# 2. Въведение в СУБД Лекционен курс "Бази от данни" Данни и информация Често се използват като синоними, но те имат различни значения. □ Данни ✓ стойно данти \* стойностите, които се съхраняват в базата данни (конкретни факти); \* необработен материал – напр. телефонен указател. Информация ✓ данни, обработени така, че да бъдат смислени и полезни за потребителя; ✓ знания, сведения, конкретна част от данни с конкретно значение; знания, сведения, конкретна част от данни с конкретно значение; нонякога информацията води до данни; обратно, информацията е често резултат от обработка на данни; структурирането на данните е полезно за намиране на информация – в примера за телефонния указател информация е телефона на личния лекар или на фризьора. "Данните са това, което съхраняваме; информацията е това, което извличаме" Данни Data... modeling, концептуално ✓ Структуриране во Структуриране на концепциите за данни в логически, разбираеми и взаимно свързани групи свързани групи «Представлява изследване на бизнеса и данните, за да се определи структурата на бизнее информацията и правилата, на които се подчинява «Тази структура може по-късно да бъде използвана за създаване на хранилище за бизнее данните - концептуалното моделиране е независимо от техниките за Отнася се за реализацията с определени софтуерна и хардуерна платформи Силно зависимо от текущото състояние на технологиите – след време моделът може да остарее base оизе Набор от данни – в хартиен или електронен формат Към момента най-популярни са релационните бази данни, но има и други модели, които все още се използват warehouse Orpomen набор от организирани данни (информация) Съставена от данни от множество източници, оптимизирана за Online Analytical Processing операции

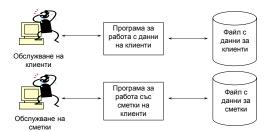
Системи	32	работа	c	файлове
Системи	3d	paoora	C	шаилове

Дори най-ранните компютърни системи са били използвани за обработка на данни с цел извличане на информация.

Te са били, в общия случай, по-точни и по-бързи от алтернативната им обработка на информацията на ръка.

Тези системи са записвали данните в отделни файлове и затова са наречени системи за работа с файлове.

# Пример за финансова файлова система



# Общ вид на система за работа с файлове



Въпреки че тези системи са голяма стъпка напред спрямо ръчната обработка, те имат някои от		
изброените по-долу недостатъци:		
□ Данните често се дублират		
	-	
Означава едни и същи данни да бъдат съхранени повече от веднъж. Това е проблем в системите за работа с файлове		
заради децентрализирания подход. Основните недостатъци от дублирането са:		
✓ Прахосване на памет – това директно се отразява на		
разходите, които ще нараснат; √ Загуба на интегритет – при промяна на данни, записани		
на повече от едно място, може да се пропусне промяна на някое от местата и по този начин да се получат		
противоречиви данни.		
Зависимост на данните		
Означава, че приложните програми, работещи с тях, са		
зависими от формата на данните.		
✓ Ако трябва да бъдат правени модификации във формата		
на данните приложенията, които ги използват, трябва да бъдат пренаписани;		
<ul> <li>✓ Ако приложенията са независими от структурата на</li> </ul>		
съхранение говорим за независимост на данните. Но в		
системите за работа с файлове е налице зависимост на данните.		
A		
<ul> <li>Данните са разделени и съответно изолирани</li> </ul>		
<b>—</b> даните са расдотени и сво восто пострани		
Данните се съхраняват в отделни файлове и съответно са		
изолирани – това затруднява достъпа до данните. В приложенията трябва да е добре направена		
синхронизацията при обработката на данни от два файла,		
за да е сигурно, че извлечените данни са коректни.		
□ Трудно е да се представят сложни обекти като		
се използват системи за работа с файлове		

Технологията на базите данни е разработена да преодолее ограниченията на системите за работа с файлове.	
Някои от предимствата при използването на система за управление на бази от данни:	
<ul> <li>Излишеството на данни може да бъде намалено</li> <li>В системите, които не използват бази от данни,</li> <li>приложенията съхраняват данните обикновено в</li> </ul>	
собствени файлове, като по този начин може да се стигне до дублиране на информация в различните файлове;	
<ul> <li>Неконсистентността на данните може да бъде избегната</li> <li>Нека даден служител работи в определен отдел и този</li> </ul>	
факт се съхранява в повече от един обект. Тогава при промяна на отдела, в който работи служителят, е възможен пропуск при актуализацията във всички обекти, в които се съхранява този факт, което ще	
доведе базата данни до неконсистентно състояние, т.е. информацията в нея ще бъде противоречива. Това би могло да бъде избегнато, ако всеки факт се записва на	
едно място, т.е. <u>излишъкът бъде намален;</u>	
<ul> <li>Данните могат да бъдат споделени</li> </ul>	
Различни приложения биха могли да работят с данните без да е необходимо всяко от тях да използва собствени структури за съхраняването им;	
<ul> <li>Могат да бъдат наложени стандарти</li> <li>Това би улеснило обмена на данни между различни</li> </ul>	
системи, също и мигрирането на данни от една система към друга;	

$\triangleright$	Могат д	а бъдат	наложени	ограничения	с цел
	сигурно	СТ			

Различни правила могат да се налагат относно всеки достъп до определена част от данните (добавяне, промяна, изтриване, извличане);

#### > Данните са интегрирани

Това е проблемът данните в базата да бъдат коректни дори когато има конкурентен достъп до тях от множество потребители;

- Премахва се зависимостта на приложенията от формата на данните във файла
- Позволява сложни обекти да бъдат лесно представяни и извличани

На практика програмите за работа с файлове осъществяват достъп до файловете с данни, а програмите за работа с бази данни осъществяват връзка със системата за управление на бази данни, която осъществява достъп до съхранените данни.

Разликата е съществена, защото тя освобождава програмиста от отговорността за това как се съхраняват данните физически и той може да се концентрира върху продуктивността на програмата за работа с бази данни.

### Достъп при използване на БД



# Общ вид на система за работа с БД



# Обобщение на разликите между системите за работа с файлове и СУБД

	С-ми за работа с файлове	СУБД
Повторение на данни	Да	Намалено е
Зависимост на данните	Оперия	Гарантирана независимост
Неконсистентност на данните	Налице е заради повторението	Може да бъде избягната
Централизация на данните	Липсва такава	Постигната е, улеснява и въпроса за сигурността
Интегритет на данните	Едновременният достъп е под въпрос	Гарантиран от наличието на транзакции
Поддръжка на различни изгледи (views)	Няма	Налице е, улеснява и сигурността
Споделяне на данни	Само чрез добра синхронизация	Основна функция, чрез налагане на стандарти

#### Основни понятия

В най-общия смисъл една <u>база данни (БД)</u> е компютърна система за регистриране и поддържане на <u>логически</u> свързани помежду си структури от данни.

Самата БД може да бъде разглеждана като място за съхраняване на данни.

Една баз	а данни	съдържа:
----------	---------	----------

- Съдържателни (оперативни) данни
- Описание на собствената си структура

# Това описание се среща под различни наименования:

- Речник на данните
- Каталог на данните
- Метаданни

(всички те означават едно и също – данни за данните)

Базата данни също така е и колекция от интегрирани записи, като освен данните и речника на данни тя включва и описание на връзки между отделните записи.

Тези връзки се съхраняват и се използват по време на работа на базата данни.

Потребителите на такива системи обикновено могат да извършват следните операции с данните:

- добавяне на нови данни в БД
- променяне на съществуващи данни
- извличане на съществуващи данни
- отстраняване на ненужни данни

Примерна база данни с таблица CELLAR, съдържаща данни за винарни и тяхната продукция.

ID	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Шардоне	Стара Изба	2004	100	2006
3	Шардоне	Карнобат	2003	350	2005
6	Шардоне	Пещера	2004	76	2006
12	Мерло	Стамболово	2001	1200	2007
21	Мерло	Вини	2003	230	2006
22	Каб.Совиньон	Стамболово	2002	120	2005
30	Каб.Совиньон	Сунгурларе	2003	50	2004
43	Каб.Совиньон	Търговище	1999	80	2002
45	Мавруд	Асеновград	2005	220	2007
47	Пино Ноар	Сунгурларе	1998	100	2001

Извличане	на	данни
-----------	----	-------

SELECT WINE, ID, PRODUCER FROM CELLAR WHERE READY = 2006

WINE	ID	PRODUCER
Шардоне	2	Стара Изба
Шардоне	6	Пещера
Мерло	21	Вини

#### Въвеждане на данни

INSERT INTO CELLAR(ID, WINE, PRODUCER, YEAR, BOTTLES, READY) VALUES (53, 'Пино Hoap', 'Търговище', 92, 1, 96)

#### Промяна на данни

UPDATE CELLAR SET BOTTLES = 40 WHERE ID = 3

#### Изтриване на данни

DELETE FROM CELLAR WHERE ID = 2

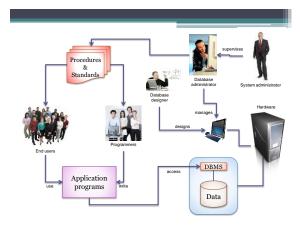
Компоненти на една система	38
управление на база данни	

- <u>Хардуер</u> този компонент е необходим, за да е налице платформа на която ще работи СУБД.
- Софтуер за поддръжка на една СУБД са необходими разнообразни програми, напр. операционна система и софтуер, изграждащ самата СУБД.

- <u>Данни</u> информация за съхраняваните в базата идентичности.
- <u>Процедури</u> действията, които потребителите могат да извършват, като напр.:
  - ✓ включване в системата;
  - ✓промяна на дизайна й;
  - √ въвеждане, редактиране, изтриване на данни;
  - √стартиране, спиране, възстановяване и управление на производителността.

- Потребители потребителите са последният компонент от една СУБД. Съществуват различни видове потребители:
  - Крайни потребители използват системата за извършване на различни информационни дейности;
  - Приложни програмисти разработчиците на приложения за работа с базата данни;
  - Администратори поддържат функционирането на БД.

-		
-		



# Режими на работа

Спрямо режимите на работа БД могат да бъдат:

- <u>Еднопотребителски</u> във всеки момент със системата може да работи само един потребител;
- <u>Многопотребителски</u> във всеки момент със системата могат да работят повече от един потребители, конкурирайки се за ресурсите на машината

Обикновено при многопотребителски режим СУБД предлагат този комфорт на потребителите, че всеки от тях има усещането, че работи сам със системата.

Сложните проблеми по разпределение, заемане и освобождаване на ресурсите на системата, решаване на конфликтни ситуации и др., се решават на системно ниво от СУБД.

Данни	
✓ Оперативни (експлоатационни) данни	
Данните, които се съхраняват за по-	
продължително време в БД. В този контекст една	
БД може да се дефинира като множество от съхранени оперативни данни, които се използват	
(обработват) от приложните системи на	
различни потребители.	
√ Входни данни	
Представляват първична информация, която се	
въвежда в системата обикновено еднократно с помощта на клавиатура, скенер, светлинна	
писалка и т.н.	
Могат ли входните	
данни да предизвикат	
необходимост от	
промяна на някои	
оперативни данни?	
Могат ли входните	
данни да станат	
част от БД (т.е.	
оперативни)?	
оперативни):	
- 1	



#### ✓ Изходни данни

Това са резултати от обработката на заявките, които се извеждат на печат или на екран. Те се получават като резултат от определени операции с оперативните данни и по принцип не са част от самата БД.



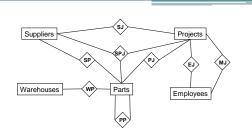


#### Обекти и взаимоотношения

Производствена компания:

- Проекти
- Части, използвани в проектите
- Доставчици на частитеСкладове, съхраняващи частите
- Служители, работещи по проектите

Изброени са основните обекти, за които компанията ще съхранява информация.



Примерна Entity/Relationship диаграма

## Взаимоотношения (Relationships)

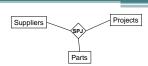
Важно е да се разбере, че обектите имат взаимоотношения, напр.:

- SP всеки доставчик доставя определена част, и обратно, всяка част се доставя от определен доставчик;
- РЈ частите се използват в проекти; проектите използват части;
- WP частите се съхраняват в складове; складовете съхраняват части.

Изброените взаимоотношения са двупосочни (bidirectional):

- Ако имаме доставчик можем да намерим частите, които той доставя
- Ако имаме част можем да намерим доставчиците, които я доставят

Важно е да се отбележи, че изброените взаимоотношения са също част от данните, както и обектите, така че трябва също да бъдат представени в базата данни.



Повечето взаимоотношения на диаграмата включват два обекта — те са бинарни (двоични). Но има едно, което включва три обекта — SPJ (тройно).

Интерпретацията му е следната – конкретни доставчици доставят определени части за конкретни проекти.

SPJ ≠ SP + PJ + SJ	
Тройното взаимоотношение не е еквивалентно на комбинацията на трите двоични:	
на комоинацията на трите двоични.  ✓ "доставчици доставят части"  ✓ "части се използват в проекти"	
√ "проекти са обслужвани от доставчици"	
Пример	
<ul><li>а) Петър доставя болтове за проект "Балкан"</li><li>b) Петър доставя болтове</li></ul>	
с) Болтове се използват в проект "Балкан" d) Проектът "Балкан" се обслужва от Петър	
Не можем да стигнем до а) знаейки само	
b), c) и d). Знаейки тях можем да заключим, че:	
<ul> <li>✓ Петър доставя болтове за проект X</li> <li>✓ Някой доставчик Y доставя болтове за проект "Балкан"</li> </ul>	
✓ Петър доставя части Z за проект "Балкан"  Но не можем вярно да заключим, че:	
✓ X = "Балкан" ✓ Y = Петър	
$\checkmark \ Z = \mbox{болтове}$ Подобни грешни заключения понякога се наричат "капани на	
връжите".	

Заключения	
<ol> <li>В диаграмата има и едно взаимоотношение, включващо само</li> </ol>	
един обект – PP – една част е съставена от други части. Това взаимоотношение е	
унарно или рекурсивно.	
Друг пример:	
EMPLOYEES  MENARHMIX  NODPHHIMM  NODPHHIMM	
Reports To	
• Служители са подчинени на други служители;	
<ul> <li>Взаимоотношението ReportsTo има вида (emp1, emp2),</li> <li>където и emp1, и emp2 са елементи от EMPLOYEES;</li> </ul>	
<ul> <li>Обаче имат различни роли – emp1 се отчита пред контролиращият го emp2, което определя ролите им на</li> </ul>	
подчинен и началник.	-
	_
II. Взаимоотношенията също могат да	
бъдат разглеждани като обекти в случаите, когато носят информация,	
която трябва да бъде съхранена в базата данни.	
диния.	

Примери	П	ри	ме	ри
---------	---	----	----	----

- Частта Р4 се съхранява в склад W8 взаимоотношение, от което бихме искали да съхраним информация, напр. какво е количеството от частта Р4, което се съхранява в склад W8.
- Доставчик Петър доставя болтове за проект "Балкан" – може да се съхрани информация за това на каква цена са били доставени болтовете конкретно за този проект.

### Независимост на данните

- По-старите системи в повечето случаи са зависими от данните – в смисъл на това как данните са били съхранени във вторичната памет и техниките за достъп до тях – това е вградено в логиката и кода на приложенията
- Такива приложения са зависими от данните, защото всяка промяна в структурата на съхранение на данните афектира приложението, осъществяващо достъп до тях

В базите данни, обаче, е крайно нежелателно приложенията да бъдат зависими от данните, поне поради следните причини:

 Различните приложения ще имат нужда от различни гледни точки върху едни и същи данни;

Например: приложение А записва телефонните номера в определено поле като низ, а В – като числа (в същото поле). Тогава ако в базата данни полето е от тип низ, при всеки достъп да В до това поле ще се налага преобразуване на данни.

<ul> <li>Администраторът на СУБД трябва да има свободата</li> </ul>
да променя структурата на съхраняване на данните
без това да налага промяна на приложенията,
работещи с тях
Например:
<ul> <li>Добавяне на нови видове данни в базата;</li> <li>Налагане на нови стандарти;</li> </ul>
<ul> <li>налагане на нови станоарти;</li> <li>Използване на нови типове устройства за съхранение на данни и</li> </ul>
т.н.
Ако приложенията са зависими от данните подобни промени биха предизвикали нужда от промяна и на
приложенията.
Следователно осигуряването на независимост
на данните е една от основните цели на базите
данни.
Независимостта на данните може да бъде
дефинирана като имунитет на приложенията
към промяната на структурата на съхранение и
техниките за достъп до данните.
Видове бази данни
видове оази данни
• Операционни (транзакционни) – Online
Transaction Processing (OLTP)
<ul> <li>Този вид бази данни се използват при обработка на транзакции</li> </ul>
в реално време, т.е. в ситуации, където е налице ежедневно
събиране, променяне и поддържане на данни.
<ul> <li>Типът данни, който се съхранява в такава база данни, е</li> </ul>
динамичен, което означава, че той се променя постоянно и
винаги отразява най-новата информация.
<ul> <li>Такъв тип бази данни използват магазини и складове за</li> </ul>
продажба на стоки, банки, телекомуникационни компании и др.

•	• Аналитични – Online Analytical Processing (OLAP, Decision Support Systems)
<b>A</b>	Аналитичните бази данни се използват главно при аналитична обработка в реално време, където има нужда да се съхраняват и проследяват минали и зависещи от времето данни.
7	Една такава база данни е ценно предимство, когато има нужда да се
_	следят тенденциите, да се извлича статистическа информация за
	определен период и да се правят тактически и стратегически бизнес прогнози.
1	•
×	<ul> <li>Този вид бази данни съхраняват статични данни, което означава, че данните не се променят или се променят много рядко. Намират</li> </ul>
	приложение в компаниите за маркетингови анализи, изследователски лаборатории, състезателни отбори от Формула 1 и
	др.
	□ Аналитичните бази данни често използват операционните бази
	данни като основен източник на данни, затова често съществува
	връзка между двата типа бази данни на дадена компания.
	□ Но въпреки това тези два типа бази данни са предназначени да
	задоволяват съвсем различни нужди и създаването на техните
	структури изисква коренно различни дизайнерски методологии.
	□ ∏o prove no wrose we oppose pure services.
	□ По време на курса ще спрем вниманието си на операционните бази данни, защото те са най-широко използвания вид в
	световен мащаб към момента.
	Примери за аналитични бази данни:
	ווף וויים אוויים
	<ul> <li>Data warehouse database – могат да бъдат моделирани по подобен на транзакционните начин, макар че често са</li> </ul>
	денормализирани. Те обикновено съдържат данни от години
	назад с цел прогнозиране на бъдещи резултати и събития;
	- $Data\ mart$ – обикновено подмножество от данни на горните;
	• Reporting database – често са data warehouse системи, но
	съдържащи само активни (текущи) данни, не от предишни
	периоди или архивирани.

Хибридни
Комбинация от транзакционна и аналитична база от данни;
•
Подходяща за компании с по-малки обеми от данни (и брой транзакции);
Икономически по-изгоден вариант от две отделни – по-малко хардуер, софтуер и поддръжка.