. Створення бази даних за допомогою SQL-запитів

Необхідно створити базу даних, яку ми назвемо *comp_firm*. Для цього в SQL існує оператор *create database*. Створення бази даних має наступний синтаксис:

```
CREATE DATABASE ім'я_бази_даних ;
```

Максимальна довжина імені БД складає 64 знака і може включати букви, цифри, символ "_" і символ "\$". Ім'я може починатися з цифри, але не повинно повністю складатися з цифр. Будь-який запит до БД закінчується крапкою з комою (цей символ називається роздільником - delimiter). Отримавши запит, сервер виконує його і в разі успіху видає повідомлення "Query OK ...".

Тепер в цей базі даних нам треба створити 4 таблиці (в дужках вказані найменування стовпців таблиці) :

PRODUCT (maker, model, type);

PC (code, model, speed, ram, hd , cd, price);

LAPTOP (code, model, speed, ram, hd , prise , screen);

PRINTER (code, model, color, type, price).

Таблиця **PRODUCT** представляє виробника (maker), номер моделі (model), і тип (type) ('PC' - ПК, 'Laptop' - ноутбук , 'Printer' - принтер). Передбачається, що номери моделей в таблиці Product унікальні для всіх виробників і типів продуктів.

У таблиці **PC** для кожного ПК, однозначно визначається унікальним кодом (code), вказана модель (model), швидкість процесора в мегагерцах (speed), обсяг пам'яті в мегабайтах (ram), розмір диска в гігабайтах (hd), швидкість пристрою, що зчитує (cd) і ціна (price).

Таблиця **LAPTOP** аналогічна таблиці PC за винятком того, що замість швидкості cd містить розмір екрану в дюймах (screen).

У таблиці **PRINTER** для кожної моделі вказується, чи є він кольоровим - color ('y', якщо кольоровий), тип принтера - type (лазерний - Laser, струменевий - Jet або матричний - Matrix) і ціна - price.

Для створення таблиць в SQL існує оператор *create table*. Створення таблиць має наступний синтаксис:

```
CREATE TABLE ім'я_таблиці  
(ім'я_першого_стовпчика тип,  
ім'я_другого_стовпчика тип,  
...,  
ім'я_останнього_стовпчика тип );
```

Вимоги до імен таблиць і стовпців такі ж, як і для імен БД. До кожного колонку прив'язаний певний тип даних, який обмежує характер інформації, яку можна зберігати в стовпці (наприклад, запобігати введення букв в числове поле). SQL підтримує кілька типів даних: числові, строкові, календарні і спеціальний тип NULL, що означає відсутність інформації.

Типи даних та схема для нашої бази наведені на рис.1.

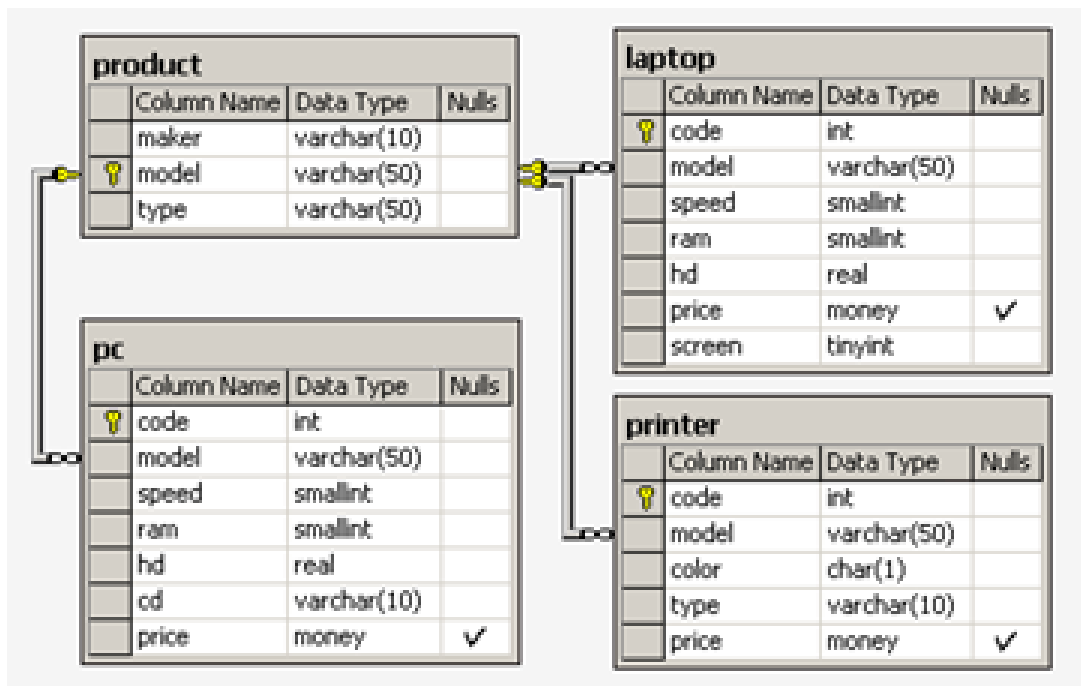


Рисунок 1 - Типи даних та схема для бази *comp_firm*

Не забуваємо вказувати первинні і зовнішні ключі.

2. Внесення даних в таблиці

Після створення таблиць необхідно внести дані в наші таблиці. Для цього використовується оператор **INSERT** :

```
INSERT ім'я_таблиці
VALUES
( 'значення_першого_стовпчика',
'значення_другого_стовпчика',
...,
'значення_останнього_стовпчика ');
```

Щоб додати відразу кілька рядків, треба просто перераховувати дужки зі значеннями через кому:

Заповнимо таблицю **PRODUCT** даними представленими на рис.2.

maker	model	type
A	1232	PC
A	1233	PC
A	1276	Printer
A	1298	Laptop
A	1401	Printer
A	1408	Printer
A	1752	Laptop
B	1121	PC
B	1750	Laptop
C	1321	Laptop
D	1288	Printer
D	1433	Printer
E	1260	PC
E	1434	Printer
E	2112	PC
E	2113	PC

Рисунок 2 - Дані для заповнення таблиці **PRODUCT**

У таблицях нашої БД все поля обов'язкові для заповнення, але перші поля таблиць **PC**, **LAPTOP**, **PRINTER** має ключове слово - **IDENTITY** (тобто воно заповнюється автоматично), тому ми можемо пропустити цей стовпець.

Якби у нас були поля з типом **NULL**, тобто необов'язкові для заповнення, ми б теж могли їх проігнорувати. А ось якщо спробувати залишити порожнім поле зі значенням **NOT NULL**, то сервер видасть повідомлення про помилку і не виконає запит. Крім того, при внесенні даних сервер перевіряє зв'язки між таблицями. Тому вам не вдасться внести в поле, що є зовнішнім ключем, значення, відсутнє в пов'язаній таблиці. У цьому ви переконаєтеся, вносячи дані в таблиці що залишилися.

Заповнюємо таблицю **PC** згідно рис.3, таблицю **LAPTOP** (рис.4) і **PRINTER** (рис.5).

code	model	speed	ram	hd	cd	price
1	1232	500	64	5.0	12x	600.0000
10	1260	500	32	10.0	12x	350.0000
11	1233	900	128	40.0	40x	980.0000
12	1233	800	128	20.0	50x	970.0000
2	1121	750	128	14.0	40x	850.0000
3	1233	500	64	5.0	12x	600.0000
4	1121	600	128	14.0	40x	850.0000
5	1121	600	128	8.0	40x	850.0000
6	1233	750	128	20.0	50x	950.0000
7	1232	500	32	10.0	12x	400.0000
8	1232	450	64	8.0	24x	350.0000
9	1232	450	32	10.0	24x	350.0000

Рисунок 3 - Дані для заповнення таблиці PC

code	model	speed	ram	hd	price	screen
1	1298	350	32	4.0	700.0000	11
2	1321	500	64	8.0	970.0000	12
3	1750	750	128	12.0	1200.0000	14
4	1298	600	64	10.0	1050.0000	15
5	1752	750	128	10.0	1150.0000	14
6	1298	450	64	10.0	950.0000	12

Рисунок 4 - Дані для заповнення таблиці LAPTOP

code	model	color	type	price
1	1276	n	Laser	400.0000
2	1433	y	Jet	270.0000
3	1434	y	Jet	290.0000
4	1401	n	Matrix	150.0000
5	1408	n	Matrix	270.0000
6	1288	n	Laser	400.0000

Рисунок 5 - Дані для заповнення таблиці PRINTER

Перевірте правильність заповнення всіх таблиць.

КОНТРОЛЬНІ ПИТАННЯ:

1. Які операції можуть програмуватися за допомогою мови SQL?
2. Які ключові слова використовуються при організації запитів?
3. Які мови програмування підтримує SQL?
4. Який символ дає повідомлення про те, що команда сформована і готова до виконання?