МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра «Вычислительная техника»

|  |
| --- |
| Утверждено на заседании кафедры  «Вычислительная техника»  «29» января 2019г., протокол № 6 |
| Заведующий кафедрой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Н. Ивутин |

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**по выполнению курсовой работы**

**по дисциплине (модулю)**

**«Базы данных»**

**основной профессиональной образовательной программы**

**высшего** **образования – программы бакалавриата**

по направлению подготовки

09.03.01 «Информатика и вычислительная техника»

с профилем

**«Программное обеспечение средств вычислительной техники и автоматизированных систем»**

Форма обучения: очная

Идентификационный номер образовательной программы: 090301-01-18

Тула 2019 год**Разработчик(и) методических указаний**

\_\_\_\_Набродова И.Н., доцент, к.т.н.\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО, должность, ученая степень, ученое звание) (подпись)

1. **Цель выполнения курсовой работы**

Приобретение навыков работы с системами управления базами данных (СУБД). Изучить принципы организации и построения БД. Выбрать предметную область и спроектировать БД. Разработать БД в среде MSSQLServer. Осуществить заполнение БД. Разработать SQL запросы к БД .

**2. Порядок выполнения курсовой работы**

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

1. Корректность схемы БД, т.е. база должна быть гомоморфным образом моделируемой ПО, где каждому объекту ПО соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных.

2. Обеспечение ограничений (на объёмы внешней и оперативной памяти и другие ресурсы вычислительной системы).

3. Эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных).

4. Защита данных (от сбоев и несанкционированного доступа).

5. Простота и удобство эксплуатации.

6. Гибкость, т.е. возможность развития и адаптации к изменениям ПО и/или требований пользователей.

Удовлетворение первых 4-х требований обязательно для принятия проекта.

Процесс проектирования БД включает в себя следующие этапы:

1. Информационно-логическое (инфологическое) проектирование.

2. Определение требований к операционной обстановке, в которой будет функционировать информационная система.

3. Выбор СУБД и других инструментальных программных средств.

4. Логическое проектирование БД.

5. Физическое проектирование БД.

**Этап 1. Инфологическое проектирование**

Инфологический подход не предоставляет формальных способов моделирования реальности, однако он закладывает основы методологии проектирования БД.

Первой задачей инфологического проектирования является определение ПО системы, позволяющее изучить информационные потребности будущих пользователей. Другая задача этого этапа – анализ ПО, который призван сформировать взгляд на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО. Анализ ПО выполняется разработчиком логической базы данных – специалистом в данной ПО.

Инфологическая модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей системы в терминах, понятных пользователю и независимых от реализации системы. Более того, инфологическая модель ПО не должна зависеть от модели данных, которая будет использована при создании БД.

Обычно описание ПО выражается в терминах не отдельных объектов и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов ПО, которые приводят к переходу ПО из одного состояния в другое. Такое описание может быть представлено любым способом, допускающим однозначную интерпретацию.

В простых случаях описание ПО представляется на естественном языке, в более сложных используется также математический аппарат: таблицы, диаграммы, графы и т.п.

**При выполнении курсовой работы необходимо придерживаться следующих обозначений:**

* 1. Имя отношения выделяется курсивом и подчеркиванием и пишется прописными буквами, например:      *СОТРУДНИКИ*.
  2. Имя атрибута отношения выделяется курсивом и подчеркиванием и пишется с большой буквы, например:      *Оклад*.
  3. Ключевые атрибуты отношения выделяются полужирным шрифтом, например:     ***Табельный номер***.
  4. Имя связи между отношениями выделяется курсивом и подчеркиванием и пишется строчными буквами, например: *работает*.

**Построение ифологической модели базы данных необходимо производить с использованием метода "сущность–связь".**

Необходимо разбить ПО на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения информационных потребностей одной группы будущих пользователей или решения отдельной задачи. Каждое локальное представление моделируется отдельно, а затем выполняется их объединение. Выбор локального представления зависит от масштабов ПО. Обычно ПО разбивается на локальные области так, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей (т.е. объектов, о которых в системе будет накапливаться информация).

Для каждой сущности определяются атрибуты, которые делятся на два типа: *идентифицирующие* и *описательные*. Идентифицирующие атрибуты входят в состав ключа (или ключей) и позволяют однозначно распознавать экземпляры сущности. Первичный ключ базовой сущности не может содержать неопределённые значения атрибутов (null). Первичный ключ должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Описательные атрибуты заключают в себе свойства сущности, интересующие пользователей.

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений, которые может принимать данный атрибут.

Далее осуществляется спецификация связей: выявляются связи между сущностями внутри локального представления. Каждая связь именуется. Кроме спецификации связей типа "сущность – сущность", выполняется спецификация связей типа "сущность – атрибут" и "атрибут – атрибут" для отношений между атрибутами, которые относятся к одной и той же сущности или к одной и той же связи типа "сущность – сущность".

При объединении проектировщик может формировать конструкции, производные по отношению к тем, которые были использованы в локальных представлениях. Цель введения подобных абстракций:

* объединение в единое целое фрагментарных представлений о различных свойствах одного и того же объекта;
* введение абстрактных понятий, удобных для решения задач системы, установление их связи с более конкретными понятиями модели;
* образование классов и подклассов подобных объектов (например, класс «изделие» и подклассы типов изделий, производимых на предприятии).

При небольшом количестве локальных областей (не более пяти) объединение выполняется за один шаг. В противном случае обычно выполняют бинарное объединение. При объединении представлений используют три основополагающие концепции:

1. **Идентичность**. Два или более элементов модели идентичны, если они имеют одинаковое семантическое значение.

2. **Агрегация**. Позволяет рассматривать связь между элементами как новый элемент. Например, связь *экзамен* между сущностями *СТУДЕНТ*, *ДИСЦИПЛИНА*, *ПРЕПОДАВАТЕЛЬ* может быть представлена агрегированной сущностью *ЭКЗАМЕН* с атрибутами *Название дисциплины*, *Фамилия преподавателя*, *Фамилия студента*, *Оценка*.

3. **Обобщение**. Позволяет образовывать многоуровневую иерархию обобщений.

По завершении объединения результаты проектирования представляют собой концептуальную инфологическую модель ПО. Модели локальных представлений – это внешние инфологические модели.

Например, БД «Publications» должна хранить сведения о печатных изданиях, а также ссылки на интересные ресурсы в Internet. И те, и другие источники информации будут касаться одной темы, а именно «баз данных». Выделим интересующие нас сущности и определим связи между ними.

Прежде всего займемся понятием «печатное издание». Мы знаем, что объект «печатное издание» воплощается в виде книги, которую можно полностью описать с помощью следующих характеристик: название, автор, год издания и издатель (издательство). Можно ли на основании этого ввести сущность «книга», а названные характеристики определить в качестве ее атрибутов? Прежде чем сделать это рассмотрим более внимательно отношения между книгой и ее характеристиками:

Один автор может написать несколько книг, и, в то же время, одна книга может быть написана несколькими авторами. Следовательно, «книга» и «автор» в данном случае выступают как различные сущности, объединяемые связью N: M. Для того, чтобы определить класс принадлежности сущностей в связи, отметим, что книг без авторов не бывает, как и авторов без книг. Значит, каждая сущность должна иметь обязательный класс принадлежности (кардинальность связи(1,N) : (1,N)).

Точно так же один издатель может издавать сразу несколько книг, но каждая конкретная книга издается только в одном месте. Следовательно, мы должны ввести сущность «издатель», ассоциируемую с «книгой» связью типа 1:N. Т.к. каждая книга кем-то издана, класс принадлежности сущности "издатель" в данной связи будет (1,1), но в то же время мы допускаем хранение сведений об издательствах, чьих книг в нашей базе данных пока нет. Соответственно, класс принадлежности сущности «книга» в этой связи (0,N).

По поводу характеристики книги «название» можно сказать следующее: как правило авторы, пишущие на одну тему, стараются придумывать для своих произведений оригинальные названия. Поэтому, можно уверенно предположить, что каждое название обязательно связано только с одной книгой (и каждая книга имеет только одно название). Следовательно, «название» нужно оставить в списке атрибутов «книги».

Те же рассуждения можно повторить и для характеристики «год издания». Ее мы тоже оставим в списке атрибутов «книги».

Таким образом, мы определили, что у сущности «книга» имеется два атрибута «название» и «год издания». Как уже говорилось, название, скорее всего, будет однозначно определять данную книгу, чего не скажешь о годе издания. Поэтому объявим ключом сущности атрибут «название» (или «имя\_книги»).

Что касается всех возможных авторов, то нас интересует только одна их характеристика– имя. Поэтому, сущность «автор» имеет только один атрибут «имя\_автора», который и является ключом.

С сущностью "издатель" дел обстоит несколько сложнее. Практически все крупные издательства имеют сейчас собственные web-страницы, которые могут содержать информацию полезную для пользователей проектируемой базы данных. Поэтому, нужно рассмотреть две характеристики этого объекта: «имя\_издателя» и «URL» (uniformresourcelocator– универсальный указатель ресурсов, с помощью которого в Internet определяется путь к web– странице). Ясно, что каждый издатель имеет уникальное имя и уникальный url, но прежде чем внести их в список атрибутов, вспомним, что наша база данных должна также содержать ссылки и на другие Internet-ресурсы. Возможно, при дальнейшем анализе возникнет необходимость во введении отдельной сущности «URL». Поэтому «имя\_издателя» внесем в список атрибутов сущности «издатель», а «URL» будем считать атрибутом отдельной сущности «web–страница», ассоциируемой с «издателем» связью (1,1):(1,1).

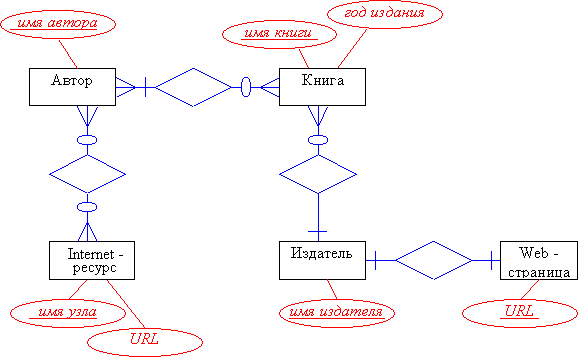
Теперь настала пора заняться объектом «ресурс Internet». Его мы можем описать с помощью понятий «имя ресурса», «url», «автор». Внимательно рассмотрев связи этих понятий с описываемым объектом, можно прийти к заключению, что «имя\_ресурса» и «url» однозначно с ним связаны, т.е. являются атрибутами. В то же время, «автор» является отдельной сущностью (один ресурс может иметь много авторов, и один автор может быть создателем многих web– страниц). Т.к. мы уже ранее ввели сущность «автор» просто определим характеристики ее связи с сущностью «Internet-ресурс». Из сказанного выше следует, что эти сущности объединяются связью n:m, в то же время, автор какой-либо книги может не иметь собственной web–страницы, а авторы некоторых Internet ресурсов не указывают своих имен (т.е. можно формально сказать, что эти ресурсы не имеют авторов). Следовательно, класс принадлежности обеих сущностей будет необязательным.

Прежде чем объявить нашу модель готовой, проверим еще раз определение каждой сущности. Внимательный анализ покажет, что построенная модель имеет несколько ошибок:

Сущность «автор» имеет обязательный класс принадлежности в связи с сущностью «книга». Это означает, что мы не сможем добавить в базу данных сведения о человеке, который создал собственный web-сайт, но не написал ни одной книги. Для того, чтобы устранить это ограничение изменим класс принадлежности сущности «книга» в рассматриваемой связи «автор»– «книга» на необязательный.

При анализе объекта «издатель» мы предположили, что сущность «web-страница» может быть объединена с сущностью «Internet-ресурс». Однако, мы видим, что эти сущности имеют разный набор атрибутов, следовательно, выполнить такое объединение нельзя. Вспомним, что в противном случае, предполагалось единственный атрибут сущности «web–страница» присоединить к атрибутам сущности «издатель». Тем не менее, не будем этого делать, в следующем разделе мы увидим, что с помощью правил порождения реляционных отношений из модели «сущность-связь» в том и в другом случае мы получим одинаковый результат.

Готовая модель «сущность-связь» представлена на следующем рисунке:



На этапе анализа ПО также необходимо решить следующие задачи:

1. Определить правила (ограничений целостности), которым должны удовлетворять сущности ПО, атрибуты сущностей и связи между ними. Часть этих правил реализуется в схеме базы данных (возможности реализации ограничений целостности в схеме БД определяются моделью данных той СУБД, которая будет выбрана для реализации проекта). Остальные правила реализуются с помощью программного обеспечения.

2. Выделить группы пользователей системы. Каждая группа выполняет определённые задачи и обладает разными правами доступа к системе.

3. Создать внешнюю спецификацию тех функций (процессов), которые эта система будет выполнять. Например, для той же библиотечной системы это задачи поиска книг (по определённым критериям), выдачи/приёма книг, определение списка должников и т.д.

**Этап 2. Определение требований к операционной системе**

На этом этапе производится оценка требований к вычислительным ресурсам, необходимым для функционирования системы, выбор типа и конфигурации ЭВМ, типа и версии операционной системы.

Выбор зависит от таких следующих показателей:

* примерный объём данных в БД;
* динамика роста объёма данных;
* характер запросов к данным (извлечение и обновление отдельных записей, групп записей, обработка отдельных отношений или соединение отношений);
* интенсивность запросов к данным по типам запросов;
* требования к времени отклика системы по типам запросов.

**Этап 3. Выбор СУБД и инструментальных программных средств**

Выбор СУБД является одним из важнейших моментов в разработке проекта БД, так как он принципиальным образом влияет на весь процесс проектирования БД и реализации информационной системы.

Теоретически при осуществлении этого выбора нужно принимать во внимание десятки факторов. Но на практике разработчики руководствуются лишь собственной интуицией и несколькими наиболее важными критериями, к которым, в частности, относятся:

* тип модели данных, которую поддерживает данная СУБД, адекватность модели данных структуре, рассматриваемой ПО;
* характеристики производительности СУБД;
* запас функциональных возможностей для дальнейшего развития информационной системы;
* степень оснащенности СУБД инструментарием для персонала администрирования данными;
* удобство и надежность СУБД в эксплуатации;
* стоимость СУБД и дополнительного программного обеспечения.

В связи с тем, что в данной работе СУБД задается в задании, на данном этапе решается задача выбора инструментального ПО.

**Этап 4. Логическое проектирование БД**

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая инфологической модели ПО. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. Результатом выполнения этого этапа являются схемы БД концептуального и внешнего уровней архитектуры, составленные на языках определения данных (DDL) выбранной СУБД.

При этом разработка логической структуры базы данных должна производиться с учетом соблюдения условий нормализации отношений.

На начальном шаге логического проектирования таблицы строятся таким образом, чтобы минимизировать количество таблиц в базе данных. Очевидно, что при таком подходе отношения не будут находиться ни в одной из нормальных форм, либо находиться в 1й нормальной форме. Необходимо последовательно привести каждую таблицу к 4-й нормальной форме.

Ниже рассматривается отношение *КНИГИ* (табл. 3.1) и последовательность его приведения к 4й нормальной форме.

***Id***      – идентификатор (первичный ключ),

*Code*  – шифр рубрики,

*Theme*– название рубрики,

*Title*   – название книги,

*Author*– автор,

*Editor* – редактор,

*Type*  – тип издания (учебник, учебное пособие, сборник и.т.п.),

*Year*   – год издания,

*Pg*     – количество страниц.

Таблица 3.1. Исходное отношение *КНИГИ*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***ID*** | *Code* | *Theme* | *Author* | *Title* | *Editor* | *Type* | *Year* | *Pg* |
| 200 | 681.3 | ПО ВТ | Бочков С. | Язык СИ | Садчиков П. | учебник | 1990 | 384 |
| Субботин Д. |
| 100 | 681.3 | ПО ВТ | Джехани Н. | Язык АДА |  | учебник | 1960 | 552 |
| 300 | 621.5 | МО | Крон Г. | Диакоптика | Баранов А. | учебник | 1972 | 544 |
| 876 | 007 | ИИ | Гик Е.Я. | Шахматы и математика | Кикоин И. | учебное пособие | 1983 | 176 |
| Капица С. |
| 440 | 32.97 | ВТ |  | ПУ для ПЭВМ | Витенберг А. | справочник | 1992 | 208 |
| 385 | 001.8 | Инфор-матика | Фролов Г. | Элементы информатики | Храмов А. | учебное пособие | 1989 | 304 |
| Кузнецов Э. | Рожков П. |

**Примечание.** В таблице 3.1 используются следующие сокращения:

ВТ – вычислительная техника;

ПО ВТ – программное обеспечение вычислительной техники;

МО – математическое обеспечение;

ИИ – искусственный интеллект.

**Первая нормальная форма (1НФ).**

Отношение приведено к 1НФ, если все его атрибуты простые.

Отношение *КНИГИ* содержит сложные атрибуты *Author* («Авторы») и *Editor* («Редакторы»). Для приведения к 1НФ требуется сделать ключ отношения составным – атрибуты *ID*, *Author* и *Editor* (табл. 3.2).

Таблица 3.2. Отношение *КНИГИ*, приведённое к 1НФ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***ID*** | *Code* | *Theme* | ***Author*** | *Title* | ***Editor*** | *Type* | *Year* | *Pg* |
| 200 | 681.3 | ПО ВТ | Бочков С. | Язык СИ | Садчиков П. | учебник | 1990 | 384 |
| 200 | 681.3 | ПО ВТ | Субботин Д. | Язык СИ | Садчиков П. | учебник | 1990 | 384 |
| 100 | 681.3 | ПО ВТ | Джехани Н. | Язык АДА |  | учебник | 1960 | 552 |
| 300 | 621.5 | МО | Крон Г. | Диакоптика | Баранов А. | учебник | 1972 | 544 |
| 876 | 007 | ИИ | Гик Е.Я. | Шахматы и математика | Кикоин И. | учебное пособие | 1983 | 176 |
| 876 | 007 | ИИ | Гик Е.Я. | Шахматы и математика | Капица С. | учебное пособие | 1983 | 176 |
| 440 | 32.97 | ВТ |  | ПУ для ПЭВМ | Витенберг А. | Спра-вочник | 1992 | 208 |
| 385 | 001.8 | Инфор-матика | Фролов Г. | Элементы информатики | Храмов А. | учебное пособие | 1989 | 304 |
| 385 | 001.8 | Инфор-матика | Кузнецов Э. | Элементы информатики | Рожков П. | учебное пособие | 1989 | 304 |

**Вторая нормальная форма (2НФ).**

Введём понятие **функциональной зависимости**. Пусть X и Y – атрибуты (группы атрибутов) некоторого отношения. Говорят, что Y функционально зависит от X, если в любой момент времени каждому значению X=х соответствует единственное значение Y=y (X→Y). (При этом любому значению Y=y может соответствовать несколько значений Х=(х1, х2,…)).

Атрибут X в функциональной зависимости X→Y называется *детерминантом* отношения.

В нормализованном отношении все не ключевые атрибуты функционально зависят от ключа отношения. Говорят, что не ключевой атрибут функционально полно зависит от составного ключа, если он функционально зависит от ключа, но не находится в функциональной зависимости ни от какой части составного ключа.

Отношение находится во 2НФ, если оно приведено к 1НФ и каждый не ключевой атрибут функционально полно зависит от составного ключа.

Для того чтобы привести отношение ко 2НФ, нужно:

* построить его проекцию, исключив атрибуты, которые не находятся в функционально полной зависимости от составного ключа;
* построить дополнительные проекции на часть составного ключа и атрибуты, функционально зависящие от этой части ключа.

Ключом отношения *КНИГИ* (табл. 3.2) является комбинация полей (***ID*, *Author*, *Editor***). Все поля, не входящие в состав ключа, зависят только от идентификатора книги. Поэтому отношение должно быть разбито на два: *КНИГИ* (табл. 3.3) и *КНИГИ–АВТОРЫ–РЕДАКТОРЫ* (табл. 3.4). Эти отношения связаны по внешнему ключу, которым является поле ID.

Таблица 3.3. Отношение *КНИГИ*, приведённое к 2НФ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***ID*** | *Code* | *Theme* | *Title* | *Type* | *Year* | *Pg* |
| 200 | 681.3 | ПО ВТ | Язык СИ для ПК | Учебник | 1990 | 384 |
| 100 | 681.3 | ПО ВТ | Язык АДА | Учебник | 1960 | 552 |
| 300 | 621.5 | МО | Диакоптика | Учебник | 1972 | 544 |
| 876 | 007 | ИИ | Шахматы и математика | учебное пособие | 1983 | 176 |
| 440 | 32.97 | ВТ | ПУ для ПЭВМ | Справочник | 1992 | 208 |
| 385 | 001.8 | Информатика | Элементы информатики | учебное пособие | 1989 | 304 |

.

Таблица 3.4. Отношение *КНИГИ–АВТОРЫ–РЕДАКТОРЫ* (2НФ)

|  |  |  |
| --- | --- | --- |
| ***ID*** | ***Author*** | ***Editor*** |
| 200 | Бочков С. | Садчиков П. |
| 200 | Субботин Д. | Садчиков П. |
| 100 | Джехани Н. |  |
| 300 | Крон Г. | Баранов А. |
| 876 | Гик Е.Я. | Кикоин И. |
| 876 | Гик Е.Я. | Капица С. |
| 440 |  | Витенберг А. |
| 385 | Фролов Г. | Храмов А. |
| 385 | Кузнецов Э. | Рожков П. |

**Третья нормальная форма (3НФ).**

Рассмотрим понятие **транзитивной зависимости**. Пусть X, Y, Z – атрибуты некоторого отношения. При этом X→Y и Y→Z, но обратное соответствие отсутствует, т.е. Z не зависит от Y или Y не зависит от X. Тогда говорят, что Z транзитивно зависит от X (X→→ Z).

Отношение находится в 3НФ, если оно находится во 2НФ и в нем отсутствуют транзитивные зависимости.

Для отношения *КНИГИ* (табл. 3.3) атрибут *Theme* зависит от атрибута *Code*, а не от ключа (хотя название рубрики, естественно, соответствует её шифру). Поэтому для приведения отношения к 3НФ (табл. 3.5) нужно выделить из него ещё одно отношение *РУБРИКАТОР* (табл. 3.6).

Таблица 3.5. Отношение *КНИГИ*, приведённое к 3НФ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | *Code* | *Title* | *Type* | *Year* | *Pg* |
| 200 | 681.3 | Язык СИ для ПК | Учебник | 1990 | 384 |
| 100 | 681.3 | Язык АДА | Учебник | 1960 | 552 |
| 300 | 621.5 | Диакоптика | Учебник | 1972 | 544 |
| 440 | 32.97 | ПУ для ПЭВМ | Справочник | 1992 | 208 |
| 876 | 007 | Шахматы и математика | учебное пособие | 1983 | 176 |
| 385 | 001.8 | Элементы информатики | учебное пособие | 1989 | 304 |

.

Таблица 3.6. Отношение *РУБРИКАТОР*, приведённое к 3НФ

|  |  |
| --- | --- |
| ***Code*** | *Theme* |
| 681.3 | ПО ВТ |
| 621.5 | МО |
| 007 | ИИ |
| 32.97 | ВТ |
| 001.8 | Информатика |

**Четвертая нормальная форма (4НФ).**

Введём понятие **многозначной зависимости**. Многозначная зависимость существует, если заданным значениям атрибута X соответствует множество, состоящее из нуля (или более) значений атрибута Y (X→Y). Если в отношении присутствуют многозначные зависимости, то схема отношения должна находиться в 4НФ.

Различают тривиальные и нетривиальные многозначные зависимости. **Тривиальной** называется такая многозначная зависимость X→→Y, для которой Y ⸦ X или X U Y = R, где R – рассматриваемое отношение. Тривиальная многозначная зависимость не нарушает 4НФ. Если хотя бы одно из двух этих условий не выполняется (т.е. Y не является подмножеством X или X U Y состоит не из всех атрибутов R), то такая многозначная зависимость называется нетривиальной.

Отношение находится в 4НФ, если оно находится в 3НФ и в нем отсутствуют нетривиальные многозначные зависимости.

Для отношения *КНИГИ–АВТОРЫ–РЕДАКТОРЫ* (табл. 3.4) атрибуты *Author* и *Editor* зависит образуют две многозначные зависимости от первичного ключа, и при этом значения этих атрибутов не зависят друг от друга. Поэтому для приведения отношения к 4НФ нужно разбить его на два отношения *КНИГИ–АВТОРЫ* и *КНИГИ–РЕДАКТОРЫ* (табл. 3.7, 3.8).

Таблица 3.7. Отношение *КНИГИ–АВТОРЫ* (4НФ)

|  |  |
| --- | --- |
| ***ID*** | ***Author*** |
| 200 | Бочков С. |
| 200 | Субботин Д. |
| 100 | Джехани Н. |
| 300 | Крон Г. |
| 876 | Гик Е.Я. |
| 385 | Фролов Г. |
| 385 | Кузнецов Э. |

.

Таблица 3.8. Отношение *КНИГИ–РЕДАКТОРЫ* (4НФ)

|  |  |
| --- | --- |
| ***ID*** | ***Editor*** |
| 200 | Садчиков П. |
| 300 | Баранов А. |
| 876 | Кикоин И. |
| 876 | Капица С. |
| 440 | Витенберг А. |
| 385 | Храмов А. |
| 385 | Рожков П. |

**Этап 5. Физическое проектирование БД**

Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т.е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам "физической" БД. Результаты этого этапа документируются в форме схемы хранения на языке определения хранимых данных. Принятые на этом этапе решения оказывают определяющее влияние на производительность системы.

Фактически проектирование БД имеет итерационный характер. В процессе функционирования системы становится возможным измерение её реальных характеристик, выявление "узких" мест. И если система не отвечает предъявляемым к ней требованиям, то обычно она подвергается реорганизации, т.е. модификации первоначально созданного проекта.

Важным шагом на данном этапе необходимо проработать вопрос использования индексов.

В системах, поддерживающих язык SQL, индекс создаётся командой CREATE INDEX. Индексы повышают производительность запросов, которые выбирают относительно небольшое число строк из таблицы. Для определения целесообразности создания индекса нужно проанализировать запросы, обращённые к таблице, и распределение данных в индексируемых столбцах.

Система может воспользоваться индексом по определённому полю, если в запросе на значение этого поля накладывается условие, например:

SELECT \* FROM authors WHERE name = 'Даль';

Но даже при наличии такой возможности система не всегда обращается к индексу. Очевидно, что если запрос выбирает больше половины записей отношения, то извлечение данных через индекс потребует больше времени, чем последовательная обработка данных. В подобных случаях использование индекса нецелесообразно.

Обращение к составному индексу возможно только в том случае, если в условиях выбора участвуют столбцы, представляющие собой лидирующую часть составного индекса. Например, если индекс строится по столбцам (X, Y, Z), то обращение к индексу будет происходить в тех случаях, когда в условии запроса участвуют столбцы XYZ, XY или X.

При создании индекса большое значение имеет понятие селективности. **Селективность** определяется процентом строк, имеющих одинаковое значение индексируемого столбца: чем выше этот процент, тем меньше селективность.

Выбор индексируемых столбцов определяется следующими соображениями:

* В первую очередь выбираются столбцы, которые часто встречаются в критериях поиска.
* Стоит индексировать столбцы, которые используются для соединения таблиц или являются внешними ключами. В последнем случае наличие индекса позволяет обновлять строки подчиненной таблицы без блокировки основной таблицы, когда происходит интенсивное конкурентное обновление связанных между собою таблиц.
* Нецелесообразно индексировать столбцы с низкой селективностью. Если селективность столбца низкая, то индексирование проводится только в том случае, если выборка чаще производится по редко встречающимся значениям.
* Не индексируются столбцы, которые часто обновляются, т.к. команды обновления ведут к потере времени на обновление индекса.
* Не индексируются столбцы, которые часто используются как аргументы функций или выражений: как правило, такие функции не позволяют использовать индекс.

Необходимо произвести экспериментальные исследования, показывающие повышение (или снижение) быстродействия индексированной базы данных при выборке/добавлении/удалении данных. Исследования производить многократным выполнением повторяющихся запросов со случайными параметрами. На основании результатов исследования сделать выводы о необходимости индексирования тех или иных столбцов таблиц базы данных.

**Этап 6.Разработка программного обеспечения**.

Программное обеспечение разрабатывается на любом языке программирования с использованием программных средств, выбранных на этапе 2 (MicrosoftVisualStudio, Borland/CodeGearDelphi/CBuilder, J2RE, PHPи т.п.). Разработанное ПО должно быть выполнено в форме дистрибутива.

**Этап 7. Оформление курсовой работы**

Результаты выполнения курсовой курсовой работы должны быть оформлены в виде пояснительной записки. Пояснительная записка должна быть оформлена в соответствии с требованиями ГОСТ 7.32-2001.

Объем пояснительной записки курсовой работы 20-30 стр. Текстовая часть должна содержать ссылки на используемую литературу. Список используемой литературы приводится в соответствии с требованиями ГОСТ 34.662.89.

Текст должен быть отформатирован с параметрами: отступы – нулевые, красная строка –1.27 см, межстрочный интервал – полуторный, выравнивание – по ширине. Текст должен содержать переносы. Все страницы текста должны содержать верхний колонтитул с нумерацией страниц с размещением по центру. Размер основного шрифта – 14.

Пояснительная записка должна состоять из нескольких обязательных разделов, каждый из которых соответствует одному из этапов проектирования.

Файл пояснительной записки должен иметь имя, соответствующее шифру курсовой работы *Шифр\_группы*\_*NN*.doc (*NN*– номер студента в списке группы), файл Readme.txt должен содержать описание всех файлов работы.

Титульный лист приведен в приложении.

**3. Библиографический список**

#### *ОСНОВНОЙ*

1. Советов, Б.Я. Базы данных. Учебник: учебник для вузов / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. – Юрайт: Профессиональное образование, 2017. – 463с. – Библиогр. в конце кн. – ISBN 978-5-534-01653-6 /в пер./: 193.12.
2. Смирнов Олег Игоревич. Базы данных: модели и проектирование: учеб. пособие / О. И. Смирнов; ТулГУ. – Тула: Изд-во ТулГУ, 2007. – 106 с.: ил. – в дар от каф. ПМиИТулГУТулГУ: 1308588-1308597. – Библиогр. в конце кн. – ISBN 978-5-7679-1108-0.
3. Семенчев Е.А. Распределенные и объектно-ориентированные базы данных: учеб. пособие / Е. А. Семенчев; ТулГУ. – Тула: Изд-во ТулГУ, 2008. – 136 с.: ил. – ISBN 978-5-7679-1356-5: 72.00.

#### *ДОПОЛНИТЕЛЬНЫЙ*

1. Автоматизированные базы данных и банки знаний: Учеб. пособие для вузов / В.Ю. Анцев, А.Н. Иноземцев, Д.И. Троицкий; ТулГУ. – Тула, 2002. – 96с.: ил. – Библиогр. в конце кн. — ISBN 5-7679-0309-3: 22.00.
2. [Малыхина М.П.](http://library.tsu.tula.ru/cgi-bin/zgate.exe?ACTION=follow&SESSION_ID=7216&TERM=%D0%9C%D0%B0%D0%BB%D1%8B%D1%85%D0%B8%D0%BD%D0%B0,%20%D0%9C.%20%D0%9F.%5B1,1004,4,101%5D&LANG=rus) Базы данных: основы, проектирование, использование: учеб. пособие для вузов / М. П. Малыхина. – 2-е изд. – СПб: БХВ-Петербург, 2006. – 517 с.: ил. – Библиогр. в конце кн. – Предм. указ.: с.513-517. – ISBN 5-94157-941-1 ((в пер.)): 299,00.
3. Базы данных: проектирование и использование: учебник для вузов / С.М. Диго. – М.: Финансы и статистика, 2005. – 592с.: ил. – Библиогр.в конце кн. – ISBN 5-279-02571-2 /в пер./: 204.00.
4. Базы данных / С.В. Глушаков, Д.В. Ломотько. – М.: АСТ, 2002. – 504с.: ил. – (Учебный курс). – ISBN 966-03-1257-1 /в пер./: 75.00.

ПРИЛОЖЕНИЕ А

**Варианты заданий на курсовую работу**

|  |  |  |
| --- | --- | --- |
|  | | |
| **Номер варианта** | **Модель** | **Используемая СУБД** |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |
| 5 | 5 | 5 |
| 6 | 6 | 1 |
| 7 | 7 | 2 |
| 8 | 8 | 3 |
| 9 | 9 | 4 |
| 10 | 10 | 5 |
| 11 | 11 | 1 |
| 12 | 12 | 2 |
| 13 | 13 | 3 |
| 14 | 14 | 4 |
| 15 | 15 | 5 |
| 16 | 16 | 1 |
| 17 | 17 | 2 |
| 18 | 1 | 3 |
| 19 | 2 | 4 |
| 20 | 3 | 5 |
| 21 | 4 | 1 |
| 22 | 5 | 2 |
| 23 | 6 | 3 |
| 24 | 7 | 4 |
| 25 | 8 | 5 |
| 26 | 9 | 1 |
| 27 | 10 | 2 |
| 28 | 11 | 3 |
| 29 | 12 | 4 |
| 30 | 13 | 5 |
| 31 | 14 | 1 |
| 32 | 15 | 2 |

**Варианты СУБД:**

**1 – MicrosoftSQLServer**

**2 –** OpenOffice.org Base

**3 – Interbase/Firebird**

**4 – MySQL**

**5 –** PostgreSQL

**Варианты модели:**

**Вариант 1**

Спроектировать базу данных для работников библиотеки. В БД должны храниться сведения об имеющихся в библиотеке книгах, о читателях библиотеки  и  читальных  залах.

Для каждой книги в БД должны храниться сведения об авторе, названии, годе издания и числе экземпляров, имеющихся в каждом зале библиотеки, а также шифр книги и дата закрепления книги за читателем. Сведения о читателях библиотеки должны включать фамилию, номер телефона и уникальный номер читательского билета.  Читатели закрепляются за определенным залом и могут записываться и выписываться из библиотеки. Библиотека имеет несколько читальных залов, которые характеризуются номером, названием и вместимостью. Библиотека может получать новые книги и списывать старые. Шифр книги может изменяться в результате переклассификации.

Библиотекарю могут потребоваться следующие сведения о текущем состоянии библиотеки:

* какие книги закреплены за читателем;
* как называется книга с заданным шифром;
* какой шифр у книги с заданным названием;
* когда книга была закреплена за читателем:
* какое число читателей пользуется библиотекой.

Библиотекарь может вносить следующие изменения:

* запись нового читателя в библиотеку;
* списывание старой книги;
* изменение шифра книги.

Необходимо предусмотреть возможность выдачи справки о количестве книг определенного автора в читальном зале и отчета о работе библиотеки в течение месяца (общее количество читателей, количество записавшихся в этот месяц, какие книги и сколько раз были взяты, кто из читателей не брал книг в этот месяц).

**Вариант 2**

Спроектировать базу данных «Картотека Интерпола». В БД должны храниться данные по каждому зарегистрированному преступнику: фамилия, имя, кличка, рост, цвет волос и глаз, особые приметы, гражданство, место и дата рождения, последнее место жительства, знание языков, преступная профессия, последнее дело и так далее. Преступные и мафиозные группировки (данные о подельщиках).

Могут потребоваться сведения по любому подмножеству признаков.

Возможны следующие изменения в БД:

* перенос «завязавших» в архив;
* удаление – только после смерти.

Необходимо предусмотреть возможность выдачи справки освободившемуся преступнику (Ф.И.О. преступника, сроки отсидки и по какой статье, дата освобождения, таблица для отметки освободившегося) и отчета о группировке (ФИО, кличка, гражданство, место и дата рождения, входящих преступников в данную группировку).

**Вариант 3**

Спроектировать базу данных для завуча школы. Для каждого учителя в ВД должны храниться сведения о предметах, которые он преподает, номере кабинета, который за ним закреплен, номерах классов, в которых он ведет занятия, номере урока и дне, когда он ведет уроки. Существуют учителя, которые не имеют собственного кабинета. Об учениках должны храниться следующие сведения: фамилия и имя, в каком классе учится, какая оценка по каждому предмету получена. Ученик может исправить полученную оценку. Завуч может добавить информацию о новом учителе или ученике, а также удалить о выбывших. Завучу могут потребоваться следующие сведения:

* какой предмет будет в заданном классе, например, во вторник на заданном уроке;
* кто из учителей преподает в заданном классе;
* в каком кабинете будет 5-й урок в среду у некоторого класса;
* в каких классах преподает определенный учитель заданный предмет;
* расписание на заданный день недели для класса.

Завуч может вносить следующие изменения:

* ввести информацию о новом учителе;
* удалить запись об ученике;
* изменить оценку ученику.

Необходимо предусмотреть возможность выдачи справки о количестве учеников, в данном классе и отчета о работе школы (количество учителей по предметам, количество кабинетов, число учеников в каждом классе, число учащихся по классам и по школе).

**Вариант 4**

Спроектировать базу данных «Бюро знакомств». В БД должны храниться сведения о потенциальных женихах и невестах: пол, регистрационный номер, дата регистрации, сведения о себе, требования к партнеру.

Работнику бюро могут потребоваться следующие сведения:

* выбор подмножества подходящих кандидатур;
* подготовка встреч.

Возможны следующие изменения в БД:

* перенос в архив пар, решивших свои семейные проблемы;
* удаление клиентов, отказавшихся от услуг.

Необходимо предусмотреть возможность выдачи приглашения для знакомства и отчета о созданных парах (сколько пар было создано за месяц, ФИО жениха и невесты, сколько клиентов отказалось от услуг).

**Вариант 5**

Спроектировать базу данных для работников приемной комиссии. В БД должны храниться сведения об абитуриентах, датах экзаменов и консультаций, номерах аудиторий.

Для каждого абитуриента в БД должны храниться сведения об абитуриенте, номере экзаменационного листа, который он получает, о кафедре и факультете, куда он собирается поступать, о номере группы и потоке, в котором он будет сдавать экзамен (группы объединены в потоки, по 3-4 группы на поток); оценка по каждому предмету, которая может быть изменена на апелляции. Абитуриенты могут подавать и забирать документы, а также перевести их на другую кафедру. Также должны храниться даты консультаций и экзаменов по предметам для каждого потока и номера аудиторий. Могут потребоваться сведения:

* список абитуриентов на заданный факультет:
* полученные оценки для абитуриента;
* дата консультации и экзамена для абитуриента по данному предмету;
* номера аудиторий, где будут экзамены у заданной группы:
* список групп, которые будут заниматься в заданной аудитории в заданное время.
* ввести информацию о новом абитуриенте;
* изменить оценку абитуриенту;
* удалить запись об абитуриенте.

Необходимо предусмотреть возможность выдачи справки о том, что данный абитуриент поступает в институт на факультет, и отчета о работе приемной комиссии факультета (количество поступающих, на какие кафедры и сколько, количество абитуриентов в каждой группе, в какие дни и где проводятся экзамены, сколько сдало на 2,3,4,5 по предметам).

**Вариант 6**

Спроектировать базу данных для почтовых работников. В БД должны храниться сведения о газетах, почтовых отделениях, получающих газеты, о типографиях, выпускающих газеты. Сведения о газетах должны включать в себя: название газеты, шифр издания (может быть изменен), цену экземпляра газеты, ФИО редактора, номера типографий, где печатается эта газета. Возможно появление новых газет. Для типографий: адрес типографии, количество газет данного наименования, печатающихся в этой типографии (в одной типографии может печататься несколько газет). Типография может быть закрыта. Для почтового отделения: адрес отделения, название и количество экземпляров, поступающих на каждое почтовое отделение.

Работникам может потребоваться следующая информация:

* по каким адресам печатаются газеты данного наименования:
* какая фамилия у редактора газеты, которая печатается в указанной типографии самым большим тиражом;
* на какие почтовые отделения (адреса) поступает газета, имеющая цену больше указанной;
* какие газеты и куда (номер почты) поступают в количестве меньшем, чем заданное;
* на какую почту поступает данная газета, печатающаяся по данному адресу типографии.

Работник может вносить следующие изменения:

* добавить информацию о новой газете,
* изменить цену газеты;
* удалить информацию о типографии.

Необходимо предусмотреть возможность выдачи справки об индексе и цене данной газеты и отчета о работе типографии (общее количество газет, печатающихся в ней, название, индекс и количество экземпляров для каждой газеты, ФИО редактора).

**Вариант 7**

Спроектировать базу данных Биржа труда. База безработных: анкетные данные, профессия, образование, место и должность последней работы, причина увольнения, семейное положение, жилищные условия, контактные координаты, требования к будущей работе. База вакансий: фирма, должность, условия труда и оплаты, жилищные условия, требования к специалисту.

При работе с БД могут потребоваться:

* поиск вариантов со стороны «базы вакансий» и со стороны «базы безработных»;
* регистрация со стороны «базы вакансий» и со стороны «базы безработных»;
* удаление в архив после трудоустройства;
* полное удаление при отказе от услуг.

Необходимо предусмотреть возможность формирования объявлений для печати и отчета о работе Биржи труда (общее количество безработных, сколько безработных нашли работу, общее количество вакансий, какие вакансии наиболее «популярны»).

**Вариант 8**

Спроектировать базу данных для коллекционера марок, собирающего тематическую   коллекцию. В БД должны храниться сведения о марках, имеющихся в коллекции, и сведения об их положении в коллекции.

Сведения о марках должны включать в себя: номер марки, страну выпуска марки, номер серии, тему серии, год выпуска, цвет марки, размер марки, цену марки, тему марки. Марки расположены в коллекции по темам. Сведения о положении марки в коллекции включают в себя: номер раздела коллекции (разбит на тома по темам и странам), в котором хранится марка, номер тема, номер страницы и уникальное положение марки на странице. Возможно открытие новых или закрытие старых тем. Место расположения марки может измениться. Коллекционеру могут потребоваться следующие сведения;

* марки каких стран содержатся в данном разделе;
* в каком томе коллекции находится марка определенной серии;
* в каких местах коллекции находятся марки указанной темы;
* какие темы у серий, включающих марки определенного размера;
* марка какой страны находится в данном месте.

Коллекционер может вносить следующие изменения:

* добавление марки новой темы;
* удаление всех марок одной темы;
* изменение места расположения марки в коллекции.

Необходимо предусмотреть возможность выдачи справки о странах, чьи марки находятся в данной теме, и отчета по коллекции (количество и названия тем и стран по разделам, количество марок каждой страны для каждой темы, количество страниц в коллекции).

**Вариант 9**

Спроектировать базу данных для работников управления торговли. В БД хранятся сведения о магазинах города; товарах, имеющихся в магазинах; о торговых базах и товарах, хранящихся на базах. Каждый магазин закреплен за одной торговой базой.  Магазин характеризуется классом, номером и имеет несколько отделов. Каждый товар в каждом магазине продается, по крайней мере, в одном отделе. Каждый отдел имеет одного заведующего отделом. Товары, имеющиеся в магазине и хранящиеся на базах, характеризуются ценой, сортом и количеством. Розничные цены в магазине зависят от класса магазина и сорта товара и могут изменяться. Магазин может открыть новый отдел или закрыть старый. В этом случае товар передается в другие отделы.

При работе с БД могут потребоваться следующие сведения:

* какие товары имеются в магазине (на базе);
* какие отсутствующие товары может заказать магазин на базе;
* какие товары и в каком количестве имеются в отделе магазина;
* список заведующих отделом магазина;
* в каких отделах магазина продается одинаковый товар.
* закупка нового товара;
* закрытие отдела в магазине;
* изменение цены товара.

Необходимо предусмотреть возможность выдачи справки о наличии товаров в отделе магазина и отчета по магазину (количество и наименование товаров в отделах, ФИО заведующих отделами, номер базы, за которой закреплен магазин).

**Вариант 10**

Спроектировать базу данных для диспетчера автобусного парка. В БД хранятся сведения о водителях, маршрутах автобусов и характеристиках автобусов.

Каждый водитель характеризуется: ФИО, классом, стажем работы и окладом, причем оклад зависит от его класса и стажа работы. Маршрут автобуса характеризуется: номером маршрута, временем начала и конца движения, интервалом движения и протяженностью. Характеристиками автобусов являются: номер автобуса, его тип и вместимость, причем вместимость автобуса зависит от его типа (каждый водитель закреплен за отдельным автобусом, а каждый автобус прикреплен к определенному маршруту. Необходимо предусмотреть возможность корректировки БД в случаях поступления на работу нового водителя, списывания старого автобуса, введения нового или изменения старого маршрута и т.п.

Диспетчеру автопарка могут потребоваться следующие сведения:

* список водителей, работающих на определенном маршруте,
* какие номера автобусов обслуживают данный маршрут,
* когда начинается или заканчивается движение автобусов на всех или отдельных маршрутах,
* какова протяженность всех или определенных маршрутов автобусов,
* на каких автобусах работает водитель.

Диспетчер может вносить следующие изменения:

* прием на работу нового водителя
* списание старого автобуса;
* изменение протяженности маршрута.

Необходимо предусмотреть возможность выдачи справки о протяженности маршрута и отчета по автопарку (количество автобусов и их тип, номера маршрутов, время начала движения и интервал, ФИО водителей и их класс).

**Вариант 11**

**Спроектировать базу данных «Аптека».** В базе необходимо хранить разнообразную информацию об аптеках, препаратах и изготовителях данных препаратов, чтобы оперативно можно было определить информацию о принадлежности того или иного препарата к определенному изготовителю и определить аптеку, где можно приобрести данный препарат, а также список лекарств-заменителей, которые можно рекомендовать покупателям при отсутствии основного лекарства.

Аптекарю может потребоваться:

* систематизировать базу по препаратам;
* распределять препараты по аптекам;
* добавлять новые препараты и удалять просроченные;
* вести учет лекарств, отпускаемых строго по рецепту;
* изменять стоимость препаратов.

Покупателю может потребоваться:

* просматривать информацию о препаратах;
* получать информацию об аптеках, в которых данный препарат можно приобрести.

Необходимо предусмотреть возможность выдачи справки о наличии лекарств на конец дня и отчета о количестве и стоимости реализованного товара.

**Вариант 12**

Спроектировать базу данных для работников справочной службы кинотеатров города.

В БД должны храниться сведения о кинотеатрах: название, район города, где расположен кинотеатр, категория (может быть изменена), вместимость (определяется категорией); о фильмах: название, производство, режиссер, жанр; кроме того, должна храниться информация о цене билета, количестве свободных мест на данный сеанс. На разных сеансах в одном кинотеатре могут идти разные фильмы. Кинотеатр может ввести новый фильм в репертуар или снять старый с проката. Цена билета определяется прокатной стоимостью (названием) фильма и категорией кинотеатра.

Справочной службе могут потребоваться следующие сведения о текущем состоянии проката фильмов в городе:

* репертуар кинотеатра (по названию кинотеатра);
* адрес и район кинотеатра (по названию кинотеатра);
* число мест (свободных) на данный сеанс (название кинотеатра и сеанс);
* цена билетов на данный сеанс (название кинотеатра и сеанс),
* жанр, производство и режиссер данного фильма (по названию);
* вместимость данного кинотеатра (по названию, кинотеатра).
* открытие нового кинотеатра;
* снятие фильма с проката:
* изменение репертуара, кинотеатра.

Необходимо предусмотреть возможность выдачи справки о сеансе фильма в указанном кинотеатре и отчета о прокате фильмов в районах города (названия фильмов, в каких кинотеатрах они демонстрировались, цена билета в каждом кинотеатре на них).

**Вариант 13**

Спроектировать базу данных Касса автовокзала. В БД должны храниться сведения о расписании автобусов: номер рейса, конечный и промежуточный пункты, время отправления; о количестве свободных мест на каждом рейсе. Пассажир может изменить дату и время отъезда.

Кассиру могут потребоваться следующие сведения:

* выбор ближайшего рейса до заданного пункта (при наличии свободных мест);
* вывод списка рейсов в один и тот же город с указанием времени пути и стоимости билета;
* выдача информации о наличии свободных билетов на рейс.

Кассир может производить:

* предварительную продажу, возврат билетов.

Необходимо предусмотреть возможность выдачи электронного билета и оформление посадочной ведомости.

**Вариант 14**

Спроектировать базу данных для работников регистратуры поликлиники.

В БД должны храниться сведения о больных: ФИО, адрес, диагноз (может быть уточнен), дата заболевания; сведения о врачах: ФИО, номер кабинета, номер участка, дни и часы приема, описание болезней: название (диагноз), симптомы, лекарство, возможно появление новых больных. Врач может уволиться из поликлиники.

Работникам регистратуры могут потребоваться следующие сведения:

* адрес, дата заболевания, диагноз данного больного;
* ФИО лечащего врача данного больного;
* номер кабинета, дни и часы приема данного врача;
* больные, находящиеся на лечении у данного врача;
* симптомы данного заболевания и рекомендуемое лекарство.

Администратор БД может вносить следующие изменения:

* появление нового больного;
* увольнение врача;
* изменение диагноза.

Необходимо предусмотреть возможность выдачи справки о болезни некоторого больного и отчета о работе поликлиники (количество больных, ФИО каждого врача и число лечащихся у него больных, количество заболеваний по каждому виду болезни, расписание работы врачей поликлиники).

**Вариант 15**

Спроектировать базу данных для диспетчера станции техобслуживания.

В БД должны храниться сведения о владельцах автомобилей: ФИО, адрес, марка автомобиля, номер госрегистрации характеристики автомобилей: год выпуска, изготовитель, перечень устраненных неисправностей; ФИО работника станции и время устранения каждой неисправности. Возможно введение в БД сведений о новых владельцах и новых неисправностях.

Диспетчеру могут потребоваться следующие сведения:

* ФИО и адрес владельца автомобиля с данным номером госрегистрации;
* изготовитель, марка и год выпуска автомобиля данного владельца;
* перечень устраненных неисправностей в автомобиле данного владельца;
* ФИО работника станции, устранявшего данную неисправность автомобиля данного владельца, и время устранения;
* какие автомобили ремонтировал данный работник станции;
* ФИО владельцев автомобилей с указанным типом неисправности.

Диспетчер может вносить следующие изменения:

* добавить информацию о владельце ремонтируемого автомобиля;
* удалить информацию о работнике станции;
* изменить номер госрегистрации автомобиля.

Необходимо предусмотреть возможность выдачи справки о наличии неисправности автомобиля некоторого владельца и отчета о работе станции техобслуживания (количество ремонтируемых автомобилей, время ремонта каждого автомобиля, ФИО работника, который их ремонтировал, список неисправностей для каждой марки автомобиля).

**Вариант 16**

Спроектировать базу данных для менеджера музыкальных групп. В БД должны храниться сведения о группах: название, год образования, страна, состав исполнителей, положение в последнем хит-параде (может изменяться); репертуаре каждой группы: названия песен, композитор, автор текста; данные о последних гастролях группы: название гастрольной программы, место, дата начала и окончания гастролей, цена билета (зависят от места гастролей и положения в хит-параде). Возможно появление новой группы и изменение состава исполнителей. Каждая песня может быть в репертуаре только одной группы.

Менеджеру могут потребоваться следующие сведения: год образования и страна группы данного названия;

* репертуар наиболее популярной группы;
* автор текста, композитор и дата создания песни с данным названием;
* место и продолжительность гастролей группы с данным названием;
* цена билета на концерт указанной группы;
* состав исполнителей группы данного названия, их возраст и амплуа;
* ввод новой группы;
* изменение положения группы в хит-параде;
* удаление информации об исполнителе, покинувшем группу.

Необходимо предусмотреть возможность выдачи справки о лучших трех группах в хит-параде и отчета о гастролях групп (название группы, место и сроки гастролей, репертуар с указанием авторов песен).

**Вариант 17**

Спроектировать базу данных для работников гостиницы. В БД должны храниться сведения о проживающих клиентах, служащих гостиницы, убирающих в номерах.  Имеются номера трех типов: одноместный, двухместный и трехместный, отличающиеся стоимостью проживания в сутки. Количество номеров в гостинице известно. О каждом проживающем должна храниться следующая информация: номер паспорта, ФИО, город, из которого он прибыл, дата поселения гостинице, выделенный гостиничный номер, на сколько дней выделен номер. Каждый номер характеризуется типом, стоимостью проживания, номером телефона. Номера упорядочены по этажам. О служащем гостиницы должна храниться следующая информация: ФИО, номер этажа, где он убирает, день недели, когда он убирает данный этаж. Служащий гостиницы убирает все номера на одном этаже в определенные дни недели.

Работа с БД предполагает обслуживание следующих запросов:

* получение списка фамилий, проживающих в заданном номере;
* вычисление счета за проживание в гостинице;
* определение количества свободных мест и свободных номеров;
* получение списка прибывших из заданного города;
* установление ФИО служащего, убиравшего номер заданный день недели у некоторого клиента;
* освобождение номера проживающим;
* изменение расписания уборки для служащего в указанный день недели;
* увольнение служащего гостиницы.

Необходимо предусмотреть возможность выдачи справки о счете за проживание в гостинице определенного клиента и отчета о работе гостиницы за последний квартал (число клиентов, сколько дней был занят и свободен каждый номер, сумма дохода гостиницы).

Необходимо предусмотреть возможность выдачи справки о счете за проживание в гостинице определенного клиента и отчета о работе гостиницы за последний квартал (число клиентов, сколько дней был занят и свободен каждый номер, сумма дохода гостиницы).

ПРИЛОЖЕНИЕ Б «Титульный лист»

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра вычислительной техники

Курсовая работа по дисциплине

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

на тему: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дисциплина и тема КР пишутся заглавными буквами

Студент группы \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

(Ф.И.О.) (подпись, дата)

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

(Ф.И.О., должность) (подпись, дата)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тула, 20\_\_ г.