

**Министерство образования и науки Российской Федерации
Федеральное государственное учреждение высшего образования
Московский государственный университет технологий и управления
(первый казачий университет)
Университетский колледж информационных технологий**

**Специальность 09.02.03 Программирование в компьютерных системах
ПМ02 Разработка и администрирование баз данных
Сборник практических заданий
для проведения
УПО2.01 учебной практики**

Содержание

1 Введение	3
2 Практические задания учебной практики	5
3 Дополнительные требования	8
4 Параметры, учитываемые в оценке	8
5 Отчет	9
5.1 Структура отчета	9
5.2 Титульный лист	9
5.3 Оформление отчета	9
5.3.1 Требования машинописи	10
5.4 Введение	10
5.5 Основная часть	10
5.5.1 Разработка объектов базы данных;	10
5.5.2 Реализовывать базы данных в конкретной СУБД	10
5.5.3 Решать вопросы администрирования базы данных и реализовывать методы и технологии защиты информации в базах данных	10
5.6 Заключение	10
6 Список литературы	11
Приложение А. Титульный лист	12

1 Введение

«Учебная практика направлена на формирование у обучающихся умений, приобретение первоначального практического опыта и реализуется в рамках профессиональных модулей ОПОП СПО по основным видам профессиональной деятельности для последующего освоения ими общих и профессиональных компетенций по избранной специальности¹».

Данная учебная практика входит в профессиональный модуль «ПМ2 Разработка и администрирование баз данных» и направлена на выполнение следующих требований к студенту²:

- иметь практический опыт:
 - работы с объектами базы данных в конкретной системе управления базами данных;
 - использования средств заполнения баз данных;
 - использования стандартных методов защиты базы данных;
- 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 баз 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 баз 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;
 - модели и структуры информационных систем;
 - основные типы сетевых технологий, приемы работы в компьютерных сетях;
 - информационные ресурсы компьютерных сетей;
 - технологии передачи и обмена данными в компьютерных сетях;
 - основы разработки приложений баз данных

Результатом освоения профессионального модуля должно быть формирование следующих компетенций:

- разрабатывать объекты базы данных;
- реализовывать базы данных в конкретной СУБД;
- решать вопросы администрирования базы данных;
- реализовывать методы и технологии защиты информации в базах данных.

Кроме того, изучение профессионального модуля должно способствовать формированию следующих общих компетенций:

¹ст. 6 положения о практике обучающихся, осваивающих основные профессиональные образовательные программы среднего профессионального образования, утвержденного приказом Министерства образования и науки Российской Федерации (Минобрнауки России) от 18 апреля 2013 г. N 291

²Федеральный государственный стандарт среднего специального образования по специальности 09.02.03 Программирование в компьютерных системах, утвержденный приказом Министерства образования и науки Российской Федерации от 28 июля 2014 г. №804

- понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес;
- организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество;
- принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность;
- осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития;
- использование информационно-коммуникационных технологий в профессиональной деятельности;
- работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями;
- брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий;
- самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации;
- ориентироваться в условиях частой смены технологий в профессиональной деятельности;

Автор благодарит студентов группы П-303 Строкова Сергея, Чермошенцева Александра, Брунса Владислава и студента группы П-304 Орищенко Андрея за вклад в формирование списка советов по выполнению заданий. В новой версии расширено количество заданий с учетом увеличения количества часов.

2 Практические задания учебной практики

При прохождении учебной практики осуществляется с работой с той же предметной областью и (в целом) с той же базой данных, что разрабатывалась в учебной практике ПМ.01. При этом надо задание дополнить ещё двумя таблицами.

Работа может осуществляться в любой СУБД из следующего списка (при условии выполнения условий лицензионных соглашений):

- Microsoft SQL Server
- MariaDB
- PostgreSQL

Среда разработки и язык программирования осуществляется студентом самостоятельно исходя из возможности разработки оконных приложений или web-приложений и возможности организации работы с выбранной СУБД.

Следует учесть, что на компьютерах вычислительного центра установлены только Microsoft SQL Server.

При разработке обязательно необходимо использовать систему контроля версий и базу данных дефектов (например, на базе сайта bitbucket.org).

Ниже приведены задания, сгруппированные по двум парам. Предполагается, что за семестр пройдёт 54 пары, из них пары, посвящённые выполнению заданий – 52.

1. Осуществите выбор используемых инструментальных средств. Подробно изучите команду `create table` (включая информацию об индексах, ограничении целостности). Изучение осуществляется на базе курса <http://www.intuit.ru/studies/courses/5/5/info> (лекции 2-3, 8-9)³.

Письменно (от руки) напишите конспект изученного материала. Вместо конспекта можно представить копии экрана, свидетельствующие отработку всего материала.

В случае закрепления темы до окончания первой недели и соответствующего желания, можно выполнять задания начиная с 3 недели и соответственно до 19-ой.

2. Осуществите изучение представлений, назначения и методов создания хранимых процедур, работу с транзакциями (лекции 10; 16; 11-12).

Хранимые процедуры изучите также по лекции <http://www.intuit.ru/studies/courses/68/68/lecture/2024>

Письменно (от руки) напишите конспект изученного материала. Вместо конспекта можно представить копии экрана, свидетельствующие отработку всего материала.

3. Осуществите разработку схемы базы данных с учетом правил нормализации. Обоснуйте и явно выберите правила ограничений целостности. Разработку схемы базы данных необходимо осуществить в специализированном инструментальном средстве. Соответствующую диаграмму (**логическую модель**, не физическую) необходимо будет включить в отчет.

Отчет включает следующие элементы содержания:

- диаграмма логической модели
- SQL-запросы создания таблиц
- обоснование использования (с указанием конкретных характеристик)/не использования следующих конструкций: PRIMARY KEY, UNIQUE, CHECK, FOREIGN KEY, ON UPDATE, ON DELETE, CONSTRAINT, DEFAULT, NULL, NOT NULL, DEFAULT, IDENTITY или ROWGUIDCOL (или их аналоги).

Для mysql следует формально указать конструкции с ключевым словом CHECK, но потом осуществить собственно реализацию с помощью триггеров.

Диаграмму можно строить в следующих программах:

- Toad Data Modeler Freeware Edition (только под Windows)
- CA ERwin® Data Modeler Community Edition (конвертация до 25 объектов, только под Windows)

³Если вы используете СУБД, отличную от MSSQL, вы можете изучать курсы, соответствующие выбранной вами СУБД

4. Прочитайте лекцию №17 курса <http://www.intuit.ru/studies/courses/5/5/info>.

Разработайте UML-диаграмму вариантов использования вашего приложения. Дополните схему базы данных представлениями в соответствии с диаграммой, осуществите настройку прав доступа к представлениям и таблицам.

Для аутентификации от имени пользователя mssql необходимо в настройках СУБД разрешить аутентификацию средствами MSSQL (а не только Windows, как это установлено по умолчанию). Кроме того, пользователю необходимо назначить разрешение «Разрешение на подключение к ядру СУБД» (такое разрешение также можно задать с помощью запроса GRANT: это действие называется CONNECT).

5. Осуществите разработку последовательностей SQL-команд для реализации основных функций приложения, включая их, при необходимости, в транзакции.

6. Осуществите разработку хранимых процедур так, чтобы в будущем приложении использовались бы только они и представления

Обратите внимание на обработку ситуаций, когда часть запросов выполнялась, а часть нет. В MSSQL существует много различных способов сделать данную обработку. Проще всего выполнить запрос

```
SET XACT_ABORT, COUNT ON
```

который обозначает, что выполнение процедуры следует прекратить после первой ошибки и откатить транзакцию, кроме того, оно подавляет вывод сообщений из успешно обработанной, но в будущем откатанной части транзакции.

В MySQL следует в начале хранимой процедуры указать текст следующего содержания

```
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
    ROLLBACK;
END;
```

7. Изучите методику организации взаимодействия с СУБД в выбранной вами среде (платформе).

- В Lazarus или Delphi следует изучить: T*Connection, где * обозначает конкретный драйвер (ODBC, ADO, MYSQL56, MSSQL и т. п.).

Для работы достаточно изучить дополнительно класс TSqlQuery⁴ (в части ExecSQL, Open, EOF, Next, Last, isEmpty, getField, Fields), а также свойства As* класса TField).

*Можно также изучить классы T*StoredProc с целью использования более специализированных запросов на вызов хранимых процедур (отметим, что если используется один объект этого типа для запуска различных процедур, то необходимо вызывать метод Refresh после смены названия процедуры).*

По желанию, можно также изучить работу с TDataSource и с компонентами вкладки Data Controls (это позволит снизить количество самописного кода, однако этот прием очень специфичен для Delphi). Этому вопросу посвящен курс <http://www.intuit.ru/studies/courses/614/470/info> (лекции 2-7, 9-13)

Для осуществления соединения программы, написанной на Lazarus с MSSQL, можно осуществить следующие действия

- (a) скачать и записать в папку с Lazarus и в папку с проектом dblib (<ftp://ftp.freepascal.org/fpc/contrib/windows/>) (при этом желательно установить MS Visual Studio, так как dblib требует дополнительные библиотеки);
- (b) в диспетчере конфигурации MSSQL указать возможность доступа по протоколу TCP/IP, а также указать, что доступ осуществляется с использованием порта 1433.

- При работе с .net изучите ADO.net
- При работе с php изучите PDO

Осуществите разработку приложения, использующего вашу базу данных.

Обратите внимание на обработку ошибочных ситуаций (когда хранимая процедура выполняется неуспешно) /перехват EDatabaseError с помощью try except для Lazarus/Delphi/

Минимальный работоспособный код, предназначенный для получения информации из базы данных при использовании Lazarus/Delphi выглядит следующим образом:

⁴TQuery в Delphi

```

try
  SqlQuery1.SQL.Add('SELECT name FROM test where length(name)=:len');
  SqlQuery1.Params.ParamByName('len').AsInteger:=3;
  SqlQuery1.Open;
  while not SqlQuery1.EOF do
  begin
    Listview1.AddItem(SqlQuery1.FieldByName('name').AsString,NIL);
    SqlQuery1.Next;
  end;
  SqlQuery1.Close;
  SqlQuery1.SQL.Clear;

except on d: EDatabaseError do
  begin
    Application.MessageBox(PChar(d.Message),'');
  end;
end;

```

Для его работы необходимо с помощью инспектора объектов (или иным путём) осуществить настройку соединения с СУБД, соединить объект `SqlQuery1` с объектом соединения и объектом транзакции.

8. Осуществите доработку приложения, использующего вашу базу данных
9. Осуществите разработку отчётов, позволяющих отпечатать полное содержимое базы данных. Отчёт необходимо выполнить с помощью специализированного программного средства (указаны примеры):
 - LazReport (http://wiki.freepascal.org/LazReport_Tutorial) для Lazarus
 - Rave Reports для Delphi 7
 - Birt Project для Java
 - Reportico для PHP (задание можно просрочить)
10. Осуществите разработку отчёта, позволяющих отпечатать статистические сведения по хранимой в вашей базе данных информации.
11. Для каждого сложного запроса (с подзапросами или JOIN) создайте запрос с аналогичной функциональностью, а также осуществите исследование, какая версия запроса более оптимальна. Для этого создайте небольшую программу, которая будет выполнять такие запросы в большом количестве и даёт возможность сравнить время выполнения запросов.
12. Реализуйте в приложении возможность аутентификации и возможность работы в режиме администратора. В этом режиме дайте возможность регистрации новых пользователей, включения их в ту или иную категорию, удаления пользователей. Дополните соответственно диаграмму вариантов использования.
Добавление нового пользователя обозначает (для MSSQL) выполнения запросов `CREATE LOGIN`, `CREATE USER`, `GRANT CONNECT`, `sp_addrolemember` (приведён один из возможных вариантов).
13. Реализуйте в приложении возможность работы в режиме администратора. В этом режиме дайте возможность регистрации новых пользователей, включения их в ту или иную категорию, удаления пользователей.
14. Осуществите резервное копирование базы данных. Осуществите автоматизацию этого процесса.
На собственном компьютере это можно осуществить посредством MS SQL агента (<https://msdn.microsoft.com/ru-ru/library/ms191002%28v=sql.120%29.aspx>), на компьютерах колледжа путем явного запуска команды `CREATE BACKUP` из специально написанного приложения; это приложение следует включить в планировщик заданий.
Резервные копии должны создаваться ежедневно, имена файлов должны различаться (включать дату).
15. Изучите лекцию 14 курса «Основы SQL». С использованием триггеров осуществите реализацию протоколирования действий пользователей (изменение, удаление, добавление информации в таблицы) <++>.
16. С использованием триггеров осуществите реализацию протоколирования действий пользователей. Реализуйте интерфейс доступа к протоколам для администратора.
17. Реализуйте интерфейс доступа к протоколам для администратора.

18. Изучите лекцию 13 курса «Основы SQL». Осуществите «сжатие протоколов» с переносом их в новую базу данных путем объединения сходных по типу подряд идущих записей в одну (если производится удаление нескольких объектов, то оставьте одну запись с информацией об удалении всех объектов; если производится редактирование одного и того же объекта несколько раз подряд, то оставьте одну запись об изменениях, включающую всю информацию о проделанной работе).
19. Закончите выполнение предыдущего задания.
20. Разработайте функцию записи XML-файла, содержащего всю информацию из базы данных (кроме протоколов). Реализуйте ее с помощью курсоров, работающих на стороне клиента.
21. Закончите выполнение предыдущего задания.
22. Репликация и зеркалирование. Осуществите изучение лекций 26–28 <http://www.intuit.ru/studies/courses/68/68/lecture/2034> Либо подготовьте конспект по этим лекциям, либо осуществите настройку всех трех видов репликации.
23. Репликация и зеркалирование. Осуществите изучение лекций 26–28 <http://www.intuit.ru/studies/courses/68/68/lecture/2034> Либо подготовьте конспект по этим лекциям, либо осуществите настройку всех трех видов репликации.
24. Кластеризация. Конспект?
25. Сравнительный анализ СУБД. Для проекта, который вас интересует с целью его последующей разработки, обоснованно выберите СУБД путем сравнения следующих: MSSQL, MySQL, MariaDB, Oracle, PostgreSQL, Cache, SQLite,
26. Оформите отчет.

Для получения положительной оценки необходимо выполнить 21 пункт из перечисленных выше без грубых ошибок, обязательно необходимо оформить отчет.

3 Дополнительные требования

Ниже перечислены требования стиля, которые будут проверяться при промежуточном и итоговом оценивании:

- все названия таблиц и полей должны быть англоязычными, слова разделяться знаком подчеркивания, регистр букв – строчный;
- все используемые команды (в том числе создания и заполнения таблиц) необходимо сохранять для возможности переноса базы данных между компьютерами;
- при использовании PHP необходимо использовать PDO для связи с СУБД;
- поля с автоматически формируемым ключом в виде целого числа должны содержать в себе подстроку `id`.

4 Параметры, учитываемые в оценке

- Сложность задания на курсовое проектирование
- Сдача в срок пунктов задания (опоздание на неделю считается нормой, в остальном будет просуммировано количество недель опоздания)
- Качество реализации
- Доля выполненного задания
- Наличие сертификата курсов [intuit.ru](http://www.intuit.ru), указанных выше (или аналогичных при использовании СУБД, отличных от MSSQL).
- Качество прохождения тестирований.

5 Отчет

Отчет по учебной практике представляет собой документ, отражающий всю работу, проделанную студентом в ходе прохождения практики. Целью составления отчета является с одной стороны облегчение общей оценки практики руководителем практики, с другой стороны позволяет сформировать целостный образ о проделанной работе у студента.

5.1 Структура отчета

Отчет состоит из следующих элементов:

1. Титульный лист
2. Содержание
3. Введение
4. Основная часть
 - (a) Разработка спецификаций различных компонент (ПК1.1)
 - (b) Осуществление разработки кода программного продукта на основе готовых спецификаций на уровне модуля (ПК1.2)
 - (c) Выполнение отладки программных модулей с использованием специализированных программных средств (ПК1.3)
 - (d) Выполнение тестирования программных модулей (ПК1.4)
 - (e) Осуществление оптимизации программного кода модуля (ПК1.5)
 - (f) Разработка компонентов проектной и технической документации с использованием графических языков спецификаций (ПК1.6)
5. Заключение
6. Приложения

Оформление отчета производится в соответствии с оформлением данного документа.

5.2 Титульный лист

Пример титульного листа приведен в приложении А.

5.3 Оформление отчета

Отчет оформляется на листах формата А4 (допускается его выполнение и рукописным способом). Поля: слева – 30 мм; сверху, снизу – 20 мм; справа – 10 мм.

Основные параметры: шрифт – Times New Roman (или аналогичный по характеристикам), размер – 14, полторный интервал, выравнивание по ширине, первая строка абзаца оформляется с отступом 1,5 см. Заголовки выравниваются по центру. Заголовки верхнего уровня располагаются на новых страницах, после них отступ 18 пт, используется полужирное начертание и все символы в них – заглавные. Заголовки второго уровня имеют те же характеристики, но отступы после них – 12 пт и располагаться они могут в любой части страницы. При этом следующие строки не должны печататься на следующей странице, а заголовок не может разрываться.

Все страницы (кроме титульного листа и задания) должны быть пронумерованы (номер указывается в правом верхнем углу). В нумерации титульный лист учитывается. Расстояние от номера страницы до текста документа должно составлять не менее 10 мм.

Рисунки имеют сквозную нумерацию и подписываются снизу, например «Рисунок 1» (без кавычек); рисунки, размещенные в приложениях имеют нумерацию в пределах приложения и подписываются в соответствии с примером: «Рисунок А.1», где А – обозначение приложения, а 1 – номер рисунка. После номера рисунка указывается название рисунка.

Аналогично, таблицы имеют сквозную нумерацию, подписываются сверху с выравниванием вправо («Таблица 1 Ошибки»). На все таблицы должны быть ссылки из текста отчета; обычно, таблица имеет нарисованные рамки, ограничивающие как саму таблицу, так и ячейки.

Таблицы и рисунки должны иметь сквозную нумерацию и названия, в основном тексте на каждый рисунок и таблицу должна быть ссылка. При необходимости можно нумеровать так же некоторые формулы.

Перед списком обязательно должно быть пояснение. Ненумерованный список должен оформляться с использованием тире.

Далее приводятся требования, являющиеся выдержками из государственных стандартов или общепринятых правил. Приведенные требования регламентируют те ситуации, что наиболее часто встречаются при выполнении работ. В остальных случаях Вы можете руководствоваться государственными стандартами ЕСКД (единая система конструкторской документации), правилами, принятыми при выполнении других работ в колледже и в других учебных заведениях.

5.3.1 Требования машинописи

Текст должен быть оформлен в соответствии с основными требованиями машинописи:

- перед основными знаками препинания (кроме тире) пробел не ставится, после — ставится всегда; тире и дефисы должны отличаться друг от друга (тире длиннее, дефис короче);
- дробная и целая часть числа отделяются друг от друга запятой (кроме исходного текста программы);
- сноски оформляются способом, который по умолчанию применяется в редакторах Microsoft Word или Libreoffice Writer, но в конце сносок устанавливается закрывающаяся скобка (как в данном документе).
- не допускается использование двух и более пробелов вместо одного (в редакторах можно отследить выполнение этого требования путем включения просмотра невидимых символов)

5.4 Введение

Во введении студент излагает постановку задачи (краткую формулировку заданий), цель и задачи учебной практики (изложение ведется на основе анализа введения сборника практических заданий).

5.5 Основная часть

В основной части студент излагает обзор всей проделанной работы.

5.5.1 Разработка объектов базы данных;

В разделе «Разработка объектов базы данных» студент описывает окончательную схему базы данных (в виде диаграммы).

5.5.2 Реализовывать базы данных в конкретной СУБД

В разделе «Реализовывать базы данных в конкретной СУБД» студент ссылается на код на языке sql, содержащий запросы на создание базы данных, таблиц, представлений, хранимых процедур, триггеров. В основном тексте проводится краткое их описание.

5.5.3 Решать вопросы администрирования базы данных и реализовывать методы и технологии защиты информации в базах данных

В разделе «Решать вопросы администрирования базы данных и реализовывать методы и технологии защиты информации в базах данных» студент включает диаграмму вариантов использования, а также ссылается на запросы на создание пользователей и настройки прав доступа. Кроме того, он описывает, как он проводил отладку своей программы. В частности, приводит примеры наиболее содержательных (для него) ошибок и способов их устранения и использованные им функции, предоставляемый средой разработки.

5.6 Заключение

В заключении студент описывает достигнутые им результаты на основе введения: он показывает, что все цели и задачи, приведенные во введении достигнуты.

В отдельные приложения в качестве дневника студент включает отчет системы контроля версий, отчет базы данных дефектов, программный код и распечатку(-и) сертификата(-ов) intuit.ru (при их наличии).

Кроме того, студент описывает **личный** взгляд на результаты практики: чему он научился, в чем изменился, какие сделал личностные и профессиональные выводы и так далее.

6 Список литературы

1. Полякова Лариса. Основы SQL / URL: <http://intuit.ru/studies/courses/5/5/info>
2. SQL Server 2000 / URL: <http://intuit.ru/studeis/courses/68/68/info>
3. MSDN / URL: <http://msdn.microsoft.com/ru-ru/library/bb545450.aspx>
4. Программирование баз данных в Delphi / URL: <http://www.intuit.ru/studies/courses/614/470/info>

Приложение А. Титульный лист

**Министерство образования и науки Российской Федерации
ФГБОУ ВО «МГУТУ им. К. Г. Разумовского (ПКУ)»
Университетский колледж информационных технологий**

Отчет по учебной практике

ПМ.02 «ПМ2 Разработка и администрирование баз данных»,
студента группы П-203, обучающегося специальности
09.02.03 «Программирование в компьютерных системах»

Выполнил студент: Иванов И. И.
Преподаватель: Глускер А. И.