

УДК 004.65:030(477)

В. В. Тарасюк

м. н. с. відділу науково-інформаційного

та науково-технічного забезпечення

Інституту енциклопедичних досліджень НАН України

ЕЛЕКТРОННА БАЗА ЕНЦИКЛОПЕДІЇ СУЧАСНОЇ УКРАЇНИ І ЇЇ ОСНОВНІ КОМПОНЕНТИ ТА ФУНКЦІЇ

У публікації подано класифікацію баз даних за їх будовою. Визначено поняття «База даних», «Система управління базою даних». Висвітлено історію створення бази даних Енциклопедії сучасної України, її основні елементи та функціональне призначення.

Ключові слова: база даних, класифікація, система управління базою даних, робоча форма бази даних.

The publication provides a classification database for their structure. Definition of concepts such as «Database», «database management system». Submitted the history of the database Encyclopedia of modern Ukraine, her main elements and functionality.

Key words: database, classification, database management system, working form database.

База даних (БД) – сукупність даних, організованих відповідно до концептуальної структури, яка описує характеристики цих даних і взаємовідносини між ними, причому таке зібрання даних, яке підтримує одну або більше сфер застосування. Стандарт ISO/IEC TR 10032:2003.

Модель даних – це абстрактне, самодостатнє, логічне визначення об'єктів, операторів та інших елементів, які у сукупності являють собою абстрактну машину доступу до даних, з якою взаємодіє користувач. Ці об'єкти дозволяють моделювати структуру даних, а оператори – поведінку даних.

Виходячи з особливостей будови, бази даних можна поділити на три типи: ієрархічні, мережеві, реляційні. Найбільш поширеними є реляційні БД, тому що в них інформація подається у найбільш зручній для сприйняття спосіб – у вигляді таблиць та відношень між ними.

Ієрархічна модель бази даних складається з об'єктів з покажчиками від батьківських об'єктів до нащадків, з'єднуючи разом зв'язану інформацію. Ієрархічні бази даних можуть бути представлені як дерево, що складається з об'єктів різних рівнів. Верхній рівень займає один об'єкт, другий – об'єкти другого рівня і т.д. Між об'єктами існують зв'язки, кожний об'єкт може містити в собі кілька об'єктів більш низького рівня. Такі об'єкти перебувають відносно предка (об'єкт більш близький до кореня) до нащадка (об'єкт більш низького рівня), при цьому можливо, що об'єкт-предок не має потомків або має їх декілька, тоді як в об'єкта-нащадка обов'язково тільки один предок. Об'єкти, що мають спільного предка, називаються близнюками.

До основних понять **мережевої моделі бази даних** належать: рівень, елемент (вузол), зв'язок. Вузол – це сукупність атрибутів даних, що описують певний об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. У мережній структурі кожний елемент може бути пов'язаний з будь-яким іншим елементом. Мережеві бази даних подібні до ієрархічних, за винятком того, що їхні зв'язки функціонують в обох напрямках. Незважаючи на те, що ця модель вирішує деякі проблеми, пов'язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом. Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, тому ця модель не є повністю не-

залежною від додатка. Інакше кажучи, якщо необхідно змінити структуру даних, то потрібно змінити й додаток.

Реляційна модель. Термін «реляційний» означає, що теорія заснована на математичному понятті відношення (relation). У якості неформального синоніма терміна «відношення» частіше трапляється слово таблиця. Необхідно пам'ятати, що «таблиця» є досить абстрактним поняттям і часто означає не «відношення» як загальноживане поняття, а візуальне розміщення відношень на папері або екрані. Некоректне й нестроге використання терміна «таблиця» замість терміна «відношення» часто приводить до непорозуміння. Найчастіше помилка полягає в міркуваннях про те, що реляційна модель даних має справу з «плоскими», або «двовимірними» таблицями, тоді як такими можуть бути тільки візуальні представлення таблиць. Відношення ж є абстракціями, і не можуть бути ні «плоскими», ні «неплоскими».

Система управління базою даних (СУБД) – це програмне забезпечення, що дозволяє створювати БД й оновлювати інформацію, яка зберігається в ній. СУБД забезпечує зручний доступ до інформації з метою перегляду, пошуку, аналізу й обробки за допомогою спеціального програмного та апаратного забезпечення.

Класифікація СУБД за моделями даних.

Кожна БД і СУБД будується на основі деякої явної або неявної моделі даних. Усі СУБД, побудовані на тій самій моделі даних, відносять до одного типу. Наприклад, основою реляційних СУБД є реляційна модель даних, мережевих СУБД – мережева модель даних, ієрархічних СУБД – ієрархічна модель даних і т.д.

Ієрархічна модель підтримує деревоподібну організацію інформації. Зв'язки між записами виражаються у вигляді відношень предок/нащадок, а в кожного запису є рівно один батьківський запис. Це допомагає підтримувати цілісність посилань. Коли запис видаляється з дерева, усі його нащадки також повинні бути вилучені. Ієрархічні бази даних мають централізовану структуру, тобто безпеку даних легко контролювати. На жаль, певні знання про фізичний порядок зберігання записів все-таки необхідний, тому що відношення предок/нащадок реалізуються у вигляді фізичних покажчиків з одного запису на інший. Це означає, що пошук запису здійснюється методом прямого обходу дерева. Записи, розташовані в одній половині дерева, знаходяться швидше ніж в іншій. Звідси випливає необхідність правильно впорядковувати записи, щоб час їх пошуку був мінімальним. Це важко, тому що не всі відношення, що існують у реальному світі, можна виразити за допомогою ієрархічної моделі даних. Відношення «один до багатьох» є природними, але практично неможливо описати відношення «багато до багатьох» або ситуації, коли запис має декілька предків. Доти, поки в додатках будуть кодуватися відомості про фізичну структуру даних, будь-які зміни цієї структури будуть перепрограмуватися.

Мережева модель розширює ієрархічну модель, дозволяючи групувати зв'язки між записами. З логічної точки зору зв'язок – це не сам запис. Зв'язки лише виражають відношення між записами. Як і в ієрархічній моделі, зв'язки ведуть від батьківського запису до дочірнього, але цього разу підтримується множинне спадкування. Виходячи зі специфікації CODASYL, мережева модель підтримує DDL (Data Definition Language – мова визначення даних) і DML (Data Manipulation Language – мова обробки даних). Це спеціальні мови, призначені для визначення структури бази даних і складання запитів. Незважаючи на їхню наявність, програміст, як і раніше, повинен знати структуру бази даних. У мережевій моделі допускаються відношення «багато до багатьох», а записи не залежать один від одного. При видаленні запису видаляються й усі його зв'язки, а не самі зв'язані записи. У мережевій моделі потрібно, щоб зв'язки встановлювалися між наявними записами, щоб уникнути дублювання й спотворення цілісності. Дані можна ізолювати у відповідних таблицях і зв'язати із записами в інших. Програмісту не потрібно думати про те, як організовано фізичне зберігання даних на диску. Це зменшує залежність між програмами і даними. Але в мережевій моделі

потрібно, щоб програміст пам'ятав структуру даних при формуванні запитів. Оптимальну структуру бази даних складно сформувати, а готову структуру важко змінити. Якщо вид таблиці піддається змінам, усі відношення з іншими таблицями повинні бути встановлені заново, щоб не порушилася цілісність даних. Складність подібного завдання призводить до того, що програмісти найчастіше скасовують деякі обмеження цілісності заради спрощення додатків.

Реляційна модель, порівняно з розглянутими вище, вимагає від СУБД набагато вищого рівня складності. У ній робиться спроба позбавити програміста виконання рутинних операцій з керування даними, характерних для ієрархічної та мережевої моделей. У реляційній моделі база даних являє собою централізоване сховище таблиць, що забезпечує одночасний безпечний доступ до інформації з боку багатьох користувачів. У рядках таблиць частина полів містить дані, що ставляться безпосередньо до запису, а частина – посилання на записи інших таблиць. Отже, зв'язки між записами є невід'ємною властивістю реляційної моделі. Кожний запис таблиці має однакову структуру. Наприклад, у таблиці, що містить опис автомобілів, у всіх записів буде той самий набір полів: виробник, модель, рік випуску, пробіг і т.д. Такі таблиці легко зображувати в графічному вигляді. У реляційній моделі досягається інформаційна й структурна незалежність. Записи не зв'язані між собою настільки, щоб зміна одного з них торкнулася інших, а зміна структури бази даних не обов'язково приводить до перекомпіляції додатків, які з нею працюють. У реляційних СУБД застосовується мова SQL (Structured query language – мова структурних запитів), що дозволяє формулювати довільні, нерегламентовані запити. Це мова четвертого покоління, тому будь-який користувач може швидко навчитися робити запити. До того ж, існує безліч додатків, що дозволяють будувати логічні схеми запитів у графічному вигляді. Усе це відбувається за рахунок жорсткості вимог до продуктивності комп'ютерів. На щастя, сучасних обчислювальних потужностей більш ніж достатньо.

Реляційні бази даних страждають від відмінностей у реалізації мови SQL, хоча це – не проблема реляційної моделі. Кожна реляційна СУБД реалізує певну підмножину стандарту SQL та набір унікальних команд, що ускладнює завдання програмістам, які намагаються перейти від однієї СУБД до іншої. Доводиться робити нелегкий вибір між максимальною поширеністю та максимальною продуктивністю. У першому випадку потрібно дотримуватися мінімального загального набору команд, підтримуваних СУБД. В іншому випадку програміст просто зосереджується на роботі конкретної СУБД, використовуючи переваги її унікальних команд і функцій.

SQL – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних. Сам по собі SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не будучи мовою програмування в тому розумінні, як C або Pascal, SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати в якості інструкцій для керування даними. Стандарт SQL, крім того, вміщує функції для визначення зміни, перевірки і захисту даних. SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції. SQL також містить CLI (Call Level Interface) для доступу і управління базами даних дистанційно.

Об'єктно-орієнтовані (ОО) – дозволяють програмістам, які працюють із мовами третього покоління, інтерпретувати всі свої інформаційні властивості як об'єкти, що зберігаються в оперативній пам'яті. Додатковий інтерфейсний рівень забезпечує перехоплення запитів, що звертаються до тих частин бази даних, які перебувають у постійному сховищі на диску.

Зміни, внесені в об'єкти, оптимально переносяться з пам'яті на диск. Перевагою ОО СУБД є спрощений програмний код. Додатки одержують можливість інтерпретувати дані в контексті тієї мови програмування, якою вони написані. Реляційна база даних повертає значення всіх полів у текстовому вигляді, а потім вони приводяться до локальних типів даних. В ОО БД цей етап ліквідований. Методи маніпулювання даними завжди залишаються однаковими незалежно від того, перебувають дані на диску чи у пам'яті. Дані в ОО СУБД здатні набутися вигляду будь-якої структури, яку можна виразити використовуюваною мовою програмування. Відношення між сутностями також можуть бути довільно складними. ОО БД керує кеш-буфером об'єктів, переміщаючи при потребі об'єкти між буфером і дисковим сховищем.

За допомогою ОО СУБД вирішуються дві проблеми. По-перше, складні інформаційні структури виражаються в них краще, ніж у реляційних базах даних, по-друге, усувається необхідність перетворювати дані з того формату, який підтримується в СУБД. Наприклад, у реляційній СУБД розмірність цілих чисел може становити 11 цифр, а у використовуваній мові програмування – 16. Програмістові доведеться враховувати цю ситуацію.

Об'єктно-орієнтовані СУБД виконують багато додаткових функцій. Це окупається сповна, якщо відношення між даними дуже складні. У такому випадку продуктивність ОО СУБД вища, ніж у реляційних СУБД. Якщо ж дані менш складні, додаткові функції виявляються надлишковими.

В об'єктній моделі даних підтримуються нерегламентовані запити, але мовою їх написання не обов'язково є SQL. Логічне подання даних може не відповідати реляційній моделі, тому застосування мови SQL стає безпідставним. Найчастіше зручніше обробляти об'єкти в пам'яті, виконуючи відповідні види пошуку. Більшим недоліком об'єктно-орієнтованих баз даних є їхній тісний зв'язок із використовуюваною мовою програмування. До даних, що зберігаються в реляційній СУБД, можуть звертатися будь-які додатки, тоді як, наприклад, Java-об'єкт, розміщений в ОО СУБД, становитиме інтерес лише для додатків, написаних на Java.

Об'єктно-реляційні СУБД поєднують у собі риси реляційної й об'єктної моделей. Їхнє виникнення пояснюється тим, що реляційні бази даних добре працюють із вбудованими типами даних і набагато гірше – із нестандартними типами даних користувача. Коли з'являється новий важливий тип даних, доводиться або робити його підтримку в СУБД, або змушувати програміста самостійно керувати даними в додатку.

Не будь-яку інформацію є сенс інтерпретувати у вигляді послідовності символів або цифр. Уявімо собі музичну базу даних. Пісню, закодовану у вигляді аудіо-файлу, можна помістити в текстове поле великого розміру, але як у такому випадку буде здійснюватися текстовий пошук?

Перебудова СУБД із метою додавання до неї підтримки нового типу даних – не кращий вихід із ситуації. Замість цього об'єктно-реляційна СУБД дозволяє завантажувати код, призначений для обробки «нетипових» даних. Отже, база даних зберігає свою табличну структуру, але спосіб обробки деяких полів таблиць визначається ззовні, тобто програмістом.

Класифікація баз даних за способом доступу.

У **файл-серверних** СУБД файли даних розташовуються безпосередньо на сервері. Ядро СУБД розміщено на кожному клієнтському комп'ютері. Доступ до даних здійснюється через локальну мережу. Синхронізація читання і оновлення здійснюється за допомогою файлових блокувань. Перевагою цієї архітектури є низьке навантаження на центральний процесор сервера, а недоліком – високе навантаження локальної мережі.

Клієнт-серверні СУБД складаються із клієнтської частини (яка входить до складу прикладної програми) і сервера (див. Клієнт-сервер). Клієнт-серверні СУБД, на відміну від файл-серверних, забезпечують розмежування доступу між користувачами й мало завантажують мережу і клієнтські машини. Сервер є зовнішньою стосовно клієнта програмою, і за потребою його можна замінити іншим. Недоліком клієнт-серверних СУБД є сам факт існу-

вання сервера (що погано для локальних програм, у яких СУБД уже вбудовано) і більших обчислювальних ресурсів, споживаних сервером.

Централізована (вбудована) СУБД – бібліотека, яка дозволяє уніфіковано зберігати більші обсяги даних на локальному комп'ютері. Доступ до даних може відбуватися через SQL або через особливі функції СУБД. Вбудовані СУБД швидші від звичайних клієнт-серверних і не вимагають встановлення сервера, тому затребувані в локальному програмному забезпеченні, яке має справу з більшим обсягом даних (наприклад, геоінформаційні системи).

СУБД Access. Основні можливості та функції.

Microsoft Access – це система управління реляційними базами даних (СУБД), призначена для роботи на окремому ПК або в локальній обчислювальній мережі під керуванням Windows. Іншими словами, MS Access – це набір інструментальних засобів для створення й експлуатації інформаційних систем. Вона дозволяє вирішувати широке коло завдань користувачів без навиків програмування.

СУБД Access має значну кількість спеціальних програм, які отримали назву «майстри». Так, є майстер таблиць, кнопок, форм і т. д. Майстри здійснюють діалог із користувачем, у процесі якого визначаються дані, необхідні для розв'язування відповідного завдання. Access має розвинуту систему запитів, яка дозволяє отримувати на екрані різні відомості з таблиць. Користувач може задавати умови запиту, відповідно до яких з бази вибираються визначені дані чи формується нова таблиця. У СУБД Access реалізовані дві мови програмування: SQL та Visual Basic.

1. Проектування базових об'єктів інформаційних систем – двовимірних таблиць з різними типами даних.
2. Встановлення зв'язків між таблицями, з підтримкою цілісності даних, каскадного оновлення та знищення записів.
3. Уведення, зберігання, перегляд, сортування, модифікація і вибірка даних із таблиць з використанням різних засобів контролю інформації, індексування таблиць і апарату логічної алгебри (для фільтрації даних).

Історія створення бази даних Енциклопедії сучасної України (БД ЕСУ).

Тотальна комп'ютеризація суспільства зумовила необхідність у переході від паперового подання інформації до цифрового. Широке впровадження комп'ютерних технологій та Інтернету дали можливість створення електронних баз даних, які значно прискорили обробку та класифікацію інформації будь-якого типу.

Потреба швидкого перегляду та обробки великих масивів даних викликала необхідність певної структуризації даних за необхідними параметрами. Перехід від паперових носіїв до цифрових дає можливість економії людських ресурсів та часу. Але накопичення великої кількості файлів різних типів без певного упорядкування призводить до збільшення часу їх оброблення.

Тому в Інституті енциклопедичних досліджень НАН України створено БД ЕСУ, яка дозволила систематизувати усі дані та мати до них швидкий доступ. БД ЕСУ створено на основі СУБД Access, що є реляційною за своєю будовою та має файл-серверну архітектуру.

Основні елементи БД ЕСУ (опис інтерфейсу головної форми)

БД складається із 3 основних частин:

1. Робоча форма (Головна БД). Містить 24 поля.
2. Форма списків.
3. Панель управління БД. Складається з 11 спеціальних вкладок та 5 груп кнопок для керування БД.

Робоча форма відображає основну таблицю БД ЕСУ, яка містить такі поля:

З (закрито) – показує, чи готова стаття для верстки, якщо закрита, якщо ні – то потребує доопрацювання чи редагування;

Л (людина) – показує, що позиція відповідає персоналії;

Стаття – назва позиції в словнику Енциклопедії;

Галузь – тематичний галузевий розподіл, до якого віднесено ту чи іншу позицію;

Автор – автор статті;

Є – наявність статті (необов'язково в електронному вигляді);

Н (новий документ) – встановлюється автоматично для щойно створених документів, якщо їх назви починаються на поточні літери;

П (посилання) – відзначає наявність посилань на поточну статтю з інших статей;

Докум. – наявність статті в електронному вигляді (дозволяє відкрити відповідну позиції статтю);

Ф (фото) – показує, чи запланована в енциклопедії ілюстрація до цієї позиції;

Е (електронне фото) – відображає наявність чи відсутність електронної фотокартки до поточної позиції;

О (оригінал фотокартки) – відображає наявність чи відсутність оригіналу фотокартки (паперовий варіант);

К (кількість оригіналів) – кількість наявних оригіналів фотокарток;

М (мистецтво) – показує, чи наявні в Інституті фото, репродукції мистецьких робіт (лише для позицій з галузі «мистецтво»);

П (погані якості) – показує, що наявні в Інституті фото мають погану якість;

Т (текст) – фіксує, чи представлена позиція фотокарткою у тексті видання;

В (вклейка) – фіксує, чи представлена позиція фотокарткою у кольоровій вклейці видання;

Праців. – відображає прізвище редактора, що працює над статтею;

Г (готово) – наявність в Інституті повної інформації про позицію (встановлений прапорець Г показує, що статтю не потрібно надсилати на доопрацювання авторів);

Н (надіслано) – встановлений прапорець показує, що стаття була надіслана для підготовки чи коригування (автору, якщо встановлено прапорець А, установі, якщо було встановлено прапорець У, або в інше місце, якщо встановлено прапорець І), але відповідь досі не надійшла;

А (автору) – показує, що статтю необхідно надіслати автору (якщо встановлено прапорець Н, то статтю вже було надіслано);

У (установі) – показує, що статтю необхідно надіслати установі (якщо встановлено прапорець Н, то статтю вже було надіслано);

І (іншому) – показує, що статтю необхідно надіслати не автору або установі, а в якесь інше місце;

Надісл. – відображає, куди саме потрібно надіслати статтю.

Головна форма БД ЕСУ містить таблицю, де подано повністю весь словник статей ЕСУ (від А до Я). Ця таблиця має 24 поля. Розглянемо деякі з них, які безпосередньо допомагають формувати макет видання ЕСУ. Основне з них – поле «Стаття». У ньому подана назва статті, що в майбутньому увійде до складу відповідного тому ЕСУ. Кожна стаття має свою власну тематику, отже і відповідає конкретній галузі, наприклад: медицина, спорт, механіка і техніка, мистецтво та культура. За редактором, який працює в ЕСУ, закріплено одну або декілька галузей. У роботі над томом це допомагає редактору бачити увесь обсяг статей за галузями, які йому потрібно опрацювати, щоб підготувати певний том ЕСУ до друку (за це відповідає поле «Працівн.»). Кожна стаття має поле «Докум.», де знаходиться безпосередньо стаття у текстовому форматі. Редактор або коректор вносить необхідні зміни (правки) у кожному статтю, формуючи таким чином текст майбутнього видання.

За допомогою такого інтерфейсу користувач БД може отримати багато інформації про окремий запис (майбутню статтю) або певну вибірку статей за потрібним критерієм, що знаходяться у БД ЕСУ. Також БД дає повну інформацію щодо рівня опрацьованості конкретного тому ЕСУ, а також повністю усіх томів.

Форма списків.

Робоча форма. Перелік усіх статей БД ЕСУ, що відповідає словнику ЕСУ (головна форма).

Вхідні та вихідні листи. Таблиця вхідної та вихідної кореспонденції. Дає змогу переглядати листи, стан та статистику їх відправлення.

Списки. Тут знаходиться таблиця усіх геопунктів, установ, персоналій, яка дозволяє формувати наявний у БД ЕСУ матеріал за певними критеріями.

Адреси. Містить таблицю адрес авторів та інші дані для зв'язку (кореспонденції) із ними.

Повертання фотокарток.

Пріоритетні автори.

Авторські фото.

Також у всіх цих списках реалізований пошук за ключовими словами або за іншими необхідними параметрами та додавання, видалення, редагування записів.

Панель управління БД.

Тут знаходяться інструменти, за допомогою яких можна керувати записами бази. Наявні додаткові сервіси, які допомагають обробляти велику кількість записів БД. Також є можливість формувати звіти змін, які відбулися у БД ЕСУ, та виводити їх на друк або зберігати у файл.

Функціональне призначення БД ЕСУ.

1. Отримання інформації про стан та параметри конкретної позиції (готова до друку чи ще редагується; наявність фото та текстового матеріалу, що прикріплений до позиції; прізвище редактора та галузь; необхідність надсилання статті автору або установі та інші параметри).
2. Перегляд, редагування та створення позицій (майбутніх надрукованих у ЕСУ гасел).
3. Створення списків (фільтрів) позицій за певними необхідними умовами, а саме: автором, редактором, установою, галуззю і т. д.
4. Пошук необхідної позиції.
5. Можливість перегляду прикріплених до обраної позиції текстових та графічних файлів.
6. Експортування списків (вбірок) до текстових файлів та експортування текстових і графічних файлів із уже створених вибірок позицій.
7. Робота зі списками. Списки установ, адрес, кореспонденції (листів). Дозволяє відслідковувати листування, фільтрувати позиції за місцем праці персоналії. Надає додаткову інформацію щодо окремої позиції або їх списку.

БД ЕСУ дозволяє систематизувати накопичену текстову та графічну інформацію для створення друкованого видання ЕСУ та інших суміжних проектів, аналізувати і порівнювати інформацію за певними параметрами та критеріями.

ЛІТЕРАТУРА

1. Стефанюк В.Л. Інформаційні системи і їх застосування. – К., 1999.
2. Гурвиц Г. Microsoft Access 2010. Разработка приложений на реальном примере. – СПб., 2010.
3. Кузин А. В., Левонисова С. В. Базы данных. – Академия, 2012.
4. Вирджиния А. Базы данных Microsoft Access. Проблемы и решения. – Эком, 2001.
5. Железняк М. «Енциклопедія сучасної України»: історія, сучасний стан та перспективи // Енциклопедичний вісник України. – 2009. – Число 1.