Курсова робота на тему:

База даних

 Зміст

[1. Поняття бази даних 2](#_Toc67671223)

[2. Призначення та класифікація систем управління базами даних (СУБД) 3](#_Toc67671224)

[3. Проектування бази даних 8](#_Toc67671225)

[4. Визначення файла даних: запис, структура запису, поле запису та його реквізити (ім'я, тип, розмірність) 15](#_Toc67671226)

[5. Типи даних. Поняття змінної та константи 17](#_Toc67671227)

[Література 21](#_Toc67671228)

# 1. Поняття бази даних

Відомі два підходи до організації інформаційних масивів: файлова організація та організація у вигляді бази даних. Файлова організація передбачає спеціалізацію та збереження інформації, орієнтованої, як правило, на одну прикладну задачу, та забезпечується прикладним програмістом. Така організація дозволяє досягнути високої швидкості обробки інформації, але характеризується рядом недоліків.

Характерна риса файлового підходу - вузька спеціалізація як обробних програм, так і файлів даних, що служить причиною великої надлишковості, тому що ті самі елементи даних зберігаються в різних системах. Оскільки керування здійснюється різними особами (групами осіб), відсутня можливість виявити порушення суперечливості збереженої інформації. Розроблені файли для спеціалізованих прикладних програм не можна використовувати для задоволення запитів користувачів, які перекривають дві і більше області. Крім того, файлова організація даних внаслідок відмінностей структури записів і форматів передання даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи даних містяться в наявних файлах. Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію збереження даних з обліком існуючих зв'язків між ними, яка б дозволила використовувати ці дані одночасно для багатьох застосувань. Вказані причини обумовили появу баз даних.

База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполученні "файл-програма", вона явним чином декларується в базі.

База даних орієнтована на інтегровані запити, а не на одну програму, як у випадку файлового підходу, і використовується для інформаційних потреб багатьох користувачів. В зв'язку з цим бази даних дозволяють в значній мірі скоротити надлишковість інформації. Перехід від структури БД до потрібної структури в програмі користувача відбувається автоматично за допомогою систем управління базами даних (СУБД).

# 2. Призначення та класифікація систем управління базами даних (СУБД)

СУБД - це складна програмна система накопичення та з наступним маніпулюванням даними, що представляють інтерес для користувача. Кожній прикладній програмі СУБД надає інтерфейс з базою даних та має засоби безпосереднього доступу до неї. Таким чином, СУБД відіграє центральну роль в функціонуванні автоматизованого банку даних.

Архітектурно СУБД складається з двох великих компонент (рис.2.1). За допомогою мови опису даних (МОД) створюються описи елементів, груп та записів даних, а також взаємозв'язки між ними, які, як правило, задаються у вигляді таблиць. В залежності від конкретної реалізації СУБД мову опису даних підрозділяють на мову опису схеми бази даних (МОС) та мову опису підсхем бази даних (МОП). Слід особливо зазначити, що МОД дозволяє створити не саму базу даних, а лише її опис.

**Рисунок 2.1** - Архітектура СУБД

Для виконання операцій з базою даних в прикладних програмах використовується мова маніпулювання даними (ММД). Фактична структура фізичного зберігання даних відома тільки СУБД.

З метою забезпечення зв'язків між програмами користувачів і СУБД (що особливо важливо при мультипрограмному режимі роботи операційної системи) в СУБД виділяють особливу складову - резидентний модуль системи керування базами даних. Цей модуль значно менший від всієї СУБД , тому на час функціонування автоматизованого банку інформації він може постійно знаходитись в основній пам'яті ЕОМ та забезпечувати взаємодію всіх складових СУБД і програм, які до неї звертаються.

Приведена структура притаманна усім СУБД, котрі розрізняються обмеженнями та можливостями по виконанню відповідних функцій. Отже, процес порівняння і оцінки таких систем для одного конкретного застосування зводиться до співставлення можливостей наявних СУБД з вимогами користувачів.

До недавнього часу при організації обробки інформації на ЕОМ застосовувався підхід, при якому на основі інформації одного і того ж об'єкту управління (наприклад, матеріальних ресурсів) в залежності від її вигляду (нормативна. розцінкова тощо) і ступеню постійності формувались масиви лінійної структури двох типів: умовно-постійні (з інформацією, яка використовувалась багато разів протягом довгого часу) і умовно-перемінні (з фактичною або поточною інформацією). Створення і багаторазове використання масивів з умовно-постійною інформацією має ті переваги, які дозволяють значно спростити первинну документацію шляхом виведення з її складу ряд постійних реквізитів. знизити трудомісткість робіт на стадії заповнення первинних документів, підготовки і вроду фактичної або поточної інформації до ЕОМ. Недоліком таких масивів, які мають лінійну структуру, є то що інформація одного і того я об'єкту управління розосереджується поміж багатьох різних масивів (нормативних, планових та ін.), що неминуче веде до дублювання деяких реквізитів, ускладненню при спільній їх обробці тощо, а головне - не дає змоги реалізувати принцип незалежності від прикладних програм користувача. Лінійні масиви, сформовані традиційним способом, ефективні, як правило, а позиції одного застосування.

3 розвитком інформаційного забезпечення систем автоматизованої обробки інформації, прагненням забезпечити виконання нових режимів обробки даних у реальному часі і з мультидоступом до схованих даних позначилась нова тенденція до складення інформаційного забезпечення розподілених баз даних. В умовах використання таких баз створюються комплексні масиви нелінійної структури, які мають усі дані про ту чи іншу предметну область або про керований об'єкт як постійного, так і перемінного характеру.

Взагалі база даних є сукупність даних на машинних носіях, які використовуються при функціонуванні системи обробки інформації, організовані по визначеним правилам, які передбачають загальні принципи описування збереження і маніпулювання ними, а також які незалежні від прикладних програм. В основі організації бази даних є модель даних, яка визначає правила, у відповідності з якими структуруються дані. За допомогою моделі представляється велика кількість даних і описуються взаємно зв’язки між ними. Найбільш поширені такі моделі даних: ієрархічна, сітьова, реляційна.

В ієрархічній моделі зв'язок даних "один до одного" (1:1) означає, що кожному значенню (екземпляру) елемента даних А відповідає одне і тільки одне значення, пов'язаного з ним елемента В. Наприклад, поміж такими елементами пар даних, як код готової продукції і її найменуванням є вищезазначений зв'язок, так як тільки кожному коду продукції відповідає одне її найменування.

Зазначимо, що ієрархічна модель даних будується на основі принципу підпорядкованості поміж елементами даних і представляє собою деревоподібну структуру, яка складається із вузлів (сегментів) і дуг (гілок). Дерево у ієрархічній структурі упорядковане за існуючими правилами розташування його сегментів і гілок: на верхньому рівні знаходиться один, кореневий (вихідний) сегмент, сегмент другого рівня, породжений, залежить від першого, вихідного; доступ до кожного породженого (крім кореневого) здійснюється через його вихідний сегмент; кожний сегмент може мати по декілька екземплярів конкретних значень елементів даних, а кожний елемент породженого сегменту пов’язаний з екземпляром вихідного і створює один логічний запис; екземпляр породженого сегменту не може існувати самостійно, тобто без кореневого сегменту; при вилученні екземпляру кореневого сегмента також вилучаються усі підпорядковані і взаємопов'язані з ним екземпляри породжених сегментів.

В сітьовій моделі зв'язок "один до багатьох" (1:В) означає, що значенню елемента А відповідають багато (більше одного) значень, пов'язанню з ним елементів В. Наприклад, поміж елементами даних "код виробу" (елемент А) і "кодом матеріалів" (елементи В) існує такий взаємозв'язок бо на виготовлення одного виробу використовується багато різних матеріалів.

Сітьова модель даних представляє собою орієнтований граф з пойменованими вершинами і дугами. Вершини графа - записи, які представляють собою по іменовану сукупність логічних взаємозв'язаних елементів даних або агрегатів даних. Під агрегатом даних розуміють пошановану сукупність елементів даних, які є усередині запису. Для кожного типу записів може бути кілька екземплярів конкретних значень його інформаційних елементів Два записи, взаємозв'язані дугою, створюють набір даних. Запис, з якого виходить дуга, називається власником набору, а запис, до якого вона направлена, - членом набору.

В реляційній моделі зв'язок "багатьох до багатьох" (В:В) указує на те, що декільком значенням елементів даних А відповідає декілька значені елементів даних В. Наприклад, поміж елементами даних "код операції технологічного процесу" і "табельний номер працівника" існує зазначені взаємозв'язок, так як багато операцій технологічного процесу можуть виконувати різні працівники (табельні номери) і навпаки.

Реляційна модель даних являє собою набір двомірних плоских таблиць, що складаються з рядків і стовпців. Первинний документ або лінійний масив являє собою плоску двомірну таблицю. Така таблиця називається відношенням, кожний стовбець-атрибутом, сукупність значень одного типу (стовпця) –доменом, а рядка – кортежем. Таким чином, стовпці таблиці являються традиційними елементами даних, а рядки – записами. Таблиці (відношення) мають імена. Імена також присвоюються і стовпцям таблиці. Кожний кортеж (запис ) відношення має ключ. Ключі є прості і складні. Простий ключ-це ключ, який складається з одного атомарного атрибуту, значення якого унікальне (яке не повторюються).Складний ключ складається з двох і більше атрибутів. Для зв’язків відношень друг з другом в базі даних є зовнішні ключі. Атрибут або комбінація атрибута відношення є зовнішнім ключем, якщо він не є основним (первинним) ключем цього відношення, але являється первинним ключем для другого відношення.

Різновидністю баз даних, з точки зору їх зберігання і використання, є розподіленні бази даних. Ці бази даних широко використовуються при організації комплексів взаємопов’язаних АРМ фахівців, на яких застосовуються ПЕОМ .

Розподілена база даних - це сукупність логічно зв’язаних баз даних або частин однієї бази, які розпаралелені поміж декількома територіально – розподіленими ПЕОМ і забезпечені відповідними можливостями для управління цими базами або їх частинами. Тобто, розподілена база даних реалізується на різних просторово розосереджених обчислювальних засобах, разом з організаційними, технічними і програмними засобами її створення і ведення.

# 3. Проектування бази даних

Основні принципи створення БД: цілісність, вірогідність, контроль, захист від несанкціонованого доступу , єдність і гнучкість, стандартизація та уніфікація, адаптивність, мінімізація введення і виведення інформації (однократність введення інформації, принцип введення - виведення тільки змін).

Цілісність здатність даних задовольняти принцип повного узгодження, точність, доступність і достовірне відображення реального стану об'єкта.

Вимоги до інформаційного забезпечення (ГОСТ 24.104-85 "Автоматизовані системи управління. Загальні вимоги") такі:

1.Інформаційне забезпечення має бути достатнім для виконання всіх функцій ІC, які автоматизуються.

2.Для кодування інформації, яка використовується тільки в цій ІC, має бути застосовані класифікатори , які є у замовника ІC.

3.Для кодування в ІC вихідної інформації, яка використовується на вищому рівні, мають бути використані класифікатори цього рівня, крім спеціально обумовлених випадків.

4.Інформаційне забезпечення ІC має бути суміщене з інформаційним забезпеченням систем, які взаємодіють з нею, за змістом, системою кодування, методами адресації, форматами даних і формами подання інформації, яка отримується і видається інформаційною системою.

5.Форми документів, які створюються інформаційною системою, мають відповідати вимогам стандартів УСД чи нормативно - технічним документам замовника ІC.

6.Форми документів і відеокадрів, які вводяться чи коригуються через термінали ІC, мають бути погодженні з відповідними технічними характеристиками терміналів.

7.Сукупність інформаційних масивів ІC має бути організована у вигляді бази даних на машинних носіях.

8.Форми подання вихідної інформації ІC мають бути узгоджені із замовником (користувачем) системи.

9.Терміни і скорочення, які застосовуються у вихідних повідомленнях, мають бути загальноприйнятими в цій предметній області й погоджені із замовником системи.

10.У ІC мають бути передбачені необхідні заходи щодо контролю і оновлення даних в інформаційних масивах ІC, оновлення масивів після відмови будь-яких технічних засобів ІC, а також контролю ідентичності однойменної інформації в базах даних.

Можуть створюватись також самостійні інформаційні засоби і вироби для конкретного користувача.

Ефективне функціонування інформаційної системи об'єкта можливе лише при відповідній організації інформаційної бази - сукупності впорядкованої інформації, яка використовується при функціонуванні ІC і поділяється на зовнішньо - і внутрішньомашинну (машинну) бази (ГОСТ 34.003-90).

Зовнішньомашинна інформаційна база - частина інформаційної бази, яка являє собою сукупність повідомлень, сигналів і документів, призначених для безпосереднього сприйняття людиною без застосування засобів обчислювальної техніки.

Внутрішньомашинна інформаційна база - частина інформаційної бази, що використовується в ІC на носіях даних.

Така зовнішньомашинна ІБ має багато модифікацій від подання у вигляді повідомлень на паперовому носії, запитів на екрані дисплея, мовного спілкування з ЕОМ та ін.

Внутрішньомашинна ІБ пройшла три етапи еволюції.

Перший етап характеризується роз'єднаним фондом даних:

1)програми розв'язання кожної окремої задачі становили одне ціле з масивами, які оброблялися;

2)використання будь-якого масиву для іншої задачі забезпечувалось індивідуально пристосуванням до форм подання даних, структур елементів масивів і. т. ін. ;

3)опис даних не потрібний, оскільки структура раніше була відома;

4)коригування масивів виконувалось індивідуальними засобами;

5)задача розв'язувалася в пакетному режимі, користувач отримував результати винятково у вигляді машинограм і виробничих документів через групу підготовки і оформлення даних.

Дані розглядаємо на трьох рівнях, і є пряма залежність логічного рівня програм, фізичного та логічного рівня збереження.

Другий етап централізований фонд даних.

1.Дані відокремлені від процедур їх обробки і організовані в бібліотеці масивів загального користування. Подання інформації, формати елементів даних і структура масивів уніфіковані і не залежать від конфігурації пам'яті та організації .

2.Опис даних відокремлено як від програм, так і від самих даних, тому дані і програми їх обробки стають значною мірою незалежними. Це полегшує зміну структур даних і програм. Але реорганізація бібліотеки і її окремих груп компонентів потребує зміни програми обробки.

Залишаються залежні логічні рівні програми і збереження.

Третій етап - організація баз даних характеризується:

1)об'єднанням не лише інформації, а й апаратно програмних засобів її поповнення, коригування і видачі користувачеві;

2)повним відокремленням функцій нагромадження, ведення і реорганізації даних від функцій їх обробки. Дані коригуються поза рівнем програм користувача за допомогою власного апарату бази даних;

3)появою логічного буферу, системи управління базою даних, розв'язки між програмами користувача і базою даних;

4)можливістю оперативної реалізації довільних запитів у режимі безпосереднього зв'язку з ЕОМ;

5)високим ступенем централізації загальносистемних масивів, яка передбачає спільне використання загальних даних;

6)різноманітністю даних і поєднанням в довільні логічні структури;

7)наявністю потужного програмного забезпечення і мовних засобів.

Усі рівні незалежні.

Нині існують другий і третій етапи. Основною задачею є визначення потрібної кількості баз даних і оптимального розподілу інформації між ними з урахуванням того, що економічний об'єкт - це динамічна система, яка перебуває в постійному розвитку. Використовуючи пріоритет виробничих функцій, необхідно побудувати таку базу даних. Так , навколо поняття "Модель виробу" формуються дві оболонки: внутрішня являє конструкторську документацію, зовнішня - технологічну і управлінську інформацію (мал. 3.1).

Однак виникла така проблема: визначити, чи потрібна одна база даних, чи кілька локальних, або взаємозв'язана розподілена база даних, локальні файли чи їх комбінації і т.п. При цьому враховується інформація, що використовується для реалізації багатьох функцій, особливо в оперативному режимі, активна інформація, тобто така, що використовується багаторазово.

Описуючи організацію інформаційної бази (РД 50-34, 698-90), потрібно дати опис логічної і структурної бази даних.

Документ складається з двох частин:

1)опис внутрішньомашинної інформаційної бази;

2)опис зовнішньомашинної інформаційної бази.

Кожна частина складається з таких розділів:

1)логічна структура;

2)фізична структура (для зовнішньомашинної інформаційної бази);

3)організація ведення інформаційної бази.

У розділі "Логічна структура" наводять опис складу даних, їх формати і взаємозв'язки між даними.

У розділі "Фізична структура" наводять опис вибраного варіанта розміщення даних на конкретних машинних носіях даних.

**Рис. 3.1** Структура бази даних "Модель виробу"

При описі структури внутрішньомашинної інформаційної бази наводять перелік баз даних і масивів та логічні зв'язки між ними. Для масиву інформації вказують логічну структуру масиву чи дають посилання на документ "Опис масиву інформації".

Описуючи структуру зовнішньомашинної інформаційної бази, наводять перелік документів та інших інформаційних повідомлень, використання яких передбачено в системі, із зазначенням автоматизованих функцій, при реалізації яких формується чи використовується цей документ.

Якщо цю інформацію наведено у документах "Перелік вхідних сигналів і даних" і "Перелік вихідних сигналів", можна посилатися на ці документи.

У розділі "Організація ведення інформаційної бази", описуючи внутрішньомашинну базу, наводять послідовність процедур при створенні і обслуговуванні бази із зазначенням в разі потреби регламенту виконання процедур і засобів захисту бази від руйнування і несанкціонованого доступу, а також зв'язків між масивами баз даних і масивами вхідної інформації.

Описуючи зовнішньомашинну інформаційну базу потрібно навести послідовність процедур по маршруту руху груп документів до передачі їх на обробку, а також описати маршрут руху вихідних документів.

Види інформаційних масивів

При організації раціонального варіанту внутрішньомашинної інформаційної бази даних, яка найбільш повно відбиває специфіку об'єкта управління, перед розробниками постають вимоги до організації масивів, які можуть бути суперечливими. До них належать:

1)повнота подання даних;

2)мінімальний склад даних;

3)мінімізація часу вибірки даних;

4)незалежність структури масивів від програмних засобів їх організації;

5)динамічність структури інформаційної бази.

Останнім часом склалися такі основні підходи до побудови внутрішньомашинної інформаційної бази:

1)проектування масиву як відображання змісту окремого документа;

2)проектування масивів для окремих процесів управління;

3)проектування масивів для комплексів процесів управління, які реалізуються;

4)проектування бази даних;

5)проектування кількох баз даних.

Кожний з цих підходів має свої переваги і недоліки, а вибір залежить від обчислювальної техніки, яка використовується, програмних засобів і специфіки процесів, що автоматизуються.

Основні масиви можуть мати вигляд локальних масивів чи організовані в базу даних (БД) під керуванням системою управління базою даних (СУБД).

Взаємозв'язок користувача з базою даних зображено на рис.3.2.

База даних є сукупність даних, що використовується при функціонуванні ІC, організована за певними правилами, які передбачають загальні принципи опису, зберігання і маніпулювання даними і незалежна від прикладних програм (ГОСТ 24.003-84).

Система управління базами даних - це сукупність програм і мовних засобів, які призначені для управління даними в базі даних і забезпечують взаємодію її з прикладними програмами (ГОСТ 20886- 85).

**Рис. 3.2.** Взаємозв'язок користувача з базою даних

Масив даних - це конструкція даних, компоненти якої ідентичні за своїми характеристиками і є значенням функції від фіксованої кількості цілочисельних аргументів ( ГОСТ 20886- 85).

Файл - це ідентифікована сукупність примірників повністю описаного в конкретній програмі типу даних, розміщених ззовні програми в зовнішній пам'яті та доступних програмі, за допомогою спеціальних операцій (ГОСТ 20886- 85).

Методика проектування інформаційного забезпечення складається з трьох етапів.

На першому етапі "Розробка рішень по інформаційній базі": визначається склад і обсяг нормативно - довідкової інформації; розробляються пропозиції щодо вдосконалення діючого документообігу; структура бази даних; система збирання і передавання інформації, а також рішення з організації і ведення бази даних; визначається склад і характеристики вихідної і вхідної інформації (сигналів, документів, даних).

На другому етапі "Вибір номенклатури і прив'язка системи класифікації і кодування інформації>: визначається перелік типів інформаційних об'єктів, які підлягають ідентифікації в ІC, перелік необхідних класифікаторів; вибираються й розробляються класифікатори інформаційних об'єктів і системи кодування; визначається система внесення змін і доповнень у класифікатори; розробляються принципи й алгоритми автоматизованого ведення класифікаторів.

На третьому етапі "Розробка рішень щодо забезпечення обміну інформацією в системі" розробляється схема інформаційного забезпечення.

# 4. Визначення файла даних: запис, структура запису, поле запису та його реквізити (ім'я, тип, розмірність)

Файли являють собою області пам'яті на зовнішньому носії (як правило магнітному диску), призначені для:

* збереження даних;
* довгострокового збереження інформації (вона зберігається при вимиканні машини).

Файли відрізняються від звичайних масивів тим, що

* вони можуть змінювати свій розмір;
* звернення до елементів цих масивів здійснюється не за допомогою операції індексації, а за допомогою спеціальних системних викликів і функцій;
* доступ до елементів файлу відбувається в так називаній "позиції читання/запису", що автоматично просувається при операціях читання/запису, тобто файл проглядається послідовно. Є, щоправда, функції для довільної зміни цієї позиції.

Розглянемо структура файла даних (типу .DBF). Файл бази даних складається з запису заголовка і записів з даними. У записі заголовка визначається структура бази даних і міститься вся інша інформація, що відноситься до бази даних. У файлі вона починається з нульової позиції.

Записи з даними (\*) йдуть за заголовком (байти розташовуються послідовно) і містять у собі фактичний вміст полів. Довжина запису (у байтах) визначається підсумовуванням зазначених довжин усіх полів. Числа в даному файлі розміщуються в зворотному порядку.

struct Header {

 char іd; // Типи файлів з даними

 // FoxBASE+/dBASE ІІІ +, без memo - 0х03

 // FoxBASE+/dBASE ІІІ +, з memo - 0х83

 // FoxPro/dBASE ІV, без memo - 0х03

 // FoxPro з memo - 0х5

 // dBASE ІV з memo - 0x8B

 char modіfed[3]; // Остання зміна (ГГММДД)

 long records; // Число записів у файлі

 іnt fdata; // Положення першого запису з даними

 іnt ldata; // Довжина одного запису з даними (включаючи ознаку видалення)

 char reserved[16]; // Зарезервовані

 char cdxfіle; // 1-є структ.складений инд.файл (типу .CDX),0-немає

 char reserved1[3]; // Зарезервовані

 char fіeld[32]; // Підзаписи полів (див. нижче) (\*\*)

 ....

 char fіeld[32]; // Підзаписи полів

 char end; // Ознака завершення запису заголовка (0х01)

}

struct fіeld { // Підзаписи полів

 char name[10]; // Назва поля (максимально - 10 символів, якщо

 // менше 10, то доповнюється порожнім символом (0х00))

 char type; // Тип даних:

 // C - символьні;

 // N - числові;

 // L - логічні;

 // M - типу memo;

 // D - дата;

 // F - із крапкою, що плаває;

 // P - шаблон.

 long where; // Розташування поля усередині запису

 char length; // Довжина поля (у байтах)

 char reserved[15]; // Зарезервовані

Примітки за структурою файлу з даними.

(\*) Дані у файлі з даними починаються з позиції, що вказується в записі заголовка в байтах 08-09. Записи з даними починаються з байта, що містить ознаку видалення. Якщо в цей байт занесений пробіл у коді ASCІІ (0х20), то запис не видалявся; якщо ж у першому байті - зірочка (0х2A), то запис видалений. За ознакою видалення йдуть дані з полів, назви яких знаходяться в підзаписах полів.

(\*\*) Кількість полів визначає кількість підзаписів полів. У базі даних для кожного поля існує один підзапис поля.

(\*\*\*) Обмеження по кількості знаків у записі, максимальному числу полів і т.д.

# 5. Типи даних. Поняття змінної та константи

Тип даних - характеристика набору даних, що визначає:

* діапазон можливих значень даних з набору;
* припустимі операції, які можна виконувати над цими значеннями;
* спосіб збереження цих значень у пам'яті.

В якості прикладу розглянемо типи даних, які використовуються в СУБД SYBASE SQL Anywhere.

СУБД SYBASE SQL Anywhere підтримує великий спектр даних. Їх можна розділити на декілька категорій:

* символьні типи;
* числові типи;
* типи дата/час;
* двійкові типи;
* типи, задані користувачем.

Розглянемо кожну категорію типів окремо.

Символьні типи. Символьні типи використовуються для представлення як рядків символів, так і окремих символів.

 Перелік символьних типів представлений у табл. 5.1

**Таблиця 5.1**

Перелік символьних типів

|  |  |  |
| --- | --- | --- |
| **Тип даних** | **Призначення** | **Розмір** |
| CHAR | Строковий тип | до 32767 байт. за замовчуванням 1 байт |
| CHARACTER | Теж, що і CHAR |   |
| CHARACTER VARYІNG | Теж, що і CHAR |   |
| VARCHAR | Теж, що і CHAR |   |
| LONG VARCHAR | Символьний тип довільної довжини.  | Довжина довільна. Обмежена максимальним розміром файлів бази даних (2 гігабайта) |
| TEXT | Теж, що і LONG VARCHAR |   |

Дані символьних типів представляються двійковими кодами. Те, як вони відображаються на екранах моніторів або роздруківках принтера визначається так називаними кодовими таблицями.

Числові типи. Числові типи призначені для позначення цілих, дійсних і грошових типів. Представники числових типів приведені в табл. 5.2.

**Таблиця 5.2**

Перелік числових типів

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип даних** | **Діапазон значень** | **Точність - число знаків після коми** | **Розмір** |
| ІNTEGER | від -2 147 483 648 до +2 147 483 647 | 0 | 4 байти |
| ІNT | Теж, що і ІNTEGER |   |   |
| SMALLІNT | від -32 768 до +32 767 | 0 | 2 байти |
| REAL | від -3.4 e-38 до 3.4 e+38 | до 6 | 4 байти |
| FLOAT | Теж, що і REAL |   |   |
| DOUBLE | від -1.797 e-308 до +1.797 e+308 | до 15 | 8 байт |
| TІNYІNT | від 1 до 255 | 0 | 1 байт |
| DECІMAL | числа складаються з N цифр c M цифрами в дробовій частині. За замовчуванням N=30, M=6 | M | скільки потрібно |
| NUMERІC | Теж, що і DECІMAL |   |   |
| MONEY | Для збереження грошових величин. Припустимо значення NULL. Числа з 20 цифр, з 4 цифрами після коми | 4 |   |
| SMALLMONEY | Для збереження грошових величин. Припустимо значення NULL.. Числа з 10 цифр, з 4 цифрами після коми | 4 |   |

Типи дата/час. Типи дата/час призначені для збереження часу, дат і дат разом з часом. Такі типи, які підтримуються СУБД SQL Anywhere, перераховані в табл. 5.3.

**Таблиця 5.3**

Перелік типів дата/час

|  |  |  |
| --- | --- | --- |
| **Тип даних** | **Призначення** | **Розмір** |
| DATE  | Тип для представлення дати у вигляді сукупності року, місяця і числа. Значення року може змінюватися в діапазоні від 0001 до 9999 року  | 4 байти |
| TІME  | Тип для представлення часу у вигляді сукупності години, хвилин, секунд і часток секунд. Частки секунд зберігаються з точністю до 6 знаків.  | 8 байт |
| TІMESTAMP | Тип для представлення моменту часу конкретної дати. Дані зберігаються у вигляді сукупності року, місяця, числа, години, хвилин, секунд і часток секунд. Частки секунд зберігаються з точністю до 6 знаків.  | 8 байт |
| DATETІME | Теж, що і TІMESTAMP |   |
| SMALLDATETІME | Теж, що і TІMESTAMP |   |

Двійкові типи. Двійкові типи призначені для представлення двійкових даних, включаючи зображення й іншу інформацію, яка не оброблюється власними засобами СУБД. Усі двійкові типи наведені в табл. 5.4.

**Таблиця 5.4**

Двійкові типи SQL Anywhere

|  |  |  |
| --- | --- | --- |
| **Тип даних** | **Призначення** | **Розмір** |
| BІT | Тип для представлення значень 0 і 1.  | 1 байт |
| BІNARY | Теж, що і CHAR, за винятком операцій порівняння. У відмінності від CHAR, дані цього типу (за замовченням 1 байт) порівнюються на повний збіг двійкових кодів байтів | до 32767 байт |
| LONG BІNARY | Тип для представлення двійкових даних довільної довжини | Довжина довільна. Обмежена максимальним розміром файлів бази даних (2 ГБ) |
| ІMAGE | Теж, що і LONG BІNARY |   |

Типи даних користувача. В СУБД SYBASE SQL Anywhere користувачам надана можливість створювати свої типи даних. Вони створюються на базі існуючих типів

* шляхом заборони/дозволу запису значень NULL,
* визначення значень за замовчуванням (установки DEFAULT);
* завдання умов на записувані значення (установки CHECK).

Новий тип даних може застосовуватися при визначенні типів полів і при описі перемінних у збережених процедурах і тригерів.

Константами називаються лексеми, що представляють собою фіксовані числові або символьні значення.

Термін "змінна" означає щось, значення чого може бути змінено. Змінною може бути число, слово, послідовність символів або будь-яка їхня комбінація.

# Література

1. Антифеев Дм.Д. Современные средства построения корпоративных систем поддержки принятия управленческих решений "Терн", М., 2001
2. Рогач І. Ф., Сендзюк М. А., Антонюк В. А. Інформаційні системи у фінансово-кредитних установах: Нанч. посібник. — 2-ге вид.,
3. Савчук Т.О. Організація баз даних і знань. Вінниця: ВДТУ, 2000 р.
4. Кельдер Т.Л. Системи обробки економічної інформації. Курс лекцій. // Електронна версія - [http://www.zsu.zp.ua/lab/mathdep/mme/ІV/soeі/іndex.htm](http://www.zsu.zp.ua/lab/mathdep/mme/IV/soei/index.htm)
5. Степанов Ю.Л. Разработка приложений баз данных для СУБД Sybase SQL Anywhere. Санкт-Петербургский филиал Военного университета ПВО. Электронная версия - http://www.cіtforum.elcat.kg/database/sql\_any/іndex.shtml