|  |  |
| --- | --- |
| **Описание: 15** | Негосударственное образовательное учреждение  высшего образования  Московский технологический институт |

|  |  |
| --- | --- |
| Факультет: Техники и современных технологий | Кафедра: Информатики и автоматизации |

**КУРСОВАЯ РАБОТА**

по дисциплине **«**Базы данных»

**на тему:**

**«МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ БД: ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ»**

Уровень образования: бакалавриат

Направление: 230100 «Информатика и автоматизация»

Профиль: «Вычислительные машины, комплексы, системы и сети»

Выполнил:

Форма обучения: очная

Ваулинский Дмитрий Валерьевич

Москва 2016

**ОГЛАВЛЕНИЕ**

Введение…………………………………………………………………………...3

1. Инфологическое моделирование предметной области………………….…..5

1.1 Цель инфологического моделирования…………………………………..5

1.2 Источники данных, базы данных и система управления ими………..…5

1.3 Принципы проектирования базы данных……………………………...…7

1.4 Пример разработки модели БД с помощью Erwin…………………...…..9

2. Логический этап проектирования базы данных………………………….…14

2.1 Сущности, типы сущностей …………………………………………..…14

2.2 Атрибуты, типы атрибутов………………………………………….……15

2.3 Связи, типы связей…………………………………………………….….19

Заключение……………………………………………………………………….23

Глоссарий……………………………………………………………………...…26

Список использованной литературы………………………………………...…27

Приложения………………………………………………………………….…..28

**ВВЕДЕНИЕ**

Базы данных составляют в настоящее время основу компьютерного обеспечения информационных процессов, входящих во все сферы человеческой деятельн6ости.

Цель моделирования данных состоит в обеспечении разработчика информационной системы концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Логическая модель базы данных – версия концептуальной модели, которая может быть реализована конкретной СУБД.

Логическая модель отображает логические связи между атрибутами объектов вне зависимости от их содержания и среды хранения. Как правило, при разработке логических структур используются в основном эвристических методы проектирования с формальной оценкой качества принимаемых решений. Выделяют три группы критериев качества проекта логической структуры.

Критерии первой группы отображают требования поддержания и сохранения в логической структуре базы данных структурных и семантических свойств и особенностей данных и связей, зафиксированных в канонической структуре. Критерии первой группы связаны с ограничениями целостности защиты базы данных от непротиворечивости при актуализации или избыточности представления данных.

Критерии второй группы связаны с вопросами производительности базы данных на временные параметры доступа и экономические показатели функционирования Автоматизированных информационно - управляющих систем (АИУС).

Поскольку разрабатываемая структура базы данных предназначается для эффективного обслуживания множества информационно - поисковых процессов, комплексная оценка результатов проектирования логической структуры базы данных должна производится с точки зрения эффективности выполнения исходных запросов и транзакции к БД отражающих информационной потребности конечных пользователей, приложений и действий на БД (операций, добавления, удаления, модификация, репликации, копирования данных). В свою очередь отдельная транзакция – это тот же запрос к БД, только содержащий помимо операции поиска и выборка данных, также операции манипулировании данными, спецификации которых задаются в описании транзакции. Поэтому для комплексной оценки проекта логической структуры базы данных информационно - поисковый процесс представляется совокупностью запросов к БД задаваемых канонической структуре БД.

В процессе разработки логическая модель данных постоянно тестируется и проверяется на соответствие требованиям пользователей, и необходимо выделить: сущность, связь и атрибуты.

Данная курсовая работа посвящена анализу логического проектирования баз данных. В качестве инструмента построения базы данных использована СУБД (ERwin) . Эту СУБД отличает простота использования в сочетании с широкими возможностями по разработке законченных приложений. В Visual FoxPro так же возможно создать базу данных, типичную той, которая будет рассмотрена в данной курсовой.

**1. ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

**1.1. Цель инфологического моделирования**

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты) [10, с. 70].

**1.2. Источники данных, базы данных и система управления ими**

Одним из условий успешного функционирования любого предприятия является наличие необходимой информации, которую необходимо обрабатывать. Именно эти требования привели к возникновению системы базы данных.

Каждой цивилизации приходится иметь дело с обработкой информации. С развитием экономики и ростом численности населения возрастает и объем взаимосвязанных данных, необходимых для решения коммерческих и административных задач. Взаимосвязанные данные называют систему. Всякая система, призванная облегчить труд человека, кроме обычных форм знаний, требует создания очень сложной модели реального мира.

Ядром информационной системы являются хранимые данные. На любом предприятии данные различных отделов, как правила, пересекаются. Например, для целей управления часто нужна информация по всему предприятию. Принятие решений по производственным вопросам невозможно без информации о товарах, о полученных заказах, о стратегии сбыта и т.д. Это означает, что описывающие конкурентную предметную область данные должны храниться в легко доступном виде.

Сегодня мы можем встретить систему обработки данных старого типа, в которой служащий в ручную помещает данные в скоросшиватель, и рядом с ней – современную систему с применением самой быстродействующей ЭВМ, сложнейшего оборудования и программного обеспечения. Но обе они обязаны предоставлять достоверную информацию в определенное время, определенному лицу, в определенном месте и за определенную плату. Такая система может потребоваться банку, заводу, предприятию сферы обслуживания, университету, больнице, универмагу и многим другим потребителям.

Базы данных – совокупность связанных данных, правила, организации которых основаны на общих принципах описания, хранения и манипулирования данными.

Для интеграции файлов базу данных и обеспечения различными пользователями различных представлений отданных, необходима система. Программное обеспечение, аппаратные средства, программируемая логика и процедуры, осуществляющие управление базой данных, образуют систему управления базы данных. СУБД создает возможность доступа к интегрированным данным, которые пересекают операционные, функциональные и организационные границы в предметной области.

Система управления БД – это специальный пакет программ, посредством которого реализуется централизованное управление базой данных, и обеспечивают доступ к данным [9, с. 60].

**1.3. Принципы проектирования базы данных**

Проектирование баз данных — процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи:

- Обеспечение хранения в БД всей необходимой информации.

- Обеспечение возможности получения данных по всем необходимым запросам.

- Сокращение избыточности и дублирования данных.

- Обеспечение целостности данных (правильности их содержания): исключение противоречий в содержании данных, исключение их потери и т.д.

В основу проектирования положено представления конечных пользователей конкретной организации – концептуальные требования. Конечный пользователь принимает решение с учетом получаемой в результате доступа к базе данных информации. Данные, помещаемые в базу данных, также предоставляет конечный пользователь.

Концептуальная (инфологическая) разработка - создание семантической модели предметной области, которая является информационной моделью большинство высокого уровня отделения. Такая модель формируется без ориентации к любому определенному DBMS и модели данных. Условия «семантическая модель», «концептуальная модель» и «инфологическая модель» является синонимами. Кроме того, в этой модели базы данных «слов окружения» и "модели предметной области" (например, «концептуальная модель базы данных» и «концептуальной модели предметной области») модель как таковая - и изображение действительности, и изображение спроектированной базы данных для этой действительности может эквивалентный использоваться.

Определенный тип и контент концептуальной модели базы данных определяются формальным устройством, выбранным с этой целью. Выразительные нотации, подобные ER-схемам, обычно используются.

Чаще концептуальная модель базы данных включает:

- Описание информационных объектов, или понятия области данных и связи промежуточный.

- Описание ограничений целостности, то есть необходимые условия к допустимым значениям данных и к связи промежуточный.

Логическая (даталогическая) разработка - создание схемы базы данных на основе определенной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель - коммутируемая из схем соотношений, это - нормаль с инструкциями первичных ключей, и также "связь" между соотношениями, представляя внешние ключи.

Преобразование концептуальной модели к логической модели, как правило, выносится формальными правилами. Этот этап может быть в основном автоматизирован.

Материальная разработка - создание схемы базы данных для определенного DBMS. Специфичность определенного DBMS может включать ограничения на именование объектах базы данных, сужения на поддерживаемые типы данных, и т.д. Кроме того, специфичность определенного DBMS при материальной разработке включает выбор решений, соединенных с физической средой хранения данных (выбор памяти на диске методов управления, совместное использование DB на файлах и устройствах, методах доступа для данных), создание индексов и т.д.

В специфичности этапа логического проектирования определенной модели данных рассматривается, но специфичность определенного DBMS нельзя рассмотреть.

Семантическая модель (концептуальная модель, инфологическая модель) - модель предметной области, имеемая в виду для представления семантики области данных на высшем уровне к отделению. Это означает, что необходимость, чтобы использовать понятие "низкого уровня", соединенного со специфичностью материального представления и хранения данных, отщепляется или минимизируется [8, с. 67].

Модель "передача аромата" (английская "Модель типа объект-отношение"), или ER-модель, предлагаемая П. Ченом в 1976, является наиболее известным представителем семантического класса (концептуальный, инфологических) модели предметной области. ER-модель обычно представлена в выразительной форме, с использованием исходной нотации П. Чена, названного ER-схемой, или с использованием других выразительных нотаций (Воронья лапа, Информационная инженерия, и т.д.).

При рассмотрении требований конечных пользователь необходимо принимать во внимание следующее:

- БД должна удовлетворять актуальным информационным потребностям.

- БД должна удовлетворять актуальным требованиям за приемлемое время, т.е. заданным требованиям производительности.

- БД должна удовлетворять выявленным и вновь возникающим требованиям конечных пользователей.

- БД должна легко расширять при реорганизации и расширении предметной области.

- БД должна легко изменяться при изменении программной и аппаратной среды.

- Загруженные в БД корректные данные должны оставаться корректными.

- Данные до включения в БД должны проверяться на достоверность.

- Доступ к данным, размещаемым в БД, должны иметь только лица с соответствующими полномочиями.

**1.4. Пример разработки модели БД с помощью ERwi**

Гипотетическая точка обмена валюты - система локальной информации (И.С), вызванный, чтобы автоматизировать процесс пристрелки транзакций покупки и продажи валюты [7, с. 17]. Создаваемая система должна обеспечить ввод, хранение и информационный поиск о транзакциях, сделанных в данной точке обмена. Каждая транзакция единственный цифровой код приспосабливается. Информация о транзакциях должна содержать сходимость на финике и время транзакции, итоги искупленных и проданных валют, фамилий, имени, патронима и числа паспорта клиента, и также о фамилии, подписывании и регистрационном номере частного дела кассира в отделе кадров. У системы должна быть возможность вычислить, денежное переворачивают в течение одного или нескольких дней, и также выносить информационный поиск о транзакциях под числом паспорта клиента. Составы задачи в разработке строения базы данных разработали автоматизированный И.С.

Пустите нам исследование области данных специально, чтобы выбрать основные объекты. Поскольку это - вопрос пристрелки транзакций покупки и продажи валюты, ясно что в модели должен быть аромат ТРАНЗАКЦИЯ. Согласно правилам, на ER-схеме заголовок объектов регистрирует прописные буквы. Мы напоминаем, что заголовок аромата - заголовок типа, вместо набора объектов. Поэтому это выражает существительное в сингулярном (ср. СДЕЛКИ), чаще.

Понятие транзакция подразумевает участие, по крайней мере, две стороны, делающие это, и как наличие сюжета транзакции. Т.к. Участник транзакции, которая клиент и кассир, и сюжет транзакции, является валютой, необходимо ввести еще три аромата: ВАЛЮТА, КЛИЕНТ и КАССИР.

Внесем перечисленные сущности в диаграмму. Название сущностей можно редактировать непосредственно на диаграмме, как это показано (см. Приложение Б).

Для внесения дополнительных сведений необходимо воспользоваться редактором сущностей. Перейти в него можно при помощи всплывающего меню (см. Приложение В), возникающего при нажатии правой клавиши мыши над любой из сущностей.

Следующим шагом в процессе создания логической модели должно стать определение связей между сущностями. Напомним, что для задания связей между двумя сущностями необходимо указать тип связи, направление связи (родительская и дочерняя сущности), мощность связи, допустимость пустых (null) значений, требования по обеспечению ссылочной целостности, а в некоторых случаях и роль.

Для задания связей между указанными сущностями сначала составим описание данной предметной области при помощи ряд истинных высказываний на естественном языке.

Любой КЛИЕНТ должен совершить одну или несколько сделок.

Каждую СДЕЛКУ должен совершить только один клиент.

Любой КАССИР может обслуживать одну или несколько СДЕЛОК, но может и не обслуживать и не одной (например, только принять на работу).

Каждую СДЕЛКУ должен обслужить только один КАССИР.

Любая ВАЛЮТА покупаться и продаваться при разных СДЕЛКАХ.

При совершении СДЕЛКИ должна покупаться одна ВАЛЮТА и продаваться другая ВАЛЮТА.

Анализ приведенных высказываний позволяет выделить четыре связи (название связи – глаголы):

КЛИЕНТ совершает СДЕЛКУ.

КАССИР обслуживает СДЕЛКУ.

ВАЛЮТА покупается при СДЕЛКЕ

ВАЛЮТА продается при СДЕЛКЕ.

Все четыре связи являются связями «один – ко - многим». Во всех четырех случаях сущность СДЕЛКА является дочерней.

Все связи неидентифицирующие, т.к. любой экземпляр сущности СДЕЛКА может быть однозначно идентифицирован по коду сделки, то есть вне зависимости от экземпляров других сущностей.

Все связи, кроме первой, могут иметь мощность 0, 1 или более. Первая сделка не может иметь мощность 0, т.к. в данном случае любой человек становится КЛИЕНТОМ только тогда, когда он совершает хотя бы одну сделку.

Во всех четырех связях родительские сущности не могут принимать пустые значения, поскольку при отсутствии экземпляра хотя бы одной из родительских сущностей экземпляр сущности СДЕЛКА перестает описывать сделку по обмену валюты.

Эти параметры задаются при помощи редактора связи. Вызвать этот редактор можно двойным нажатием левой клавиши мыши над связью.

Задание ограничений ссылочной целостности, а так же указание ролей производится на закладке Rolename/RI Action панели диалога редактора связи.

Ограничение ссылочной целостности, задаваемое по умолчанию ERwin, в данном случаи можно оставить без изменений.

Указание ролей понадобится лишь для связей между сущностями ВАЛЮТА и СДЕЛКА.

После задания связей между сущностями диаграмма будит выглядеть следующим образом (см. Приложение Г).

Теперь для каждой сущности необходимо указать первичные ключи и не ключевые атрибуты. Кроме того, для некоторых, возможно, понадобится задание альтернативных ключей и инверсных кодов.

Полезным источником информации в этом случаи может стать перечень требований хранимой информации, приведенный в задании. Рассмотрим по очереди каждую из сущностей. [[1]](#footnote-1)

Сведения о клиенте должны состоять из его фамилии, имени, отчества и номера его паспорта. Очевидно, что они и будут атрибутами сущности КЛИЕНТ. Первичным ключом можно было бы выбрать номер паспорта, поскольку он однозначно идентифицирует любой из экземпляров этой сущности. Однако номер паспорта не является числом, т.к; кроме цифр, содержит и буквы, и, следовательно, для его хранений будет использоваться строка минимум из 13 символов, что не совсем удобно. По этому введен для каждого КЛИЕНТА уникальный номер, который и будет первичным ключом, что бы обеспечить возможность быстрого поиска информации о сделках по его значениям, согласно заданию.

Сведения о кассире должны включать фамилию, инициалы и учетный номер кассира – они и будут атрибутами сущности КАССИР. Поскольку учетный номер личного дела кассира может содержать не только цифры, как и в предыдущем, случай, введем для каждого экземпляра уникальный номер, который и будет первичным ключом.

По тем же соображениям сущность ВАЛЮТА будет содержать два атрибута: код валюты и название валюты, первый из которых будет первичным ключом.

Что касается сущности СДЕЛКА, то часть атрибутов она унаследует от родительских сущностей и остается лишь добавить следующие: «дата сделки», «время сделки», «сумма покупаемой валюты» и «сумма продаваемой валюты». Очевидно, что первичным ключом следует выбрать уникальный цифровой код сделки. Поскольку в задании сказано, что создаваемая система должна позволять вычислить денежный оборот за один или несколько дней, полезно было бы сделать атрибут «дата сделки» инверсным входом, т.к. он довольно часто будет использоваться для доступа к данным.

Для задания первичных ключей и атрибутов используется редактор атрибутов. Перейти в него можно, воспользовавшись всплывающим меню (см. Приложение Д).

Для задания альтернативных ключей и инверсных входов следует воспользоваться редактором ключей. Переход в него осуществляется так же, как и в редактор атрибутов.

На этом процесс создания логической модели завершается, сама модель приобретает вид (см. Приложение Е).

**2. ЛОГИЧЕСКИЙ ЭТАП ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ**

**2.1. Сущности, типы сущностей**

Сущность – Это реальный или воображаемый объект, информация о котором представляет интерес. В диаграммах Er-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не конкретного объекта – экземпляра этого типа. Каждый экземпляр сущности должен быть отличим от любого экземпляра той же сущности. В зависимости от режима представления диаграммы прямоугольник может содержать имя сущности, ее описание, список ее атрибутов и другие требования [10, с. 25].

Каждая сущность должна обладать следующими свойствами:

иметь уникальный идентификатор;

содержать один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь с другими сущностями;

содержат совокупность атрибутов, однозначно идентифицирующих каждый экземпляр сущности;

Любая сущность может иметь произвольное количество связей с другими сущностями. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе.

Определите три ядра класса объектов: удочка, ассоциативная и характерная, и также подкласс ассоциативных объектов - назначения

Ассоциативный аромат (ассоциация) является передачей типа "многие - к - многие" между двумя или больше объектами или копиями объектов. Ассоциации рассматривают как полные объекты:

- Они могут участвовать в других ассоциациях и назначениях таким же образом, как объекты удочки,

- Может обладать свойствами, то есть иметь не только коммутируемый из ключевых атрибутов, необходимых для инструкций связи, но также и любого числа других атрибутов, характеризующих передачу. Характерный аромат (характеристика) является передачей типа "многие - к - один" или "один - к - один" между двумя объектами (особый случай ассоциации). Единственная цель характеристики в рамках продуманной области данных состоит в описании или спецификации некоторого другого аромата. Необходимость их воскресает, потому что у объектов реального мира иногда есть многозначные свойства. 0бозначающая аромат или назначение - передача типа "многие - к - один" или "один - к - один" между двумя объектами и отличаются от характеристики, которая не зависит от определяемого аромата.

Назначения используют для хранения повторяющихся значений больших текстовых атрибутов: наказания "кодификаторы", изученные студентами, именами устройств и их отделов, материально-технических ресурсов, и т.д.

Пустите нам переопределять теперь аромат удочки как аромат, который не является ни ассоциацией, ни назначением, характеристикой. У таких объектов есть независимое существование, хотя они и могут определять другие объекты как, например, инспектор назначает отдел кадров.

Есть два типа сущностей: зависимая и независимая. Если экземпляры сущности могут быть уникально идентифицированы без определения ее связей с другими сущностями, она называется независимой. В противном случае сущность называют зависимой. Зависимая сущность отображается в Erwin прямоугольником с закругленными углами.

**2.2. Атрибуты, типы атрибутов**

Атрибут – поименованный столбец отношения.

Первичный ключ – Это атрибут или набор атрибутов, уникально идентифицирующий экземпляр сущности. Если несколько наборов атрибутов могут уникально идентифицировать сущность, то выбор одного из них осуществляется разработчиком на основании анализа предметной области и учета следующих требований к первичному ключу:

- первичный ключ не должен принимать пустые (Null) значения;

- первичный ключ не должен изменяться в течение времени;

- размер первичного ключа должен быть как можно меньшим.

При этом если разработчик считает, что какой либо из оставшихся наборов будит часто использоваться для доступа к сущности, то он может объявить его альтернативным ключом.

В ERwin можно также составлять группы атрибутов, которые не идентифицируют уникально экземпляры сущности, но часто используются для доступа к данным. Они получили название инверсных входов. Одни и те же атрибуты сущности могут входить в несколько различных групп ключей.

Рассмотрим выше сказанное на примере сущности КЛИЕНТ (см. Приложение А)/

Среди атрибутов данной сущности на роль первичного ключа могут претендовать «номер клиента» и группа атрибутов «фамилия», «имя», «отчество», «дата рождения» (последний необходим, т.к. на предприятии могут работать полные тезки). Очевидно, что по соображению размера в качестве первичного ключа следует выбрать первый из вариантов.

На диаграмме атрибуты, составляющие первичный ключ, располагаются в верхней части прямоугольника и отделяющая от прочих ( не входящих в первичных ключ) горизонтальной линией.

Группа атрибутов «фамилия», «имя», «отчество», «дата рождения» может являться альтернативным ключом. Однако вряд ли кто - либо, пытающийся найти информацию о сотруднике, будет знать дату его рождения. А вот группа атрибутов «фамилия», «имя», «отчество», в полне возможно, будит достаточно часто использоваться для этих целей. По этому на основе этих атрибутов было бы логично создать инверсный вход. Инверсный вход обозначается на диаграмме символами IEn, заключенными в скобки.

Простой ключ - ключ, содержащий только один атрибут. Обычно присоединитесь, операции выполняются быстрее в этом случае, когда как ключ самое короткое и время простоя от возможных типов данных используется. С этой точки зрения целочисленный тип, у которого есть поддержка аппаратных средств для производительности по ним логических операций лучшим способом подходы. Трудный и составной ключ - ключевое строение нескольких атрибутов. Коммутируемое из свойства обладания атрибутов единственности, но не обладание minimality, вызывают как суперключ. Суперключ - трудный (составной) ключ с большим числом столбцов, которое необходимо, чтобы быть уникальным идентификатором. Такие ключи не редко используются практически, поскольку избыточность может казаться полезной для потребителя. В зависимости от, содержит ли атрибут, который является первичным ключом, любой информацией, отличают искусственные и естественные ключи [5, с. 120].

Искусственный ключ или ключ запасного игрока - ключ, создаваемый DBMS или потребителем посредством некоторой процедуры, которая сам по себе не содержит информацию. Искусственный ключ используется для создания уникальных идентификаторов строк, когда аромат должен быть описан полностью однозначно, чтобы идентифицировать определенный элемент. Искусственный ключ часто используется вместо значительного трудного ключа, который является слишком громоздким, чтобы использоваться в реальной базе данных. Система поддерживает искусственный ключ, но ее никогда не показывают потребителю.

Естественный ключ - ключ, который значительные атрибуты и которые включены, таким образом, содержит информацию.

У каждого из типов первичных ключей есть преимущества и отсутствия. Основные преимущества естественных ключей - то, что они переносят совершенно уверенный, информация и их использование не приводят к необходимости, чтобы добавить в таблице атрибуты, какое значение не имеют никакого смысла и используются только для передачи между соотношениями. По-другому, использование естественных ключей позволяет получать более компактную форму таблицы (в котором не будет никакой избыточной информации), и более естественная связь промежуточный. Основное ограничение естественных ключей, что их использование довольно неудобно в случае вариабельности области данных. Необходимо понять, что значение атрибутов первичного ключа не должно измениться. Это - как только предварительно установленное значение первичного ключа для картежа не может быть изменено позже. Такое требование помещается в ядро для поддержки целостности базы данных. Передача между соотношениями обычно устанавливается на первом ключе, и его изменение приводит к нарушению этой связи или к необходимости изменения записей для нескольких таблиц.

Второе, достаточно эфирное отсутствие естественных ключевых составов, что, как правило, естественные уникальные ключи - составной объект и содержат строковые атрибуты. Но с точки зрения быстродействующей производительности системных ключей бекара часто кажутся неоптимальными.

Не редко в соотношениях есть так называемые вторичные ключи. Вторичный ключ представляет сочетание атрибутов, которое является явным от сочетания, делая первичный ключ. И вторичные ключи не обязательны, обладают свойством единственности. При их определении после ограничений может быть установлен:

UNIQUE- ограничение уникальности, значение вторичных ключей при данном ограничении не могут дублироваться;

NOT NULL-при данном ограничении ни один из атрибутов, входящих в состав вторичного ключа, не может принимать значение NULL. Наложенные ключи - трудные ключи, у которых есть один или несколько общих столбцов. Внешний ключ - атрибут (или набор атрибутов) одно соотношение, будучи ключом другого (или то же самое) соотношения. Внешние ключи используются для установления логической связи между соотношениями. Передача между двумя таблицами устанавливается присвоением значений внешнего ключа одной таблицы к значениям ключа другой таблицы. Часто передача между соотношениями устанавливается на первичном ключе, который является к значениям внешнего ключа значений соотношения первичного ключа другого соотношения, приспосабливаются. Внешний ключ может обратиться и к той же самой таблице, которой он принадлежит. В этом случае внешний ключ вызывают как рекурсивный.

Есть все еще такое понятие как - домен. Понятие домена является более определенным для баз данных, хотя имеет некоторые аналогии к подтипам в некоторых языках программирования. В общем виде домен определяется заданием некоторого базового типа данных, какие доменные элементы, и произвольное логическое прессование применялись к элементу беспокойства типа данных. Если расчет этого логического результата результатов прессования "истина" элемент данных является доменным элементом.

Самая корректная интуитивная обработка понятия домена - понимание домена как допустимый потенциальный набор значений данного типа.

Необходимо отметить также семантическую загрузку понятия домена: данные считают сопоставимыми только в этом случае, когда они касаются одного домена.

**2.3. Связи, типы связей**

Связь в Erwin трактуется как функциональная зависимость между двумя сущностями (в частной, возможной связь сущности с сомой собой).

Если рассматривать диаграмму как графическое представление правил предметной области, то сущности являются существительными, а связи – глаголами.

ERwin поддерживается следующие основные типы связей: идентифицирующая, не идентифицирующая, полная категория, неполная категория, многие – ко – многим.

Передачу вызывают как идентификация, если копия дочернего аромата идентифицируется через его передачу с родительским ароматом. Атрибуты, делающие первичный ключ родительского аромата, таким образом включены в первичный ключ дочернего аромата. Аромат Дочерняя при идентификации передачи всегда зависит.

Передачу вызывают как не идентификация, если копия дочернего аромата идентифицируется по-другому, чем через передачу с родительским ароматом. Атрибуты, делающие первичный ключ родительского аромата, таким образом являются частью не ключевые атрибуты дочернего аромата.

Идентификация передачи представлена сплошной линией;

Идентификация - пунктирная линия. Строки заканчиваются с точкой от дочернего аромата.

При коммуникационной миграции определения атрибутов первичного ключа родительского аромата прибывает, чтобы приспособить область атрибутов дочернего аромата. Поэтому такие атрибуты не вводятся вручную.

Зависимый аромат может наследовать тот же самый атрибут больше чем от одного родительского аромата или от того же самого родительского аромата до некоторой связи.

Поскольку атрибуты первичного ключа родительского аромата по умолчанию переходят с именами, ERwin полагает, что в зависимых атрибутах аромата внешнего ключа появляется только однажды. Чтобы избежать этого сужения, ERwin позволяет поступать на них роль, то есть новые имена, под которыми переход атрибутов будет представлен в дочернем аромате. В случаях повторенной миграции атрибута такой renamings это необходимо.

Местоположение, когда к копии одного аромата соответствует одна или несколько копий второго аромата, и соответствует копии второго аромата одна или несколько копий первого аромата, отражается в логической модели передачей многие - к - очень между данными объектами. На диаграмме передача представлена сплошной линией с точками на концах.

Некоторые объекты определяют целую категорию объектов одного типа. В ERwin такие случаи там - аромат для определения категории memberwise категории, и затем категоризация передачи вводится для них. Родительский аромат категории вызывают как супертип, и дочерний элемент - подтип.

Различная часть (например, данные на платежах часа за височных работников или данные о зарплате и празднике для постоянных элементов штата) располагается в действительности - подтипы. В действительности, супертип атрибут - различитель вводится, позволяя отличать определенные существенные копии - от верхних частей.

В зависимости от, включены ли все возможные объекты - подтипы в модель, категорическая передача является полной или незамкнутой. В ERwin полная категория представлена кругом с двумя подчеркиванием, вместо полного - круг с одним подчеркиванием.

Емкость передачи представляет соотношение количества копий родительского аромата к адекватному количеству копий дочернего аромата. Емкость передачи определяется только для того, чтобы идентифицировать и не идентифицировать связь и регистрируется как 1:n. ERwin, согласно методологии IDEFIX, дает 4 разновидности для и которые представлены дополнительным символом в дочернем аромате.

С целью управления ссылочной целостности (ссылочная целостность в ERwin понята как поддержка требования, что значение внешнего ключа копии дочернего аромата там соответствовало значения первичного ключа в родительском аромате) для каждой передачи необходимые условия на обработках операций INSERT/UPDATE/DELETE для родительского и дочернего аромата, может быть установлен. ERwin представляет следующие разновидности обработки этих событий:

Отсутствие проверки;

Проверка допустимости;

Запрет операции;

Каскадное выполнение операции(DELETE/UPDATE);

Установка пустого(Null-значение) или заданного значения по умолчанию.

Преобразование концептуальной модели данных с целью удаления из нее всех структур, реализация которых в СУБД реляционного типов затруднительна.

Удаление связи типа M:N. Если в концептуальной модели присутствуют связи типа M:N («многие – ко - многим»), то их следует устранить путем определения некоторой промежуточной сущности. Связь типа M:N заменяется двумя связями типа 1:М, устанавливаемыми со вновь созданной сущностью.

Удаление сложных связей. Сложной называется связь, существующая между тремя и больше типами сущностей. Если в концептуальной модели присутствует сложная связь, ее следует устранить с помощью промежуточной сущности. Сложная связь заменяется необходимым количеством бинарных связей типа 1:М, устанавливаемых со вновь созданной сущностью.

Удаление рекурсивных связей. Рекурсивными называются такие связи, в которых сущность некоторого типа взаимодействует сама с собой. Если концептуальная модель содержит рекурсивные связи, они должны быть устранены посредством определения некоторой промежуточной сущности.

Удаление связей с атрибутами. Если в концептуальной модели присутствуют связи, имеющие собственные атрибуты, они должны быть преобразованы путем создания новой сущности.

Удаление множественных атрибутов. Множественными называют атрибуты, которые могут иметь одновременно несколько значений для одного и того же экземпляра сущности. Если в концептуальной модели присутствует множественный атрибут, его следует прообразовать путем определения новой сущности.

**ЗАКЛЮЧЕНИЕ**

В первые годы развития работы с данными, в пятидесятых - начале шестидесятых, последовательный доступа к файлам был правилом. Все данные хранились в файлах последовательного доступа, что заставляло прикладную программу обрабатывать файл целиком. В шестидесятые, когда широко распространились диски с прямым доступом, приобрели популярность файлы произвольного доступа. Этот метод доступа позволяет напрямую обращаться к нужной записи.

По мере того, как компьютерные системы обработки данных приобретали все большее значение, бизнесмены начали понимать, что информация является чрезвычайно ценным корпоративным ресурсом. Они все более ясно осознавали, что данные, необходимые для ответов на многие рабочие вопросы, содержатся в их файлах. Как следствие, они начали требовать создания информационно - управляющих систем, которые использовали бы возможности компьютера для извлечения информации из корпоративных данных.

Таким образом, понадобилось создание информационных систем. Использующих базы данных, которые обеспечивали бы более эффективный доступ и обработки данных.

В последние годы в связи с широким распространением недорогих персональных компьютеров были разработаны сетевые методы, позволяющие пользователям совместно использовать ресурсы. Сервер сети обеспечивает доступ к базе данных пользователей персональных компьютеров, что является эффективным разделением труда: сервер извлекает данные, которые клиентская машина обрабатывает и выдает конечному пользователю. Сети, организованные по принципу клиент/сервер, достигли высокого уровня развития, и все более широко распространяются в бизнесе.

В данной курсовой работе были рассмотрены: 1. цель инфологического моделирования, источники данных, базы данных и система управления ими, принципы проектирования базы данных.

Каждой цивилизации приходится иметь дело с обработкой информации. С развитием экономики и ростом численности населения возрастает и объем взаимосвязанных данных, необходимых для решения коммерческих и административных задач. Взаимосвязанные данные называют систему. Всякая система, призванная облегчить труд человека, кроме обычных форм знаний, требует создания очень сложной модели реального мира.

Ядром информационной системы являются хранимые данные. На любом предприятии данные различных отделов, как правила, пересекаются. Например, для целей управления часто нужна информация по всему предприятию. Принятие решений по производственным вопросам невозможно без информации о товарах, о полученных заказах, о стратегии сбыта и т.д. Это означает, что описывающие конкурентную предметную область данные должны храниться в легко доступном виде.

Сегодня мы можем встретить систему обработки данных старого типа, в которой служащий в ручную помещает данные в скоросшиватель, и рядом с ней – современную систему с применением самой быстродействующей ЭВМ, сложнейшего оборудования и программного обеспечения. Но обе они обязаны предоставлять достоверную информацию в определенное время, определенному лицу, в определенном месте и за определенную плату. Такая система может потребоваться банку, заводу, предприятию сферы обслуживания, университету, больнице, универмагу и многим другим потребителям.

Во второй части мы рассмотрим: сущности, типы сущностей; атрибуты, типы атрибутов; связи, типы связей. В этом разделе мы рассматриваем, что такое атрибут, сущность и связь. Какие бывают ключи и т.д.

И в третьей, заключительной части, мы рассматриваем на примере логическую модель базы данных. Гипотетическая точка обмена валюты - система локальной информации (И.С), вызванный, чтобы автоматизировать процесс пристрелки транзакций покупки и продажи валюты. Создаваемая система должна обеспечить ввод, хранение и информационный поиск о транзакциях, сделанных в данной точке обмена. Каждая транзакция единственный цифровой код приспосабливается. Информация о транзакциях должна содержать сходимость на финике и время транзакции, итоги искупленных и проданных валют, фамилий, имени, патронима и числа паспорта клиента, и также о фамилии, подписывании и регистрационном номере частного дела кассира в отделе кадров. У системы должна быть возможность вычислить, денежное переворачивают в течение одного или нескольких дней, и также выносить информационный поиск о транзакциях под числом паспорта клиента. Составы задачи в разработке строения базы данных разработали автоматизированный И.С.

инфологический моделирование база данные атрибут

**ГЛОССАРИЙ**

|  |  |  |
| --- | --- | --- |
| № п/п | Понятие | Определение |
| 1 | База знаний | Это формализованная система сведений о некоторой предметной области, содержащая данные о свойствах объектов, закономерностях процессов и явлений. |
| 2 | Базы данных | совокупность связанных данных, правила, организации которых основаны на общих принципах описания, хранения и манипулирования данными. |
| 3 | Буфер обмена | Временное место хранения вырезанных или скопированных данных |
| 4 | Индекс | Структура данных, которая помогает СУБД быстрее обнаружить отдельные записи в таблице, а потому позволяет сократить время выполнения запросов пользователя |
| 5 | Инфологического моделирования | обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных |
| 6 | Информация | Любые сведения о каком-либо событии, процессе |
| 7 | Логическое проектирование баз данных | Это процесс конструирования общей информационной модели предприятия на основе отдельных моделей данных пользователей, которая является независимой от особенностей реально используемой СУБД других физических условий. |
| 8 | Реляционная база данных | Набор нормализованных отношений |
| 9 | Система управления БД | это специальный пакет программ, посредством которого реализуется централизованное управление базой данных, и обеспечивают доступ к данным |
| 10 | Системы управления базами данных (СУБД) | Это программные средства, предназначенные для создания, наполнения, обновления и удаления баз данных |
| 11 | Сущность | Это реальный или воображаемый объект, информация о котором представляет интерес |

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Балдин, К.В., Информационные системы в экономике [Текст]/ Уткин В.Б.: Учебник. - М., Издательско-торговая корпорация «Дашков и Ко», 2012- 395 с.

2. Дейт, К.Дж. Введение в системы баз данных. [Текст]/ 8-е изд- М., Вильямс, 2015-1328 с.

3. Информатика: учебник [Текст]/ Б.В. Соболь и др.-Изд. 3-е, дополн. и перераб. - Ростов н/Д: Феникс, 2011. - 446 с.

4. Коннолли, Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика [Текст]/ М., Вильямс, 2013-1440 с.

5. Кузин, А.В. Базы данных. 2-е изд. [Текст]/ - М., Издательский центр «Академия», 2012-320 с.

6. Кузнецов, С.Д. Основы баз данных: учебное пособие[Текст]/.-М.:Бином, 2013-484 с.

7. Леонтьев, В.П. Персональный компьютер [Текст]/./В.П. Леонтьев.-М.: ОЛМА-ПРЕСС, 2014. - 900 с.

8. Марков, А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию [Текст]/.- М., Финансы и статистика, 2012- 512 с.

9. Райордан, Р. Основы реляционных баз данных [Текст]/ Пер. с англ.- М.: Издательско-торговый дом «Русская Редакция», 2011.- 384 с.

10. Фридланд, А.Я. Информатика и компьютерные технологии [Текст]/ А.Я. Фридланд, Л.С. Ханамирова.- М.: Астрель. 2013. - 204 с.

11. Фуфаев, Э.В. Базы данных. 3-е изд. [Текст]/ - М., Изд-во «Академия», 2012- 320 с.

12. Хомоненко, А.Д. и др. Базы данных: Учебник для вузов. [Текст]/ 4-изд.- СПб, Корона принт, 2014- 736 с.

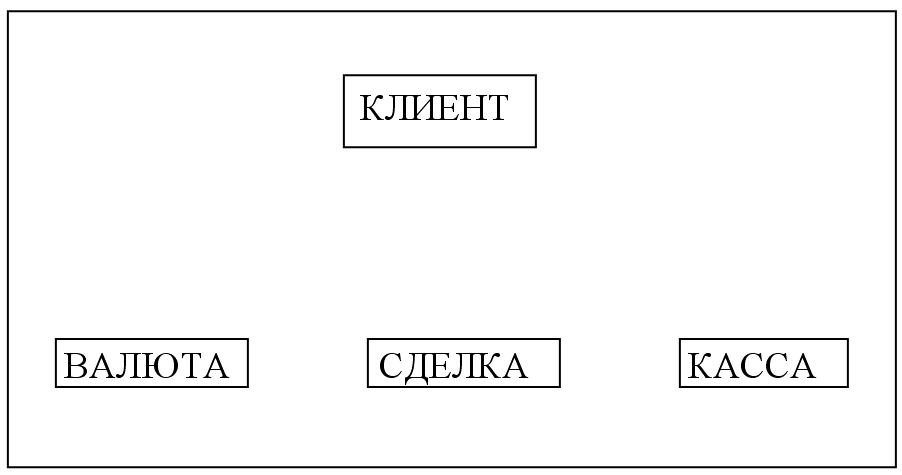
**ПРИЛОЖЕНИЕ А**

Пример сущности КЛИЕНТ

|  |
| --- |
| Номер клиента |
| Фамилия(IE1)  Имя(IE1)  Отчество(IE1)  Дата рождения  Должность |

**ПРИЛОЖЕНИЕ Б**

Внесение новых сущностей в диаграмму



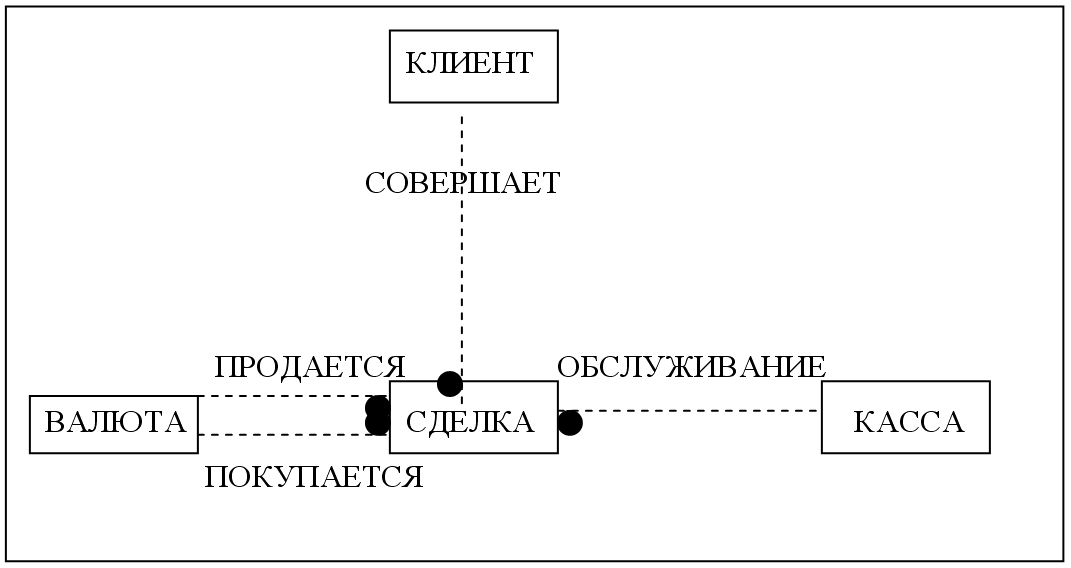
**ПРИЛОЖЕНИЕ В**

Меню вызова редакторов, относящихся к сущности

|  |
| --- |
| Entity Editor…  Attribute Editor… |
| Key Group Editor… |
| Object Font/Color... |

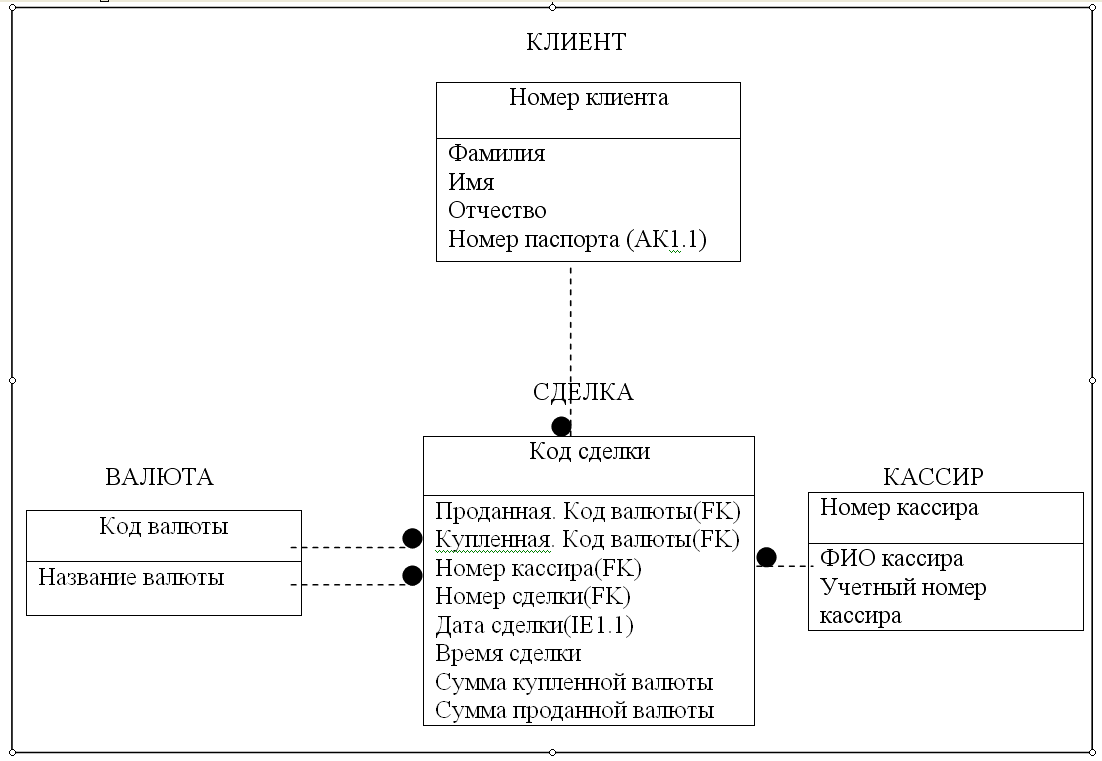
**ПРИЛОЖЕНИЕ Г**

Логическая модель со связями.



**ПРИЛОЖЕНИЕ Д**

Логическая модель с атрибутами



1. Кузнецов С.Д. Основы баз данных: учебное пособие.-М.:Бином, 2007.-С.91. [↑](#footnote-ref-1)