|  |  |  |  |
| --- | --- | --- | --- |
| № | Модель данных | Преимущества | Недостатки |
| 1 | Иерархическая модель данных | *Достоинствами* этой модели являются простота понимания и использования. Такая модель удобна для работы с иерархически упорядоченной информацией. Иерархическая модель данных, как показала практика, позволяет эффективно использовать память компьютера и демонстрирует достаточно высокую скорость выполнения основных операций над данными. | *Недостатки* модели – в первую очередь, ее не универсальность: для большинства задач требуется дублирование данных, возможна потеря данных, связи «многие - ко - многим» (см. ниже) могут быть реализованы только искусственно при избыточности данных; а во вторую очередь – допустимость только навигационного принципа доступа к данным, (последовательным перемещением по БД для нахождения требуемой записи), записи извлекаются по одной, и чтобы извлечь некое множество данных, нужно повторять операции извлечения повторно. Непосредственный *доступ по ключу* (см. ниже), как правило, возможен только к объекту самого высокого уровня (*корневому*). Для обработки информации с достаточно сложными логическими связями иерархическая модель подходит плохо, поскольку становится громоздкой и сложной для понимания обычного пользователя. |
| 2 | Сетевая модель данных | Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. | Недостатком сетевой модели данных являются высокая сложность и жесткость схемы БД, построенной на её основе. Поскольку логика процедуры выборки данных зависит от физической организации этих данных, то эта модель не является полностью независимой от приложения. Другими словами, если необходимо изменить структуру данных, то нужно изменить и приложение. |
| 3 | Реляционная модель данных | - Простота. В реляционной модели всего одна информационная конструкция, которая формализует табличное представление данных, привычное для пользователей.  - Теоретическое обоснование. Наличие теоретически обоснованных методов нормализации отношений позволяет получать БД с заданными характеристиками.  - Независимость данных. Когда необходимо изменить структуру реляционной БД, это, как правило, приводит к минимальным изменениям в прикладных программах. | - Низкая скорость при выполнении операции соединения.  - Большой расход памяти для представления реляционной БД. |
| 4 | Модель «сущность — связь» | Достоинства ER– модели:  1. Язык обозначений простой и понятный, как для разработчиков, так и для заказчиков БД,  2. От ER – диаграммы возможен переход к схеме реляционной БД по четко определенным правилам. | Недостатки ER– модели:  1. недостаточная детализация  2. ненормализованность отношений  3. избыточное дублирование |
| 5 | Расширенная реляционная модель данных | Основными преимуществами расширенной реляционной модели данных являются повторное и совместное использование компонентов. Например, в приложении может понадобиться использование данных пространственного типа, представляющие собой точки, линии, и многоугольники, со связанными с ними функциями, которые вычисляют расстояние между точками, расстояние между точкой и линией, проверяют наличие точки в многоугольнике и т.д. При правильном проектировании с учетом новых возможностей подобный подход позволяет организациям воспользоваться преимуществами новых расширений эволюционным путем без утраты преимуществ, получаемых от использования компонентов и функций уже существующей базы данных.  Другое очевидное преимущество заключается в том, что расширенный реляционный подход позволяет воспользоваться обширным объемом накопленных знаний и опыта, связанных с разработкой реляционных приложений. Э'ГО очень важное досто инство, поскольку многие организации не хотели бы тратить средства на достаточно дорогой переход к системе нового типа. При правильном проектировании с учетом новых функциональных возможностей подобный подход позволяет организациям воспользоваться преимуществами новых расширений эволюционным путем без утраты преимуществ, получаемых от использования компонентов и функций уже существующей базы данных. Таким образом, переход к ОРСУБД можно организовать в интегрирующем, эволюционном стиле посредством выполнения некоторых пробных и экспериментальных проектов. Будущий стандарт SQL3 проектируется с учетом совместимости снизу вверх с текущим стандартом SQL, а потому любые ОРСУБД, совместимые с SQL3, должны удовлетворять этому условию. | Очевидным недостатком подхода с использованием ОРСУБД являются сложность и связанные с ней повышенные расходы. Простора и ясность, присущая реляционной модели, утрачивается при использовании подобных типов расширения. Некоторые считают, что расширения РСУБД предназначены для незначительного количества приложений, причем в последних не может быть достигнута оптимальная производительность при использовании имеющейся реляционной технологии. И многие другие, вплоть до терминологии. |
| 6 | Семантическая модель | Достоинства семантических сетей:  – универсальность, достигаемая за счет выбора соответствующего набора отношений. В принципе с помощью семантической сети можно описать сколь угодно сложную ситуацию, факт или предметную область;  – наглядность системы знаний, представленной графически;  – близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке;  – соответствие современным представлениям об организации долговременной памяти человека. | Недостатки семантических сетей:  – сетевая модель не дает (точнее, не содержит) ясного представления о структуре предметной области, поэтому формирование и модификация такой модели затруднительны;  – сетевые модели представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода;  – проблема поиска решения в семантической сети сводится к задаче поиска фрагмента сети, соответствующего подсети, отражающей поставленный запрос. Это, в свою очередь, обуславливает сложность поиска решения в семантических сетях;  – представление, использование и модификация знаний при описании систем реального уровня сложности оказывается трудоемкой процедурой, особенно при наличии множественных отношений между ее понятиями. |
| 7 | Объектно-ориентированная база данных (ООБД) | **Преимущества:**  1.                  Объекты в СУООБД могут хранить произвольное количество простых типов и других объектов. Поэтому можно организовать модель данных, как большой класс, содержащий подмножество меньших классов, содержащих в свою очередь другие подмножества классов и так далее. Использование реляционной модели приведет к созданию многочисленных таблиц, при работе с которыми придется постоянно организовывать объединения таблиц. Объект является наилучшей моделью отображения реального мира, нежели реляционные картежи. Особенно это касается сложных и многогранных объектов. СУООБД больше подходит для обработки комплексных, сложно взаимосвязанных данных и в зависимости от сложности данных может превосходить СУРБД по производительности в десятки, а то и в тысячи раз.  2.                  Данные в реальном мире обычно имеют иерархические характеристики. Известный пример с Сотрудниками, используемый в большинстве СУРБД, гораздо проще описать в СУООБД. Чтобы определить для сотрудника, является ли он менеджером или нет, в СУРБД обычно вводят дополнительное поле в таблице Сотрудников, ссылающееся на идентификатор сотрудника-менеджера или создают отдельную таблицу для определения взаимоотношения между Сотрудниками. В СУООБД класс Сотрудник просто является родительским классом для класса Менеджера.  3.                  Для доступа к данным из СУООБД не обязателен отдельный язык запросов, поскольку доступ происходит непосредственно к объектам. Тем не менее, возможность использовать запросы существует.  4.                  В типичном приложении, построенном на использовании объектно-ориентированного языка и СУРБД, значительное количество времени обычно тратится на взаимосвязывание таблиц и объектов. Также существуют различные проблемы, связанные с неполной совместимостью типов данных. При использовании СУООБД данная проблема полностью отпадает. | **Недостатки:**  1.                  В СУРБД изменение схемы данных в результате создания, изменения или удаления таблиц обычно не зависит от приложения. В приложениях, работающих с СУООБД, изменение схемы класса обычно означает, что изменения должны быть сделаны и в других классах приложения, которые взаимодействуют с экземплярами данного класса. Это ведет к необходимости перекомпиляции всей системы.  2.                  СУООБД обычно привязана к отдельному языку с помощью отдельного АПИ и данные доступны только через этот АПИ. СУРБД в этом плане имеет большие возможности, благодаря общему языку запросов.  3.                  В СУРБД, реляционная природа данных позволяет конструировать ad-hoc запросы, где можно объединять различные таблицы. В СУООБД невозможно дублировать семантику соединения двух таблиц соединением двух классов, поэтому в данном случае СУООБД уступает СУРБД в гибкости. Запросы, которые могут исполняться над данными в СУООБД, в большей мере зависят от дизайна системы. |
| 8 | Объектно-реляционная модель данных | **Достоинства:**   * Расширенные возможности SQL, в особенности, средства серверного программирования, обеспечивающие возможности определения UDT, хранимых процедур и функций, триггеров и т.д. позволяют переносить на сервер баз данных все большую часть логики приложений. * При проектировании приложения базы данных имеется три альтернативы: можно реализовать логику приложения на стороне клиента, на сервере приложений и на сервере баз данных. Очевидно, что каждая альтернатива имеет право на жизнь, и каждая из них может оказаться выигрышной в конкретной ситуации. | **Недостатки:**+   * Обширные возмодности можно так же отнести и к недостаткам так как некоторые возможности в значительной степени противоречит учению Эвгара Кодда, в котором обосновывалась целесообразность независимости базы данных от приложений. Независимость базы данных от приложений часто выглядит очень привлекательной идеей, но для ее применения разумно отказаться от многих расширений SQL. * Еще одной проблемой, не связанной с возможностями серверной реализации логики приложений, является использование в базах данных типов коллекций. Поддержка в стандарте SQL типов мультимножеств, элементами которых могут быть значения анонимных строчных типов, обеспечивает теперь возможность определения вложенных таблиц с произвольным (теоретически, неограниченным) уровнем вложенности. Поскольку все значения, хранимые в базе данных, продолжают оставаться строго типизированными, такая возможность не противоречит базовому требованию первой нормальной формы, унаследованному из реляционной модели данных, но, по существу, обеспечивает подход к прямому моделированию иерархических структур. |