

## Резюме предыдущих лекций

1. Система баз данных (СБД) – это набор данных, относящихся к определённой организованной деятельности.
2. Доступ к СБД предоставляется одновременно многим пользователям – участникам деятельности.
3. СБД управляется специальным программным комплексом: СУБД + приложения.
4. СУБД управляет доступом приложений к хранимым данным.
5. Для каждого приложения создаётся своя проекция хранимых данных.
6. Пользователь приложения видит только эту проекцию.  
Это и есть его база данных.
7. СБД содержит информацию о самой себе — метаданные.
8. Метаданные необходимы СУБД  
для управления данными конечных пользователей.
9. Технология баз данных решает проблемы накопления и обработки данных предприятий.

## 4. Архитектура и функции СУБД

О первой попытке создания общей архитектуры СБД  
[https://ru.wikipedia.org/wiki/Data\\_Base\\_Task\\_Group](https://ru.wikipedia.org/wiki/Data_Base_Task_Group)

### 4.1 Архитектура ANSI/SPARC

Предложена в 1975 году Рабочей группой по базам данных (РГБД) при Комитете планирования стандартов и норм (SPARC) Национального института стандартизации США (ANSI).

Цель

Скрыть от прикладных программ детали размещения данных во внешней памяти.



## Отображение концептуальный ↔ внутренний

### Концептуальная схема

Концепт.запись1

A	B	C
---	---	---

Концепт.запись2

D	E	F	G
---	---	---	---

Концепт.запись3

H	K	L
---	---	---

Внутр.запись1

H	E	A	L	C	s1
---	---	---	---	---	----

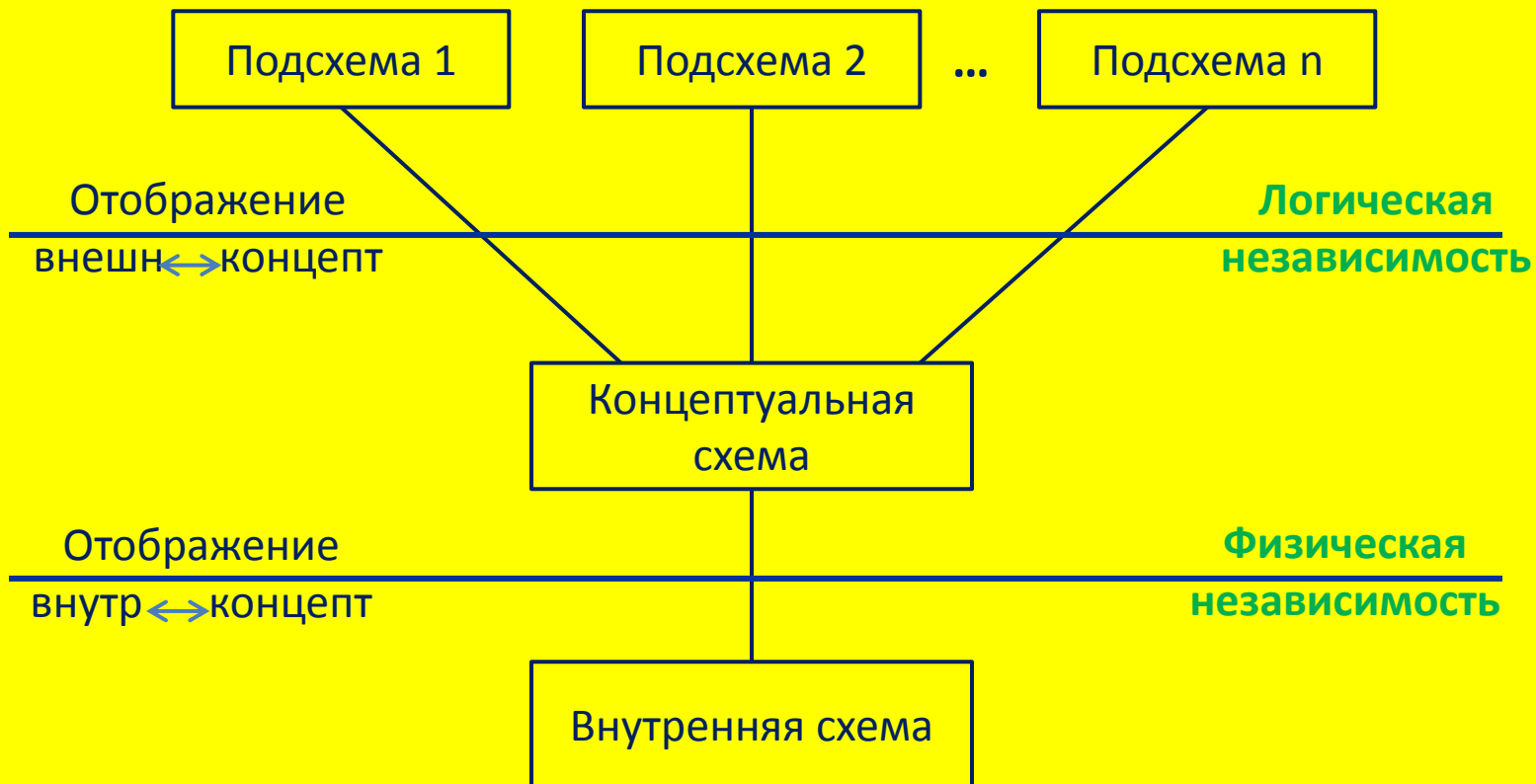
Внутр.запись2

B	K	F	G	D	s2
---	---	---	---	---	----

Служебные поля

### Внутренняя схема

## Уровни независимости от данных



## 4.2. Функции СУБД

**ГЛАВНАЯ** — обеспечить прикладным программам доступ к данным на логическом уровне.

### **ДОПОЛНИТЕЛЬНЫЕ**

Обеспечить:

- *целостность* (согласованность) базы данных;
- управление *многопользовательским доступом* к данным;
- *защиту данных* от несанкционированного доступа;
- *восстановление данных*, разрушенных вследствие аварий;
- поддержку *среды разработки* прикладных программ;
- поддержку *среды обслуживания* набора данных.

#### 4.2.1 Обеспечение доступа к данным

**Обмен данными между ПП и ФБД только через рабочий буфер СУБД.**

**Рабочий буфер — это актуальная часть БД.**

По запросу приложения СУБД

- загружает в буфер БД нужные страницы внешней памяти,  
если они ещё не загружены;
- формирует требуемые записи внешнего уровня;
- выполняет в буфере все запрошенные обновления записей.

**Эти действия изменяют ТОЛЬКО состояние рабочего буфера.**

Если новое состояние буфера удовлетворяет ОЦ, оно становится частью БД.  
В противном случае обновления отменяются.

Данные НЕ ВЫГРУЖАЮТСЯ из буфера по факту завершения приложения.  
В ФБД выгружаются редко используемые данные по факту переполнения буфера.

#### **Буферизация**

во-первых, уменьшает среднее время реакции СБД на запрос и  
во-вторых, обеспечивает возможность поддержания целостности БД.



#### 4.2.2. Операции над базой данных

##### RETRIEVE

Считывание в рабочий буфер совокупности записей ФБД и формирование в буфере совокупности внешних записей.

Возможны: *сортировка, группирование записей, агрегирование данных*

##### INSERT

Создание в рабочем буфере новых экземпляров записей, содержащих новые значения данных.

##### UPDATE

Замена существующих значений указанных полей в извлечённом множестве записей новыми значениями.

##### DELETE

Удаление указанного подмножества записей.

Записи могут *уничтожаться* или *помечаться как удалённые*, но физически сохраняться в буфере.



При исполнении операций **RETRIEVE, INSERT, UPDATE, DELETE**

**никаких изменений в физической базе данных не происходит.**

Изменяется только состояние буфера

Если новое состояние буфера удовлетворяет ограничениям целостности,  
оно становится частью БД.

В противном случае обновления отменяются.

Буфер возвращается в исходное состояние.

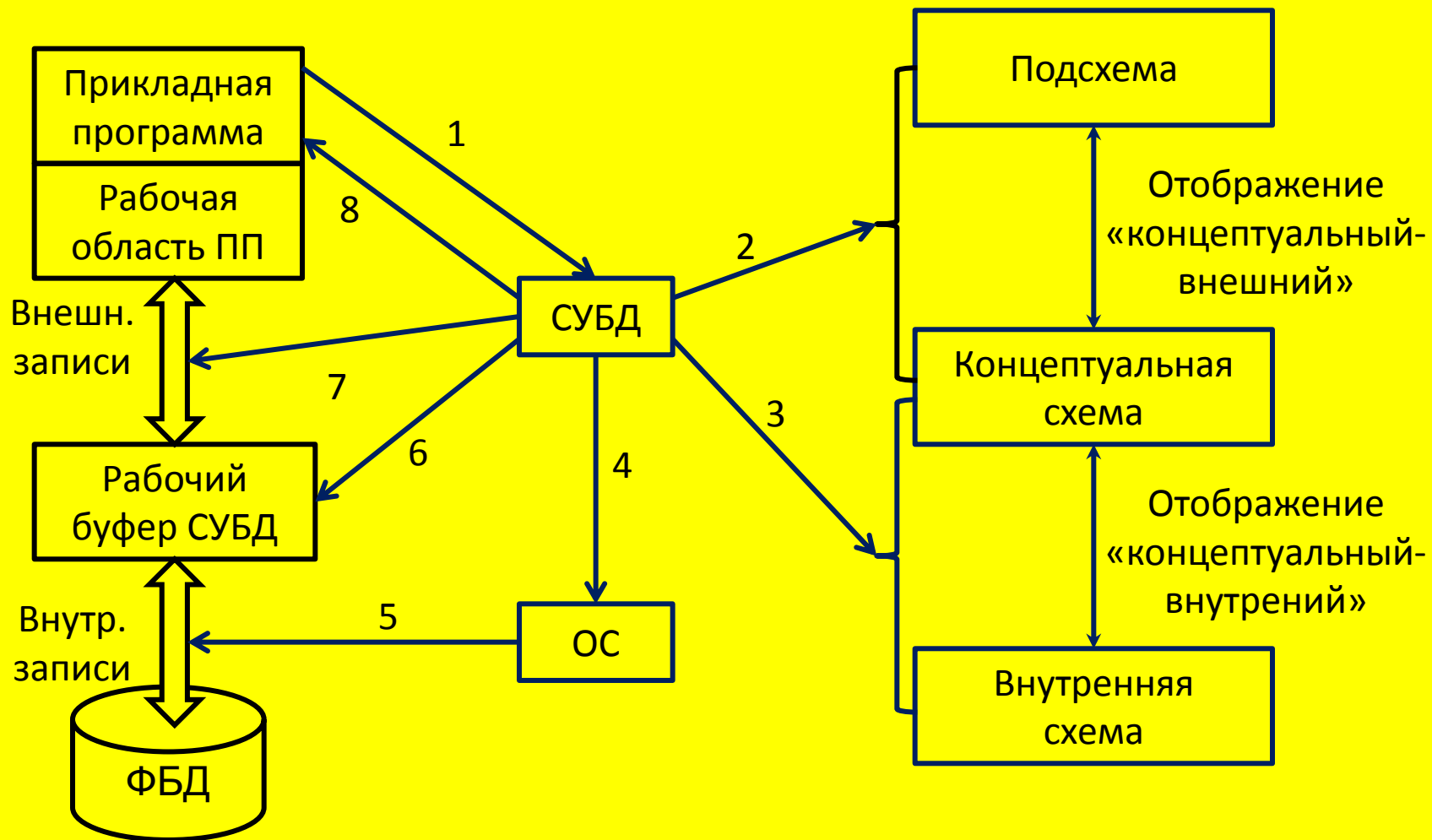
Данные **НЕ ВЫГРУЖАЮТСЯ** из буфера

**по завершении приложения.**

В ФБД выгружаются редко используемые данные

**по факту переполнения буфера.**

#### 4.2.3. Схема обработки запроса на извлечение данных



## 4.3. Обеспечение целостности данных

### 4.3.1. Понятие целостности данных

Данные в БД должны быть *полными и достоверными.*

**Гарантировать это невозможно.** Однако...

Полные и достоверные данные

*всегда имеют осмысленную интерпретацию.*

**Обратное неверно!!!**

Интерпретируемый в терминах ПО набор данных называют

**ЦЕЛОСТНЫМ**

**Целостный набор данных ограничен**

- свойствами объектов ПО,
- свойствами отношений объектов,
- правилами ПО (бизнес-правилами)
- соображениями здравого смысла.

### Примеры (ограничения целостности)

1. Возраст солдата срочной службы не может быть меньше 18 и больше 25 лет.
2. Численность мотострелкового отделения не превышает 8 человек.
3. Число единиц личного оружия в отделении не превышает численности отделения.
4. Номер мотострелкового отделения уникален в пределах роты.

Ограничение  
целостности

Предикат,  
определённый на объектах БД.

СУБД может обеспечить целостность БД в *пределах заданных ОЦ*.

### **УРОВНИ ОЦ**

Уровень атрибута

Уровень экземпляра записи

Уровень таблицы

Уровень базы данных

**СУБД должна гарантировать  
целостность набора данных пользователя  
в пределах заданных ОЦ.**

#### 4.3.2. Механизмы поддержки целостности

##### Внутренний

Реализован в ядре СУБД.

Обеспечивает контроль

целостности атрибута,  
уникальности записи и  
целостности ссылок.

ОЦ объявляются средствами DDL,

Сохраняются как объекты БД.

Используются СУБД при попытках обновления состояния буфера.

Date  
Definition  
Language

##### Внешний

Реализуется проектировщиком БД.

Обеспечивает контроль ОЦ,

которые невозможно объявить средствами DDL.

## Виды ОЦ

### Немедленно проверяемое

Проверяется в момент завершения операции обновления, которая может нарушить это ограничение.

### С отложенной проверкой

*Заведомо нарушается* одиночной операцией обновления. Проверка возможна только по завершении **серии** обновлений.

### Пример

Перевод денег с одного банковского счёта на другой

- а) извлечь запись счёта-источника (**RETRIEVE**),
- б) уменьшить его остаток на переводимую сумму  $S$  (**UPDATE**),
- в) извлечь запись счёта-приёмника (**RETRIEVE**),
- г) увеличить его остаток на сумму  $S$  (**UPDATE**).

**Только после операции г)** состояние счетов будет согласованным.

**В общем случае** целостность БД может быть проверена лишь по исполнении *логически завершённой последовательности* операций обновления.

Подробнее <http://citforum.ru/database/dblearn/> главы 3, 9