

# Этапы развития СУБД

Подготовила Нечаева Н.

# Введение

До того, как появились СУБД, вся информация хранилась так, как решал конкретный разработчик.

Для этого могли использовать функции операционной системы или же устройства ввода-вывода, что не было очень удобным вариантом: функции ОС не позволяли сортировать данные, хранящиеся в файлах, ограничивать запись в них; также был вопрос о совместном доступе к одним и тем же данным без проблем со стороны системы. Все эти проблемы послужили развитию СУБД.

В итоге можно выделить следующие этапы развития СУБД:

1. Централизованное хранение данных
2. Развитие персональных компьютеров
3. Переход от персонализации к интеграции
4. Развитие Интернета

# Первый этап

## “Большие” машины

Базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи, запускаемые в основном в пакетном режиме.

Интерактивный режим доступа обеспечивался с помощью консольных терминалов, которые не обладали собственными вычислительными ресурсами и служили только устройствами ввода-вывода для центральной ЭВМ. Программы доступа к БД писались на различных языках и запускались как обычные числовые программы.

Функции управления распределением ресурсов в основном осуществлялись операционной системой.

В это же время проводилось очень много работ по обоснованию и формализации реляционной модели данных, и была создана первая система (System R), реализующая идеологию реляционной модели данных.

Также появляется определение "транзакций" и первые языки высокого уровня для работ с реляционной моделью данных, но отсутствуют стандарты.

# Второй этап

## Появление компьютеров в домах

Так как пользователи в большинстве своем не имели специальных знаний, а домашние компьютеры обладали меньшей производительностью, то разработанные СУБД им не подходили. Поэтому начали появляться настольные СУБД. Они были рассчитаны на создание БД в основном с монопольным доступом.

Большинство СУБД имели развитый и удобный пользовательский интерфейс. В большинстве существовал интерактивный режим работы с БД как в рамках описания БД, так и в рамках проектирования запросов. Кроме того, большинство СУБД предлагали развитый и удобный инструментарий для разработки готовых приложений без программирования (на основе готовых шаблонов форм, конструкторов запросов).

Во всех СУБД поддерживался только внешний табличный вид структур данных.

При наличии высокоуровневых языков манипулирования данными типа реляционной алгебры и SQL в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.

В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных: эти функции должны были выполнять приложения.

Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД и в связи с этим — к отсутствию инструментальных средств администрирования БД.

# Третий этап

## Распределенные БД. Нынешнее время

Появились технологии "клиент-сервер".

Большинство современных СУБД рассчитаны на многоплатформенную архитектуру, то есть они могут работать на компьютерах с разной архитектурой и под разными операционными системами.

Необходимость поддержки многопользовательской работы с базой данных и возможность децентрализованного хранения данных потребовали развития средств администрирования БД с реализацией общей концепции средств защиты данных.

Для того чтобы не потерять клиентов, которые ранее работали на настольных СУБД, практически все современные СУБД имеют средства подключения клиентских приложений, разработанных с использованием настольных СУБД, и средства экспорта данных.

# Четвертый этап Развитие Интернета.

Основное отличие этого подхода от технологии клиент-сервер состоит в том, что отпадает необходимость использования специализированного клиентского программного обеспечения. Для работы с удаленной базой данных используется стандартный браузер.

При этом встроенный в загружаемые пользователем HTML-страницы код, написанный обычно на языке Java, Java-script, Perl и других, отслеживает все действия пользователя и транслирует их в низкоуровневые SQL-запросы к базе данных, выполняя, таким образом, ту работу, которой в технологии клиент-сервер занимается клиентская программа.

# Некоторые важные замечания

С развитием СУБД выяснилось, что не все задачи решаются с использованием реляционных СУБД и языка для работы с ними - SQL. Исходя из этого появилась необходимость в разработке чего-то нового, позволяющего все-таки разрешить подобные задачи.

В связи с этим появилась и активно развивается идеология NoSQL (Not only SQL), которая объединяет ряд подходов и проектов, связанных с использованием и созданием нереляционных баз данных.