

Глава 1

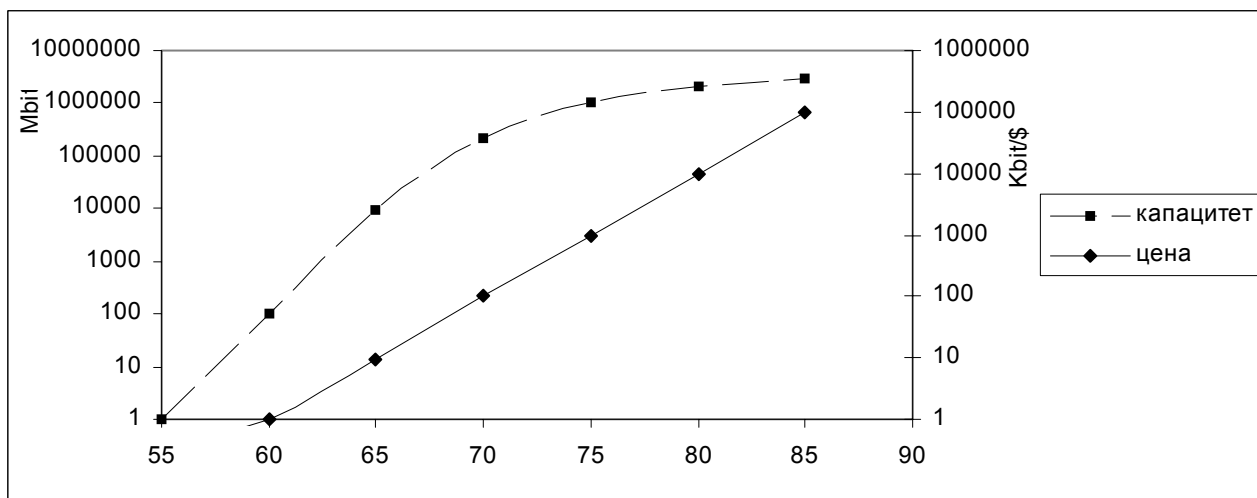
Бази от данни (БД) и системи за управление на бази от данни (СУБД)

Въпросът за архивиране на външни устройства и извличане на данните от там е основен още от зората на информатиката. В началото, когато магнитните ленти са били единствените външни носители, данните са били групирани във физически записи, представляващи еднотипни обекти, и образуващи структура наречена **файл**. Единственият възможен достъп до тези обекти е бил *последователния*, но по-късно с развитието на външните (периферни) запомнящи устройства стават възможни и други методи на достъп като *директния(прекия)* и *индексирания*. Тези методи се използват и до днес.

Но макар че в повечето случаи тези файлове притежават общи елементи, това не може да бъде използвано в общия случай поради отделното и независимо проектиране, и различните и независими една от друга процедури за достъп.

Но макар че в повечето случаи тези файлове притежават общи елементи, това не може да бъде използвано в общия случай поради отделното и независимо проектиране, и различните и независими една от друга процедури за достъп.

В основата на развитието на базите от данни, освен горепосоченият факт за затруднено използване на отделни файлове, лежи от една страна развитието на технологията страна, позволяващо запомнянето на все по-голям обем от данни за все по-ниски цени, и от друга страна нуждите на предприятията и организациите от все по-детайлно познание на тяхната дейност.



Фигура 1.1. Максимален капацитет и цени на външна памет

Дефиниции на БД

- БД е структурирано множество от данни, записано на носител, който е достъпен за четене и запис чрез компютър с цел да обслужва едновременно много потребители поотделно и за разумно време.
- Множество от логически свързани и съхранявани заедно данни без излишък, които могат да бъдат използвани от различни приложения; начинът на съхранение е независим от приложенията и се използват общи процедури за промяна, добавяне и за търсене.
- Множество от данни структурирани според една схема от данни.
- Данни, които директно и едновременно са достъпни от различни, взаимно конкуриращи се потребители.

Информатика II – 1. Бази от данни и СУБД

БД се създава за да регистрира факти и събития настъпили в живота на дадена организация и да ги възстанови по желание или да извлече заключения, като съпоставя няколко елементарни факта. По-рано, всяко ново приложение изискваше специфични файлове и програми. БД използва противоположен подход, който позволява централизацията, интеграцията и разпространението на събраната информация.

Недостатъците на стария подход (приложения със специфични за тях файлове) са:

- Голям излишък на информация;
- Аномалии при обновяването, добавянето и изтриването на информация – данните губят своята кохерентност;
- Трудности при добавяне на обработки не планирани в самото начало – липса на гъвкавост.

Предимствата на подхода БД са:

- намален излишък на информация;
- запазва се кохерентността, тъй като поддръжката се прави за всички;
- администрирането е централизирано, което позволява да се съблюдают определени стандарти даващи възможност за обмен на данни с други системи;
- управление на правата на достъп, което дава по-голяма сигурност;
- съществува глобална нормализация на системите.

Система за управление на база от данни (СУБД)

СУБД позволява на потребителя да взаимодейства с БД. Тя управлява съхранението на данните във външната памет и предоставя процедурите за търсене и избор на същите данни.

Потребителят описва абстрактно това, което иска да върши, като оставя на системата задачите за представяне и организация на данните върху физическите носители. Другите функции и характеристики на СУБД са:

1. Описание на данните

СУБД трябва да предоставя на потребителя средство за описание на множеството на данните. Има няколко нива за описание: концептуално, логическо, физическо.

2. Използване

Тази функция предлага на потребителя взаимодействие с БД. Съществуват два основни типа потребители:

- Специалисти, които използват и създават алгоритмични процедури
- Неспециалисти, които съблюдают предварително създадени правила

3. Интегритет (логическа непротиворечивост)

Колкото повече данни има в една БД, толкова е по-голям рискът тези данни да съдържат грешки. За да се намали този риск СУБД трябва да предлага на потребителите възможност да дефинират ограничения за интегритет (правила, позволяващи да се поддържа интегритета). Тези ограничения дефинират контрола, който СУБД може да упражнява върху данните. Освен това СУБД трябва да поддържа управляем излишък на данни.

4. Секретност

Когато една БД има много потребители, известни подмножества от данните трябва да са достъпни само за упълномощени лица. СУБД трябва да предлага средства за проверка на правата на достъп на потребителите.

5. Конкурентен (едновременен) достъп

Когато потребителите ползват едновременно една и съща информация от БД, СУБД трябва да разрешава конфликтите при достъпа и да ги обработва правилно.

6. Сигурност на работа

При произшествие с хардуера или софтуера, БД не е в работещо състояние. С цел да може да се пусне наново след като причината е отстранена, СУБД трябва позволява записа на контролни точки, за да бъде приведена БД в задоволително състояние.

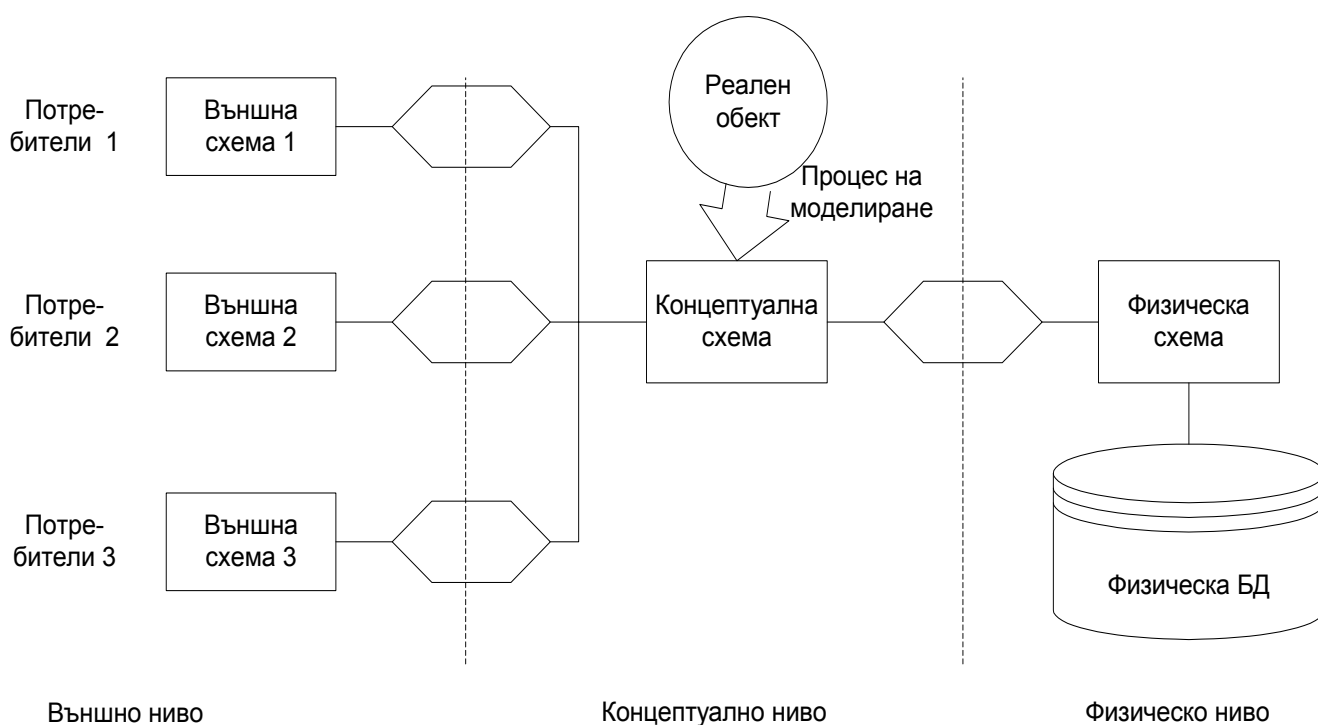
Нива на представяне на БД

Вътрешно (физическо) ниво

Физическата схема има за цел да покаже как данните се съхраняват на физическите носители

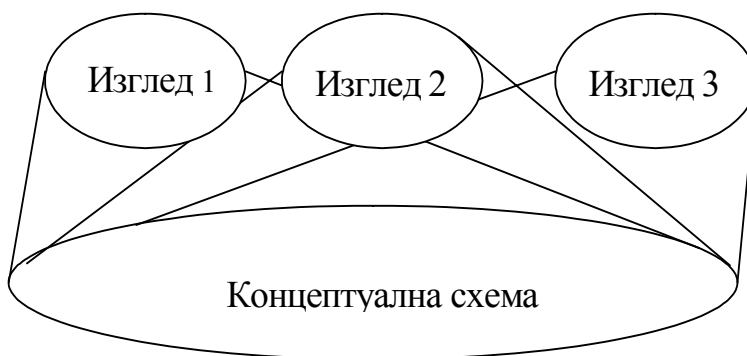
Концептуално ниво

Това е основната част на архитектурата на една СЪБД. Има за цел опише в абстрактни но релевантни термини известна част от една организация или предприятие и управляващите процеси, които изискват разработката на БД. Това преминаване от от реалния свят в концептуална схема отговаря на процес на моделиране, където обектите от реалния свят са класифицирани и наименовани. СУБД предоставя език за дефиниция на данните, който позволява да се опише концептуалната схема. Съществуват три големи групи модели, които се различават по същността на отношенията, които позволяват да бъдат моделирани: йерархични, мрежови и релационни.



Фигура 1.2. Нива на представяне

Фигура 1.3. Подсхеми

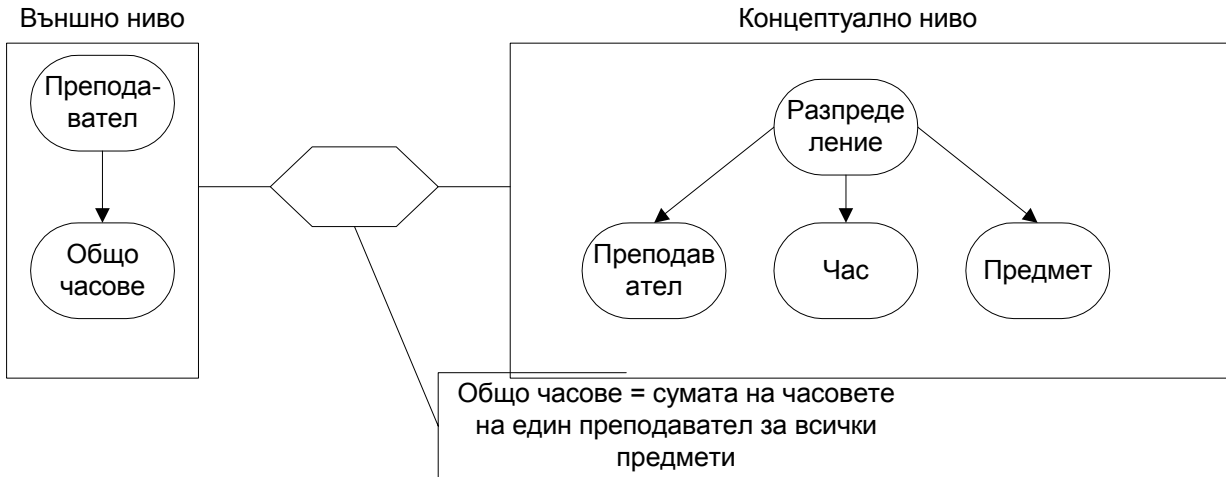


Информатика II – 1. Бази от данни и СУБД

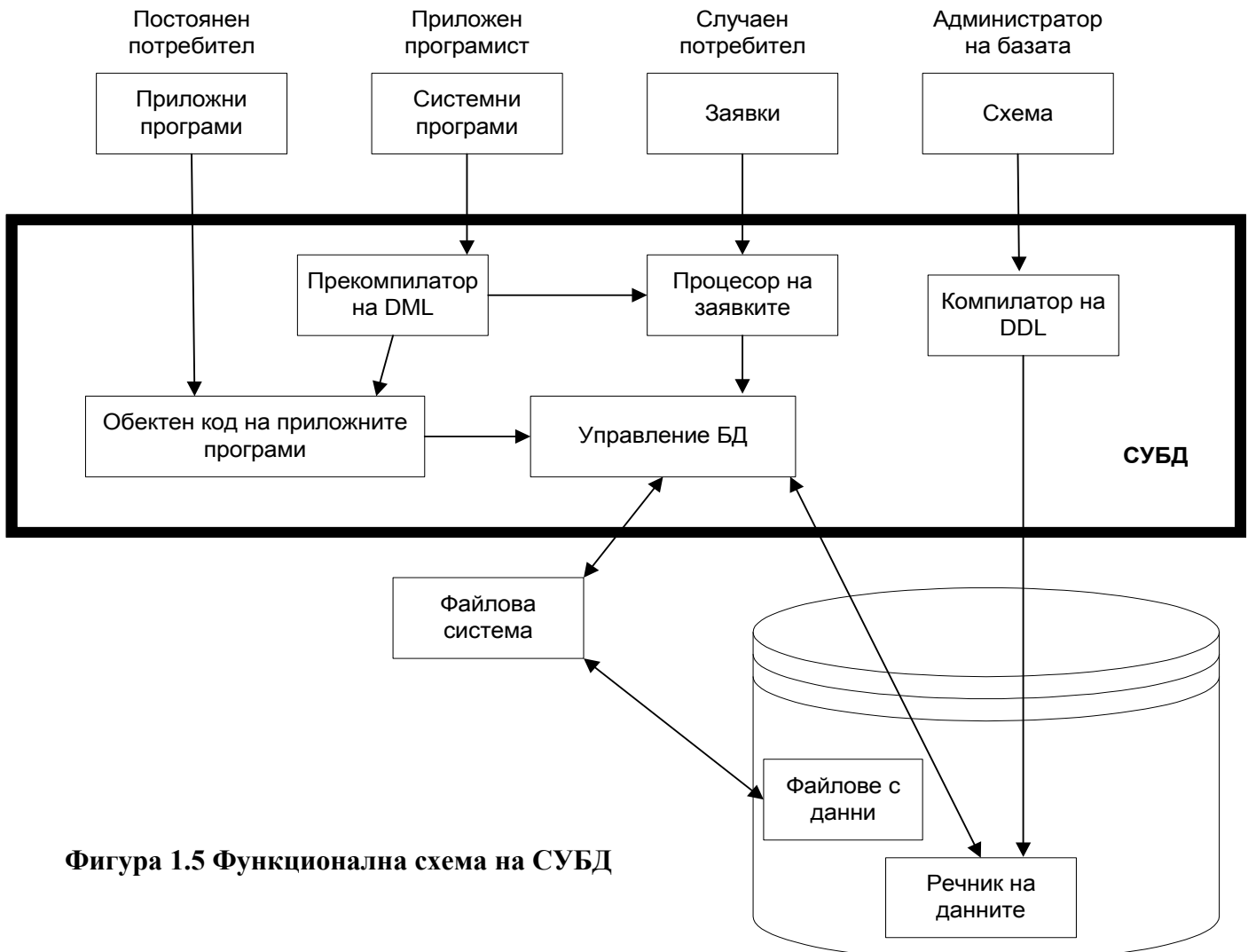
Външно ниво

Това ниво кореспондира на виждането на концептуалната схема (цялата или част от нея) от страна на група потребители, използващи едно приложение. Външната схема или изглед може да бъде разглеждан като подсхема на концептуалната схема (фиг. 1.3).

Пример: Може да се види, че докато на външното ниво информацията отнасяща се за един преподавател се състои от името му и от общия брой на часовете му, докато във физическата БД и в концептуалната схема тази информация я няма, но може да бъде изчислена. (фиг. 1.4)



Фигура 1.4. Външна схема



Фигура 1.5 Функционална схема на СУБД

Създаване на една БД

Език за дефиниция на данните

Проектирането на БД изисква усилията на многобройни специалисти. Първата задача да се създаде концептуалната схема с помощта на есика за дефиниция на данните (DDL).

Администраторът на БД (или аналита) играе централна роля и при проектирането и при експлоатацията. Той отговаря за:

- Изработване на концептуалната схема;
- Изработване на външните схеми;
- Дефиниране на правата за достъп на всеки потребител;
- Задаването на физическата организация на данните, както и на методите за достъп, които ще се използват
- Създаване на процедурите, осигуряващи задоволително ниво на сигурност

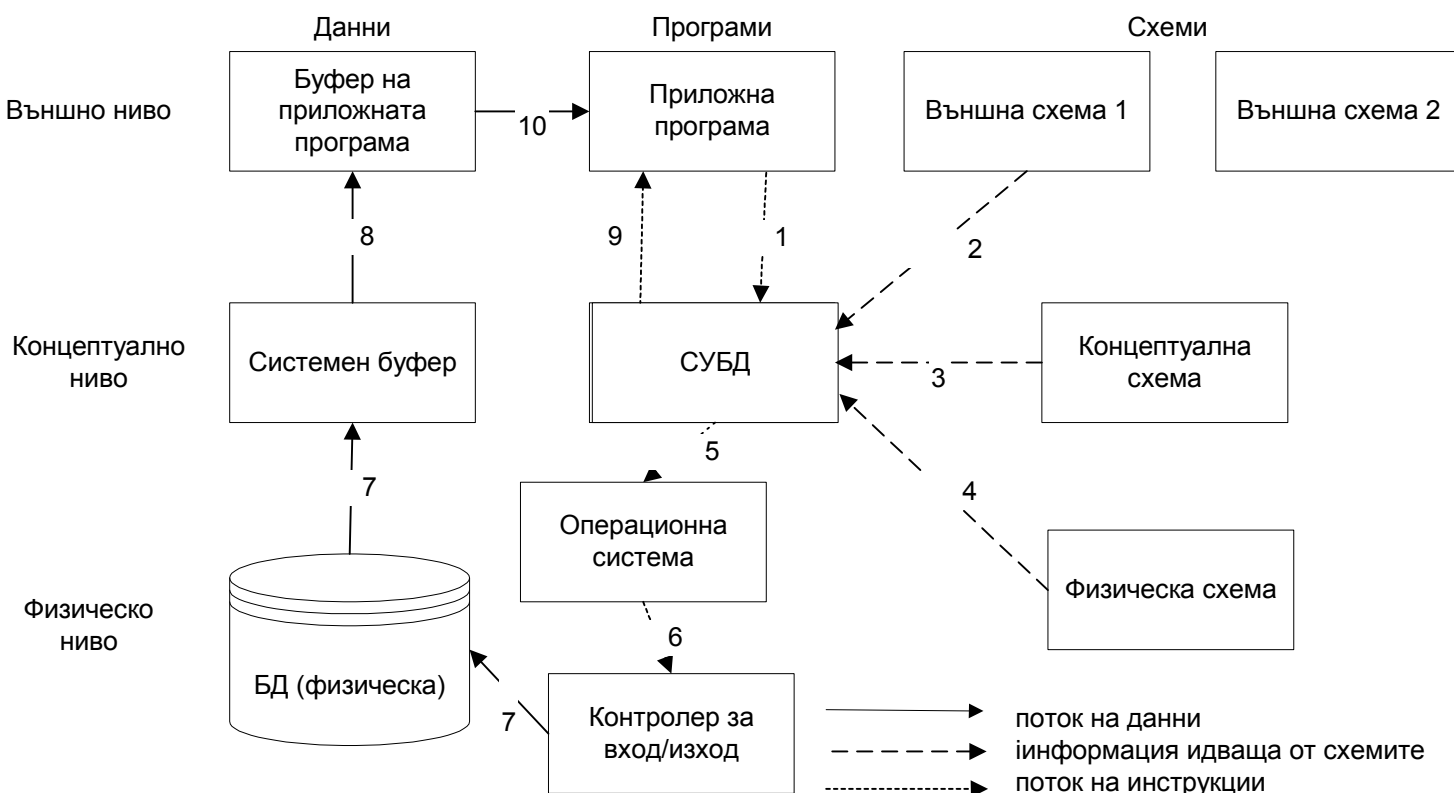
Език за манипулация на данните (DML)

DML се използва за се пишат приложните програми. Един DML има всички характеристики на език за програмиране: условни и итеративни конструкции и присвояване, Освен това той съдържа специфични оператори, които позволяват да се обръща към данните за манипулация и да определя типа на манипулацията: добавяне, обновяване, изтриване, търсене и запитване.

В някои СУБД се използва универсален език, обогатен със специфични оператори и процедури. Други СУБД използват автономен език.

Изпълнение на приложна програма от СУБД

Когато програмист пише приложна програма, той се води само от собственото си знание за БД, т.е. от външната схема. СУБД ще трябва да интерпретира инструкциите, изразени с понятията на външната схема, за да ги преобразува в тези на концептуалната схема и по-нататък в



Фигура 1.6. Изпълнение на оператор за четене

Информатика II – 1. Бази от данни и СУБД

команди на физическото ниво, Един пример: (фигура 1.6):

- 1) Искане за четене е изпратено на СУБД
 - 2) Искането е анализирано с помощта на външната схема, за да провери от една страна, че потребителят има право на достъп до конкретните данни, и от друга страна да предаде характеристиките на тези данни чрез каталога съдържащ външна схема 1.
 - 3) СУБД преглежда концептуалната схема, за да извлече логическия тип на данните за четене.
 - 4) СУБД преглежда физическата схема и намира физическия запис, който трябва да прочете.
 - 5) СУБД изпраща команда за четене на операционната система.
 - 6) Операционната система получава команда и я анализира, като преглежда някои елементи на физическата схема. След това изпраща команда на контролера, управляващ устройството, на което се намират данните.
 - 7) Намерените данни са прехвърлени в системния буфер.
 - 8) Измежду данните в буфера СУБД отделя само тези, от които приложната програма се нуждае. Тя извършва всички преобразувания между концептуалната и външната схеми и прехвърля данните в буфера на приложната програма.
 - 9) СУБД информира приложната програма за евентуални грешки възникнали по време на четенето, за да може програмата да реагира съответно.
 - 10) Приложната програма разполага с исканите данни и може да започне следваща операция.
- Съвременните СУБД са като нови слоеве на операционната система.

Понятието “независимост на програмата от данните”

Това понятие е много трудно за дефиниране и е по-лесно да се дефинира противоположното. Например, ако едно множество от данни използва таблица-индекс за достъп, програмистът който иска да обработва данни от това множество, ще трябва да разработи своята програма въз основа на тази физическа организация. В този случай ние казваме, че СУБД не притежава физическа независимост, тъй като не е възможно да се промени организацията без да се промени и приложната програма. Физическата независимост позволява на администратора да промени (за повишение на производителността) физическата организация без да променя концептуалната схема и приложните програми.

Съответствието между концептуалното ниво и външното ниво отговаря на така наречената логическа независимост. Тя има за цел да позволи изменението на концептуалната схема, например, добавянето на нов тип обекти без да се променят външните програми, които не обработват този нов тип.