

Елементи теорії баз даних.

Поняття, види, етапи роботи з базою даних.

План

1. Поняття та види баз даних. Моделі організації даних
2. Загальна характеристика СУБД
3. Етапи роботи з базою даних

1. Поняття та види баз даних. Моделі організації даних

База даних (БД) – це структурована сукупність відомостей про об’єкти певної предметної області та окремих стандартизованих засобів для автоматизації їх обробки.

Існують не лише комп’ютерні БД. База даних обліку залізничних квитків – це комп’ютерна БД, журнал академічної групи коледжу – приклад некомп’ютерної БД.

За технологією обробки даних БД поділяють на *централізовані* та *розподілені*.

У *централізованих* БД всі дані зберігаються на одному комп’ютері, найчастіше на сервері, але доступ до них можливий з інших комп’ютерів мережі. В *розподілених* БД різні частини бази зберігаються на різних комп’ютерах. Наприклад, бази даних автоматизованих систем бухгалтерського обліку 1С: Підприємство та Парус-Підприємство є централізованими, а база електронних листів є розподіленою, бо електронні листи зберігаються на серверах і в клієнтах поштових систем.

За способом доступу до даних БД поділяють на *локальні* та *мережеві*.

Локальні БД призначені для обробки даних бази лише на одному комп’ютері, *мережеві* – для роботи з базою в мережі. Наприклад, автоматизовані системи бухгалтерського обліку в середніх і великих підприємствах використовують мережеві БД. Електронний телефонний довідник, встановлений на комп’ютері, базується на локальній БД.

За характером збереження даних розрізняють *документальні*, *фактографічні* та *документально-фактографічні* БД.

Документальні БД створені сукупністю неструктурованих текстових документів (статті, книги, реферати, тексти законів) та графічних об’єктів. Прикладом може бути сукупність файлів окремого диску комп’ютера.

Основна ідея *фактографічних* БД полягає в тому, що усі відомості про об’єкти мають свій формат. Інформація, яка заноситься до бази даних, має чітку структуру. Прикладом роботи з фактографічною БД є реєстрація обліку відвідувань та оцінювання знань студентів в журналі академічної групи.

Дані у базі організують відповідно до *моделі організації даних*. Розрізняють наступні моделі організації даних:

- *реляційна* – у цьому випадку БД являє собою одну чи декілька взаємозв’язаних двовимірних таблиць. Для кожного зв’язку можна виділити основну і підпорядковану таблиці. Прикладом БД з реляційною моделлю є БД, що містить таблиці з даними співробітників та здобутою ними освітою;

- **ієрархічна** – БД являє собою сукупність об'єктів різних рівнів, при цьому об'єкти нижчого рівня підпорядковуються об'єктам вищого рівня. Прикладом БД з ієрархічною моделлю є файлова структура дисків, оскільки каталоги нижчих рівнів вкладаються в каталоги вищих рівнів, аж до кореневого каталогу;
- **сіткова** – БД являє собою сукупність об'єктів різних рівнів, проте схема зв'язків між ними може бути довільною (кожен об'єкт може бути підпорядкований будь-якому іншому об'єкту).

Надалі під БД будемо розуміти реляційну БД з ієрархічними індексами, основними об'єктами якої є **таблиці** та **зв'язки між ними**. Структуру двовимірної таблиці утворюють стовпці і рядки. Їх аналогами в таблиці БД є **поля** і **записи**.

2. Загальна характеристика СУБД

Система управління базою даних (СУБД) – це комплекс програмних і мовних засобів загального і спеціального призначення, необхідних для створення БД, підтримування її в актуальному стані, маніпулювання даними і організації доступу до них різних користувачів. Іншими словами, якщо БД – це сховище даних, то СУБД – це **прикладна програма** для їх обробки.

Автоматизований банк даних (АБД) – це система спеціальним чином організованих даних (баз даних), програмних, технічних, мовних, організаційно-методичних засобів, які необхідні для забезпечення **централізованого нагромадження та колективного багаточільового використання даних**. Зрозуміло, що основними складовими АБД є БД і СУБД.

Існує багато систем управління базами даних, наприклад Oracle, Microsoft Visual FoxPro, dBase, Clipper, C++ Builder, Borland Delphi, Microsoft SQL Server, Microsoft Access та ін.

Інформаційна система (ІС) обробки баз даних – це база даних і комплекс апаратно-програмних засобів для збереження та маніпулювання ними, це **прикладна програма**, з якою працюють користувачі (наприклад, ІС бухгалтерського обліку, продажу залізничних квитків та ін.).

3. Етапи роботи з базою даних

Зазвичай, з БД працюють три категорії персоналу: постановники та проектувальники, розробники і користувачі.

Перша категорія – **постановники та проектувальники**. Їх завдання полягає, насамперед, в розробці структури таблиць бази даних і узгодженні її з замовником.

Друга категорія – **розробники**. Вони реалізують розроблену структуру таблиць та схему БД за допомогою обраної СУБД. Також розробники створюють і вдосконалюють інші об'єкти ІС, призначені для автоматизації роботи з БД.

Третя категорія – **користувачі**. Вони одержують ІС та початкову базу даних від розробників і займаються її веденням для задоволення інформаційних потреб організації. Користувачі опрацьовують лише ті дані, робота з якими передбачена на конкретному робочому місці.

Кожна з цих категорій персоналу уявляє дані на своєму рівні і в процесі проектування БД ці рівні мають бути узгоджені між собою.

Основи проектування баз даних

План

1. Рівні подання даних
2. Етапність проектування баз даних на зовнішньому рівні
3. Проектування баз даних на інфологічному рівні
4. Реляційний підхід до проектування баз даних
5. Нормалізація відношення баз даних

1. Рівні подання даних

Розробку БД, як і спорудження будинку, необхідно починати зі створення проекту. Адже неоптимальні рішення і явні помилки, закладені на етапі проектування, згодом дуже важко усуваються. *Основна мета проектування баз даних – забезпечення одноразового введення кожної одиниці інформації.* Саме цей принцип дозволяє в подальшому швидко вводити та ефективно опрацьовувати інформацію в БД. Отже, проектування БД виконують для усунення дублювань, надлишковостей та неузгодженостей між даними бази.

Проектування баз даних – це ітеративний процес прийняття обґрунтованих рішень з боку постановників, програмістів і користувачів, синтезу фізичних і логічних структурних даних, аналізу та обґрунтування вибору програмних і апаратних засобів.

Наголосимо ще раз, що етапність проектування БД пов'язана з багаторівневою організацією даних для різних категорій користувачів.

Розрізняють **чотири рівні подання даних**:

- 1) зовнішній;
- 2) інфологічний;
- 3) даталогічний;
- 4) внутрішній.

На **зовнішньому** рівні вивчається все позамашинне інформаційне забезпечення, форми документування та подання даних, методи фіксації, передавання та обробки інформації. При проектуванні на цьому рівні необхідно вивчити функціонування об'єкта управління, всю вхідну та вихідну інформацію і визначитися, які саме дані доцільно зберігати в базі.

Результатом проектування на зовнішньому рівні є перелік задач, які мають вирішуватися системою та перелік атрибутів, які будуть зберігатися в БД. Проектування на цьому рівні не включає елементів дублювання надлишковостей та неузгодженостей. Усуненням цих протиріч займаються на наступному рівні проектування. Проектуванням на зовнішньому рівні займається **постановник**.

На **інфологічному** рівні здійснюється структуризація даних. Тут будується інформаційно-логічна модель предметної області, в якій усуваються дублювання, надлишковість та неузгодженість і відображаються інформаційні особливості об'єкта управління без урахування особливостей та обмежень конкретної СУБД. Фактично на цьому рівні визначається структура таблиць та встановлюються зв'язки між ними. Розробкою БД на цьому рівні займається **проектувальник**.

При проектуванні на **дatalogічному** рівні будується інформаційно-логічна модель предметної області з урахуванням особливостей та обмежень конкретної СУБД. Тут формується концептуальна модель даних, визначаються їх тип та характеристики. Проектування на цьому рівні виконують **програмісти** та кваліфіковані **розробники**.

На **внутрішньому** рівні формується опис фізичного розміщення даних в пам'яті ПК:

- 1) визначаються формати записів;
- 2) порядок їх фізичного впорядкування;
- 3) характеристики та шляхи доступу до даних.

Від параметрів фізичної моделі залежить обсяг пам'яті, яку займає БД, та час реакції системи. Параметри фізичної моделі можуть змінюватись в процесі експлуатації, але ці зміни не повинні зумовлювати зміни на вищих рівнях. Проектування на внутрішньому рівні виконують **розробники СУБД** та **адміністратор**.

Проекти зовнішнього, інфологічного та дatalogічного рівня мають обов'язково узгоджуватись зі словником даних.

2. Етапність проектування баз даних на зовнішньому рівні

Для створення БД спочатку розробляють **технічне завдання** на зовнішньому рівні проектування. Є два підходи до проектування на зовнішньому рівні:

- 1) «від запиту»;
- 2) «від предметної області».

Основні етапи проектування на зовнішньому рівні:

1) визначення функцій предметної області і проведення аналізу її особливостей. Результатом цього етапу є перелік задач, що має вирішуватися ІС. Наприклад, перелік задач для автоматизації діяльності продуктового складу такий:

- облік товарів;
- облік постачальників;
- облік клієнтів;
- облік співробітників та відділи, де вони працюють;
- облік постачань;
- облік замовлень (реалізацій).

А перелік задач для автоматизації домашньої бібліотеки наступний:

- облік авторів;
- облік книг;
- облік читачів;
- облік книговидач;

2) вивчення та аналіз первинних документів. Результатом цього етапу є перелік атрибутів, що мають зберігатися в таблицях оперативної інформації;

3) вивчення нормативно-довідкових документів. Результатом цього етапу є перелік атрибутів, що мають зберігатися в таблицях умовно-постійної інформації;

- 4) вивчення процесу перетворення вхідної інформації у вихідну.

Якщо якусь вихідну інформацію не вдається отримати з вхідної, то виконують дообстеження предметної області, повертаючись до попередніх етапів.

3. Проектування баз даних на інфологічному рівні

Суть інфологічного проектування полягає у виділенні інформаційних об'єктів та встановленні зв'язків між ними.

Є два підходи до проектування на інфологічному рівні:

- 1) аналіз об'єктів (використовується у процесі створення документальних систем);
- 2) синтез атрибутів (використовується для розробки фактографічних систем).

Виділяють **п'ять складових інформаційної моделі**:

- 1) **інформаційний об'єкт** (таблиця) – це сутність, що має відобразитися в БД. Кожен об'єкт характеризується своїми властивостями, властивості складаються з неподільних атрибутів.

Для кожного об'єкта виділяють поняття типу та екземпляра. **Тип** — це перелік атрибутів об'єкта. **Екземпляр** – це сукупність значень атрибутів, що описує окремого представника об'єкта чи їх взаємодію.

- 2) **атрибут** (реквізит, поле) – це логічнонеподільна поіменована одиниця інформації;
- 3) **структурний зв'язок** — це зв'язок між парами інформаційних об'єктів, один з яких виступає власником а інший – підпорядкованим об'єктом;
- 4) **запит** — це словесний опис потреб користувача;
- 5) **запитальний зв'язок** — реалізація запиту з відображенням об'єктів та зв'язків між ними, які необхідні для його використання.

В основі інфологічного проектування лежить процес аналізу співвідношень між атрибутами, та виділення на основі цього аналізу окремих об'єктів і встановлення зв'язків між ними. Співвідношення встановлюється **не між значеннями**, а між **кількостями** атрибутів.

Різновиди співвідношень:

- 1) 1:1 (один до одного) – атрибути перебувають в такому співвідношенні якщо кожному представнику атрибута з сукупності A1 відповідає не більше одного представника атрибута з сукупностей A2 і навпаки;
- 2) 1:∞ (один до багатьох) – кожному представнику атрибута з сукупності A1 може відповідати один або декілька представників з сукупності A2, але кожному представнику з сукупності A2 відповідає не більше одного представника з сукупності A1.
- 3) ∞:1 – обернене до попереднього;
- 4) ∞:∞ (багато до багатьох) – атрибут перебуває у співвідношенні багато до багатьох, якщо кожному представнику атрибута з сукупності A1 може відповідати один або декілька представників з сукупності A2 і кожному представнику з сукупності A2 може відповідати один або декілька представників з сукупності A1.

Етапи проектування на інфологічному рівні:

- 1) вилучення синонімії та омонімії і проведення процесу узагальнення атрибутів;

2) аналіз співвідношень між атрибутами та виділення інформаційних об'єктів. Для цього:

- виділяють атрибути, між якими встановлено співвідношення 1:1 в окремий об'єкт, присвоюють йому назву, а також виділяють в кожному об'єкті ключове поле. Ключове поле використовується для однозначної ідентифікації екземплярів об'єкта (наприклад, в таблиці *Співробітники* – це поле *КодСпівробітника*). Ключове поле (чи сукупність ключових полів) має обов'язково бути унікальним і непорожнім;
- атрибути, між якими встановлено співвідношення 1:∞ або ∞:1, розміщують в різних таблицях і створюють між цими таблицями структурний зв'язок яких 1:∞.

Якщо об'єкт взаємодіє з іншим об'єктом, то в таблиці, що описує взаємодію, встановлюють лише значення ключових полів об'єктів, які взаємодіють між собою, та поля, що функціонально залежать від встановленого зв'язку. Як правило, для ключових полів встановлюється тип даних лічильник (счетчик), а для полів, які вказують на використання об'єкта – числовий тип даних.

- якщо між атрибутами виявляють співвідношення ∞:∞, то їх розміщують в різних таблицях та створюють одну чи декілька проміжних таблиць, між якими встановлюють співвідношення 1:∞;

3) виконання зовнішнього кодування, в процесі якого замінюють довгі назви кодами та створюють відповідні довідники;

4) виділення запитів, при якому здійснюють їх опис та опис ходу їх розв'язання;

5) побудова графічного зображення інфологічної моделі з відображенням структурних зв'язків між інформаційними об'єктами.

Для ілюстрації проекту інфологічного рівня та типів можливих зв'язків між таблицями скористаємося схемою даних учбової бази *Борей* (рис. 1), яка є базою для обліку реалізацій товарів торгівельною компанією.

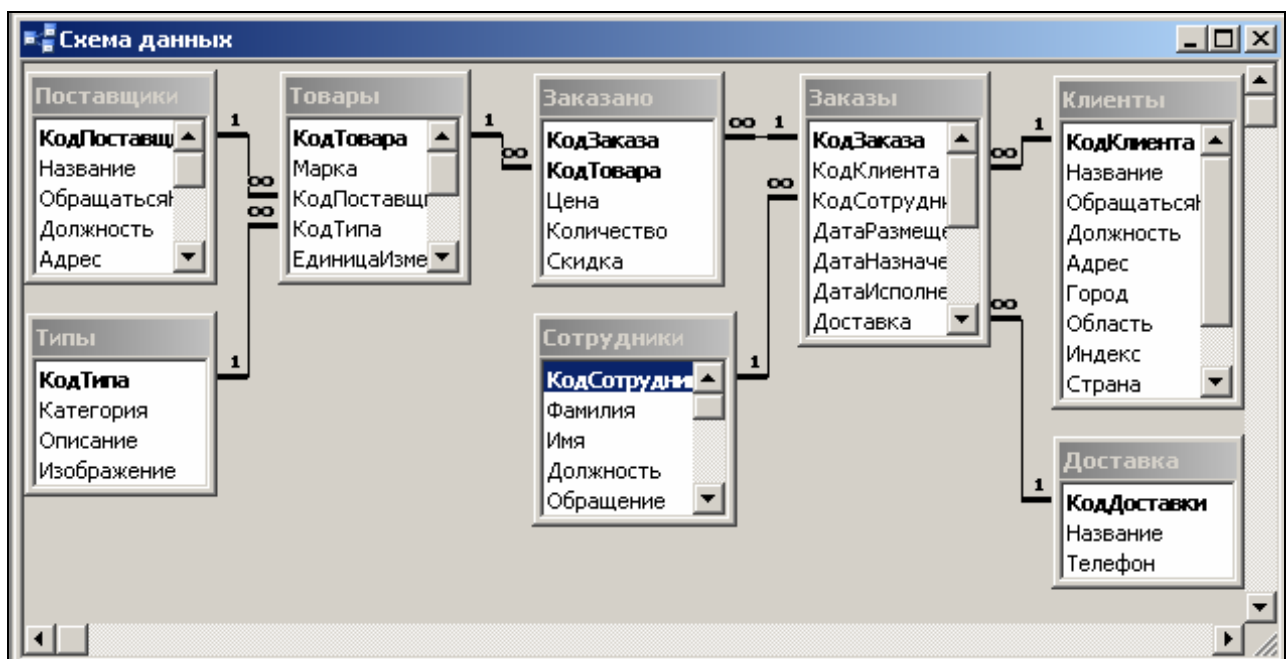


Рис. 1. Схема даних ІС торгівельної компанії Борей

Розглянемо таблицю *Типи*, призначену для зберігання можливих типів товарів. У таблиці *Типи* поле *КодТипу* є ключовим. Це зрозуміло, оскільки у кожного типу товару повинен бути свій унікальний код, який ідентифікує його однозначно. Якщо ми розглянемо таблицю *Товари*, де зберігаються дані про товари, реалізацією яких займається торгова компанія *Борей*, то побачимо, що в ній *КодТипу* вказує на тип, до якого належить товар. Воно не може бути унікальним, оскільки існує багато товарів одного і того ж типу, наприклад *молочні продукти*. На схемі даних ці поля графічно сполучені зв'язком 1:∞.

Ще один приклад. Ключовим полем в таблиці *Замовлення* є *КодЗамовлення* – воно однозначно ідентифікує, хто і коли зробив замовлення. Тут також можна дізнатися, який співробітник прийняв замовлення для виконання. Оскільки один співробітник може прийняти безліч замовлень, то поле *КодСпівробітника* в таблиці *Замовлення* не є ні унікальним, ні ключовим, зате в таблиці *Співробітники* це поле унікальне і непорожнє.

Про подібні таблиці кажуть, що вони зв'язані *реляційними відношеннями*. Відповідно, системи управління, здатні працювати із пов'язаними таблицями, називають *системами управління реляційними базами даних*.

Розробкою схеми даних закінчується "паперовий" етап роботи над технічною пропозицією. Цю схему можна погоджувати із замовником, після чого приступати до безпосереднього створення бази даних за допомогою обраної СУБД.

4. Реляційний підхід до проектування баз даних

Альтернативою інфологічному проектуванню виступає реляційний підхід.

Реляційна БД — це набір взаємозв'язаних відношень. Кожне відношення має вигляд двовимірної таблиці. Всі відношення поділяються на об'єктні та зв'язкові.

В **об'єктних відношеннях** зберігаються дані про об'єкти предметної області. Серед всіх полів такого відношення (атрибутів) має бути хоча б одне ключове (тобто унікальне і непорожнє) поле чи ключова сукупність полів. Воно однозначно ідентифікує об'єкт.

Зв'язкове відношення встановлює зв'язок між двома чи більше об'єктними відношеннями (наприклад, зв'язкове відношення *Заголовки замовлень* встановлює зв'язок між об'єктними відношеннями *Співробітники* і *Клієнти*). В такому відношенні зберігається значення ключів об'єктних відношень. Крім цього, у зв'язкових відношеннях можуть міститися інші атрибути, які функціонально залежать від встановленого зв'язку (наприклад: *ДатаЗамовлення*).

Ключі, які використовують у об'єктних відношеннях називають **первинними**, а у зв'язкових відношеннях – **вторинними**. Між відношеннями забезпечується посилкова цілісність.

Цілісність — це забезпечення достовірності та узгодженості даних в будь-який момент часу. Для зв'язків між відношеннями цілісність означає, що для будь-якого значення *вторинного* ключа має існувати *первинний* ключ в об'єктивному відношенні. Крім цього, всі відношення повинні відповідати умовам нормалізації.

5. Нормалізація відношення баз даних

Відношення перебувають у **першій** нормальній формі, якщо:

- 1) всі назви атрибутів унікальні і непорожні;
- 2) всі рядки мають однакову структуру;
- 3) значення кожного атрибуту у всіх рядках мають однаковий формат.

Виконання першої нормальної форми забезпечується СУБД автоматично.

Відношення перебуває у **другій** нормальній формі, якщо в них відсутні неповні функціональні залежності, тобто значення кожного неключового атрибута залежить від всього ключа.

Відношення перебувають у **третьій** нормальній формі, якщо в них відсутні транзитивні залежності між неключовими атрибутами.

Відношення перебувають в **четвертій** нормальній формі, якщо в них відсутні сукупності багатозначних залежностей.

Відношення перебувають у **п'ятій** нормальній формі, якщо в кожному з них присутній принаймні один ключовий і один неключовий атрибут.

Створення та робота з БД в середовищі MS Access

План

1. Особливості та типи об'єктів СУБД MS Access
2. Інтерфейс СУБД MS Access. Основні режими використання об'єктів в СУБД MS Access
3. Сервісні функції обробки БД в СУБД MS Access

1. Особливості та типи об'єктів СУБД MS Access

Не дивлячись на те що СУБД можуть по-різному працювати з різними об'єктами і надають користувачу різні функції і засоби, більшість СУБД опираються на єдиний комплекс основних понять. Це дає нам можливість розглянути одну систему і узагальнити її поняття, прийоми та методи на весь клас СУБД. В якості такої учбової системи ми використаємо СУБД MS Access, яка входить до пакету Microsoft Office поряд з MS Word та MS Excel, оскільки вона дає змогу створювати БД у візуальному інтерактивному режимі без знання основ програмування. Об'єкти БД MS Access можуть міститися в одному файлі, але кожен з них належить до одного з наперед визначених типів.

Основні типи об'єктів MS Access:

1) **таблиці** – це основні об'єкти будь-якої бази даних. По-перше, в таблицях зберігаються всі дані, наявні в базі, а по-друге, у таблицях крім даних зберігається ще й їх структура;

2) **форми** – це засоби для введення, аналізу даних та керування іншими об'єктами. Основне їх призначення – надати користувачу засоби для заповнення тільки тих полів, які йому призначено заповнювати. У формі можна розмістити спеціальні елементи управління (лічильники, списки, що розкриваються, перемикачі, прапорці та інші) для автоматизації введення.

Переваги форм розкриваються особливо наочно, коли відбувається введення даних із заповнених бланків. У цьому випадку форму роблять графічними засобами такою, щоб вона повторювала оформлення бланка – це помітно спрощує роботу оператора, знижує його втомлюваність і запобігає появі друкарських помилок.

За допомогою форм дані можна не тільки вводити, але і відображати. При виведенні даних за допомогою форм можна застосовувати спеціальні засоби оформлення.

3) **запити** використовуються для аналізу чи пакетного корегування даних таблиць. Результати аналізу запитів подаються користувачеві в табличному вигляді. Особливістю запитів є те, що вони дозволяють обробляти інформацію з багатьох таблиць одночасно.

За допомогою запитів можливо:

- виконувати відбір, сортування, фільтрацію, перетворення даних по заданому алгоритму;
- створювати нові таблиці;
- виконувати автоматичне наповнення таблиць даними, імпортованими з інших джерел;
- здійснювати обчислення над даними таблиць і багато іншого;

4) **звіти** – це паперові чи електронні документи з результатами обробки інформації БД. За своїми властивостями і структурою звіти багато в чому схожі на форми, але призначені вони тільки для виведення даних, причому не стільки для виведення на екран, скільки для виведення на друкуючий пристрій (принтер). У зв'язку з цим звіти відрізняються тим, що в них вжиті спеціальні засоби для групування даних і для виведення спеціальних елементів оформлення, характерні для друкованих документів (верхній і нижній колонтитули, номери сторінок, службова інформація про час створення звіту і т. п.);

5) **сторінки** (коректніше їх називати **сторінками доступу до даних**) – це спеціалізовані файли, виконані в коді HTML, які зберігаються поза БД і використовуються для публікації інформації БД в локальній чи глобальній мережі.


Під час завантаження сторінка доступу звертається до БД і бере звідти всю необхідну інформацію, тому такі сторінки є динамічними. Сам по собі такий об'єкт не є базою даних, але він містить компоненти, через які здійснюється зв'язок переданої Web-сторінки з базою даних, що залишається на сервері;

6) **макриси** призначені для автоматизації виконання типових операцій при роботі з БД та СУБД. В MS Access макриси складаються з послідовності внутрішніх команд СУБД.

7) **модулі** призначені для створення нових функцій засобами програмування. Модулі створюються засобами зовнішньої мови програмування. Для MS Access такою мовою є Visual Basic for Applications. Це один із засобів, за допомогою якого розробник бази може задовольнити специфічні вимоги замовника, підвищити швидкість системи управління, а також рівень її захищеності.

Наголосимо, що дані про об'єкти предметної області чи про їх взаємодію зберігаються тільки у взаємопов'язаних реляційних таблицях. Всі інші об'єкти – це інструментальні засоби для підвищення ефективності обробки даних.

2. Інтерфейс СУБД MS Access. Основні режими використання об'єктів в СУБД MS Access

Для переходів між об'єктами в БД Access використовується *Область переходов (навігації)*, яка розміщується зліва від робочого поля (рис. 2). З метою організації впорядкованого доступу до всіх об'єктів БД доцільно, по-перше, розгорнути цю область за допомогою кнопки  у її верхньому правому кутку, по-друге, згрупувати об'єкти за типами і, по-третє, відобразити всі об'єкти БД (як на рис. 2).

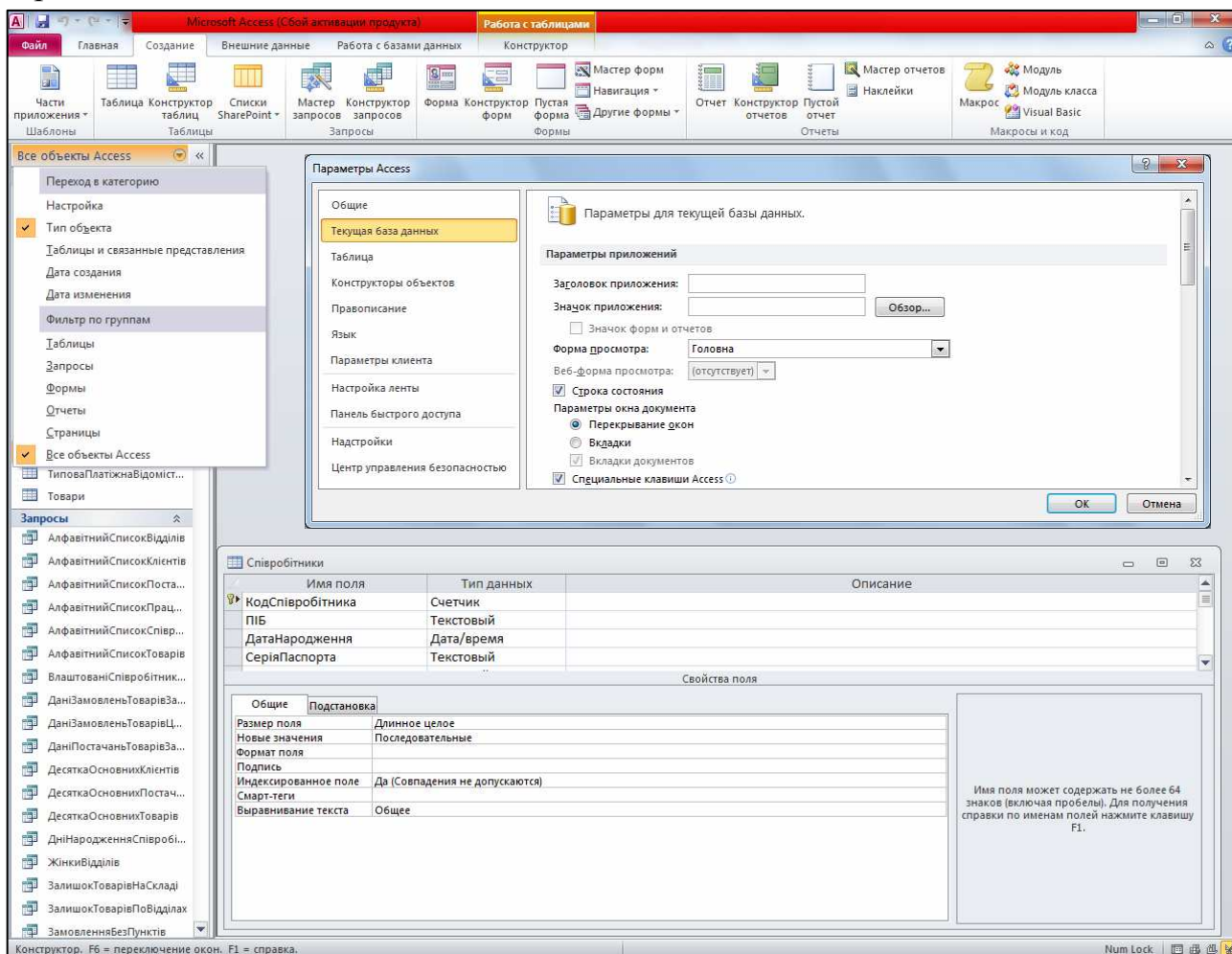



Рис. 2. Область переходів, вікно параметрів та вікно таблиці в режимі конструктора

Всі об'єкти БД Access можуть відображатися в окремих вікнах або вкладках. Встановлення режиму відображення об'єктів здійснюється у вікні параметрів Access (*Файл ► Параметри*) на вкладці *Текущая база данных* (див. рис. 2). Ми рекомендуємо відображати об'єкти у вигляді окремих вікон, оскільки це дозволяє ефективніше співставляти їх дані.

Використовуючи область переходів, кожен об'єкт БД Access можна відразу завантажити в **режимі застосовування** (для таблиць і запитів – це режим таблиці, для форм – режим форми, для звітів – режим попереднього перегляду чи його подання). Цей самий режим використовується для об'єктів БД Access при виборі в контекстному меню їх назви пункту *Открыть*. Структура візуальних об'єктів БД Access змінюється в режимі **конструктора** (див. нижню частину рис. 2). Для переходу в цей режим потрібно використати однойменний пункт контекстного меню об'єкта в області переходів чи заголовка вікна завантаженого об'єкта. Інші режими використання об'єктів залежать від їх типу (наприклад, для форм і звітів

використовується також режим **макету**). Для швидкого переходу між режимами використання відкритого об'єкта ми рекомендуємо застосовувати першу кнопку *Режим* з вкладки *Главная* стрічки меню та її список (на місці кнопки *Частини приложения* на рис. 2), оскільки ця кнопка дає змогу відразу перейти до найуживанішого альтернативного режиму (наприклад, для режиму застосування –



це режим конструктора, і тому ця кнопка має вигляд , а її список – до довільного іншого режиму. При цьому об'єкт не потрібно щоразу закривати і відкривати знову.

3. Сервісні функції обробки БД в СУБД MS Access

Серед сервісних функцій обробки БД, які надає СУБД MS Access, використаємо в цій лабораторній роботі дві найуживаніших:

- **стиснення БД** – це автоматичне виконання модифікацій БД на фізичному рівні з метою зменшення розміру файлу БД без внесення змін в логічну модель. При цьому самі дані не стискаються, а з файлу вилучаються об'єкти і дані, що вже не використовуються (наприклад, при видаленні записів з таблиць вони лише помічаються як видалені, а фізично знищуються при стисненні БД). За рахунок зменшення розміру файлу БД стиснення пришвидшує обробку її об'єктів. Ми рекомендуємо стискувати БД в процесі розробки після кожного сеансу роботи, а під час експлуатації – хоча б раз на тиждень;
- **конвертування БД** – це автоматичне зберігання вмісту БД у новому файлі іншого формату без внесення змін в дані та логічну модель. Зрозуміло, що СУБД може виконувати конвертування до і з одного з попередніх форматів. Найефективніше СУБД працює з активним форматом, а для перетворення БД до попереднього формату версій Access необхідно вилучити з неї всі складові частини, що використовують нові можливості, які не підтримуються цим попереднім форматом.