



ВВЕДЕНИЕ

Восприятие реального мира можно соотнести с последовательностью разных, хотя иногда и взаимосвязанных, явлений. С давних времен люди пытались описать эти явления (даже тогда, когда не могли их понять). Такое описание называют данными.

Традиционно фиксация данных осуществляется с помощью конкретного средства общения, например, с помощью естественного языка на конкретном носителе.

В настоящее время успешное функционирование различных фирм, организаций и предприятий просто не возможно без развитой информационной системы, которая позволяет автоматизировать сбор и обработку данных. Обычно для хранения и доступа к данным, содержащим сведения о некоторой предметной области, создается база данных.

База данных (БД) — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Под предметной областью принято понимать некоторую область человеческой деятельности или область реального мира, подлежащих изучению для организации управления и автоматизации, например, предприятие, вуз и т.д.

Система управления базами данных (СУБД) — совокупность языковых и программных средств, предназначенных для создания, наполнения, обновления и удаления баз данных.

Программы, с помощью которых пользователи работают с БД, называются приложениями.

1. ПРИНЦИПЫ ПОСТРОЕНИЯ БАЗ ДАННЫХ

К современным базам данных, а, следовательно, и к СУБД, на которых они строятся, предъявляются следующие основные требования.

1. Высокое быстродействие (малое время отклика на запрос).

Время отклика - промежуток времени от момента запроса к БД до фактического получения данных. Похожим является термин время доступа - промежуток времени между выдачей команды записи (считывания) и фактическим получением данных. Под доступом понимается операция поиска, чтения данных или записи их. Часто операции записи, удаления и модификации данных называют операцией обновления.

2. Простота обновления данных.

3. Независимость данных.

4. Совместное использование данных многими пользователями.

5. Безопасность данных - защита данных от преднамеренного или непреднамеренного нарушения секретности, искажения или разрушения.

6. Стандартизация построения и эксплуатации БД (фактически СУБД).

7. Адекватность отображения данных соответствующей предметной области.

8. Дружелюбный интерфейс пользователя.

Важнейшими являются первые два противоречивых требования: повышение быстродействия требует упрощения структуры БД, что, в свою очередь, затрудняет процедуру обновления данных, увеличивает их избыточность.

Независимость данных - возможность изменения логической и физической структуры БД без изменения представлений пользователей.

Независимость данных предполагает инвариантность к характеру хранения данных, программному обеспечению и техническим средствам. Она обеспечивает минимальные изменения структуры БД при изменениях стратегии доступа к данным и структуры самих исходных данных. Это достигается «смещением» всех изменений на этапы концептуального и логического проектирования с минимальными изменениями на этапе физического проектирования.

Безопасность данных включает их целостность и защиту.

Целостность данных - устойчивость хранимых данных к разрушению и уничтожению, связанных с неисправностями технических средств, системными

ошибками и ошибочными действиями пользователей.

Она предполагает:

1. отсутствие неточно введенных данных или двух одинаковых записей об одном и том же факте;
2. защиту от ошибок при обновлении БД;
3. невозможность удаления (или каскадное удаление) связанных данных разных таблиц;
4. неискажение данных при работе в многопользовательском режиме и в распределенных базах данных;
5. сохранность данных при сбоях техники (восстановление данных).

Целостность обеспечивается триггерами целостности – специальными приложениями-программами, работающими при определенных условиях. Защита данных от несанкционированного доступа предполагает ограничение доступа к конфиденциальным данным и может достигаться:

1. введением системы паролей;
2. получением разрешений от администратора базы данных (АБД);
3. запретом от АБД на доступ к данным;
4. формированием видов - таблиц, производных от исходных и предназначенных конкретным пользователям.

Три последние процедуры легко выполняются в рамках языка структуризованных запросов Structured Query Language - SQL, часто называемого SQL2.

Стандартизация обеспечивает преемственность поколений СУБД, упрощает взаимодействие БД одного поколения СУБД с одинаковыми и различными моделями данных. Стандартизация (ANSI/SPARC) осуществлена в значительной степени в части интерфейса пользователя СУБД и языка SQL. Это позволило успешно решить задачу взаимодействия различных реляционных СУБД как с помощью языка SQL, так и с применением приложения Open DataBase Connection (ODBC). При этом может быть осуществлен как локальный, так и удаленный доступ к данным (технология клиент/сервер или сетевой вариант).

2. КОНЦЕПЦИЯ ПОСТРОЕНИЯ БАЗЫ ДАННЫХ

Существует два подхода к построению БД, базирующихся на двух подходах к созданию автоматизированной системы управления (АСУ).

Первый из них, широко использовавшийся в 80-е годы и потому получивший название классического (традиционного), связан с автоматизацией документооборота (совокупность документов, движущихся в процессе работы предприятия). Исходными и выходными координатами являлись документы, как это видно из примера¹.

Использовался следующий тезис. Данные менее подвижны, чем алгоритмы, поэтому следует создать универсальную БД, которую затем можно использовать для любого алгоритма. Однако вскоре выяснилось, что создание универсальной БД проблематично. Господствовавшая до недавнего времени концепция интеграции данных при резком увеличении их объема оказалась несостоятельной. Более того, стали появляться приложения (например, текстовые, графические редакторы), базирующиеся на широко используемых стандартных алгоритмах.

К 90-м годам сформировался второй, современный подход, связанный с автоматизацией управления. Он предполагает первоначальное выявление стандартных алгоритмов приложений (алгоритмов бизнеса в зарубежной терминологии), под которые определяются данные, а стало быть, и база данных. Объектно-ориентированное программирование только усилило значимость этого подхода.

В работе БД возможен одно- и многопользовательский (несколько пользователей подключаются к одному компьютеру через разные порты) режимы.

Используют восходящее и нисходящее проектирование БД. Первое применяют в распределенных БД при интеграции спроектированных локальных баз данных, которые могут быть выполнены с использованием различных моделей данных. Более характерным для централизованных БД является нисходящее проектирование.