Министерство образования Московской области

Государственное бюджетное профессиональное

образовательное учреждение Московской области

Воскресенский колледж

**Методические указания и контрольные задания**

**по дисциплине «Технология разработки и защиты баз данных»**

для обучающихся по заочной форме

по специальности **09.02.03 Программирование в компьютерных системах**

**Воскресенск, 2015 г.**

Разработчик(и):

ГБПОУ СПО МО «Воскресенский колледж» преподаватель спецдисциплин Комиссаров С.А.

(место работы) (занимаемая должность) (инициалы)

Одобрено на заседании предметно-цикловой комиссии

компьютерных дисциплин

Протокол №\_\_\_\_\_\_\_\_от «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

Председатель ПЦК\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

**Пояснительная записка**

Методические указания для студентов заочного отделения по выполнению практических заданий по дисциплине «Технология разработки и защиты баз данных» предназначены для реализации ФГОС по специальности 09.02.03 «Программирование в компьютерных системах».

«Методы проектирования программного обеспечения» входит в профессиональный модуль ПМ02 «Разработка и администрирование баз данных».

Обучающийся в ходе освоения учебной дисциплины должен:

***Уметь:***

-создавать объекты баз данных в современных СУБД и управлять доступом к этим объектам;

- работать с современными Case-средствами проектирования баз данных;

- формировать и настраивать схему базы данных;

- разрабатывать прикладные программы с использованием языка SQL;

- создавать хранимые процедуры и триггеры на базах данных;

- применять стандартные методы для защиты объектов базы данных

***Знать:***

- основные положения теории баз данных, хранилищ данных, баз знаний;

- основные принципы построения концептуальной, логической и физической модели данных;

- современные инструментальные средства разработки схемы базы данных;

- методы описания схем баз данных в современных СУБД;

- структуры данных СУБД, общий подход к организации представлений, таблиц, индексов и кластеров;

- методы организации целостности данных;

- способы контроля доступа к данным и управления привилегиями;

- основные методы и средства защиты данных в базах данных;

- основы разработки приложений баз данных.

Каждая контрольная работа разделена на 10 вариантов. Вариант работы определяется по последней цифре индивидуального шифра.

Перед выполнением контрольной работы необходимо изучить теоретический материал по данной теме, который достаточно полно представлен в списке литературы. Список разделов учебной дисциплины для изучения приведен после примера выполнения задания.

В процессе выполнения работы рекомендуется тщательно анализировать имеющиеся в учебниках иллюстрации, схемы, диаграммы и таблицы.

Необходимо обязательно выяснять значение встречающихся в тексте незнакомых или непонятных слов и терминов.

Также при подготовке к выполнению контрольной работы нужно пользоваться источниками, доступными в сети интернет. К ее выполнению следует приступать после того, как вы освоите достаточное количество материала, чтобы выполнить все задания по возможности одновременно.

**Контрольная работа** включает изучение разделов «Общие сведения о программных продуктах», «Технологии разработки программных продуктов» и «Современные технологии программирования».

**Требования к оформлению работы.**

Работа выполняется на листах формата А4 в печатном виде. Шрифт – Times New Roman, размер шрифта – 12, междустрочный интервал – полуторный. Размеры полей: слева – 2 см (под подшивку), остальные – по 1 см (сверху, снизу, справа).

В начале работы, после титульного листа, указывается содержание работы, в конце – список использованной литературы. При оформлении любого задания необходимо сначала указать вопрос.

**Пример правильного ответа**.

**Вопрос: Первая нормальная форма.**

**Ответ.**

**Первая нормальная форма (1NF)**

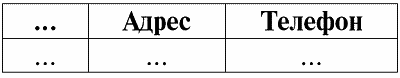
На ранних стадиях проектирования баз данных и разработки схем их управления использовались простые и однозначные атрибуты как наиболее продуктивные и рациональные единицы кода. Тогда применяли наряду с простыми и составные атрибуты, а также наряду с однозначными и многозначные атрибуты. Поясним значения каждого из этих понятий.

**Составные атрибуты,** в отличие от простых, – это атрибуты, составленные из нескольких простых атрибутов.

**Многозначные атрибуты,** в отличие от однозначных, – это атрибуты, представляющие множество значений.

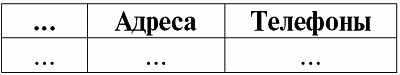
Приведем примеры простых, составных, однозначных и многозначных атрибутов.

Рассмотрим следующую таблицу, представляющую отношение:



Здесь атрибут «Телефон» – простой, однозначный, а атрибут «Адрес» – простой, но многозначный.

Теперь рассмотрим другую таблицу, с другими атрибутами:



В этом отношении, представленном таблицей, атрибут «Телефоны» – простой, но многозначный, а атрибут «Адреса» – и составной, и многозначный.

Вообще возможны различные комбинации простых или составных атрибутов. В разных случаях таблицы, представляющие отношения, могут выглядеть следующим общим образом:



При нормализации схем базовых отношений программистами может быть использована одна из четырех наиболее распространенных видов нормальных форм: первая нормальная форма (1NF), вторая нормальная форма (2NF), третья нормальная форма (3NF) или нормальная форма Бойса – Кодда (NFBC). Поясним: сокращение NF – это аббревиатура от англоязычного словосочетания Normal Form. Формально, кроме вышеназванных, существуют и другие виды нормальных форм, но вышеназванные – одни из самых востребованных.

В настоящее время разработчики баз данных стараются избегать составных и многозначных атрибутов, чтобы не усложнять написание кода, не перегружать его структуру и не запутывать пользователей. Из этих соображений логически и вытекает определение первой нормальной формы.

**Определение**. Любое базовое отношение находится в **первой нормальной форме** тогда и только тогда, когда схема этого отношения содержит только простые и только однозначные атрибуты, причем обязательно с одной и той же семантикой.

Для наглядного объяснения различий нормализованных и ненормализованных отношений рассмотрим пример.

Пусть, имеется ненормализованное отношение, со следующей схемой.

Итак, **вариант 1** схемы отношения с заданным на ней простым первичным ключом:

*Сотрудники (****№ табельный****, Фамилия Имя Отчество, Код должности, Телефоны, Дата приема или увольнения);*

*Primary key (№ табельный);*

Перечислим, какие в этой схеме отношения имеются ошибки, т. е. назовем те признаки, которые и делают собственно эту схему ненормализованной:

1) атрибут «Фамилия Имя Отчество» является составным, т. е. составленным из разнородных элементов;

2) атрибут «Телефоны» является многозначным, т. е. его значением является множество значений;

3) атрибут «Дата приема или увольнения» не имеет однозначной семантики, т. е. в последнем случае не понятно, какая именно дата внесена.

Если, например, ввести дополнительный атрибут, чтобы поточнее определить смысл даты, то для этого атрибута значение будет семантически понятно, но тем не менее остается возможность хранения только какой-то одной из указанных дат для каждого сотрудника.

Что необходимо сделать для приведения этого отношения к нормальной форме?

Во-первых, необходимо провести разбиение составных атрибутов на простые, для того, чтобы исключить эти самые составные атрибуты, а также атрибуты с составной семантикой.

Во-вторых, провести декомпозицию этого отношения, т. е. нужно разбить его на несколько новых самостоятельных отношений, с тем чтобы исключить многозначные атрибуты.

Таким образом, с учетом всего вышесказанного после приведения отношения «Сотрудники» к первой нормальной форме или 1NF путем его декомпозиции мы получим систему следующих отношений с заданными на них первичными и внешними ключами.

Итак, **вариант 2** отношения:

*Сотрудники (****№ табельный****, Фамилия, Имя, Отчество, Код должности, Дата приема, Дата увольнения);*

*Primary key (№ табельный);*

*Телефоны (****№ табельный****,****Телефон****);*

*Primary key (№ табельный, Телефон);*

*Foreign key (№ табельный) references Сотрудники (№ табельный);*

Составного атрибута «Фамилия Имя Отчество» больше в нашем отношении нет, вместо него присутствуют три простых атрибута «Фамилия», «Имя» и «Отчество», поэтому эта причина «ненормальности» отношения исключилась.

Кроме того, вместо атрибута с неясной семантикой «Дата приема или увольнения» у нас появилось два атрибута «Дата приема» и «Дата увольнения», каждый из которых имеет однозначную семантику. Следовательно, вторая причина того, что наше отношение «Сотрудники» не находится в нормальной форме, также благополучно устранена.

И, наконец, последняя причина того, что отношение «Сотрудники» не было приведено к нормальной форме, – это наличие многозначного атрибута «Телефоны». Чтобы избавиться от этого атрибута, и необходимо было провести декомпозицию всего отношения. Из исходного отношения «Сотрудники» в результате этой декомпозиции был исключен атрибут «Телефоны» вообще, но зато образовалось второе отношение – «Телефоны», в котором присутствуют два атрибута: «№ табельный» сотрудника и «Телефон», т. е. все атрибуты – опять-таки простые, условие принадлежности к первой нормальной форме выполняется. Эти атрибуты «№ табельный» и «Телефон» образуют составной первичный ключ отношения «Телефоны», а атрибут «№ табельный», в свою очередь, является внешним ключом, ссылающимся на одноименный атрибут отношения «Сотрудники», т. е. в отношении «Телефоны» атрибут первичного ключа «№ табельный» является одновременно внешним ключом, ссылающимся на первичный ключ отношения «Сотрудники». Таким образом, обеспечивается связь между этими двумя отношениями. Посредством этой связи можно по номеру табельному любого сотрудника без особого труда и затрат времени вывести весь список его телефонов, не прибегая к использованию составных атрибутов.

Заметим, что в случае наличия в отношении системы ограничений функциональных зависимостей после всех вышеприведенных преобразований нормализация не была бы завершена. Однако в данном конкретном примере нет ограничений функциональных зависимостей, поэтому дальнейшая нормализация этого отношения не требуется.

**ВАРИАНТ №1**

1. Семантическая модель проектирования БД. *В ответ включить определение модели, ее главные аспекты, достоинства и недостатки*.
2. Нормальные формы. *В ответ включить определение нормальных форм, значение нормализации, последствия отсутствия нормализации форм, разновидности нормальных форм.*
3. Типы данных языка SQL. *В ответ включить подробное описание типов данных языка и примеры их использования.*

**ВАРИАНТ №2**

1. Структурированный язык запросов SQL. *В ответ включить определение языка, его назначение, достоинства и недостатки, краткую историю развития и его основные функции*.
2. Вторая нормальная форма. *В ответе необходимо подробно охарактеризовать вторую НФ с указанием методик приведения и анализом необходимости такого приведения.*
3. Даталогические модели БД. *В ответе подробно описать даталогическую модель БД, ее назначение и место в проектировании БД.*

**ВАРИАНТ №3**

1. Физические модели БД. *В ответе подробно описать физическую модель БД, ее назначение и место в проектировании БД*.
2. Теория нормальных форм: функциональные и многозначные зависимости. *В ответе необходимо подробно описать вышеуказанные зависимости и их место в обеспечении целостности БД.*
3. Основные операторы языка SQL. *В ответ включить краткое описание самых распространенных операторов языка с примерами использования.*

**ВАРИАНТ №4**

1. Работа с таблицами в SQL. *В ответе описать соответствующие операторы с примерами и пояснениями*.
2. Процедура нормализации таблиц. *В ответе необходимо описать технологию приведения таблиц к одной из нормальных форм.*
3. ER-диаграммы. *В ответе необходимо дать определение ER-диаграмм, привести их примеры с пояснениями, описать значение и место ER-диаграмм в проектировании БД.*

**ВАРИАНТ №5**

1. Запросы на выборку в SQL. *В ответ включить инструкции языка SQL с описанием и примерами запросов с пояснениями.*
2. Особенности реляционных БД. *В ответе необходимо дать определение реляционных БД, указать их отличия от других моделей БД, описать достоинства и недостатки.*
3. Индексирование в БД. *В ответе подробно пояснить необходимость в индексации таблиц, описать преимущества и недостатки.*

**ВАРИАНТ №6**

1. Типы запросов в SQL. *В ответ включить подробное описание различных типов запросов с примерами*. *Примеры снабдить пояснениями*.
2. Технология клиент-сервер. *В ответе необходимо подробно объяснить суть технологии, ее достоинства и недостатки, а также разместить упрощенную схему организации такой технологии на практике.*
3. Стандарт ODBC. *В ответе подробно описать область применения стандарта, его суть, указать аналоги*.

**ВАРИАНТ №7**

1. Запросы на изменение данных в SQL. *В ответ включить инструкции языка SQL с описанием и примерами запросов с пояснениями.*
2. Технология обмена данными с использованием сервера приложений. *В ответе изложить особенности этой технологии, ее главные преимущества и недостатки, указать значение в современной сфере проектирования БД и работы с ними, а также разместить упрощенную схему организации такой технологии на практике.*
3. Понятие транзакции. *В ответе необходимо дать развернутое определение транзакции, физический смысл блокировки записи, привести пример на языке SQL с подробным описанием действий.*

**ВАРИАНТ №8**

1. Технологии COM, ADO, MTS. *В ответе подробно описать область применения технологий, их суть, указать аналоги*.
2. Технология распределенных БД. *В ответе дать подробное определение распределенных БД, их отличия от локальных, особенности работы с ними, описание технологических решений для работы с ними и примеры таких решений.*
3. SQL: вычисления и групповые операции в запросах. *В ответ включить инструкции языка SQL с описанием и примерами запросов с пояснениями*.

**ВАРИАНТ №9**

1. Технология репликации БД. *В ответ включить определение репликации БД, ее физический смысл, а также подробно описать типы репликации и применяемые для этого технологии. Также необходимо включить краткое схематическое пояснение различных технологий репликации.*
2. Стандарт Microsoft OLE DB. *В ответе подробно описать область применения стандарта, его суть, указать аналоги.*
3. SQL: операторы для работы с индексами. *В ответ включить инструкции языка SQL с описанием и примерами запросов с пояснениями*.

**ВАРИАНТ №10**

1. Проектирование межтабличных связей. *В ответ включить понятие ключевых полей, подробно описать возможные типы связей и привести примеры их использования.*
2. Основные компоненты реляционных БД. *В ответе подробно описать основные компоненты, дать их развернутые определения, указать связи между ними.*
3. Связь с источниками данных в SQL. *В ответ включить инструкции языка SQL с описанием и примерами запросов с пояснениями*.

**Литература:**

1. Федорова Г.Н. Основы проектирования баз данных: учеб. пособие для студ. учреждений сред. проф. образования. – М.: Издательский дом «Академия», 2014. – 224 с.

2. Фуфаев Э.В. Разработка и эксплуатация удаленных баз данных: учебник для студ. сред. проф. образования. – 3-е изд., стер. – М.: Издательский центр «Академия», 2010. – 256 с.