

# 1. Архитектура на система за управление на бази данни

## 1.1. Понятие за база данни

База от данни (БД) е интегрирана съвкупност от взаимосвързани данни, съхранявани на електронен носител, които представят всички факти за дадена предметна област, представляващи интерес и използвани от много потребители посредством приложни програми. Предметна област (ПО) е тази част от реалния свят, за която се съхранява информация в БД. Това може да е промишлено предприятие, държавно ведомство, университет, болница, малка фирма или голяма корпорация. Основните характеристики на данните в БД, които следват от това определение са:

- Интегрирани – В БД са събрани данни за ПО, обединени от различни източници, например различни подразделения или дейности на ПО.
- Общи – Едни данни, в едно и също време са достъпни за много потребители за различни цели.

Какви преимущества дава използването на БД? Ще разгледаме други важни характеристики на БД, които ни позволяват да разглеждаме БД като нещо повече от група файлове на диска.

- **Централизирано управление на данните за ПО**

Преди появата на БД, за различните нужди на ПО (приложни програми или системи) се създават и поддържат собствени файлове и никой не се грижи за синхронизиране на данните, използвани в различните приложения. В системите за управление на БД има лице/група хора, наречено администратор на БД (АБД, DBA), който отговаря за цялостното управление на данните. В групата на АБД влизат технически специалисти в областта на информационните технологии, които определят съдържанието на БД така, че тя най-точно да представя ПО и поддържат БД в коректно състояние.

- **Намалено и контролирано излишество на данните**

Преди появата на БД е характерно съществуването на неконтролирано излишество на данните в ПО, една и съща информация за ПО се съхранява за различни нужди и цели в различни файлове от различни подразделения на предприятието. Например персонални данни за служителите в предприятието, като имена, адрес, дата на раждане, и др. са необходими както в отдел кадри така и във финансово-счетоводен отдел (ФСО). Всеки от тези отдели има собствена информационна система, която използва собствени файлове, съдържащи необходимата ѝ информация. Недостатъците от такава повторемост са много: разход на външна памет; многократно и скъпо обновяване на данните; възможна противоречивост на информацията, получавана от различните приложения. В БД с интегриране на данните се намалява повторението, но все още е възможно да съществува минимално излишество с цел повишаване на ефективността или надеждността на системата. Но дори и да съществува, това е управлявано излишество, т.е. АБД знае за него и системата за управление на базата данни (СУБД) го поддържа.

- **Осигуряване на сигурност на данните**

Сигурност на данните (Data Security) е понятие, свързано със защитата на данните в БД. Под сигурност се разбира защита на данните от неправомерен достъп, изменение или унищожаване. АБД определя правата на потребителите за достъп до БД, като описва за всеки потребител към кои данни какъв достъп има право. Това са така наречените ограничения за сигурност. СУБД следи за спазването на тези ограничения, като извършва проверка при всеки опит за достъп до БД.

- **Осигуряване на цялостност на данните**

И цялостност на данните (Data Integrity) е понятие, свързано със защитата на данните в БД. Под цялостност се разбира защита на данните от неправилни изменения, т.е. целта е в БД да се съхраняват само правилни (правдоподобни) и непротиворечиви данни. С отстраняване на неуправляемото излишество на данни, се намалява и опасността за противоречивост на данните, но въпреки това данните в БД може да не са верни. АБД определя ограничения за цялостност – условия, на които трябва да отговарят данните, за да се счита че са правдоподобни. Например, сумата в банкова сметка да е положително число или годината на раждане на служител да е преди текущата година. Тези ограничения се проверяват при всеки опит за обновяване на БД – добавяне, изтриване или изменение на данни, и съответната операция се изпълнява само ако не е нарушено нито едно ограничение за цялостност. Но както се досещате това не винаги гарантира правилни данни, затова използваме думата правдоподобни.

- **Независимост на данните**

Под независимост на данните (Data Independence) се разбира изолиране на програмите от изменения в структурата на данните, които обработват. За да обясним какво е независимост ще разгледаме противоположен пример. Нека имаме програма, написана на някакъв език за програмиране, без използване на база данни, която обработва файл съдържащ записи за служители. Записите във файла са в хронологичен ред (в реда на добавяне). След време се оказва, че програмата работи бавно и се налага да се промени организацията на файла, напр. записите да са сортирани по ЕГН на служителя. Или се появява ново приложение, за което е необходимо да се добави ново поле в записа на служителя, напр. email адрес. Такива промени в начина на съхраняване или структурата на данните най-вероятно ще наложат преработка на програмата, т.е. програмата зависи от данните. Защо е желателно да има независимост, особено при използване на БД?

- БД се развива

При създаване на БД е трудно да се предвидят всички възможни приложения. Освен това самата ПО се развива, появяват се нови приложения, изменя се приоритета на съществуващите, променят се нормативни актове и закони, което се отразява на дейността в ПО и естествено на БД. Всичко това налага изменения във физическата организация на данните или добавяне на нови типове данни.

- Данните в БД се използват от много потребители

Едни и същи данни се използват и е възможно да се представят различно от различните приложни програми. Например, отдел кадри иска да вижда датата на раждане на служителя, а ФСО вижда същите данни като възраст на служителя.

Има две нива на независимост:

- Физическа независимост – Осигурява изолиране на програмите от изменения във физическото представяне на данните.
- Логическа независимост – Означава измененията в логическата структура на данните, в резултат от развитието на БД, да не оказват влияние на разработените приложни програми.

Независимостта на данните е относително понятие, различните СУБД осигуряват независимост в различна степен или не я осигуряват въобще. Съвременните системи са по-развити в това отношение. В следващия раздел ще разгледаме архитектурата на ANSI/SPARC, която е основа за постигане на по-висока степен на независимост.

## 1.2. Компоненти в система за управление на БД и архитектура на ANSI/SPARC

Една система за управление на БД (СУБД) включва различни видове компоненти:

### • Програми

1. Ядро на СУБД (database server, database engine, сървър) – сложна програмна система, осигуряваща съхранение и достъп до данните в БД, сигурност и цялостност на данните в условията на многопотребителски, конкурентен достъп.
2. Сервизни програми (утилити) – средства за АБД, с които той изпълнява дейности по поддържане на БД, като:
  - настройка и тестване на системата;
  - създаване на резервни копия на БД;
  - възстановяване след аварийни ситуации и повреда;
  - реорганизация на БД;
  - събиране на статистики за функционирането и производителността.
3. Инструментални средства - средства за разработка на приложения, като:
  - процесори на езици за програмиране;
  - процесори на езици за БД;
  - генератори на отчети и др.
4. Приложни програми – програми, създадени чрез горните средства, които реализират конкретните обработки на данните в БД, осигуряващи удобен потребителски интерфейс за работа с БД, създаването на отчети и др.

### • Потребители

1. АБД – Това е групата отговорна за управлението на БД на техническо ниво, т.е. осигурява създаването, функционирането и развитието на БД.
2. Крайни потребители (end users) – Това са потребителите, за които се създава БД.
3. Приложни програмисти (DB application programmers) – Това са програмистите, които разработват приложните програми.

### • Модели на данните

Модел на данните (Data model) е съвкупност от абстрактни обекти данни, които представят обектите, свойствата и връзките в определена ПО. Един от ключовите въпроси в областта на БД е: Какви да са тези абстрактни обекти данни? Отговорът е важен, тъй като той определя понятията и начина, по който потребителят си представя данните в БД. В една система съществуват различни нива на абстрактно представяне на данните между компютъра и човека. През 1975г. ANSI/SPARC публикува документ, в който въведе три нива на представяне на БД и съответно три вида модели на данните:

1. Вътрешен модел на данните (физически). На това ниво БД се представя чрез понятия близки до физическите, напр. файлове на операционната система с определена организация, индекси и др. И вътрешният модел е абстрактен, защото не се използват машинните понятия, като бит, байт, сектор. Вътрешният модел на БД е един.

2. Концептуален модел на данните (логически). На това ниво БД се представя чрез понятия на по-високо ниво на абстракция, без да се конкретизира физическото представяне на данните. Концептуалният модел на БД е един.

3. Външни модели на данните. Това са моделите на потребителите. Всеки потребител или група потребители със сходни интереси, има свой външен модел, който определя начина, по който той вижда част от БД. Понятията на външния модел са на същото ниво на абстракция, както и в концептуалния. Разликата е, че

концептуалният модел е глобален, а външните са локални в смисъл, че съответстват на индивидуалните представи на потребителите за БД.

- **Езици за БД**

Езикът е интерфейс на потребителите към БД. Някои автори използват термина подезик на данните, с което показват, че това не е пълноценен език за програмиране, а само осигурява инструмент за работа с данните. По-често се използва наименованието Език на заявките или Query Language като общо наименование на всички езикови средства за БД (макар, че това наименование не е съвсем коректно). Езиковите средства обикновено се разделят в две основни групи:

1. Език за описание на данните (Data Definition Language)

Това са езиковите средства, с които се описват моделите данни. Формалното и пълно описание на модел на данните се нарича схема (schema). Следователно според архитектурата на ANSI/SPARC всяка БД се описва с една концептуална схема, една вътрешна схема и няколко външни схеми (използва се и термина подсхема). Освен тези три вида схеми описанието включва и два вида изображения:

- Изображението концептуален-вътрешен определя по какъв начин обектите в концептуалния модел се материализират от тези във вътрешния модел.
- Изображението външен-концептуален аналогично определя начина, по който външните обекти се конструират от концептуалните.

Наличието на междинния концептуален модел в архитектурата е основа за постигане на по-висока степен на независимост на данните. Изменение във физическото представяне на данните ще доведе до промяна във вътрешната схема и евентуално изображението концептуален-вътрешен, което означава, че концептуалната схема остава непроменена и следователно това няма да се отрази на съществуващите приложни програми. Аналогично добавянето на нови обекти в концептуалния модел ще се реализира чрез разширение на концептуалната схема и изменения в изображенията външен-концептуален, но външните модели ще останат непроменени, което означава логическа независимост.

Обикновено езикът за описание на данните включва и средства за описание на ограниченията за цялостност и ограниченията за сигурност.

2. Език за манипулиране на данните (Data Manipulation Language)

Това са езиковите средства за обработка на данните в термините на външния модел, а именно за търсене и извличане, добавяне, изтриване и изменение на данни. Езиковите средства се реализират в СУБД по два начина, а именно като:

- Независим език. Самостоятелни програми на езика на заявките се изпълняват от специални процесори на езика, които са част от инструменталните средства доставяни със СУБД.
- Вграден (Embedded Query Language) или програмен език на заявките. Програмата на езика на заявките се съдържа в друга програма, написана на традиционен език за програмиране (най-често C, C++, Java).

Да се върнем към ролята на АБД. Неговите отговорности по управление на данните включват следните основни задачи:

- Проектира БД – проектира концептуалния модел, вътрешния модел и описва концептуалната, вътрешната схема и изображението концептуален-вътрешен.
- Взаимодейства с потребителите, за да определи информационните им потребности и да опише външните схеми и изображенията външен-концептуален.
- Определя ограниченията за сигурност.
- Определя и изпълнява процедурите за резервно копиране и възстановяване на БД след повреда.

- Отговаря за настройка на системата при изменения на изискванията към нея, т.е. на етапа на експлоатация отговаря за ефективното функциониране на системата.
- Рис.1 представя основните компоненти на една система за БД и ролята на АБД.

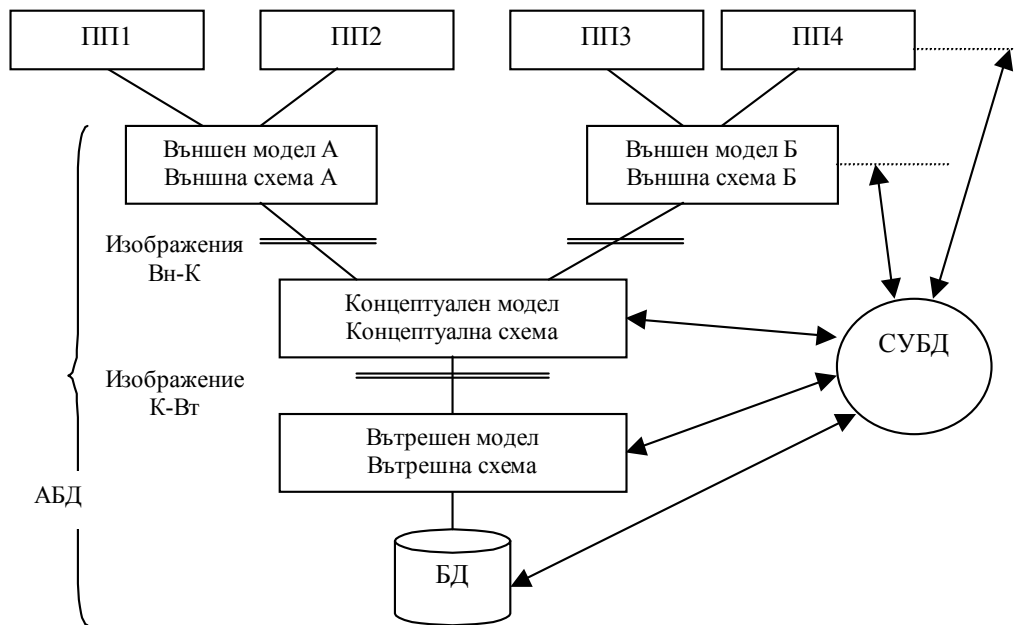


Рис. 1. Архитектура на Система за БД

### 1.3. Архитектура клиент/сървер

Ако разглеждаме архитектурата на система за БД на най-високо ниво, то тя е представена на Рис.2 и е система, включваща два типа елементи:

- Сървър – ядро на СУБД
- Клиенти – приложни програми, инструментални средства, утилити

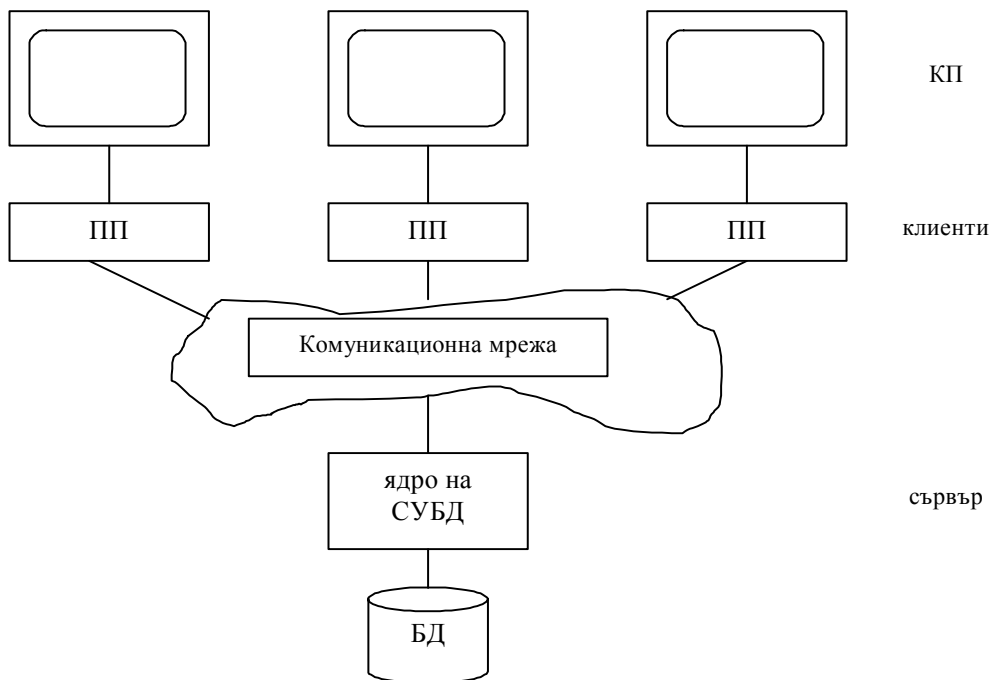


Рис.2. Архитектура клиент/сървър на Система за БД