# ТЕМА 3: Система управления базами данных

**СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ** – совокупность языковых и программных средств, предназначенных для создания, ведения и конкурентного использования базы данных многими пользователями

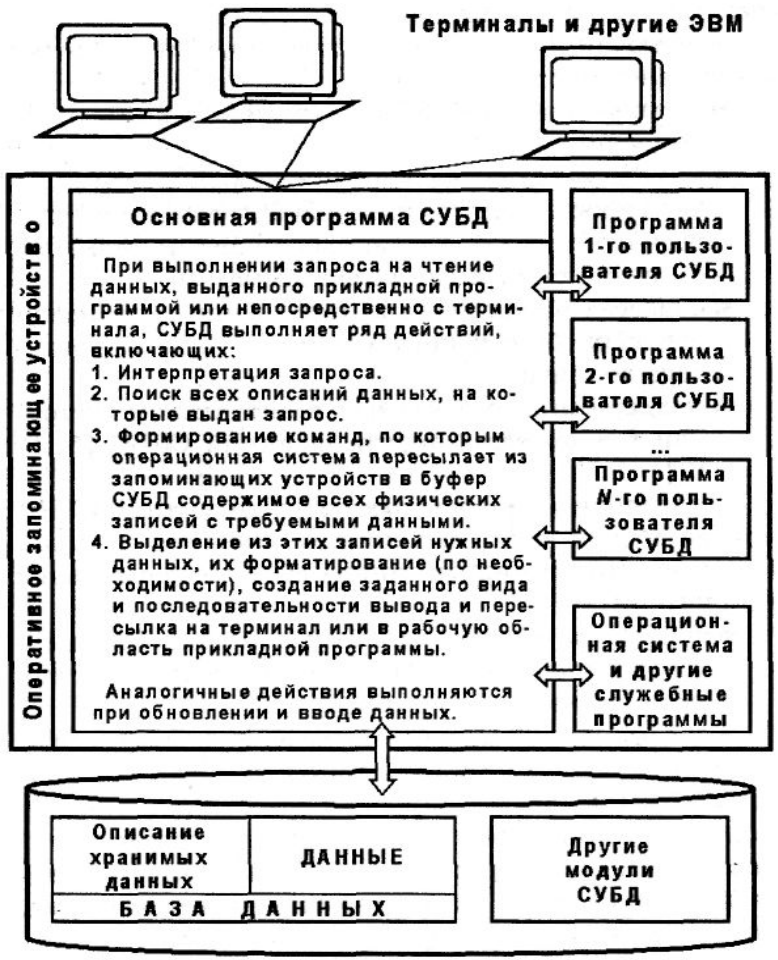
**Основная особенность СУБД** – это наличие процедур для ввода и хранения не только самих данных, но и описаний их структуры. Файлы, снабженные описанием хранимых в них данных и находящиеся под управлением СУБД, называются в свою очередь базами данных.

**Концепция СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ имеет ряд неоспоримых преимуществ:**

1. Минимальная избыточность хранимых данных.
2. Непротиворечивость хранимых данных.
3. Многоаспектное использование данных.
4. Комплексная оптимизация.
5. Возможность стандартизации представления и обмена данными.
6. Возможность обеспечения санкционированного доступа к данным.

# ТЕМА 3: Система управления базами данных

## ОБОБЩЕННАЯ АРХИТЕКТУРА СУБД




**СОВРЕМЕННЫЕ СУБД РЕАЛИЗУЮТ ЦЕНТРАЛИЗОВАННОЕ УПРАВЛЕНИЕ ДАННЫМИ И, КРОМЕ ТОГО, ОБЕСПЕЧИВАЮТ:**

а) определение данных, подлежащих хранению в БД;
б) первоначальную загрузку данных в БД;
в) обновление данных;
г) доступ к данным по различным запросам пользователя, отбор и извлечение некоторой части БД, редактирование извлеченных данных и выдачу их пользователю.

**СУБД должна предоставлять доступ к данным посредством прикладных программ любым пользователям, включая и тех, которые практически не имеют представления о:**

1) физическом размещении в памяти данных и их описаний;
2) механизмах поиска запрашиваемых данных;

3) проблемах, возникающих при одновременном запросе одних и тех же данных многими пользователями;

4) способах обеспечения защиты данных от некорректных обновлений и (или) несанкционированного доступа;

5) поддержании баз данных в актуальном состоянии и множестве других функций СУБД.

# ТЕМА 3: Система управления базами данных

## ДОСТОИНСТВА СУБД

1. Контроль за избыточностью данных;
2. Непротиворечивость данных;
3. Больше полезной информации при том же объеме хранимых данных;
4. Совместное использование данных;
5. Поддержка целостности данных;
6. Повышенная безопасность;
7. Применение стандартов;
8. Повышение эффективности с ростом масштабов системы;
9. Возможность нахождения компромисса при противоречивых требованиях;
10. Повышение доступности данных;
11. Улучшение показателей производительности;
12. Упрощение сопровождения системы за счет независимости данных;
13. Улучшенное управление параллельностью;
14. Развитые службы резервного копирования и восстановления.

## Недостатки СУБД:

1. Сложность;
2. Размер;
3. Стоимость;
4. Дополнительные затраты на аппаратное обеспечение;
5. Затраты на преобразование;
6. Производительность;
7. Серьезные последствия при выходе системы из строя.

# ТЕМА 3: Система управления базами данных

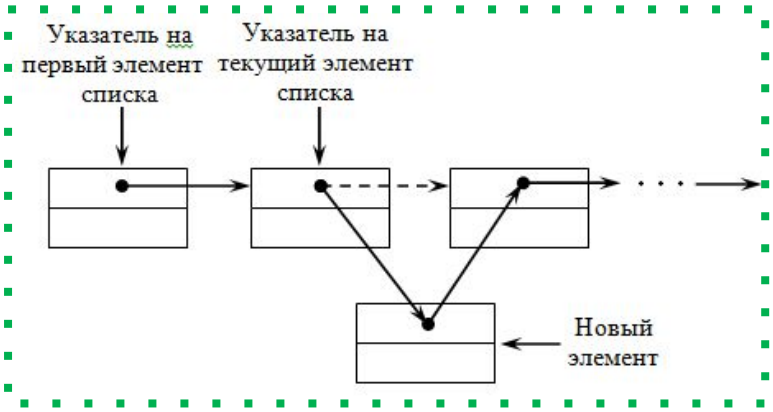
## ФИЗИЧЕСКАЯ ОРГАНИЗАЦИЯ ДАННЫХ В СУБД

### 1. ЛИНЕЙНЫЙ СПИСОК

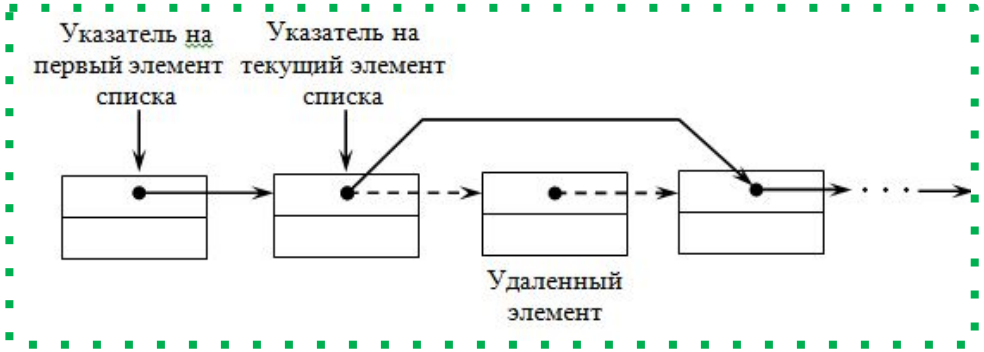
Линейный (последовательный) список – последовательность записей базы данных, сформированная по некоторым логическим принципам.

При поиске информации, соответствующей некоторым критериям (например, товаров с определенным названием или артикулом), линейный список необходимо просмотреть полностью от первой до последней записи. Это приводит к тому, что рассматриваемая структура хранения, обеспечивая оптимальные требования к минимальному объему выделяемой памяти на внешних устройствах, является неэффективной по быстродействию.


**Вставка элемента в линейный список**




**Удаление элемента из линейного списка**



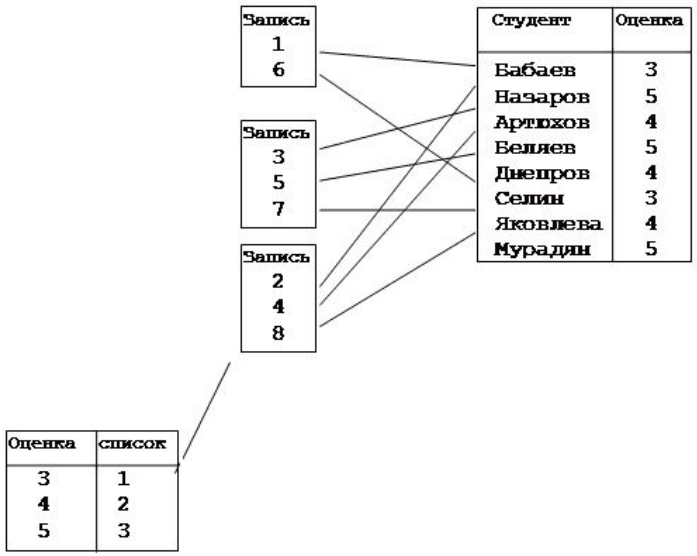
# ТЕМА 3: Система управления базами данных

## ФИЗИЧЕСКАЯ ОРГАНИЗАЦИЯ ДАННЫХ В СУБД

### 2. ИНВЕРТИРОВАННЫЙ СПИСОК

Инвертированные списки позволяют существенно ускорить процесс поиска необходимой информации по сравнению с линейными списками. Это достигается с помощью упорядочивания (сортировки) записей исходного списка по значениям данных в одном из не ключевых полей. Инвертирование исходного списка можно выполнить для отдельных (частичное инвертирование) или всех (полное инвертирование) не ключевых полей исходного списка.

Обеспечивая уменьшение времени выполнения запросов, инвертирование приводит к существенному увеличению объемов внешней памяти для хранения информации в результате ее дублирования. Это увеличение прямо пропорционально количеству полей, по которым производится инвертирование. Для частичного устранения данного недостатка применяется создание индексов.




**Например:** размещается список студентов с оценками. Требуется выбрать всех студентов, имеющих одинаковое значение оценки. Левый прямоугольник символизирует индекс, в котором находится единственный вторичный ключ – «оценка». Каждому его значению соответствует список, в котором перечислены соответствующие номера записей информационного файла. Выбор всех двоечников сводится к нахождению в индексе соответствующего значения ключа «оценка» и загрузки записей, указанных в списке. Если нужно найти тех, кто получил «4» или «5», следует найти и объединить соответствующие списки.

# ТЕМА 3: Система управления базами данных

## ФИЗИЧЕСКАЯ ОРГАНИЗАЦИЯ ДАННЫХ В СУБД

| Порядковый номер записи | Товар | Дата | Кол-во (ед) |
|---|---|---|---|
| 1 | Сахар | 10.01.03 | 10 |
| 2 | Картофель | 15.01.03 | 20 |
| 3 | Свекла | 12.02.03 | 50 |
| 4 | Сахар | 20.02.03 | 10 |
| 5 | Свекла | 10.03.03 | 50 |
| 6 | Сливы | 19.03.03 | 4 |

| Индекс по наименованию | Индекс по количеству |
|---|---|
| Index(1)=2 | Index(1)=3 |
| Index(2)=1 | Index(2)=5 |
| Index(3)=4 | Index(3)=2 |
| Index(4)=3 | Index(4)=1 |
| Index(5)=5 | Index(5)=4 |
| Index(6)=6 | Index(6)=6 |

| Индекс по дате прихода товара | | Индекс по наименованию товара | | Индекс по количеству | |
|---|---|---|---|---|---|
| Дата | Порядковый номер записи | Товар | Порядковый номер записи | Кол-во (ед) | Порядковый номер записи |
| 10.01.03 | 1 | Картофель | 2 | 50 | 3 |
| 15.01.03 | 2 | Сахар | 1 | 50 | 5 |
| 12.02.03 | 3 | Сахар | 4 | 20 | 2 |
| 20.02.03 | 4 | Свекла | 3 | 10 | 1 |
| 10.03.03 | 5 | Свекла | 5 | 10 | 4 |
| 19.03.03 | 6 | Сливы | 6 | 4 | 6 |

### 3. ИНДЕКСЫ

Индексы применяются для ускорения доступа к записям базы данных. Их можно сравнить с предметным указателем книги – упорядоченной последовательностью слов (словосочетаний) с перечнем номеров страниц, на которых встречается это слово (словосочетание).

Индекс базы данных представляет собой структуру, в которой содержатся рассортированные в заданном порядке значения данных в некотором поле и указатели адресов записей (страниц), где находятся эти значения. В отличие от инвертированных списков индексы занимают значительно меньшее место во внешней памяти.

**Поиск и выборка нужных записей в базе данных осуществляются в следующей последовательности:**

1. Выбирается индекс, соответствующий условию поиска
2. В индексе находится строка с заданным условием.
3. Из найденной строки выбираются номера страниц, где хранятся искомые записи.
4. Полученные номера страниц используются для чтения необходимой информации.

Файл, в котором созданы индексы для всех полей размещенной в нем таблицы, занимает практически вдвое больше места по сравнению с исходным файлом. Кроме того, наличие индексов затрудняет обновление информации, хранящейся в базе данных, и требует дополнительных затрат времени на данный процесс.

Поэтому не рекомендуется индексировать все поля таблиц или поля, данные в которых часто изменяются. Наиболее эффективным является создание индексов для первичных и внешних ключей, для полей, по которым в основном выполняются запросы.