

ЛАБОРАТОРНА РОБОТА № 1

СТВОРЕННЯ ФАЙЛУ БАЗИ ДАНИХ СУБД Microsoft Access

Мета: навчитись створювати та редагувати файл БД: вивчити структуру об'єкта "таблиця"; навчитися задавати полям різні типи даних, виконувати операції в таблицях; навчитися працювати з БД.

Теоретичні відомості

Визначення та класифікація БД.

База даних (БД) – впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система керування БД.

Головне завдання БД – гарантоване збереження значних обсягів інформації (так звані записи даних) та надання доступу до БД користувачеві або ж прикладній програмі. Таким чином, БД складається з двох частин: збереженої інформації та системи керування.

БД можна класифікувати за різними ознаками (табл. 1).

Окреме місце в теорії та практиці займають просторові (англ. spatial), тимчасові, або темпоральні (temporal) і просторово-часові (spatial-temporal) БД.

Системи централізованих БД з віддаленим (мережевим) доступом можуть допускати різні архітектури подібних систем:

- «файл-сервер»,
- «клієнт-сервер».

Коротка характеристика деяких СУБД.

Створення БД, обробка та пошук необхідної інформації в ній здійснюється за допомогою системи управління базами даних (СУБД). СУБД – це набір певних програмних засобів, які надають можливість користувачеві швидко та ефективно взаємодіяти з БД.

MySQL – вільна СУБД. MySQL є власністю компанії Oracle Corporation, що отримала її разом з поглиненою Sun Microsystems, яка здійснює розробку і підтримку додатку. Розповсюджується під GNU General Public License і під власною комерційною ліцензією. Крім цього розробники створюють функціональність на замовлення ліцензійних користувачів, саме завдяки таким замовленням з'явився механізм реплікації.

Таблиця 1

Класифікація БД	
За моделлю даних:	<ul style="list-style-type: none">– ієрархічні;– мережеві;– реляційні;– об'єктні;– об'єктно-орієнтовані;– об'єктно-реляційні;
За технологією фізичного зберігання:	<ul style="list-style-type: none">– БД у вторинній пам'яті (традиційні);

	<ul style="list-style-type: none"> - БД в оперативній пам'яті (in-memory databases); - БД у третинній пам'яті (tertiary databases);
За ступенем розподіленості:	<ul style="list-style-type: none"> - централізовані (зосереджені); - розподілені;
За застосуванням мови програмування:	<ul style="list-style-type: none"> - відкриті (спираються на одну з універсальних мов); - замкнуті (використовується власна мова програмування);
За функціями, які виконуються:	<ul style="list-style-type: none"> - інформаційні; - операційні;
За сферою застосування:	<ul style="list-style-type: none"> - універсальні; - спеціалізовані;
За «потужністю»:	<ul style="list-style-type: none"> - корпоративні; - настільні;
За способом доступу:	<ul style="list-style-type: none"> - з локальним доступом; - з віддаленим (мережевим) доступом.

MySQL із самого початку була дуже схожою на mSQL, проте з часом вона розширювалася і зараз MySQL – одна з найпоширеніших СУБД.

Використовується, у першу чергу, для створення динамічних веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування.

MySQL є рішенням для малих і середніх додатків. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте до дистрибутиву входить бібліотека внутрішнього сервера, що дозволяє включати MySQL до автономних програм. Вихідні коди сервера компілюються на багатьох платформах.

Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі, GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць. MySQL характеризується великою швидкістю, стійкістю і простотою.

PostgreSQL – об'єктно-реляційна СУБД. Є альтернативою як комерційним СУБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СУБД з відкритим кодом (MySQL, Firebird, SQLite).

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL дана СУБД не контролюється однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які бажають використовувати дану СУБД та впроваджувати в неї найновіші досягнення.

СУБД Oracle – це найпотужніший програмний комплекс, що дозволяє створювати додатки будь-якої складності. Ядром цього комплексу є БД, що

зберігає інформацію, кількість якої за рахунок наданих засобів масштабування практично необмежена. З високою ефективністю працювати з відповідною інформацією одночасно може практично будь-яка кількість користувачів (за наявності достатніх апаратних ресурсів) без тенденції до зниження продуктивності системи при різкому збільшенні їх кількості.

Механізми масштабування в СУБД Oracle останньої версії дозволяють збільшувати потужність і швидкість роботи сервера Oracle, додатків, додаючи нові вузли кластера. Це не вимагає зупинки працюючих додатків, переписування старих додатків, розроблених для звичайної одномашинної архітектури. Крім того, вихід з ладу окремих вузлів кластера також не призводить до зупинки програми.

Вбудовування до СУБД Oracle JavaVM повномасштабної підтримки серверних технологій (Java Server Pages, Java-сервлети, модулі Enterprise JavaBeans, інтерфейси прикладного програмування CORBA), призвели до того, що Oracle де-факто є стандартом СУБД для Internet.

СУБД Oracle поставляється практично для всіх існуючих операційних систем. Працюючи під Sun Solaris, Linux, Windows або на іншій операційній системі з продуктами Oracle не буде виникати проблем у роботі. СУБД Oracle однаково добре працює на будь-якій платформі. Таким чином, компаніям, які розпочинають роботу з продуктами Oracle не доводиться змінювати мережеве оточення. Існує лише невелика кількість відмінностей при роботі з СУБД, обумовлених операційною системою.

Microsoft SQL Server – система управління реляційними БД, розроблена корпорацією Microsoft. Основна мова запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів (SQL) із розширеннями. Використовується для роботи з БД розміром від персональних до великих БД масштабу підприємства, конкурує з іншими СУБД у даному сегменті ринку.

При взаємодії з мережею Microsoft SQL Server і Sybase ASE використовують протокол рівня додатків під назвою Tabular Data Stream (TDS, протокол передачі табличних даних). Протокол TDS також був реалізований у проекті FreeTDS з метою забезпечити різні додатки можливістю взаємодії з БД Microsoft SQL Server і Sybase.

Для забезпечення доступу до даних Microsoft SQL Server підтримує Open Database Connectivity (ODBC) – інтерфейс взаємодії додатків з СУБД. SQL Server надає можливість підключення користувачів через веб-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, кросплатформно з'єднуватися з SQL Server.

Microsoft Office Access або просто Microsoft Access – реляційна СУБД корпорації Microsoft. Основні компоненти MS Access:

- будівник таблиць;
- будівник екранних форм;
- будівник SQL-запитів (мова SQL в MS Access не відповідає стандарту ANSI);
- будівник звітів, що виводяться на друк.

Кожна нова версія Access все тісніше інтегрувалася з іншими програмними продуктами, що входять в Office. Важливим вдосконаленням стало те, що всі продукти Office і Visual Basic використовують спільну мову програмування Visual Basic for Applications (VBA).




Істотно розширює можливості MS Access з написання додатків механізм зв'язку з різними зовнішніми СУБД: «зв'язані таблиці» (зв'язок з таблицею СУБД) і «запити до сервера» (запит на діалекті SQL, який «розуміє» СУБД). Також MS Access дозволяє будувати повноцінні клієнт-серверні додатки на СУБД MS SQL Server; є можливість поєднати інструменти для управління БД і засоби розробки.

Варіанти, які пропонуються для створення нової БД:

- створення БД на основі шаблону;
- створення БД без використання шаблону;
- копіювання даних з іншого джерела до таблиці Access;
- імпортування, додавання та зв'язування з даними з іншого джерела;
- відкриття наявної бази даних Access;
- створення настроюваного пусого шаблону.

Структура, способи створення таблиць, типи полів у таблицях

В одній базі може бути безліч об'єктів різних типів – таблиць, запитів, форм, звітів. Основні дані у СУБД Microsoft Access зберігаються в таблицях. Таблиця – це сукупність записів. Стовпці в таблиці називаються полями, а рядка – записами. Кількість записів у таблиці обмежується ємністю жорсткого диска. Припустима кількість полів – 255. Таблиць у базі даних (БД) може бути кілька. Різні групи даних слід зберігати в окремих таблицях. Для роботи з таблицею її необхідно відкрити, а після закінчення роботи – закрити, попередньо зберігши всі зміни. Крім того, при створенні нової БД Access будує системні таблиці, де міститься інформація про об'єкти бази (одне з характерних властивостей БД полягає в тому, що вони самодокументовані!). Файли БД Microsoft Access 2010 мають розширення accdb.

В Access існує два основні режими роботи з таблицями – режим Таблиці та режим Конструктора. У першому з них (кнопка ) здійснюється безпосередньо робота з даними – введення, редагування, перегляд, сортування. У режимі Конструктора (кнопка ) створюється і модифікується структура таблиці: задаються імена і типи полів, формуються накладаються на них обмеження і т.д. Перехід з режиму таблиці в режим конструктора й навпаки проводиться за допомогою кнопки Режим , (вкладка Главная, група Режимы).

У новій БД автоматично створюється і відкривається в режимі таблиці Таблиця 1 (в російськомовній версії Access). Access дозволяє створювати нові поля і в режимі Таблиці, ґрунтуючись на введених користувачем значеннях.

Усі записи таблиці бази даних можна імпортувати з інших баз даних, а також з файлів *Excel* або текстових файлів.

При створенні таблиць необхідно дотримуватися наступних правил:

- усі внесені в базу даних таблиці повинні бути нормалізовані;
- кожна таблиця повинна мати ключ;

- дані в таблиці бажано вносити після встановлення зв'язку між ними, тобто після створення схеми бази даних;

- у випадку проблем із внесенням інформації, необхідно повернутися до початкового етапу створення таблиці й перевірити правильність встановлення типу полів.

Кожна таблиця повинна мати первинний ключ. Первинний ключ – це одне або кілька полів, зміст яких унікальний (не повторюється) для кожного запису. Дотримання цієї умови забезпечує цілісність даних. Первинні ключі гарантують унікальність записів у таблиці, а поля первинних ключів, які одночасно є індексами, використовуються для зв'язку таблиць.

Завдання 1. Створіть нову базу даних. Змінити назву (бажано використати назву певного родовища). Обрати для розміщення свою особисту папку.

Порядок виконання.

1. Завантажте *MS Access*, виконавши команду Пуск - Все програми - Microsoft Office - Microsoft Office Access.

2. У вікні СУБД ввести у поле Імя файла ім'я для створюваної бази даних, обрати для розміщення свою папку (переконатися, що виділений шаблон Новая база данных (рис. 1)) та натиснути на кнопку Создать. Відкриється вікно створення таблиці бази даних, у якому зліва буде відображена панель навігації, а праворуч шаблон для створення таблиці (рис. 2).

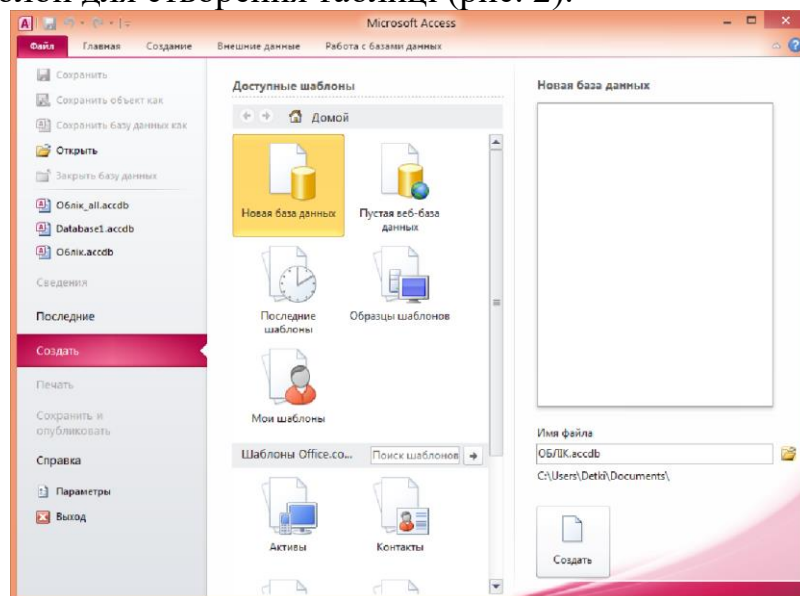


Рис. 1. Стартовое вікно СУБД MS Access

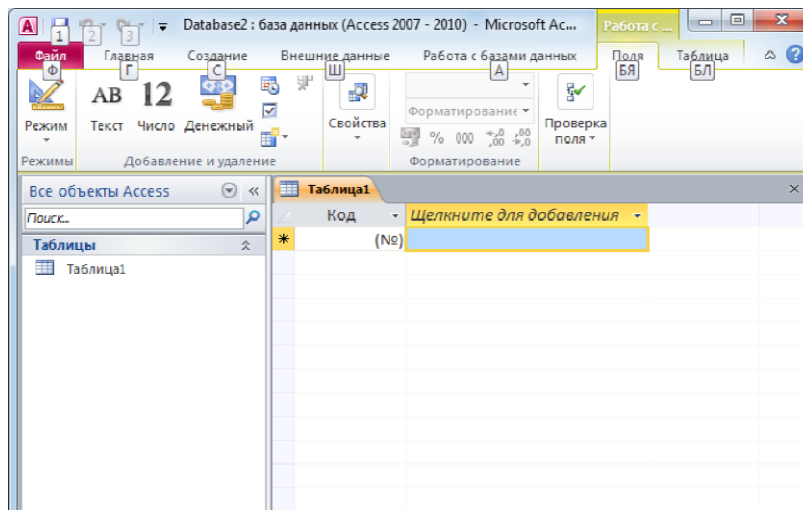


Рис. 2. Вікно створення таблиці бази даних
Завдання 2. Закрийте таблицю. Перейдіть у режим конструктора.

Порядок виконання.

Перейти на вкладку **Создание** та у групі **Таблицы** обрати команду **Конструктор таблиц** (рис. 3). З'являється вікно Конструктора.

Вікно Конструктора (рис. 4) складається із двох частин. У верхній частині відображається список полів таблиці із вказівкою їх типу й описом призначення, а в нижній – властивості виділеного поля.

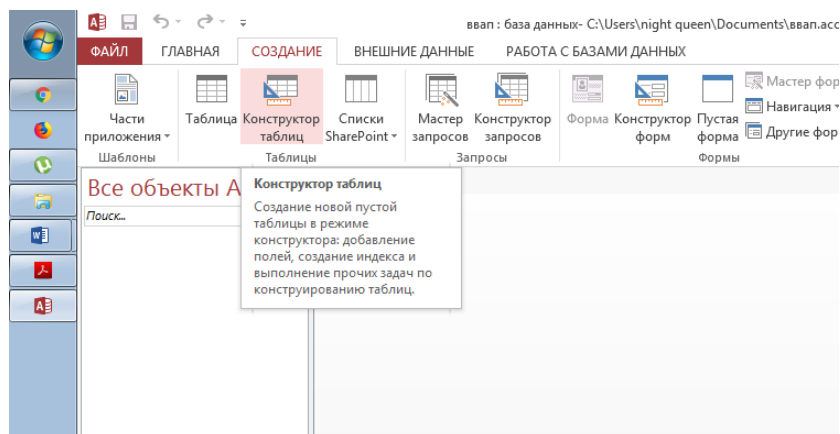


Рис. 3. Перехід в режим Конструктора

В режимі Конструктора таблиці створюються шляхом завдання імен полів, їх типів і властивостей. При завданні імен полів потрібно враховувати, що:

- імена полів повинні бути унікальними в межах таблиці;
- імена полів можуть містити не більше 64 символів, включаючи пробіли;
- небажано, щоб імена полів збігалися з іменами вбудованих функцій або властивостей Access, операторів мови SQL, так як в деяких випадках це може призвести до неправильної інтерпретації запитів до БД;
- імена полів не повинні починатися з пробілу або керуючого символу;
- імена полів можуть містити будь-які літери, цифри, знаки за винятком крапки «.», знака оклику «!», апострофа «'», квадратних дужок «[]».

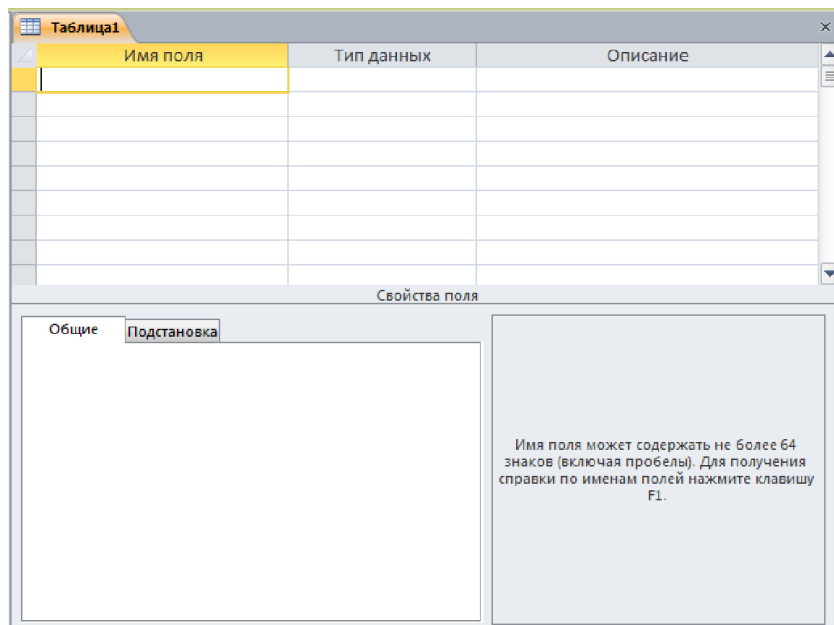


Рис. 4. Вікно Конструктора

Аналогічні правила застосовуються при завданні імен таблиць і інших об'єктів Access.

Ознайомившись з правилами іменування полів таблиць, перейдемо до розгляду типів даних. Нижче наведено перелік основних типів з назвами (табл. 2).

Для більшості полів (стовпців таблиці) можна задати додаткові властивості. Для цього служать вкладки внизу вікна конструктора. Нижче перераховані деякі з цих властивостей, але в залежності від обраного типу даних список доступних властивостей може змінюватися.

Підпис – псевдонім імені поля. Виводиться в заголовку стовпця в режимі таблиці, в формах і т.д. Наприклад, ви хочете називати стовпець «ID», а при відображенні таблиці показувати його під заголовком «Табельний номер співробітника». У цьому випадку зручно буде використовувати підпис.

Таблица 2

Типи даних

Тип	Опис
Текстовий	для введення тексту довжиною до 255 символів (за замовчуванням)
Поле МЕМО	для введення заміток або довгих описів
Числовий	для введення числових даних
Дата/час	для введення часу і дати
Грошовий	використовується для роботи з грошовими величинами
Лічильник	для введення числа, яке автоматично збільшується на одиницю при додаванні в таблицю нового запису

Логічний	для збереження логічного значення Так або Ні
Об'єкти OLE	використовуються для збереження в таблиці OLE-об'єктів (наприклад, малюнків, звуків, документів Word та ін.)
Гіперпосилання	для запису в таблицю гіперпосилань

Обов'язкове поле визначає, чи є введення даних в це поле обов'язковим або допустимо присутність записів, у яких відповідне значення не вказано. Значення за замовчуванням – «Ні»; якщо поставлено «Так», то це еквівалентно обмеження NOT NULL в SQL.

Формат поля дозволяє визначити, в якому форматі відображаються дані в режимі таблиці. Конкретний перелік форматів залежить від типу даних, яка була обрана для цього поля. Дуже часто зумовлений формат вказують для даних з типом Дата / Час.

Значення за замовчуванням дозволяє задати значення за замовчуванням для поля. Це значення буде присвоюватися в випадках, коли при внесенні запису в таблицю явно не вказано інше.

Індексовані поле – властивість, що дозволяє вказати, чи буде створюватися індекс для даного поля таблиці. Індокси використовуються для прискорення пошуку записів в БД, але для їх створення і підтримки потрібні додатковий час і дисковий простір. Поле може бути не індексованих, індексованих і допускати збіг значень, індексованих і не допускати збігу значень. Унікальний індекс (збіг значень не допускається) також є обмеженням цілісності. Він автоматично створюється для первинного ключа таблиці, і можна його створити вручну для альтернативних ключів.

Умова на значення – дозволяє задати обмеження на можливі значення поля, якщо такі обмеження присутні в предметної області. Наприклад, дата народження людини не може бути пізніше поточної дати.

Повідомлення про помилку – дозволяє визначити текст повідомлення, що виводиться Access в разі введення в поле неприпустимих значень.

Завдання 3. Створити у базі даних таблицю collar у режимі Конструктор.


Порядок виконання.


1. У режимі конструктор у поле Имя поля ввести hole_id. Клавішею Tab або за допомогою мишки перейдіть у наступне поле Тип данных. Тут за допомогою списку, що розкривається, необхідно обрати тип поля – Текстовый. У нижній частині частині вікна Конструктора Свойство поля на вкладці Общие вказати: Размер поля – 255, Обязательное поле – Да, Индексированное поле – Да (Совпадения не допускаются). Інші властивості поля залиште без змін.

2. Ввести назву наступного поля – NUMBER, обрати тип поля – Текстовий й встановити властивості поля згідно структури таблиці collar (табл 3). Діючи за аналогією створити усі поля таблиці.

Структура таблиці collar

Ім'я поля	Тип даних	Властивості поля
1	2	3
hole_id	Тектовий	Размер поля – 255 Обязательное поле – Да Индексированное поле – Да (совпадения не допускаются)
NUMBER	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
X	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
Y	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
Z	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
quarry	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
type	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
max_depth	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
hole_path	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет

3. У даній таблиці первинним ключовим полем є поле hole_id. Для того, щоб позначити його як ключове поле, необхідно встановити курсор на назву поля, потім потрібно вибрати команду Ключевое поле  (вкладка Конструктор, група Сервис) або вибрати аналогічну команду з контекстного меню поля, на сірій кнопці ліворуч від поля з'явиться знак ключа. Структура таблиці створена.

4. З контекстного меню ярлика таблиці вибрати команду Сохранить та у вікні діалогу, що відкрилося, увести ім'я таблиці collar і клацнути на кнопці ОК. Закрити вікно Конструктора, натиснувши кнопку Закреть  або обрати відповідну команду з контекстного меню ярлика таблиці. Створена таблиця з'явиться у списку таблиць вікна База даних.

5. Для заповнення таблиці collar вихідними даними вибрати команду Открыть з контекстного меню назви таблиці на панелі навігації бази даних (або зробити

подвійне клацання на імені таблиці). Увести вихідні дані у таблицю (рис. 5). Дані вводяться тільки записами, тобто по рядках.

hole_id	NUMBER	X	Y	Z	quarry	type	max_depth	hole_path
7521	1021	63468,82	23152,65	175,48	ЮГОК	Експлуатаційна	22	
7522	1022	63588,1	23498,03	184,34	ЮГОК	Експлуатаційна	31	
7523	1023	63059,18	22567,49	184,98	ЮГОК	Експлуатаційна	32,5	
7524	1024	62850,07	22657,28	186,87	ЮГОК	Експлуатаційна	33	
7525	1025	63004,79	22780,97	182,27	ЮГОК	Експлуатаційна	27	
7526	1026	62924,82	23236,87	165,26	ЮГОК	Експлуатаційна	23,5	
7527	1027	63160,88	22843,93	173,14	ЮГОК	Експлуатаційна	23	
7528	1028	62876,34	23365,41	170,64	ЮГОК	Експлуатаційна	17,3	
7529	1029	63271,01	23250,81	174,75	ЮГОК	Експлуатаційна	19,5	
7530	1030	63427,8	23374,59	184,3	ЮГОК	Експлуатаційна	31	
7531	1031	63588,28	23082,63	172,84	ЮГОК	Експлуатаційна	18	
7532	1032	63309,82	23090,72	167,7	ЮГОК	Експлуатаційна	15	
7533	1033	63124,42	23140,32	163,63	ЮГОК	Експлуатаційна	13	
7534	1034	62949,85	22262,12	175,86	ЮГОК	Експлуатаційна	22,5	
7535	1035	62901,47	22443,75	182,77	ЮГОК	Експлуатаційна	32	
7536	1036	62443,13	22589,66	188,57	ЮГОК	Експлуатаційна	36	
7537	1037	62600,91	22713,79	188,9	ЮГОК	Експлуатаційна	34,8	
7538	1038	63107,56	22383,77	181,43	ЮГОК	Експлуатаційна	30	
7539	1039	62790,79	22352,87	179,48	ЮГОК	Експлуатаційна	26	
7540	1040	62725,63	23757,97	181,44	ЮГОК	Експлуатаційна	26,5	
7541	1041	62863,43	23850,92	186,6	ЮГОК	Експлуатаційна	33	
7542	1042	62693,16	22532,92	186,02	ЮГОК	Експлуатаційна	33	
7543	1043	62593,12	22453,83	184,75	ЮГОК	Експлуатаційна	38	
7544	1044	63353,36	22805,31	168,95	ЮГОК	Експлуатаційна	28,4	
7545	1045	63153,45	22213,77	174,34	ЮГОК	Експлуатаційна	25	
7546	1046	63420,17	22632,01	175,82	ЮГОК	Експлуатаційна	22	
7547	1047	63414,66	22310,92	178,6	ЮГОК	Експлуатаційна	25,5	

Рис. 5. Вихідні дані для заповнення таблиці collar

6. По закінченню введення даних закрити вікно табличного режиму перегляду таблиці.

Прийоми роботи з таблицями БД

Зверніть увагу на рядок стану в нижній частині вікна. Цей рядок називається полем номеру запису. Це поле містить кнопки переходу, за допомогою яких можна ефективно переміщувати по таблиці. А також в цій панелі знаходиться зручний пошук по БД.

Кожний запис має зліва кнопку (маркер запису). Натиснення лівої клавіші миші на цьому маркері виділяє весь запис і готує його до копіювання, переміщення, видалення.

Натиснення правою кнопкою на виділеному записі відкривається контекстне меню для операцій із записом.

Маркер, що знаходиться в лівому верхньому кутку таблиці – це маркер таблиці. Натиснення лівої кнопки миші виділяє всю таблицю, а правої – відкриває контекстне меню для операцій із таблицею в цілому.

Поля БД подані в таблиці стовпцями. Кожний стовпець має заголовок, у якому записане ім'я поля або те значення, яке задане у властивості «Підпис».

Якщо вміст поля не повністю розміщується в комірці таблиці, стовпець можна розширити. При наведенні покажчика миші на границю між стовпцями покажчик змінює форму. Тепер границю можна переміщувати методом перетягування, а подвійне натиснення лівої кнопки миші, виконане у цей момент, автоматично встановлює ширину стовпця рівній довжині найдовшого значення в даному полі.

Натиснення лівої кнопки миші на заголовку стовпця виділяє весь стовпець, а натиснення правою кнопкою на виділеному стовпці відкриває контекстне меню. У ньому є пункти, що дозволяють відсортувати записи по даному полю, вставити новий стовпець, сховати стовпець та інше.

Схований стовпець не зникає з бази, а тільки перестає відображатися на екрані. Щоб знову його відобразити, треба навести покажчик на границю між стовпцями в тому місці, де був схований стовпець, і виконати подвійне натиснення лівої кнопки миші. Схований стовпець знову стане видимим.

Завдання для самостійної роботи.

1. Створити у базі даних таблицю *opis* у режимі Конструктор згідно структури

Таблиця 5

Структура таблиці *opis*

Ім'я поля	Тип даних	Властивості поля
1	2	3
ID	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
hole_id	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
Poroda	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
depth_from	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
depth_to	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
AL2O3	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
FE2O3	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
PPP	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
OGNEUPOR N	Числовой	Размер поля – Одинарное с плавающей точкой Обязательное поле – Нет Индексированное поле – Нет
Sort	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет
samp_id	Тектовий	Размер поля – 255 Обязательное поле – Нет Индексированное поле – Нет

2. Заповнити таблицю опис

ID	hole_id	Poroda	depth_from	depth_to	AL2O3	FE2O3	PPP	OGNEUPORI	Sort	samp_id
85821	7521	Суглинок коричневый	0,3	8					Без сортирності	
85822	7521	Песок пестроцветный	8	9,2					Без сортирності	
85823	7521	Песок кварцевый	9,2	17,7					Без сортирності	
85824	7521	Глина огнеупорная	17,7	18,1	19,46	2,65	5,36		ПК 2	
85825	7521	Глина огнеупорная	18,1	18,4	21,25	3,28	5,81		ДНПК	
85826	7521	Глина огнеупорная	18,4	18,9	33,45	1,76	9,6		ДН 2	
85827	7521	Глина огнеупорная	18,9	19,6	31,53	1,76	9,06		ДН 2	
85828	7521	Песок глинистый	19,6	22					Без сортирності	
85829	7522	Почвенно-растительный сл	0	0,5					Без сортирності	
85830	7522	Суглинок коричневый	0,5	19					Без сортирності	
85831	7522	Песок пестроцветный	19	21,3					Без сортирності	
85832	7522	Песок кварцевый	21,3	26,3					Без сортирності	
85833	7522	НК кровля	26,3	27	13,84	1,61	3,43		НК	
85834	7522	Глина огнеупорная	27	28,1	23,27	3,76	6,56		ДНПК	
85835	7522	Глина огнеупорная	28,1	28,9	34,39	4,01	9,9		ДНПК	
85836	7522	Глина огнеупорная	28,9	29,9	31,94	4,3	9,4		ДНПК	
85837	7522	Песок глинистый	29,9	31					Без сортирності	
85838	7523	Почвенно-растительный сл	0	0,5					Без сортирності	
85839	7523	Суглинок коричневый	0,5	20,1					Без сортирності	
85840	7523	Песок пестроцветный	20,1	23,7					Без сортирності	
85841	7523	Песок кварцевый	23,7	27,5					Без сортирності	
85842	7523	Глина огнеупорная	27,5	27,8	21,04	1,62	5,1		ДНПК	
85843	7523	Глина огнеупорная	27,8	28,8	22,68	3,26	5,93		ДНПК	
85844	7523	Глина огнеупорная	28,8	29,3	29,02	3,52	8,25		ДНПК	
85845	7523	Глина огнеупорная	29,3	29,7	30,9	2,42	8,77		ДН 3	
85846	7523	Глина огнеупорная	29,7	30	31,01	0,89	8,59		ДН 2	
85847	7523	Песок глинистый	30	32,5					Без сортирності	

Рис. 6. Вихідні дані для заповнення таблиці опис

Результат роботи.

Файл бази даних, який складається з двох таблиць з вихідними даними: collar та опис. Усі таблиці повинні мати ключові поля.

Питання для самоконтролю.

1. Що таке реляційна база даних?
2. Яку структуру та розширення має файл бази даних, створений у Microsoft Access?
3. Для чого призначені таблиця, форма, запит, звіт? Дайте визначення цих понять.
4. Якими способами можна створювати таблиці у базі даних Microsoft Access?
5. Які обмеження накладаються на імена полів таблиці?
6. Для чого використовується режим таблиці та режим конструктора? В чому між ними різниця?
7. Що таке нормалізація таблиць. Яка таблиця має назву нормалізованої?
8. Для чого призначено поле типу MEMO та Вложение?