Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Тверской государственный университет»

Факультет иностранных языков и международной коммуникации

Направление «Лингвистика»

Реферат

на тему: «Сетевая модель организации базы данных»

Выполнила:

Студентка 1 курса, группа 12

Бабурина Надежда Константиновна

Проверила:

к.ф.н., доцент, доцент кафедры английского языка

Е.М. Масленникова

Тверь

2020

Содержание

[Содержание 2](#_Toc37318132)

[Введение 3](#_Toc37318133)

[Сетевая модель базы данных 4](#_Toc37318134)

[Признаки 6](#_Toc37318135)

[Операции над данными в сетевой модели БД 7](#_Toc37318136)

[Объектно-ориентированные СУБД 8](#_Toc37318137)

[Объектно-реляционные СУБД 10](#_Toc37318138)

[Заключение 11](#_Toc37318139)

[Список литературы 12](#_Toc37318140)

# Введение

Прежде чем перейти к описанию процессов, которые происходят внутри сетевой модели данных, давайте ознакомимся со структурой сетевой базы данных, чтобы иметь представление о том, с чем предстоит иметь нам дело. Прежде всего, следует разобраться со словом сети, которое присутствует в название: «сетевая модель». Сети – это естественный способ представления отношений между объектами базы данных и связей между этими объектами. Под словом объекты следует понимать таблицы баз данных или сущности. В общем, как вам удобно, так и называйте, вас везде поймут правильно.

Сетевые базы данных опираются на математику графов, конкретнее, **сетевую модель данных** можно представить в виде ориентированного графа. Направленный граф состоит из узлов и ребер. Узлы направленного графа – это ни что иное, как объекты сетевой базы данных, а ребра такого графа показывают связи между объектами сетевой модели данных, причем ребра показывают не только саму связь, но и тип связи (связь один к одному или связь один ко многим).

# Сетевая модель базы данных

На разработку этого стандарта большое влияние оказал американский ученый Ч.Бахман. Основные принципы сетевой модели данных были разработаны в середине 60-х годов, эталонный вариант сетевой модели данных описан в отчетах рабочей группы по языкам баз данных (COnference on DAta SYstem Languages) CODASYL (1971 г.).

Сетевая модель данных определяется в тех же терминах, что и *иерархическая*. Она состоит из множества записей, которые могут быть владельцами или членами групповых отношений. Связь между записью-владельцем и записью-членом также имеет вид 1:N.

**Основное различие этих моделей** состоит в том, что в сетевой модели запись может быть членом более чем одного группового отношения. Согласно этой модели каждое групповое отношение именуется и проводится различие между его типом и экземпляром. Тип группового отношения задается его именем и определяет свойства общие для всех экземпляров данного типа. Экземпляр группового отношения представляется записью-владельцем и множеством (возможно пустым) подчиненных записей. При этом имеется следующее ограничение: экземпляр записи не может быть членом двух экземпляров групповых отношений одного типа (т.е. сотрудник из примера в п..1, например, не может работать в двух отделах).

Иерархическая структура рис. 1 преобразовывается в сетевую модель, следующим образом (см. рис. 2):

* деревья (a) и (b), показанные на рис. 1, заменяются одной сетевой структурой, в которой запись СОТРУДНИК входит в два групповых отношения;
* для отображения типа M:N вводится запись СОТРУДНИК\_КОНТРАКТ, которая не имеет полей и служит только для связи записей КОНТРАКТ и СОТРУДНИК, (см. рис. 2). Отметим, что в этой записи может храниться и полезная информация, например, доля данного сотрудника в общем вознаграждении по данному контракту.

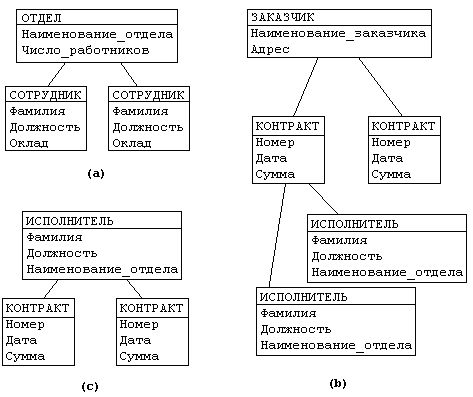


Рис. 1. Пример иерархической базы данных

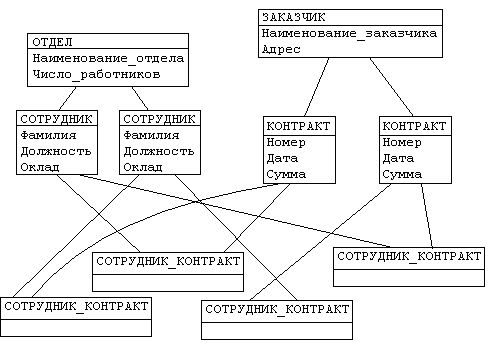


Рис. 2. Сетевая модель базы данных

# Признаки

Каждый экземпляр группового отношения характеризуется следующими признаками:

**Способ упорядочения подчиненных записей:**

* произвольный,
* хронологический /очередь/,
* обратный хронологический /стек/,
* сортированный.

Если запись объявлена подчиненной в нескольких групповых отношениях, то в каждом из них может быть назначен свой способ упорядочивания.

**Режим включения подчиненных записей:**

* автоматический – невозможно занести в БД запись без того, чтобы она была сразу же закреплена за неким владельцем;
* ручной – позволяет запомнить в БД подчиненную запись и не включать ее немедленно в экземпляр группового отношения. Эта операция позже инициируется пользователем.

**Режим исключения.**

Принято выделять три класса членства подчиненных записей в групповых отношениях:

* **Фиксированное.** Подчиненная запись жестко связана с записью владельцем и ее можно исключить из группового отношения только удалив. При удалении записи-владельца все подчиненные записи автоматически тоже удаляются. В рассмотренном выше примере фиксированное членство предполагает групповое отношение «ЗАКЛЮЧАЕТ» между записями «КОНТРАКТ» и «ЗАКАЗЧИК», поскольку контракт не может существовать без заказчика.
* **Обязательное.** Допускается переключение подчиненной записи на другого владельца, но невозможно ее существование без владельца. Для удаления записи-владельца необходимо, чтобы она не имела подчиненных записей с обязательным членством. Таким отношением связаны записи «СОТРУДНИК» и «ОТДЕЛ». Если отдел расформировывается, все его сотрудники должны быть либо переведены в другие отделы, либо уволены.
* **Необязательное.** Можно исключить запись из группового отношения, но сохранить ее в базе данных не прикрепляя к другому владельцу. При удалении записи-владельца ее подчиненные записи – необязательные члены сохраняются в базе, не участвуя более в групповом отношении такого типа. Примером такого группового отношения может служить «ВЫПОЛНЯЕТ» между «СОТРУДНИКИ» и «КОНТРАКТ», поскольку в организации могут существовать работники, чья деятельность не связана с выполнением каких-либо договорных обязательств перед заказчиками.

# Операции над данными в сетевой модели БД

Таблица 1. Операции над данными в сетевой модели БД

|  |  |
| --- | --- |
| Добавить | - внести запись в БД и, в зависимости от режима включения, либо включить ее в групповое отношение, где она объявлена подчиненной, либо не включать ни в какое групповое отношение. |
| Включить в групповое отношение | - связать существующую подчиненную запись с записью-владельцем. |
| Переключить | - связать существующую подчиненную запись с другой записью-владельцем в том же групповом отношении. |
| Обновить | - изменить значение элементов предварительно извлеченной записи. |
| Извлечь | - извлечь записи последовательно по значению ключа, а также используя групповые отношения - от владельца можно перейти к записям - членам, а от подчиненной записи к владельцу набора. |
| Удалить | - убрать из БД запись. Если эта запись является владельцем группового отношения, то анализируется класс членства подчиненных записей. Обязательные члены должны быть предварительно исключены из группового отношения, фиксированные удалены вместе с владельцем, необязательные останутся в БД. |
| Исключить из группового отношения | - разорвать связь между записью-владельцем и записью-членом. |

**Достоинства и недостатки ранних СУБД.**

Достоинства ранних СУБД**:**

* развитые средства управления данными во внешней памяти на низком уровне;
* возможность построения вручную эффективных прикладных систем;
* возможность экономии памяти за счет разделения подобъектов (в сетевых системах)

Недостатки ранних СУБД**:**

* сложность использования;
* высокий уровень требований к знаниям о физической организации БД;
* зависимость прикладных систем от физической организации БД;
* перегруженность логики прикладных систем деталями организации доступа к БД.

Как иерархическая, так и сетевая модель данных предполагает наличие высококвалифицированных программистов. И даже в таких случаях реализация пользовательских запросов часто затягивается на длительный срок.

# Объектно-ориентированные СУБД

Появление объектно-ориентированных СУБД вызвано потребностями программистов на ОО-языках, которым были необходимы средства для хранения объектов, не помещавшихся в оперативной памяти компьютера. Также важна была задача сохранения состояния объектов между повторными запусками прикладной программы. Поэтому, большинство ООСУБД представляют собой библиотеку, процедуры управления данными которой включаются в прикладную программу. Примеры реализации ООСУБД как выделеного сервера базы данных крайне редки.

Сразу же необходимо заметить, что общепринятого определения "объектно-ориентированной модели данных" не существует. Сейчас можно говорить лишь о неком "объектном" подходе к логическому представлению данных и о различных объектно-ориентированных способах его реализации.

Мы знаем, что любая модель данных должна включать три аспекта: структурный, целостный и манипуляционный. Посмотрим, как они реализуются на основе объектно-ориентированная парадигмы программирования.

#### Структура

Таблица 2. Структура объектной модели описывается с помощью трех ключевых понятий:

|  |  |
| --- | --- |
| инкапсуляция | - каждый объект обладает некоторым внутренним состоянием (хранит внутри себя запись данных), а также набором методов - процедур, с помощью которых (и только таким образом) можно получить доступ к данным, определяющим внутреннее состояние объекта, или изменить их. Таким образом, объекты можно рассматривать как самостоятельные сущности, отделенные от внешнего мира; |
| наследование | - подразумевает возможность создавать из классов объектов новые классы объекты, которые наследуют структуру и методы своих предков, добавляя к ним черты, отражающие их собственную индивидуальность. Наследование может быть простым (один предок) и множественным (несколько предков); |
| полиморфизм | - различные объекты могут по разному реагировать на одинаковые внешние события в зависимости от того, как реализованы их методы. |

#### Целостность данных

Для поддержания целостности объектно-ориентированный подход предлагает использовать следующие средства:

* автоматическое поддержание отношений наследования возможность объявить некоторые поля данных и методы объекта как "скрытые", не видимые для других объектов; такие поля и методы используются только методами самого объекта создание процедур контроля целостности внутри объекта

#### Средства манипулирования данными

К сожалению, в объектно-ориентированном программировании отсутствуют общие средства манипулирования данными, такие как реляционная алгебра или реляционное счисление. Работа с данными ведется с помощью одного из объектно-ориентированных языков программирования общего назначения, обычно это SmallTalk, C++ или Java.

# Объектно-реляционные СУБД

Разница между объектно-реляционными и объектными СУБД: первые являют собой надстройку над реляционной схемой, вторые же изначально объектно-ориентированы. Главная особенность и отличие объектно-реляционных, как и объектных, СУБД от реляционных заключается в том, что О(Р)СУБД интегрированы с Объектно-Ориентированным (OO) языком программирования, внутренним или внешним как C++, Java. Характерные свойства OРСУБД - 1) комплексные данные, 2) наследование типа, и 3) объектное поведение.

Комплексные данные могут быть реализованы через постоянно-хранимые объекты (persistent objects). Создание комплексных данных в большинстве существующих ОРСУБД основано на предварительном определении схемы через определяемый пользователем тип (UDT - user-defined type). Используются также встроенные конструкторы составных типов, например массив (ARRAY).

Иерархия структурных комплексных данных предлагает дополнительное свойство, наследование типа. То есть структурный тип может иметь подтипы, которые используют все его атрибуты и содержат дополнительные атрибуты, специфицированные в подтипе.

Объектное поведение закладывается через описание программных объектов. Такие объекты должны быть сохраняемыми и переносимыми для обработки в базе данных, поэтому они называются обычно как постоянные (или долговременные) объекты. Внутри базы данных все отношения с постоянным программным объектом есть отношения с его объектным идентификатором (OID).

Объектно-реляционными СУБД являются, к примеру, широко известные Oracle Database, Microsoft SQL Server 2005, PostgreSQL, а также Sav Zigzag, IBM Cloudscape.

# Заключение

Рассмотрены модели организации БД. Различают три основные модели базы данных - это иерархическая, сетевая и реляционная. Эти модели отличаются между собой по способу установления связей между данными.

Достоинства и недостатки ранних СУБД.

Рассмотрены более поздние модели СУБД такие как объектно-ориентированные и объектно-реляционные.

# Список литературы

НОУ ИНТУИТ. URL: <https://www.intuit.ru/studies/courses/3439/681/lecture/14023?page=1> (дата обращения 08.04.2020)

ZametkiNaPolyah.ru. URL: <https://zametkinapolyah.ru/zametki-o-mysql/setevaya-baza-dannyx-setevaya-model-dannyx.html> (дата обращения 08.04.2020)