

Створення таблиці

Створюючи базу даних, ви зберігаєте свої дані в таблицях – тематичних списках, що складаються з рядків і стовпців. Наприклад, ви можете створити таблицю "Контакти", щоб зберігати список імен, адрес і телефонних номерів, або таблицю "Товари", щоб зберігати відомості про товари.

У цій статті пояснюється, як створити таблицю, як додати поля до таблиці та як установити первинний ключ таблиці. Тут також пояснюється, як установити властивості поля та таблиці.

Оскільки інші об'єкти бази даних сильно залежать від

Огляд

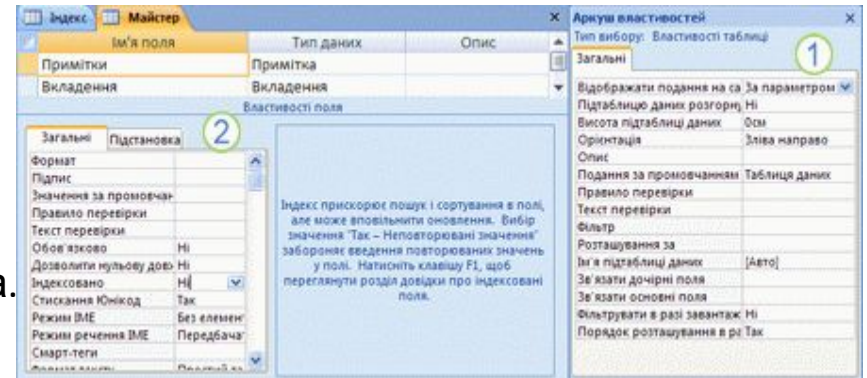
- Таблиця – це об'єкт бази даних, у якому зберігаються дані про певний предмет, наприклад про співробітників або товари. Таблиця складається із записів і полів.
- Кожний запис містить дані про один екземпляр предмета таблиці, як-от відомості про конкретного працівника. Запис також часто називають рядком або екземпляром.
- Кожне поле містить дані про один аспект предмета таблиці, наприклад ім'я або адресу електронної пошти. Поле також часто називають стовпцем або атрибутом.
- Запис складається зі значень поля, наприклад "Contoso, Ltd." або abc@example.com. Значення поля також часто називають фактом.

Іденти	Компанія	Ім'я	Прізвище
1	Компанія А	Anna	Bedecs
2	Компанія В	Antonio	Gratacos Solsona
3	Компанія С	Thomas	Axen

- 1. Запис
- 2. Поле
- 3. Значення поля
- База даних може містити багато таблиць, у кожній із яких зберігається інформація про свій предмет. Кожна таблиця може містити багато полів із різними типами даних (текст, числа, дати, гіперпосилання тощо).

Властивості таблиць і полів

- Крім того, таблиці й поля мають властивості, за допомогою яких можна керувати їхніми характеристиками та поведінкою.



- Таблиця, відкрита в режимі конструктора.
- 1. Властивості таблиці
- 2. Властивості поля
- У базі даних Access властивості таблиці – це атрибути таблиці, які впливають на вигляд і поведінку таблиці в цілому. Властивості таблиці визначаються у вікні властивостей таблиці в режимі конструктора. Наприклад, щоб указати спосіб відображення таблиці за замовчуванням, можна встановити властивість таблиці Подання за промовчанням.
- Властивість поля застосовується до конкретного поля в таблиці та визначає одну з характеристик поля або аспект його поведінки. Деякі властивості поля можна встановити в поданні таблиці. Будь-яку властивість поля також можна встановити в режимі конструктора за допомогою області Властивості поля.

Типи даних

- Кожне поле задається з певним типом даних. Тип даних поля визначає, які саме дані зберігаються в ньому, як-от великі обсяги тексту або вкладені файли

Ім'я поля	Тип даних	Опис
Примітки	Примітка	
Вкладення	Вкладення	
Властивості поля		

- Тип даних – це властивість поля, але вона дещо відрізняється від інших властивостей поля:
- Тип даних поля встановлюється в бланку таблиці, а не в області Властивості поля.
- Тип даних поля визначає, які ще властивості матиме поле.
- Тип даних поля потрібно обов'язково вказувати під час створення поля.
- Ви можете створити нове поле в Access, ввівши дані в новому стовпці у вікні табличного подання даних. Коли ви створюєте поле, вводячи дані у вікні табличного подання даних, програма Access автоматично призначає полю тип даних за введеним значенням. Якщо введені дані не відповідають жодному іншому типу, програма Access установить текстовий тип даних. За потреби тип даних можна змінити за допомогою стрічки в новому інтерфейсі користувача Microsoft Office Fluent.

Приклади автоматичного визначення типу даних

- У таблиці нижче показано, як автоматичне визначення типу даних працює у вікні табличного подання даних.

Введені дані:	Тип даних поля, створеного в програмі Office Access 2007:
Євген	Текст
http://www.contoso.com Ви можете вказати будь-який припустимий префікс протоколу Інтернету, наприклад "http://", "https://" або "mailto:".	Гіперпосилання
1	Число, довге ціле число
50 000	Число, довге ціле число
50 000,99	Число, подвійної точності
50000,389	Число, подвійної точності
12.07 Розпізнаються формати дати й часу вашої локалізації.	Дата й час
31 грудня 2006 р.	Дата й час
10:50:23	Дата й час
10:50 am	Дата й час
17:50	Дата й час
12,50 € Розпізнається символ грошової одиниці вашої локалізації.	Грошова одиниця
21,75	Число, подвійної точності
123.00 %	Число, подвійної точності

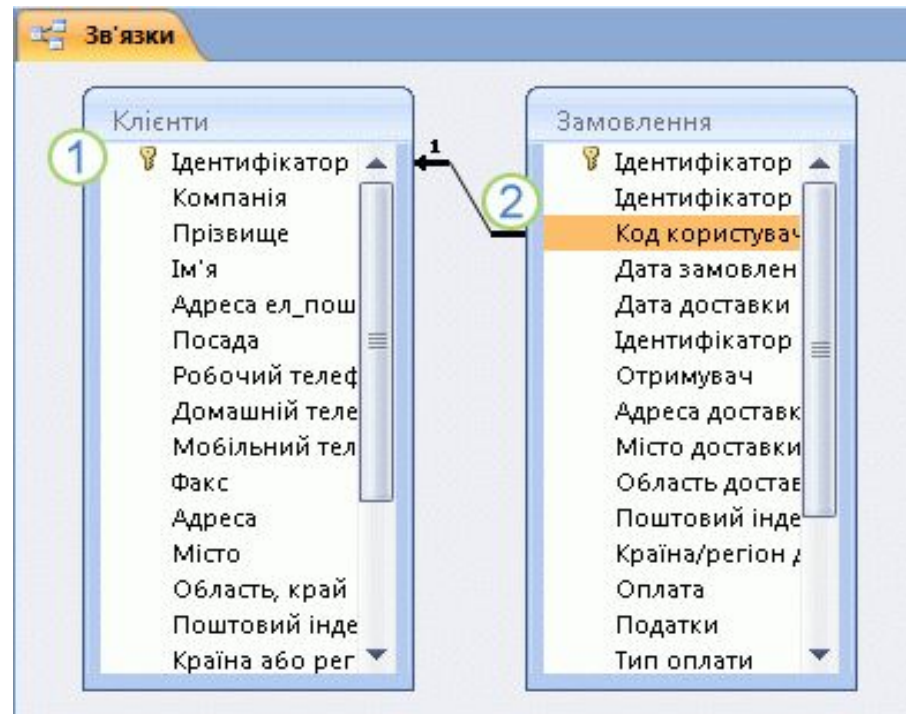
Зв'язки таблиць

- Хоча в кожній таблиці зберігаються дані про окремий предмет, зазвичай в одній базі даних містяться таблиці з взаємопов'язаними предметами. Наприклад, база даних може містити:
- таблицю клієнтів, яка містить імена клієнтів компанії та їхні адреси;
- таблицю товарів, які ви продаєте, включно з цінами та зображеннями для кожного елемента;
- таблицю замовлень, у якій відстежуються замовлення клієнтів.
- Оскільки дані про різні предмети зберігаються в окремих таблицях, потрібно певним чином зв'язати ці дані разом, щоб мати змогу легко об'єднати пов'язані дані з цих окремих таблиць. Для цього створюють зв'язки. Зв'язок – це логічне поєднання двох таблиць, яке визначає спільні поля для обох із них.

Ключі

- Поля, які використовуються для зв'язку таблиць, називаються ключами. Ключ зазвичай складається з одного поля, але може складатись і з кількох полів. Існують два типи ключів:
 - **Первинний ключ** . Таблиця може мати лише один первинний ключ. Первинний ключ складається з одного або кількох полів, які унікально ідентифікують кожен запис, що зберігається в таблиці. Як первинний ключ часто використовують певний унікальний ідентифікатор, наприклад ідентифікаційний номер, серійний номер або код. Наприклад, у вас є таблиця "Клієнти", у якій кожний клієнт має унікальний ідентифікатор. Поле такого ідентифікатора – це первинний ключ таблиці "Клієнти". Якщо первинний ключ містить кілька полів, зазвичай він складається з наявних полів, які, взяті разом, забезпечують унікальні значення. Наприклад, можна використати поєднання прізвища, імені та дати народження як первинний ключ для таблиці про людей.
 - **Зовнішній ключ** . Таблиця також може мати один або кілька зовнішніх ключів. Зовнішній ключ містить значення, які відповідають значенням первинного ключа іншої таблиці. Наприклад, у вас є таблиця "Замовлення", у якій кожне замовлення має ідентифікатор клієнта, що відповідає запису в таблиці "Клієнти". Поле такого ідентифікатора – це зовнішній ключ таблиці "Замовлення".

- Відповідність значень у полях ключів формує основу зв'язків таблиць. Зв'язок таблиць використовується для об'єднання даних із пов'язаних таблиць. Припустімо, ви маєте дві таблиці: "Клієнти" й "Замовлення". У таблиці "Клієнти" кожний запис визначається за полем первинного ключа (ідентифікатором).
- Щоб пов'язати кожне замовлення із клієнтом, додайте до таблиці "Замовлення" поле зовнішнього ключа, яке відповідає полю ідентифікатора в таблиці "Клієнти", а потім створіть зв'язок між двома ключами. Під час додавання запису до таблиці "Замовлення" слід використовувати значення ідентифікатора клієнта з таблиці "Клієнти". Відтак, якщо потрібно переглянути будь-яку інформацію про замовника, ви використовуєте зв'язок, щоб визначити, які дані з таблиці "Клієнти" відповідають потрібним записам у таблиці "Замовлення".



- Зв'язок між таблицями, відображений у вікні "Зв'язки".
- 1. Первинний ключ, позначений піктограмою ключа поряд з іменем поля.
- 2. Зовнішній ключ. Зверніть увагу: піктограма ключа відсутня.

Переваги використання зв'язків

- Зберігання даних порізно у зв'язаних таблицях дає такі переваги:
- **Узгодженість.** Оскільки кожний елемент даних записується лише один раз в одній таблиці, імовірність виникнення неоднозначності або неузгодженості менша. Наприклад, ім'я клієнта зберігається лише один раз у таблиці про клієнтів, а не багаторазово (і, імовірно, неузгоджено) у таблиці, яка містить дані про замовлення.
- **Ефективність.** Записування даних в одному розташуванні означає, що використовується менший обсяг місця на диску. Крім того, знайти дані в невеликих таблицях зазвичай можна швидше, ніж у великих. Урешті, якщо ви не використовуватимете окремі таблиці для різних предметів, у таблицях з'явиться надмірна кількість даних і Null-значення (відсутність даних). Те й друге призводить до марнування місця на диску та зниження продуктивності.
- **Зрозумілість.** Макет бази даних легше зрозуміти, якщо предмети належним чином розділено між таблицями