

Самостоятельная работа №2

1. Инвариантная часть.

Задание: для каждой модели и подходов к организации данных предложить соответствующую предметную область и описать взаимоотношения ее объектов.

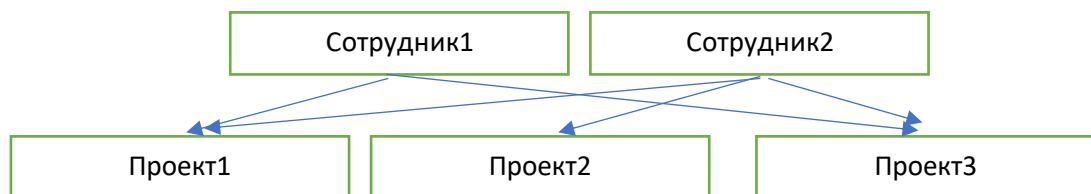
Иерархическая

Например, если иерархическая база данных содержала информацию о покупателях и их заказах, то будет существовать объект «покупатель» (родитель) и объект «заказ» (дочерний). Объект «покупатель» будет иметь указатели от каждого заказчика к физическому расположению заказов покупателя в объект «заказ».

Иерархической базой данных является файловая система, состоящая из корневого каталога, в котором имеется иерархия подкаталогов и файлов.

Сетевая

Например, есть «Сотрудник1», «Сотрудник2», «Проект1», «Проект2», «Проект3», где Сотрудник1 и Сотрудник2 работают над проектами 1, 3, а сотрудник2 еще и над проектом 2. Тогда связь показывается следующим образом:



Реляционная

В университете есть много институтов/факультетов, на каждом из которых есть множество различных специальностей, на каждой специальности есть различные курсы, на каждом курсе есть множество групп, в каждой группе есть множество студентов. Тогда последовательно можно говорить о связи данных отношений как связь один-ко-многим.

Сущность-связь

ER-модель (от англ. entity-relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает

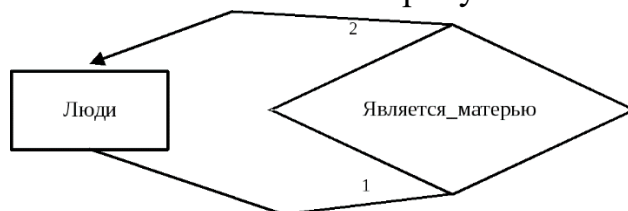
никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. entity-relationship diagram, ERD, ER-диаграмма).

Расширенная реляционная

Аналогична реляционной, за исключением того, что производится её расширение различными функциями (например, расширяемая пользователем система типов, инкапсуляция, наследование, полиморфизм, динамическое связывание методов, использование составных объектов, а также поддержка идентичных объектов). Но не существует какой-то общепринятой расширенной реляционной модели, а скорее имеется несколько таких моделей, характеристики которых зависят от способа и степени реализации внесенных расширений. Однако во всех моделях используются одинаковые базовые реляционные таблицы и язык запросов, включено понятие объекта, а в некоторых дополнительно реализована возможность сохранения методов (или процедур, или триггеров) таким же способом, что и базе данных.

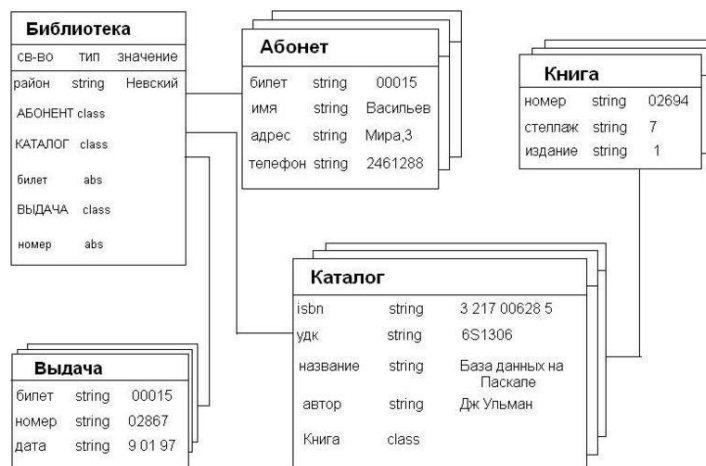
Семантическая

Допустим, имеется набор объектов ЛЮДИ и связь ЯВЛЯЕТСЯ-МАТЕРЬЮ, список наборов объектов которой есть ЛЮДИ, ЛЮДИ. Мы предполагаем, что набор связей ЯВЛЯЕТСЯ-МАТЕРЬЮ включает все пары (p_i, p_j) , такие, что человек p_i является матерью человека p_j . Диаграмма для набора объектов ЛЮДИ со связью ЯВЛЯЕТСЯ-МАТЕРЬЮ показана на рисунке:



Объектно-ориентированная

Пример логической структуры объектно-ориентированной БД библиотечного дела приведен на рисунке.



Здесь объект типа БИБЛИОТЕКА является родительским для объектов-экземпляров классов АБОНЕНТ, КАТАЛОГ и ВЫДАЧА. Различные объекты типа КНИГА могут иметь одного или разных родителей. Объекты типа КНИГА, имеющие одного и того же родителя, должны различаться по крайней мере инвентарным номером (уникален для каждого экземпляра книги), но имеют одинаковые значения свойств ISBN, УДК, название и автор. Логическая структура объектно-ориентированной БД внешне похожа на структуру иерархической БД. Основное отличие между ними состоит в методах манипулирования данными.

Объектно-реляционная

Объектно-реляционная модель данных является реляционной моделью с некоторыми свойствами объектной модели данных, или наоборот. Четкого определения не существует.

Полуструктурированная

XML и другие языки разметки, email и сообщения в форматах EDI — всё это примеры слабоструктурированных данных.

2. Вариативная часть.

Задание: заполнить таблицу "Преимущества и недостатки моделей данных".

№	Модель данных	Преимущества	Недостатки
1.	Иерархическая	Эффективное использование памяти компьютера при хранении данных	Громоздкость модели для обработки данных со сложными логическими данными
2.	Сетевая	Высокая эффективность затрат памяти; Оперативность обработки данных	Сложность и жесткость схемы базы; Сложность понимания; Ослаблен контроль целостности (в ней допускается устанавливать произвольные связи между записями)
3.	Реляционная	Простота и доступность для понимания пользователем; Строгие правила проектирования, базирующиеся на математическом аппарате; Полная независимость данных. Изменения в прикладной программе при изменении реляционной БД минимальны; Для организации запросов и написания прикладного ПО нет необходимости знать конкретную организацию БД во внешней памяти	Далеко не всегда предметная область может быть представлена в виде "таблиц"; в результате логического проектирования появляется множество "таблиц". Это приводит к трудности понимания структуры данных; БД занимает относительно много внешней памяти; Относительно низкая скорость доступа к данным
4.	Сущность-связь	Понятна, проста; Строгие правила проектирования	Не подходит для любой предметной области
5.	Расширенная реляционная	Повторное и совместное использование компонентов; Использование обширного объема накопленных знаний и опыта, связанных с разработкой реляционных приложений	Сложность разработки; Повышенные расходы
6.	Семантическая	Универсальность, достигаемая за счет выбора соответствующего набора отношений;	Не содержит ясного представления о структуре предметной области, поэтому формирование и модификация

		<p>Наглядность системы знаний, представленной графически;</p> <p>Близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке;</p> <p>Соответствие современным представлениям об организации долговременной памяти человека</p>	<p>такой модели затруднительны;</p> <p>Представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода;</p> <p>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;</p> <p>Представление, использование и модификация знаний при описании систем реального уровня сложности оказывается трудоемкой процедурой, особенно при наличии множественных отношений между ее понятиями</p>
7.	Объектно-ориентированная	<p>Возможность отображения информации о сложных взаимосвязях объектов;</p> <p>Позволяет идентифицировать отдельную запись базы данных и определять функции их обработки.</p>	<p>Высокая понятийная сложность;</p> <p>Неудобство обработки данных;</p> <p>Низкая скорость выполнения запросов</p>
8.	Объектно-реляционная	<p>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</p>	<p>Обширные возможности (так как некоторые возможности в значительной степени противоречат учению Кодда)</p> <p>Независимость базы данных от приложений часто выглядит очень привлекательной идеей, но для ее применения разумно отказаться от многих расширений SQL.</p>

		<p>большую часть логики приложений.</p>	<p>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.</p>
9.	Полуструктурированная	<p>Допускают создание более гибкой структуры по сравнению с реляционными; Требуют, чтобы все данные укладывались в заранее заданную схему</p>	<p>Структура данных обычно известна лишь частично; Структура является глубоко вложенной или даже циклической, что требует от системы управления базами данных развитых рекурсивных возможностей. Реляционная алгебра этому требованию не соответствует</p>