

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине
«Анализ данных в бизнес-системах»
направление 38.03.05 «Бизнес-информатика»
направленность (профиль) «Информационная бизнес-аналитика и
цифровые технологии»

Оглавление

Введение.....	3
Лабораторная работа №1 «Реляционная база данных, хранимые процедуры, триггеры».....	11
Лабораторная работа № 2 Подготовка базы данных SQL Server.....	19
Лабораторная работа № 3 «Служба Microsoft SQL Server Analysis Services (SSAS)».....	23
Лабораторная работа № 4. Определение и развертывание куба.....	27
Лабораторная работа № 5. Определение и развертывание куба.....	32
Лабораторная работа № 6. Изменение мер, атрибутов и иерархий.....	38
Лабораторная работа № 7. Изменение мер, атрибутов и иерархий.....	46
Лабораторная работа № 8. Определение расширенных свойств атрибутов и измерений.....	59
Лабораторная работа № 9. Определение расширенных свойств атрибутов и измерений.....	68
Список литературы, рекомендуемый к использованию.....	118

Лабораторная работа №1 Реляционная база данных, хранимые процедуры, триггеры

Формируемые компетенции или их части

п/п	№	Содержание компетенции	Ши фр
.	1	Анализ данных и обоснование решений	ПК-4

1. Практическая часть

Установка SQLServerDeveloperEdition

SQLServerDeveloper – обладающий полным набором функций бесплатный выпуск, лицензируемый, для использования в качестве базы данных, для разработки и тестирования, и непредназначенный для применения в рабочей среде.

Для установки SQLServer с сайта <https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads> выберите DeveloperEdition и нажмите кнопку «Скачать». После скачивание автоматически запускается установщик. Выберите тип установки–«Базовая»(рис.1):

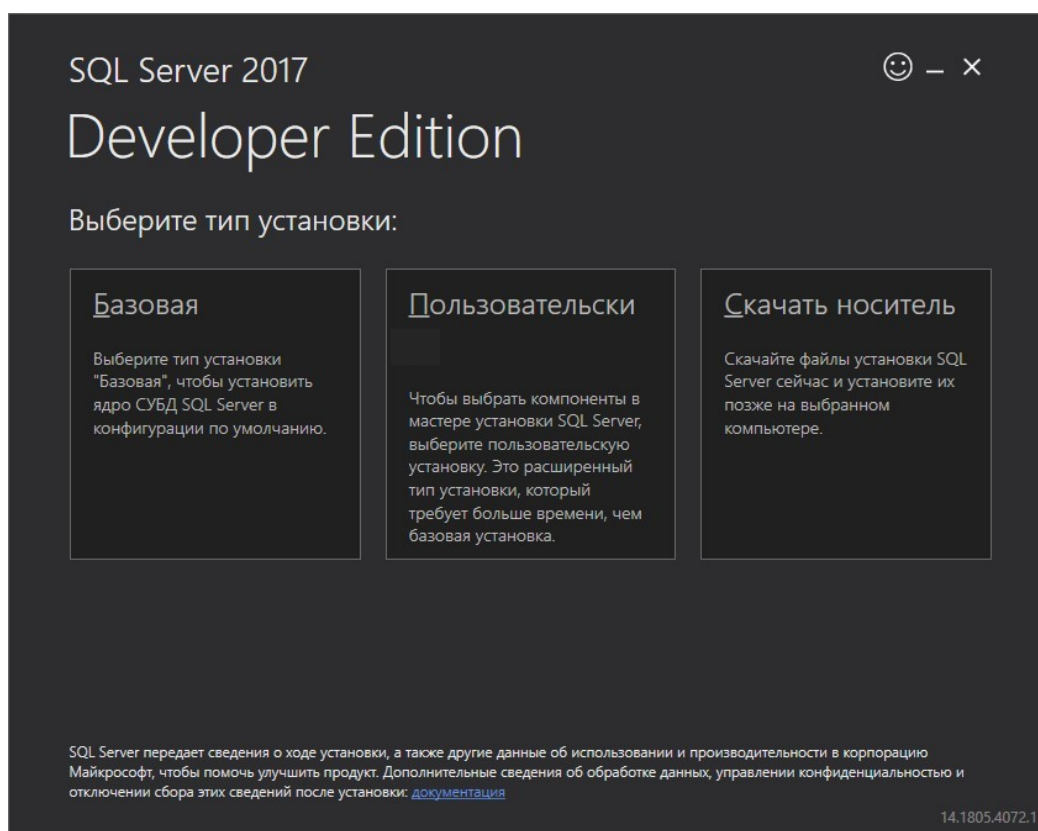


Рис.1.Выбор тип установки

После принятия условия соглашения, укажите целевое расположение SQLServer.

Установка SQLServerManagementStudio

SQLServerManagementStudio (SSMS) – это интегрированная среда для управления любой инфраструктурой SQL, от SQLServer до баз данных SQLAzure. SSMS предоставляет средства для настройки, наблюдения и

администрирования экземпляров SQL Server и баз данных. С помощью SSMS можно развертывать, отслеживать и обновлять компоненты уровня данных, используемые вашими приложениями, а также создавать запросы и скрипты.

Скачайте SQL Server Management Studio (SSMS) с сайта: <https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

Открытие среды SQL Server Management Studio

Запускайте SQL Server Management Studio.

В диалоговом окне Соединение с сервером подтвердите заданные по умолчанию параметры и нажмите кнопку Подключиться. Для соединения необходимо, чтобы поле Имя сервера содержало имя компьютера, на котором установлен SQL Server. Если компонент Database Engine представляет собой именованный экземпляр, то поле «Имя сервера» должно также содержать имя экземпляра в формате <имя_компьютера>\<имя_экземпляра>.

Выбрать имя сервера (в данном случае это REM-DESKTOP).

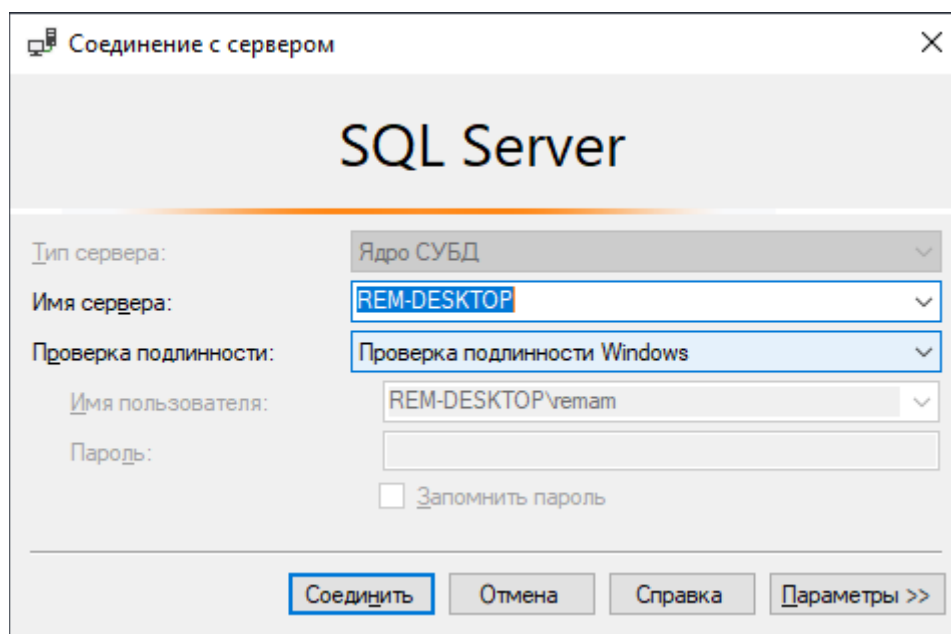


Рис.2. Соединение с SQL Server

MSSQL Server использует два режима аутентификации: аутентификацию Windows и аутентификацию MSSQL Server (SQL Server authentication). Для аутентификации по умолчанию используется аутентификация Windows. Windows Authentication mode (Проверка подлинности Windows) – это режим, который использует для подключения к серверу только логины Windows (рис.2). В этом случае пользователям не требуется вводить как имя, так и пароль при подключении к SQL Server, если они уже вошли в сеть Windows.

Для использования аутентификации SQL Server вначале необходимо создать учетную запись в самом SQL Server. В отличие от логинов Windows, логины SQL Server – это самостоятельные учетные записи со своими именами и паролями, информация о которых хранится в системной базе данных SQL Server. При подключении к серверу при помощи логина SQL Server вам придется указать имя логина и пароль.

Создание учетной записи

Создайте новую учетную запись с именем STUDENT. В ManagementStudio список учетных записей, сконфигурированных на сервере, содержится в папке «\Безопасность \Имена для входа». Чтобы добавить новую учетную запись, необходимо выделить узел «Имена для входа» в контекстном меню и выбрать пункт «Создать имя для входа...».

В открывшемся окне (рис. 3) в поле «Имя для входа» введите STUDENT. Далее выберите переключатель «Проверка подлинности SQL Server» и в поле «Пароль» наберите пароль. Снимите флажок «Пользователь должен сменить пароль при следующем входе». Остальные поля оставьте без изменений.

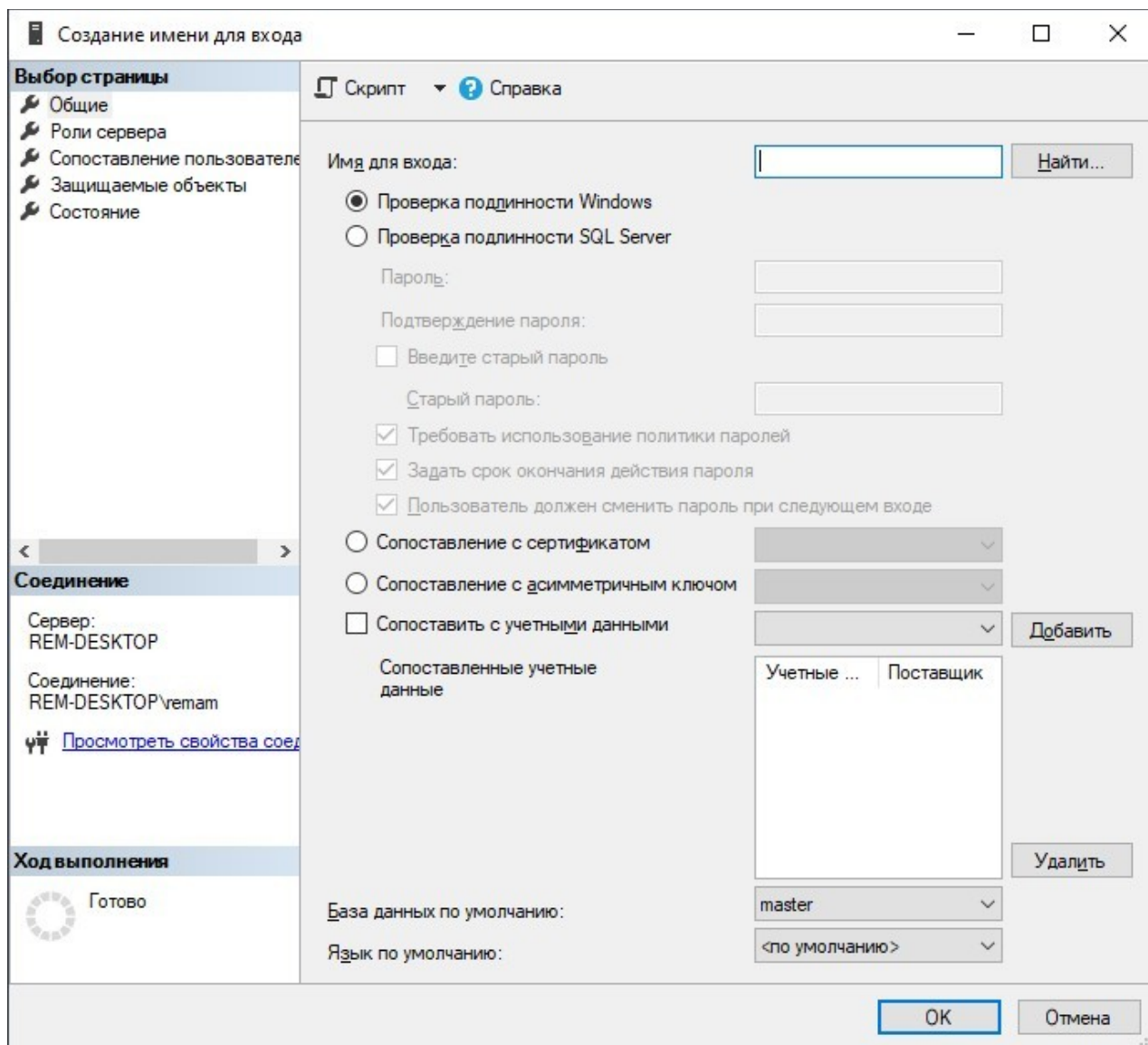


Рис.3.Создание новой учетной записи

Создание базы данных

Базу данных может создать только пользователь с правами администратора. Выберите в контекстном меню папки «Базы данных» команду «Создать базу данных...»(рис.4). В поле «Имя базы данных» введите имя создаваемой базы данных (БД)—«Учебная» (рис.5). Здесь также можно изменить путь сохранения созданной БД. Обратите внимание, что создаётся пустая база данных (контейнер).

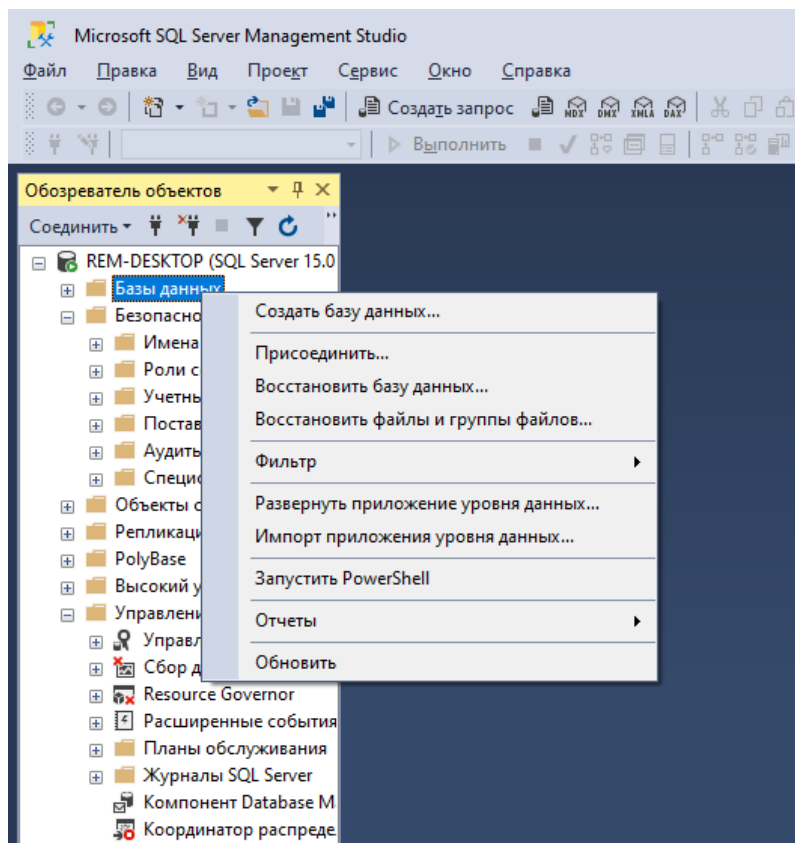


Рис.4. Пункт контекстного меню «Создать базу данных...»

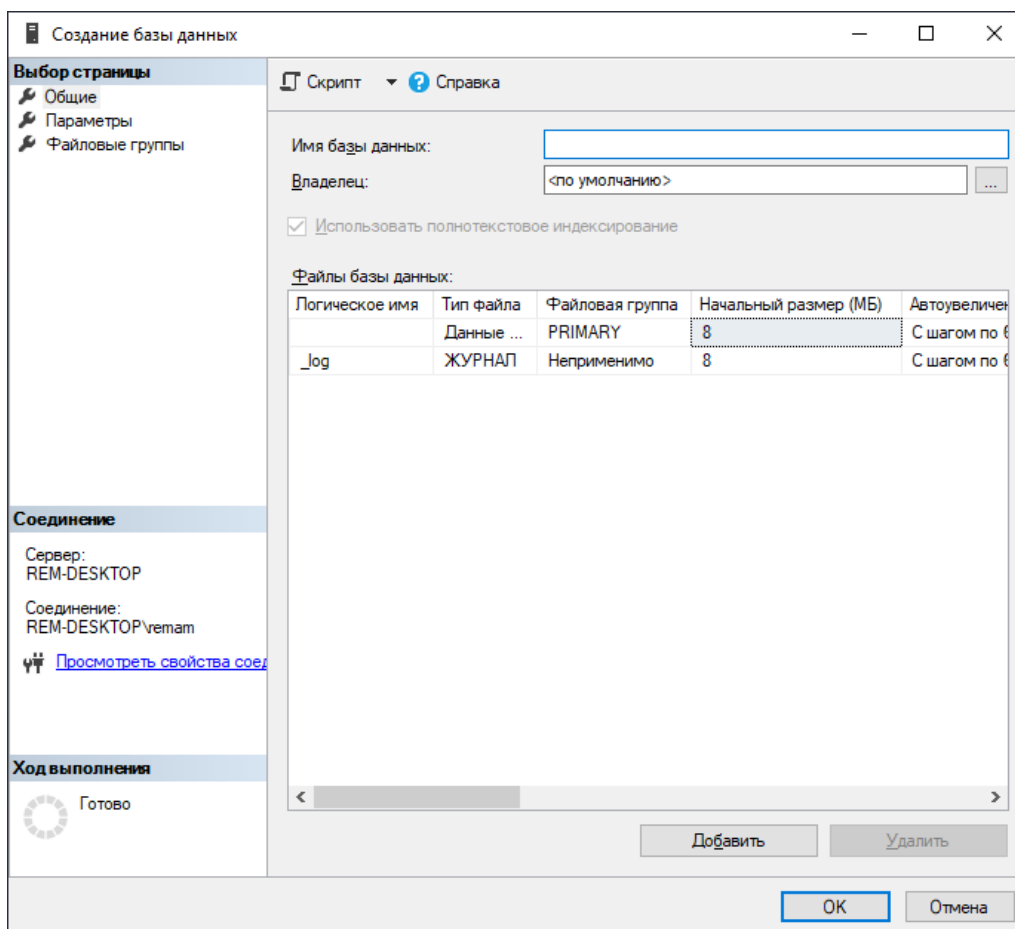


Рис.5. Создание новой базы данных

После создания БД окно «Обозреватель объектов» обновится. В ветке «Базы данных» появится новая база данных «Учебная». Эта база данных принадлежит пользователю по имени sysadmin(системный администратор).

Создание нового пользователя

Создание пользователя в SQLServer осуществляется посредством создания учетной записи. Пользователи БД–это специальные объекты, которые создаются на уровне базы данных и используются для предоставления разрешений в базе данных (на таблицы, представления, хранимые процедуры). Логины и пользователи БД–это совершенно разные объекты.

Чтобы добавить в базу данных «Учебная» нового пользователя, надо: выделить узел «\Базы данных\Учебная\Безопасность\Пользователи» и в контекстном меню выбрать пункт«Создать пользователя...»(рис. 6).

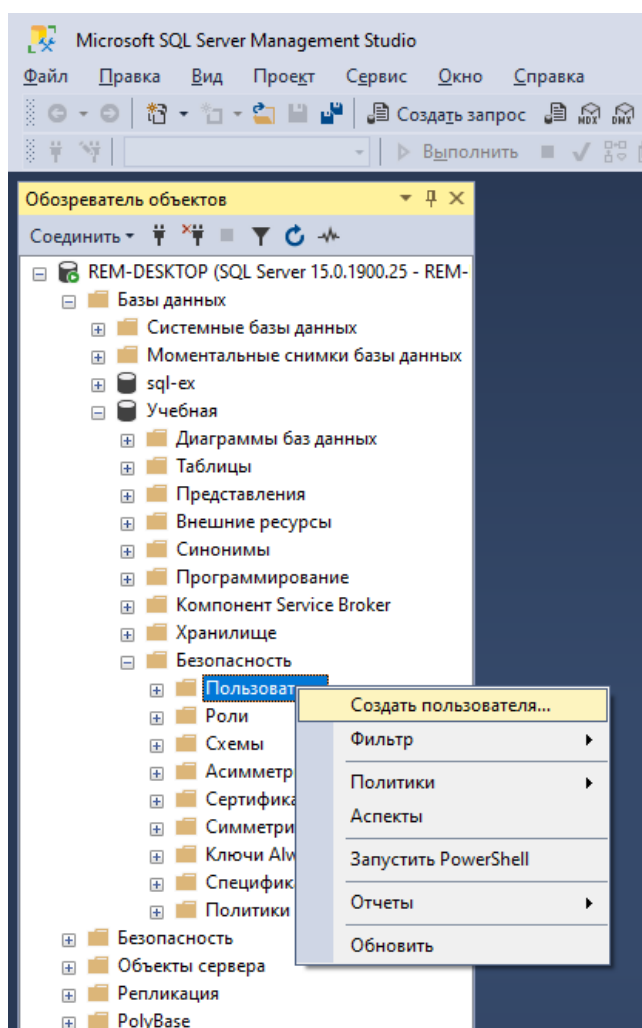


Рис.6.Пункт контекстного меню «Создатьпользователя...»

В открывшемся окне в поля «Имя пользователя» и «Имя для входа» ввести имя –Student. На странице «Собственные схемы» в списке «Схемы,

принадлежащие данному пользователю:» установить флажок db_owner→ОК (рис. 7).

Таким образом, на основе учетной записи Student будет создан новый пользователь для БД «Учебная» с правами владельца этой базы данных.

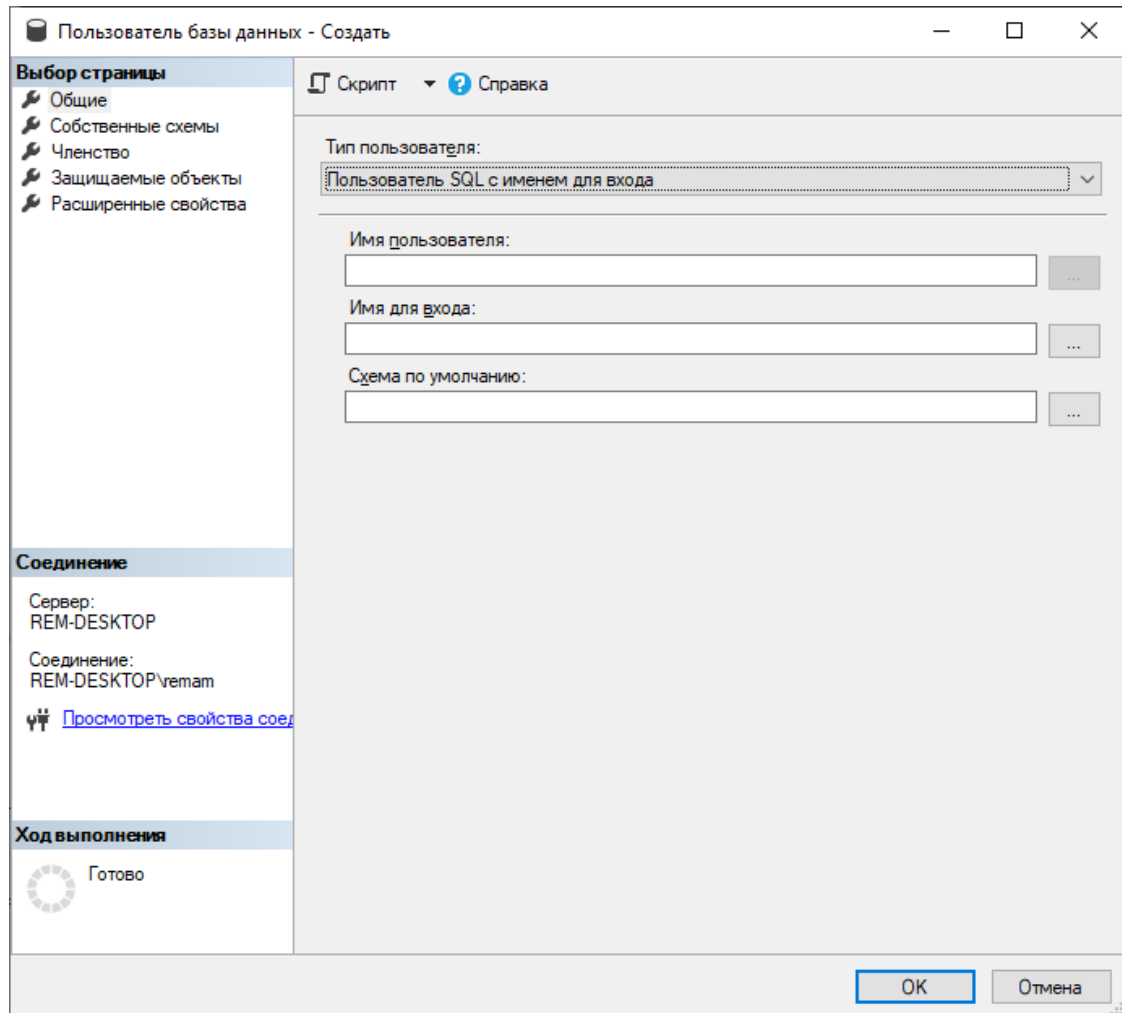
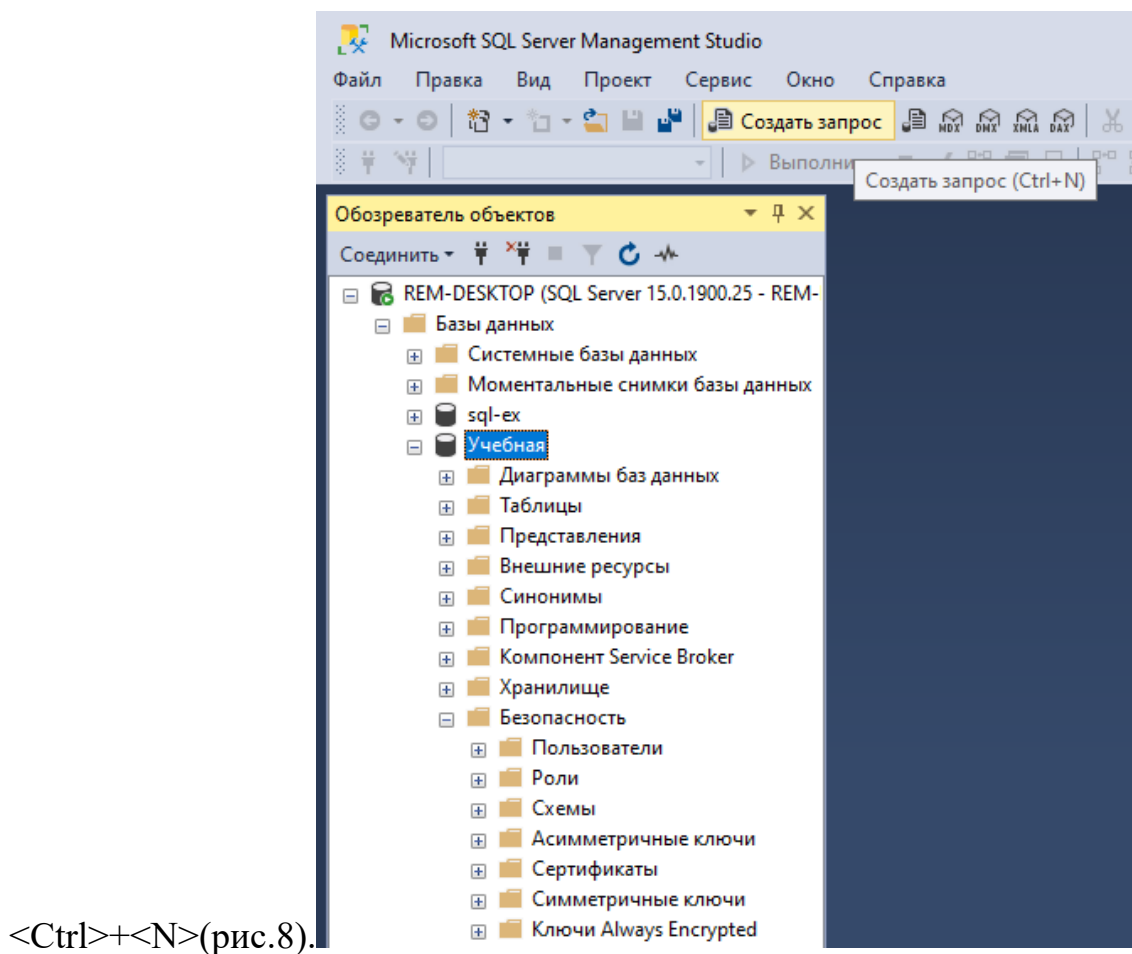


Рис.7.Добавление в БД нового пользователя

Создание и заполнение таблиц

В «Обозревателе объектов» выберите базу данных «Учебная». На строке инструментов нажмите «Создать запрос» или используйте горячие клавиши



<Ctrl>+<N>(рис.8).

Рис.8.Выбор БД «Учебная»

На открывшем окне редактора наберите код из приложения.

На строке инструментов нажмите «Выполнить» или используйте горячую клавишу <F5>для выполнения команд.

Резервное копирование БД

Сведения, хранящиеся в базах данных, являются стратегической ценностью любой компании, поэтому их сохранность очень важна.

Выберем в контекстном меню базы данных «Учебная» пункт «Задачи»→ «Создать резервную копию»(рис. 9).

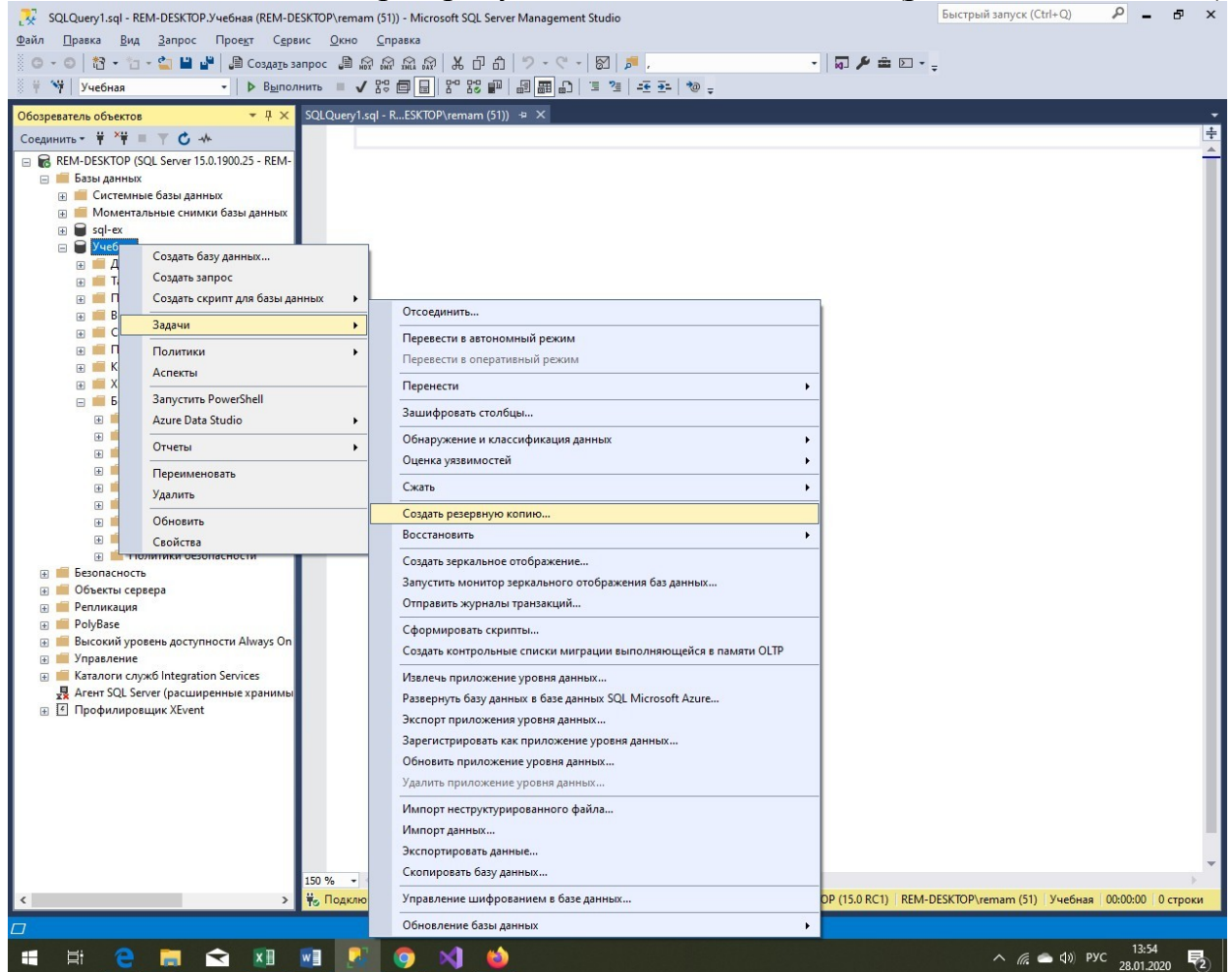


Рис.9. Пункт контекстного меню «Создать резервную копию...»

В открывшемся диалоге «Резервное копирование базы данных–Учебная» (рис.10) установите некоторые параметры:

База данных–Учебная (это база данных для резервного копирования);

Тип архивной копии–Полная (тип резервной копии–полный; для первого резервного копирования применяется только такой; если резервная копия уже имеется, и вы желаете не создавать заново полную копию, а внести изменения в старую, выберите Разностная);

Компонент архивной копии–База данных;

Архивировать в:–Диск.

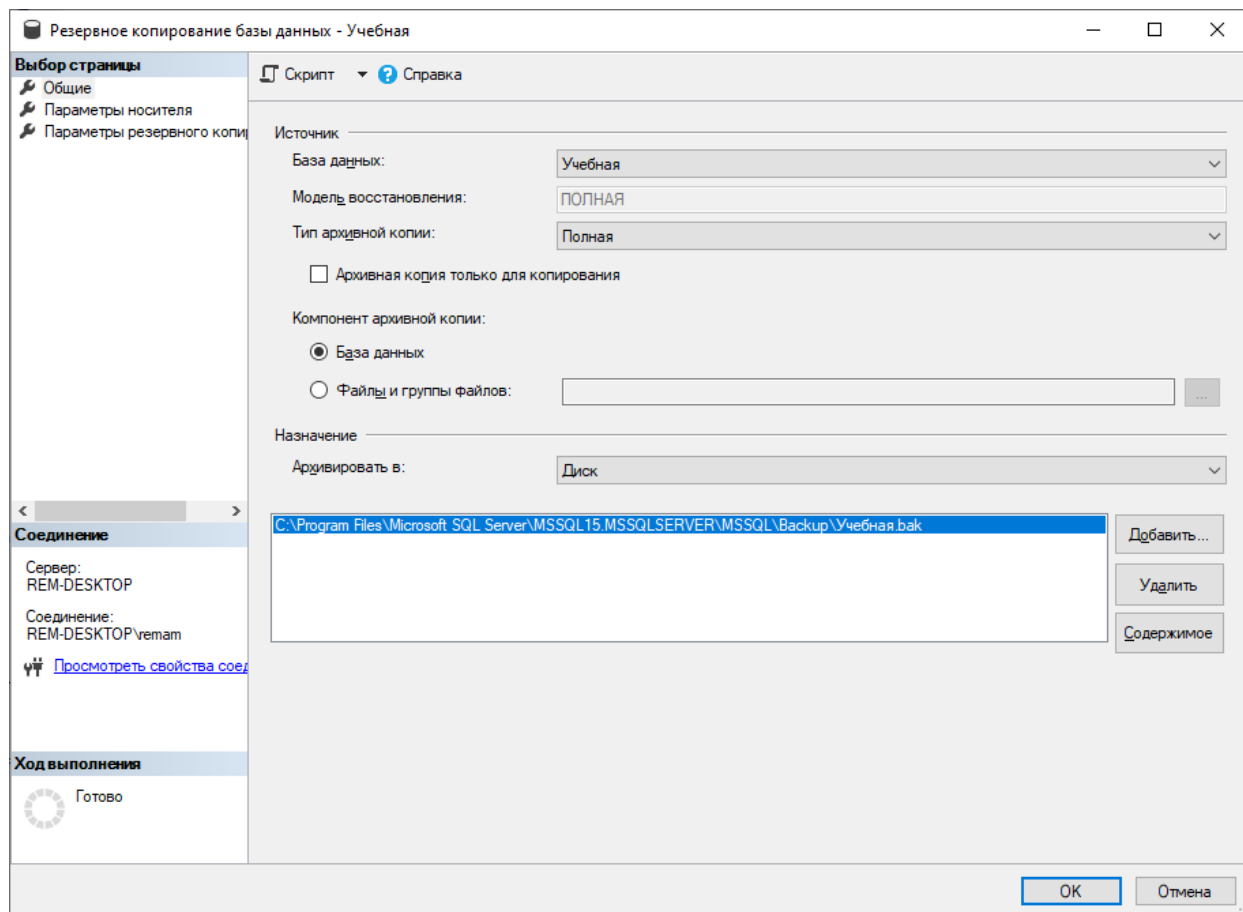


Рис.10.Диалоговое окно BackUpDatabase

Нажмите кнопку «Добавить...» в этом диалоге. Откроется диалоговое окно «Выбор места расположения резервной копии» (рис.11). В поле окна «Имя файла» введите имя файла, в который будет выполнено резервное копирование.

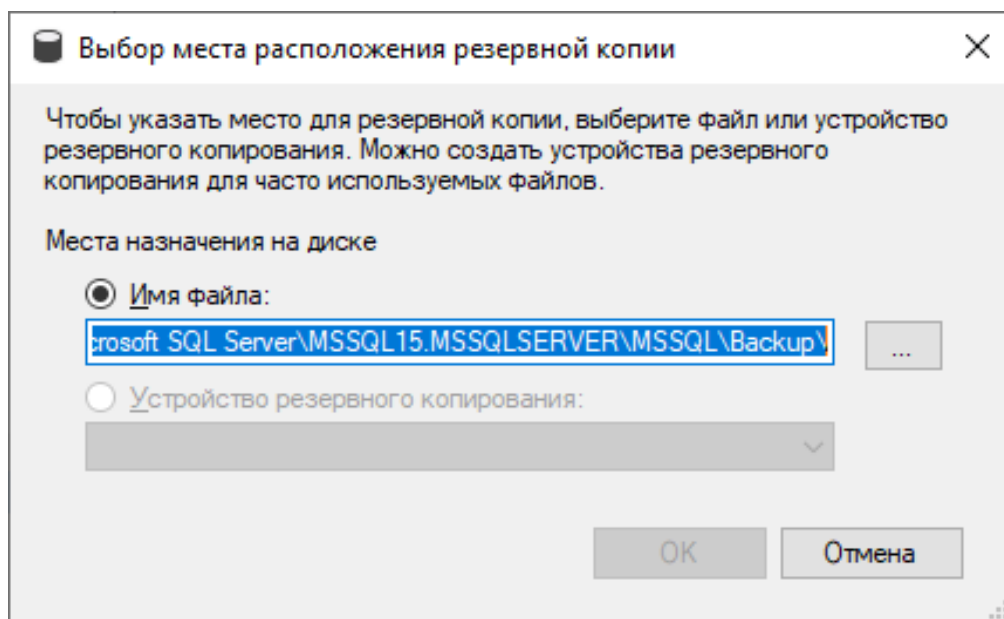


Рис.11.Диалоговое окно «Выбор места расположения резервной копии»

Нажмите ОК в диалоговом окне «Выбор места расположения резервной копии». Оно закроется, а в списке «Назначение» появится строка, которую ввели. Выберите ее и нажмите кнопку ОК. Произойдет резервное копирование по указанному пути.

Задание

Установите SQLServer DeveloperEdition.

Установите SQLServer ManagementStudio(SSMS).

Создайте учетную запись пользователя MSSQL-Server.

Создайте новую базу данных.

Создайте нового пользователя базы данных.

Соединитесь с базой данных под учетной записью пользователя.

Создайте структуру базы в соответствии с порядком выполнения работы.

Выполните резервное копирование базы данных.

Контрольные вопросы:

Каковы основные конструкции языка SQL?

Как создается хранимая процедура? Для чего они нужны?

Как создается триггер? Для чего они нужны?

Какие операторы языка SQL допустимы внутри триггера?

Как реализуется механизм событий?

Как указать действия, которые следует предпринять в случае ошибки?

Как удалить хранимую процедуру, триггер из базы данных?

С какими моделями транзакций связано исполнение триггеров?

Объясните структуру файла хранимой процедуры (триггера)? Из каких конструкций он состоит?

Лабораторная работа № 2 Подготовка базы данных SQL Server

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Подавляющее большинство пользователей используют SQL для организации выборки данных. Для выборки данных из БД используется запрос SELECT. Он позволяет фильтровать выбранные данные и преобразовать их к нужному виду. Результатом выполнения запроса SELECT является другая таблица, к которой снова может быть применен запрос SELECT.

Полный синтаксис инструкции SELECT сложен, однако основные предложения можно в кратце описать следующим образом:

```
[ WITH { [ XMLNAMESPACES ,]  
[ <common_table_expression> ] } ]SELECTselect_list [INTOnew_table]  
[FROMtable_source]  
[WHEREsearch_condition]  
[ GROUP BY group_by_expression ][ HAVING search_condition]  
[ ORDER BYorder_expression [ ASC |DESC]]
```

Обработка элементов запроса SELECT выполняется в следующей последовательности:

1. FROM—определяет имена используемых таблиц;
2. WHERE—фильтрует строки таблицы в соответствии с заданными условиями;
3. GROUPBY—группирует строки, имеющие одинаковые значения в указанном столбце;
4. HAVING—фильтрует группы строк в соответствии с указанным условием;
5. SELECT—форматирует выходные данные;
6. ORDERBY—сортирует результаты выполнения запроса.

Порядок предложений в запросе SELECT не может быть изменен. Предложения SELECTи FROM являются обязательными, присутствие остальных зависит от контекста.

В предложении SELECT указывается список столбцов, которые должны быть возвращены запросом. Можно указать исходные элементы или вычисляемые поля во время выполнения запроса.

КонструкцияDISTINCT|ALLисключает/разрешает вывод повторяющихся строк.

КонструкцияALLиспользуется по умолчанию.

* означает вывод всех столбцов указанной таблицы. В случае, если

выборка производится из нескольких таблиц, перед символом звездочки может указываться имя таблицы.

SQL-запрос может содержать вычисляемые столбцы, значения которых могут определяться на основе значений данных, хранящихся в БД конструкции. Вычисляемым столбцам следует давать название с помощью ключевого слова AS.

Вычисляемый столбец можно создать как: <Новое поле>=<выражение>

Если название столбца состоит из нескольких слов, разделенных пробелами, следует их записать в квадратных скобках:[].

Сортировка данных выполняется с помощью команды ORDERBY, которая добавляется в конец запроса, после чего перечисляется список столбцов. Для каждого столбца указывается тип сортировки ASC | DESC (ascending – по возрастанию | descending – по убыванию).ASC – по умолчанию, можно не указывать.

Конструкция TOP<N> позволяет выбрать определенное количество строк из таблицы. Дополнительный оператор PERCENT позволяет выбрать процентное количество строк из таблицы. Дополнительный оператор WITHTIES позволяет выбрать все строки с такими же свойствами.

Конструкция OFFSET<N>ROWS указывает число строк, которые необходимо пропустить, прежде чем будет начат возврат строк из выражения запроса.

Конструкция FETCHNEXT<N>ROWSONLY указывает число строк, возвращаемых после обработки предложения OFFSET.

На языке T-SQL регистр не имеет значения(case insensitive).

2. Практическая часть

Дана таблица *Академики*:

ФИО	Дата_рождения	Специализация	Год_присвоения_звания
Аничков Николай Николаевич	1885-11-03	медицина	1939
Бартольд Василий Владимирович	1869-11-15	историк	1913
Белопольский Аристарх Аполлонович	1854-07-13	астрофизик	1903
Бородин Иван Парфеньевич	1847-01-30	ботаник	1902
Вальден Павел Иванович	1863-07-26	химик-технолог	1910
Вернадский Владимир Иванович	1863-03-12	геохимик	1908
Виноградов Павел Гаврилович	1854-11-30	историк	1914
Ипатьев Владимир Николаевич	1867-11-21	химик	1916
Истрин Василий Михайлович	1865-02-22	филолог	1907
Карпинский Александр Петрович	1847-01-07	геолог	1889
Коковцов Павел Константинович	1861-07-01	историк	1906
Курнаков Николай Семёнович	1860-12-06	химик	1913
Марр Николай Яковлевич	1865-01-06	лингвист	1912

Насонов Николай Викторович	1855-02-26	зоолог	1906
Ольденбург Сергей Фёдорович	1863-09-26	историк	1903
Павлов Иван Петрович	1849-09-26	физиолог	1907
Перетц Владимир Николаевич	1870-01-31	филолог	1914
Соболевский Алексей Иванович	1857-01-07	лингвист	1900
Стеклов Владимир Андреевич	1864-01-09	математик	1912

Пример 1: Вывести список академиков: **SELECT***

FROM

Академики

Пример 2: Вывести ФИО и дату рождения всех академиков: **SELECT**

ФИО, Дата_рождения

FROM

Академики

Пример 3: Создайте вычисляемое поле «Информация», содержащее информацию об академиках в таком виде: «Академик Петров Петр Петрович, специализация: математика»:

SELECT

'Академик'+ФИО+', специализация: '+Специализация **AS** Информация

FROM

Академики

Пример 4: Вывести ФИО академиков и номер следующего года после присвоения звания: **SELECT**

ФИО

, [Черезгод] = Год_присвоения_звания + 1

FROM

Академики

Пример 5: Выведите список специализаций, убрав дубликаты: **SELECT DISTINCT**

Специализация

FROM

Академики

Пример 6: Вывести список академиков, отсортированный по возрастанию года присвоения звания:

SELECT

*

FROM

Академики

ORDERBY

Год_присвоения_звания

Пример 7: Вывести список академиков, отсортированный в обратном алфавитном порядке по полю «Специализация» и в алфавитном порядке по полю «ФИО»:

SELECT

*

FROM

Академики

ORDERBY

СпециализацияDESC

,ФИОASC

Пример 8: Вывести первые две строки из списка академиков, отсортированного в алфавитном порядке по полю «ФИО»:

SELECTTOP 2

*

FROM

Академики

ORDERBY

ФИОASC

Пример 9: Вывести первые 30% строк из списка академиков, отсортированного по возрастанию года присвоения звания:

SELECTTOP30PERCENT

*

FROM

Академики

ORDERBY

Год_присвоения_звания

Пример 10: Вывести из таблицы «Академики», отсортированной по возрастанию года присвоения звания, список академиков, у которых год присвоения звания – один из первых четырех в отсортированной таблице:

SELECTTOP4WITH TIES

*

FROM

Академики

ORDERBY

Год_присвоения_звания

Пример 11: Вывести, начиная с третьего, список академиков, отсортированный в алфавитном порядке ФИО:

SELECT

*

FROM

Академики

ORDERBY

ФИО

OFFSET2ROWS

Пример12: Вывести, начиная с третьего и до десятого, список академиков, отсортированный в алфавитном порядке ФИО:

SELECT

*

FROM

Академики

ORDERBY

ФИО

OFFSET2ROWS

FETCHNEXT8ROWS ONLY

Задание

1. Вывести ФИО, специализацию и дату рождения всех академиков.
2. Создать вычисляемое поле «О присвоении звания», которое содержит информацию об академике в виде: «Петров Петр Петрович получил звание в1974».
3. Вывести ФИО академиков и вычисляемое поле «Через 5 лет после присвоения звания».
4. Вывести список годов присвоения званий, убрав дубликаты.
5. Вывести список академиков, отсортированный по убыванию даты рождения.
6. Вывести список академиков, отсортированный в обратном алфавитном порядке специализаций, по убыванию года присвоения звания, и в алфавитном порядке ФИО.
7. Вывести первую строку из списка академиков, отсортированного в обратном алфавитном порядке ФИО.
8. Вывести фамилию академика, который раньше всех получил звание.
9. Вывести первые 10% строк из списка академиков, отсортированного в алфавитном порядке ФИО.
10. Вывести из таблицы «Академики», отсортированной по возрастанию года присвоения звания, список академиков, у которых год присвоения звания – один из первых пяти в отсортированной таблице.
11. Вывести, начиная с десятого, список академиков, отсортированный по возрастанию даты рождения.

12. Вывести девятую и десятую строку из списка академиков, отсортированного в алфавитном порядке ФИО.

Контрольные вопросы:

1. Какие сведения хранятся в БД AdventureWorksDW?
2. Выделите основные этапы подготовки данных для работы с моделями Data Mining.

Лабораторная работа № 3. Служба Microsoft SQL Server Analysis Services (SSAS)

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Для фильтрации данных применяется оператор WHERE. Синтаксис: WHERE<условие>. В условиях поле таблицы сравнивается с константой или с выражением. Символьные константы пишутся в одинарных кавычках. Числовые константы и названия столбцов пишутся без кавычек.

Чтобы строка попала в результат, условие должно быть истинно. В условиях используются операции сравнения. В Transact-SQL применяются следующие операции сравнения:

=—равенство;
<>или !=— неравенство;
<— меньше;
>— больше;
!<— не меньше;
!>— не больше;
<=—меньше или равно;
>=—больше или равно.

Можно использовать несколько условий для фильтрации данных. Для объединения их в одно выражение используются логические операторы. В Transact-SQL применяются следующие логические операторы:

AND – логическое умножение или конъюнкция (И). Бинарный оператор, объединяет два условия; если оба условия истинны, результат– истина, иначе – ложь.

OR – логическое сложение или дизъюнкция (ИЛИ). Бинарный оператор, объединяет два условия; если хотя бы одно из этих условий истинно, то общее условие оператора OR также будет истинно.

NOT – логическое отрицание или инверсия (НЕ). Унарный оператор, применяется к одному условию. Если выражение в этой операции ложно, то общее условие истинно.

Самый высокий приоритет у оператора NOT. Самый низкий – у OR. Если эти операторы встречаются в одном выражении, то сначала выполняется NOT, потом AND, а затем OR. При записи условий использование скобок–хороший тон программирования.

Оператор BETWEEN используется для сравнения с диапазоном от начального и до конечного значения. Начальное и конечное значения включены в промежуток.

Оператор LIKE используется для сравнения с шаблоном строки. Для определения шаблона применяются специальные символы:

%–любая последовательность символов, в том числе пустая;

–любой символ;
 []– символ, который указан в квадратных скобках; [-] – символ из определенного диапазона;
 [^] –символ, который не указан после символа^.

В базах данных, для обозначения неизвестного значения, используется понятие NULL. Для проверки неизвестного значения нельзя использовать операторы сравнения. Допускается только ISNULL или ISNOTNULL.

Оператор IN используется для сравнения с набором значений. Список значений указывается в скобках.

2. Практическая часть

Дана таблица **Страны**:

Название	Столица	Площадь	Население	Континент
Австрия	Вена	83858	8741753	Европа
Азербайджан	Баку	86600	9705600	Азия
Албания	Тирана	28748	2866026	Европа
Алжир	Алжир	2381740	39813722	Африка
Ангола	Луанда	1246700	25831000	Африка
Аргентина	Буэнос-Айрес	2766890	43847000	ЮжнаяАмерика
Афганистан	Кабул	647500	29822848	Азия
Бангладеш	Дакка	144000	160221000	Азия
Бахрейн	Манама	701	1397000	Азия
Белиз	Бельмопан	22966	377968	Северная Америка
Белоруссия	Минск	207595	9498400	Европа
Бельгия	Брюссель	30528	11250585	Европа
Бенин	Порто-Ново	112620	11167000	Африка
Болгария	София	110910	7153784	Европа
Боливия	Сукре	1098580	10985059	ЮжнаяАмерика
Ботсвана	Габороне	600370	2209208	Африка
Бразилия	Бразилиа	8511965	206081432	ЮжнаяАмерика
Буркина-Фасо	Уагадугу	274200	19034397	Африка
Бутан	Тхимпху	47000	784000	Азия
Великобритания	Лондон	244820	65341183	Европа
Венгрия	Будапешт	93030	9830485	Европа
Венесуэла	Каракас	912050	31028637	ЮжнаяАмерика
ВосточныйТимор	Дили	14874	1167242	Азия
Вьетнам	Ханой	329560	91713300	Азия

Пример 1: Вывести список стран, площадь которых больше 1 млн. кв. км:**SELECT**Название

,Столица

,Площадь

,Население

,Континент**FROM**

Страны**WHERE**

Площадь>1000000

Пример 2: Вывести список стран, население которых не больше 1 млн. чел.:
`SELECT` Название
, Столица
, Площадь
, Население
, Континент `FROM`
Страны `WHERE`
Население!>1000000

Пример 3: Вывести список африканских стран:
`SELECT` Название
, Столица
, Площадь
, Население
, Континент `FROM`
Страны `WHERE`
Континент='Африка'

Пример 4: Вывести список всех стран, кроме европейских:
`SELECT` Название
, Столица
, Площадь
, Население
, Континент `FROM`
Страны `WHERE`
Континент!='Европа'

Пример 5: Вывести список стран, население которых больше 1 млн. чел., а площадь меньше 100 тыс. кв.км:
`SELECT` Название
, Столица
, Площадь
, Население
, Континент `FROM`
Страны `WHERE`
(Население>1000000) `AND`(Площадь <100000)

Пример 6: Вывести список стран, которые находятся в Европе и их население больше 10 млн. чел., или находятся в Азии, а население больше 50 млн. чел.:
`SELECT` Название
, Столица

,Площадь
,Население
 ,КонтинентFROM
 СтраныWHERE
(Континент = 'Европа') AND (Население >10000000)OR
(Континент = 'Азия')AND(Население>50000000)

Пример 7: Вывести список стран, население которых от 10 до 100 млн.чел., а площадь от100 до 200 тыс. кв.км:

SELECTНазвание
,Столица
,Площадь
,Население
 ,КонтинентFROM
 СтраныWHERE
(НаселениеBETWEEN 10000000 AND 100000000)AND
(Площадь>=100000) AND(Площадь <=200000)

Пример 8:Вывести отсортированный в алфавитном порядке список стран от Бенина доВатикана:

SELECTНазвание
,Столица
,Площадь
,Население
 ,КонтинентFROM
 СтраныWHERE
 Название BETWEEN'Бенин' AND'Ватикан'ORDERBY
Название

Пример 9: Вывести список стран, название которых начинается с буквы «С»:
SELECTНазвание

,Столица
,Площадь
,Население
 ,КонтинентFROM
 СтраныWHERE
НазваниеLIKE'C%'

Пример 10: Вывести список стран, в названии которых вторая буква—«а», а

последняя–я»: **SELECT** Название
, Столица
, Площадь
, Население
 , Континент **FROM**
 Страны **WHERE**
Название **LIKE** ' _a%я'

Пример 11: Вывести список стран, в названии которых третья буква—«а», «о» или «у»: **SELECT** Название

, Столица
, Площадь
, Население
 , Континент **FROM**
 Страны **WHERE**
Название **LIKE** '[aou]%'

Пример 12: Вывести список стран, название которых начинается с буквы от «А» до «Г»: **SELECT** Название

, Столица
, Площадь
, Население
 , Континент **FROM**
 Страны **WHERE**
Название **LIKE** '[А-Г]%'

Пример 13: Вывести список стран, название которых не начинается с буквы от «А» до «Г» или с буквы «С»:

SELECT Название
, Столица
, Площадь
, Население
 , Континент **FROM**
 Страны **WHERE**
Название **LIKE** '[^А-Г,^С]%'

Пример 14: Вывести список стран, столицы которых не введены в базу: **SELECT** Название

, Столица
, Площадь
, Население

,КонтинентFROM
СтраныWHERE
СтолицаISNULL

Пример 15: Вывести список европейских, азиатских и африканских стран:SELECTНазвание

,Столица

,Площадь

,Население

,КонтинентFROM

СтраныWHERE

КонтинентIN('Европа','Азия','Африка')

Задание

1. Вывести названия и столицы пяти наибольших стран по площади.
2. Вывести список африканских стран, население которых не превышает 1 млн.чел.
3. Вывести список стран, население которых больше 5млн.чел., а площадь меньше 100тыс. кв.км, и они расположены не в Европе.
4. Вывести список стран Северной и Южной Америки, население которых больше 20млн. чел., или стран Африки, у которых население больше30 млн. чел.
5. Вывести список стран, население которых составляет от 10 до100млн.чел.,а площадь не больше500 тыс.кв.км.
6. Вывести список стран, названия которых не начинаются с буквы «К».
7. Вывести список стран, в названии которых третья буква–«а», а предпоследняя «и».
8. Вывести список стран, в названии которых вторая буква–гласная.
9. Вывести список стран, названия которых начинаются с букв от «К» до «П».
10. Вывести список стран, названия которых начинаются с букв от «А» до «Г», но не с буквы «Б».
11. Вывести список стран, столицы которых есть в базе.
12. Вывести список стран Африки, Северной и Южной Америки.

Контрольныевопросы:

- 1) Что такие OLTP и OLAP системы?
- 2) Средства OLAP?
- 3) OLAP клиент с локальным кубом?
- 4) OLAP клиент и ROLAP сервер?
- 5) OLAP клиент с OLAP машиной?

Лабораторная работа № 4. Определение и развертывание куба

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Типы данных

На языке Transact-SQL используется множество различных типов данных. Всех их можно разделить на следующие группы:

Числовые типы данных: BIT (значение 0 или 1), TINYINT (от 0 до 255), SMALLINT (от -32768 до 32767), INT (от -2147483648 до 2147483647), BIGINT (от -9223372036854775808 до 9223372036854775807), DECIMAL (числа с фиксированной точностью), NUMERIC: (аналогичен типу DECIMAL), SMALLMONEY (дробные значения от -214748.3648 до 214748.3647), MONEY (дробные значения от -922337203685477.5808 до 922337203685477.5807), FLOAT (от -1.79E+308 до 1.79E+308), REAL (числа от -340E+38 до 3.40E+38);

Типы данных, представляющие дату и время: DATE (дата от 01/01/0001 до 31/12/9999), TIME (время в диапазоне от 00:00:00.0000000 до 23:59:59.9999999), DATETIME (дата и время от 01/01/1753 до 31/12/9999), DATETIME2 (дата и время от 01/01/0001 00:00:00.0000000 до 31/12/9999 23:59:59.9999999), SMALLDATETIME (дата и время от 01/01/1900 до 06/06/2079), DATETIMEOFFSET (дата и время от 01/01/0001 до 31/12/9999);

Строковые типы данных : CHAR (фиксированная строка длиной от 1 до 8000 символов), VARCHAR (переменная строка длиной от 1 до 8000 символов), NCHAR (Unicode –фиксированная строка длиной от 1 до 4000 символов), NVARCHAR (Unicode–переменная строка длиной от 1 до 4000 символов), TEXT и NTEXT (устаревшие, не рекомендуется использовать);

Бинарные типы данных: BINARY (фиксированные бинарные данные от 1 до 8000 байт), VARBINARY (переменные бинарные данные от 1 до 8000 байт), IMAGE (устаревшая, не рекомендуется использовать);

Другие типы данных: UNIQUEIDENTIFIER (уникальный идентификатор GUID), TIMESTAMP (номер версии строки в таблице), CURSOR (набор строк таблицы), HIERARCHYID (позиция в иерархии), SQL_VARIANT (данные любого типа), XML (документы или фрагменты XML), TABLE (таблица), GEOGRAPHY (географические данные, такие как широта и долгота), GEOMETRY (координаты на плоскости).

Встроенные функции Transact-SQL

Функции SQL производят действия с данными и возвращают результат. Встроенные функции делятся на три основные группы:

Скалярные функции–обрабатывают одиночное значение и возвращают одно значение. Их можно использовать везде, где допускается применение выражений.

Агрегатные функции–используются для получения обобщающих значений. Они, в отличие от скалярных функций, оперируют значениями столбцов множества строк;

–функции для списка значений.

Скалярные функции бывают следующих категорий:

строковые функции—выполняют определенные действия над строками и возвращают строковые или числовые значения;

числовые функции — возвращают числовые значения на основании заданных в аргументе значений того же типа;

функции времени и даты—выполняют различные действия над входными значениями времени и даты и возвращают строковое, числовое значение или значение в формате даты и времени;

функции преобразования типа.

Список часто используемых строковых функций:

LEN(строка)	возвращает количество символов в заданной строке
TRIM(строка) TRIM([символFROM]строка)	удаляет символ пробела или другие заданные символы в начале и в конце строки.
LTRIM(строка)	удаляет начальные пробелы из заданной строки
RTRIM(строка)	удаляет конечные пробелы из заданной строки
CHARINDEX (подстрока, строка) CHARINDEX (подстрока, строка, начальная позиция)	возвращает индекс, по которому находится первое вхождение подстроки в строке.
PATINDEX('%шаблон%', строка)	возвращает индекс, по которому находится первое вхождение определенного шаблона в строке
LEFT(строка,число)	возвращает с начала строки определенное количество символов
RIGHT(строка,число)	возвращает с конца строки определенное количество символов
SUBSTRING (строка, начальная позиция, длина)	возвращает под строку заданной длиной, начиная с данной позиции
REPLACE(строка,подстрока,замена)	заменяет одну подстроку другой
REVERSE(строка)	переворачивает строку наоборот
CONCAT(строка1,строка2[, строкаN])	объединяет заданные строки в одну
LOWER(строка)	переводит строку в нижний регистр
UPPER(строка)	переводит строку в верхний регистр
SPACE(число)	возвращает заданное количество пробелов
REPLICATE(строка,число)	повторяет значение строки указанное число раз
STUFF(строка, начальная позиция, количество, замена)	удаляет указанное количество символов первой строки в начальной позиции и вставляет на их место замену.

Список часто используемых *числовых функций*:

ABS(число)	возвращает абсолютное значение числа
CEILING(число)	возвращает наименьшее целое, большее или равное заданного числа.
FLOOR(число)	возвращает наибольшее целое число, меньшее или равное заданного числа
POWER(число,степень)	возвращает значение указанного выражения, возведенное в заданную степень
RAND([начальноезначение])	возвращает псевдослучайное значение от 0 до 1
ROUND(число,точность)	возвращает число, округленное до указанной точности

SIGN(число)	возвращает положительное (+1), нулевое (0) или отрицательное (-1) значение, обозначающее знак заданного выражения
SQRT(число)	возвращает квадратный корень данного числа
SQUARE(число)	возвращает квадрат указанного числа
PI()	возвращает константное значение π
ACOS(число)	возвращает угол в радианах, косинус которого задан арккосинус.
ASIN(число)	возвращает угол в радианах, синус которого задан арксинус.
ATAN(число)	возвращает угол в радианах, тангенс которого задан арктангенс.
COS(число)	возвращает косинус указанного угла в радианах.
SIN(число)	возвращает синус указанного угла в радианах.
TAN(число)	возвращает тангенс указанного угла в радианах.
COT(число)	возвращает котангенс указанного угла в радианах
DEGREES(число)	возвращает для значения угла в радианах соответствующее значение в градусах.
RADIANS(число)	возвращает для значения угла в градусах соответствующее значение в радианах
EXP(число)	возвращает экспонент заданного числа
LOG(число)	возвращает натуральный логарифм указанного числа
LOG(число,основа)	возвращает логарифм указанного числа
LOG10(число)	возвращает десятичный логарифм указанного числа

Список часто используемых *функций времени и даты*:

GETDATE()	возвращает текущую дату и время
CURRENT_TIMEZONE()	возвращает имя часового пояса
GETUTCDATE()	возвращает текущую дату и время по Гринвичу (UTC/GMT)
DAY(дата)	возвращает день месяца указанной даты
MONTH(дата)	возвращает номер месяца указанной даты
YEAR(дата)	возвращает год указанной даты
DATEPART(часть, дата)	возвращает целое число, представляющее указанную часть заданной даты
DATENAME(часть, дата)	возвращает строку символов, представляющую указанную часть заданной даты
DATEADD(часть, число, дата)	добавляет указанное целое число со знаком к части входного значения даты, а затем возвращает это измененное значение
DATEDIFF(часть, начальная дата, конечная дата)	возвращает разницу как целое число со знаком между частями заданных дат
EOMONTH(дата)	возвращает последний день месяца, заданной даты

Для функций времени и даты используются следующие аргументы как часть даты и времени:

Часть даты и времени	Сокращения
-----------------------------	-------------------

year	yy,yyyy
quarter	qq, q
month	mm,m
dayofyear	dy,y
day	dd, d
week	wk,ww
weekday	dw
hour	hh
minute	mi,n
second	ss, s
millisecond	ms
microsecond	mcs
nanosecond	ns
tzoffset	tz
iso_week	isowk, isoww

Список часто используемых *функций преобразования*:

CAST(выражение ASтип)	Преобразуют выражение в заданный тип
CONVERT(тип, выражение[,стиль])	
ASCII(строка)	возвращает кодASCII первого символа указанного символьноговыражения
UNICODE(строка)	возвращает код Юникод первого символа указанного символьноговыражения
CHAR(число)	возвращает символ ASCII с указанным кодом
NCHAR(число)	возвращает символ Юникода с указанным кодом
STR(число)	возвращает символьные данные, преобразованные из числовых данных

Список часто используемых *функций проверки значений*:

ISDATE(выражение)	возвращает 1, если выражение имеет допустимое значение типа даты и времени, иначе возвращает значение 0
ISNUMERIC(выражение)	возвращает 1,если выражение имеет допустимое значение числовой тип данных, иначе возвращает 0
ISNULL(выражение,замена)	заменяет значениеNULLуказанным замещающим значением
COALESCE(выражение[,...n])	вычисляет аргументы по порядку и возвращает текущее значение первого выражения, изначально не вычисленного как NULL.

Особое место среди встроенных скалярных функций языка SQL занимают функции вывода, которые являются разновидностью CASE-выражений. Функция CASE проверяет значение некоторого выражения, и в зависимости от результата проверки может возвращать тот или иной результат.

ВыражениеCASEимеет два формата:

- простое выражение CASE для определения результата сравнивает выражение с набором простых выражений;
- поисковое выражение CASE для определения результата вычисляет набор

логических выражений.

Оба формата поддерживают дополнительный аргумент ELSE.

Функция IF(условие, выражение_если_истина, выражение _ если_ложь) – возвращает одно из двух значений в зависимости от того, принимает логическое выражение значение true или false. Практическая часть

Дана таблица Академики:

ФИО	Дата_рождения	Специализация	Год_присвоения_звания
Аничков Николай Николаевич	1885-11-03	медицина	1939
Бартольд Василий Владимирович	1869-11-15	историк	1913
Белопольский Аристарх Аполлонович	1854-07-13	астрофизик	1903
Бородин Иван Парфеньевич	1847-01-30	ботаник	1902
Вальден Павел Иванович	1863-07-26	химик-технолог	1910
Вернадский Владимир Иванович	1863-03-12	геохимик	1908
Виноградов Павел Гаврилович	1854-11-30	историк	1914
Ипатьев Владимир Николаевич	1867-11-21	химик	1916
Истрин Василий Михайлович	1865-02-22	филолог	1907
Карпинский Александр Петрович	1847-01-07	геолог	1889
Коковцов Павел Константинович	1861-07-01	историк	1906
Курнаков Николай Семёнович	1860-12-06	химик	1913
Марр Николай Яковлевич	1865-01-06	лингвист	1912
Насонов Николай Викторович	1855-02-26	зоолог	1906
Ольденбург Сергей Фёдорович	1863-09-26	историк	1903
Павлов Иван Петрович	1849-09-26	физиолог	1907
Перетц Владимир Николаевич	1870-01-31	филолог	1914
Соболевский Алексей Иванович	1857-01-07	лингвист	1900
Стеклов Владимир Андреевич	1864-01-09	математик	1912

Пример 1: Вывести ФИО академиков и длину ФИО: **SELECT**

ФИО

, **LEN**(ФИО) **AS** Количество_символов

FROM

Академики

Пример 2: Вывести список академиков, убрать лишние пробелы в ФИО: **SELECT**

TRIM(ФИО) **AS** ФИО

, Дата_рождения

, Специализация

, Год_присвоения_звания

FROM

Академики

Пример 3: Найти позиции буквы «о» в ФИО каждого академика. Вывести ФИО и позицию:

```
SELECT
ФИО
,CHARINDEX('о',ФИО)ASПозиция_о
FROM
Академики
```

Пример 4: Вывести ФИО и первые три буквы специализации каждого академика:SELECT

```
ФИО
,LEFT(Специализация,3)ASСпец_3
FROM
Академики
```

Пример 5:Вывести ФИО и от второй до пятой буквы специализации каждого академика:

```
SELECT
ФИО
,SUBSTRING(Специализация,2,4)ASСпец_2_5
```

FROM

Академики

Пример 6: Вывести список академиков, заменить специализацию «ЛИНГВИСТ» на «ЯЗЫКОВЕД»:

SELECT

ФИО

,Дата_рождения

,REPLACE(Специализация,'лингвист','языковед')ASСпец

,Год_присвоения_звания

FROM

Академики

Пример 7: Вывести список академиков, специализацию на верхнем регистре:SELECT

ФИО

,Дата_рождения

,UPPER(Специализация)ASСпец

,Год_присвоения_звания

FROM

Академики

Пример 8: Вывести ФИО академиков в правильном и обратном виде:SELECT

ФИО

,REVERSE(ФИО)ASФИО_Обр

FROM

Академики

Название

Пример 9: Вывести каждую специализацию 4 раза в одной строке. Убрать дубликаты:SELECTDISTINCT

REPLICATE(Специализация,4)ASСпец_4

FROM

Академики

Пример 10: Вывести абсолютное значение тригонометрических функций на точке π :SELECT

ABS(COS(PI()))ASKосинус_Пи

```
,ABS(SIN(PI()))ASСинус_Пи  
,ABS(TAN(PI()))ASТангенс_Пи  
,ABS(COT(PI()))ASКоТангенс_Пи
```

Пример 11: Вывести число 132.456, округленное с точностью от 3 до -3:
`SELECT`

```
ROUND(123.456,3)ASОкр3  
,ROUND(123.456,2)ASОкр2  
,ROUND(123.456,1)ASОкр1  
,ROUND(123.456,0)ASОкр0  
,ROUND(123.456,-1)ASОкр_1  
,ROUND(123.456,-2)ASОкр_2  
,ROUND(123.456,-3)ASОкр_3
```

Пример 12: Вывести наименьшее целое число, которое больше или равно 123.456, и наибольшее целое число, которое меньше или равно 123.456:

```
SELECT  
CEILING(123.456)ASБольше  
,FLOOR(123.456)ASМеньше
```

Пример 13: Вывести квадратный корень, квадрат и куб числа 25:
`SELECT`

```
SQRT(25)ASКорень  
,SQUARE(25)ASКвадрат  
,POWER(25,3)ASКуб
```

Пример 14: Вывести текущую дату и время:

```
SELECT  
GETDATE()ASСейчас
```

Пример 15: Вывести день, месяц, год, час, минуту, секунду, номер квартала, номер недели, день года, день недели для текущей даты и времени:

```
SELECT  
DAY(GETDATE())ASДень  
,MONTH(GETDATE())ASМесяц  
,YEAR(GETDATE())ASГод
```

```
,DATEPART(HOUR,GETDATE())ASЧас
,DATEPART(MINUTE,GETDATE())ASМинута
,DATEPART(SECOND, GETDATE())ASСекунда
,DATEPART(QUARTER,GETDATE())ASKвартал
,DATEPART(WEEK,GETDATE())ASНеделя
,DATEPART(DAYOFYEAR,GETDATE())ASДень_года
,DATEPART(WEEKDAY,GETDATE())ASДень_недели
```

Пример 16: Вывести дату 100 дней назад от текущей:

```
SELECT
DATEADD(DAY,-100, GETDATE())ASДень_100_Назад
```

Пример 17: Академик Игорь Евгеньевич Тамм родился 8 июля 1895 года. И.Е. Тамм скончался 12 апреля 1971 года. Вывести количество прожитых дней:

```
SELECT
DATEDIFF(DAY,'18950708','19710412')ASКоличество_прожитых_дней
```

Пример 18: Вывести ФИО и время года рождения каждого академика: SELECT

```
ФИО
, CASEMONTH(Дата_рождения)WHEN3 THEN'Весна'WHEN4
THEN'Весна'WHEN5 THEN'Весна'WHEN6 THEN'Лето'WHEN7
THEN'Лето'WHEN8 THEN'Лето'WHEN9 THEN'Осень'WHEN10
THEN'Осень'WHEN11 THEN'Осень'ELSE'Зима'
ENDASВремени_года
FROMАкадемики
```

Пример 19: Вывести ФИО, дату рождения и знак зодиака каждого академика: SELECT

```
ФИО
,Дата_рождения
,CASE
WHEN(MONTH(Дата_рождения)=3ANDDAY(Дата_рождения) >=
21)OR(MONTH(Дата_рождения)=4ANDDAY(Дата_рождения)<=20)THEN'Ове
н'
WHEN(MONTH(Дата_рождения)=4ANDDAY(Дата_рождения) >=
21)OR(MONTH(Дата_рождения)=5ANDDAY(Дата_рождения)<=21)THEN'Тел
```

ещ'

```
WHEN(MONTH(Дата_рождения)=5ANDDAY(Дата_рождения) >=
22)OR(MONTH(Дата_рождения)=6ANDDAY(Дата_рождения)<=21)THEN'Бли
знецы'
```

```
WHEN(MONTH(Дата_рождения)=6ANDDAY(Дата_рождения) >=
22)OR(MONTH(Дата_рождения)=7ANDDAY(Дата_рождения)<=22)THEN'Рак'
```

```
WHEN(MONTH(Дата_рождения)=7ANDDAY(Дата_рождения) >=
23)OR(MONTH(Дата_рождения)=8ANDDAY(Дата_рождения)<=21)THEN'Лев'
```

```
WHEN(MONTH(Дата_рождения)=8ANDDAY(Дата_рождения) >=
22)OR(MONTH(Дата_рождения)=9ANDDAY(Дата_рождения)<=23)THEN'Дев
а'
```

```
WHEN(MONTH(Дата_рождения)=9ANDDAY(Дата_рождения) >=
24)OR(MONTH(Дата_рождения)=10ANDDAY(Дата_рождения)<=23)THEN'Ве
сы'
```

```
WHEN(MONTH(Дата_рождения)=10 ANDDAY(Дата_рождения) >=
24)OR(MONTH(Дата_рождения)=11ANDDAY(Дата_рождения)<=22)THEN'Ск
orpion'
```

```
WHEN(MONTH(Дата_рождения)=11 ANDDAY(Дата_рождения) >=
23)OR(MONTH(Дата_рождения)=12ANDDAY(Дата_рождения)<=22)THEN'Ст
релец'
```

```
WHEN(MONTH(Дата_рождения)=12 ANDDAY(Дата_рождения) >=
23)OR(MONTH(Дата_рождения)=1ANDDAY(Дата_рождения)<=20)THEN'Коз
ерог'
```

```
WHEN(MONTH(Дата_рождения)=1ANDDAY(Дата_рождения) >=
21)OR(MONTH(Дата_рождения)=2ANDDAY(Дата_рождения)<=19)THEN'Вод
олей'
```

```
WHEN(MONTH(Дата_рождения)=2ANDDAY(Дата_рождения) >=
20)OR(MONTH(Дата_рождения)=3ANDDAY(Дата_рождения)<=20)THEN'Рыб
ы'
```

```
ENDASЗнак_зодиакаFROMАкадемики
```

Пример 20: Вывести список академиков. Для каждого академика, в зависимости от возраста, при присвоении звания вывести «молодой» или «старый» в дополнительном столбце:

```
SELECT
ФИО
```

```

,Дата_рождения
,Специализация
,Год_присвоения_звания
,IF(Год_присвоения_звания - Year(Дата_рождения) <= 45,
'Молодой','Старый')ASВозраст_при_присвоении
FROMАкадемики

```

Задание

1. Вывести список академиков, отсортированный по количеству символов в ФИО.
2. Вывести список академиков, убрать лишние пробелы в ФИО.
3. Найти позиции «ов» в ФИО каждого академика. Вывести ФИО и номер позиции.
4. Вывести ФИО и последние две буквы специализации для каждого академика.
5. Вывести список академиков, ФИО в формате Фамилия и Инициалы.
6. Вывести список специализаций в правильном и обратном виде. Убрать дубликаты.
7. Вывести свою фамилию в одной строке столько раз, сколько вам лет.
8. Вывести абсолютное значение функций $\sin 2(\pi) - \cos(3\pi)$ с точностью два знака после десятичной запятой.
9. Вывести количество дней до конца семестра.
10. Вывести количество месяцев от вашего рождения.
11. Вывести ФИО и високосность года рождения каждого академика.
12. Вывести список специализаций без повторений. Для каждой специализации вывести «длинный» или «короткий», в зависимости от количества символов.

Контрольные вопросы:

- 1) Что такое измерения и для чего они нужны? Какие типы измерений Вы использовали в этой лабораторной работе?
- 2) Что такое источник данных, представление источника данных и куб?
- 3) Что такие пользовательские иерархии и для чего они нужны?

Лабораторная работа № 5. Определение и развертывание куба

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Агрегатные функции используются для получения обобщающих значений. Они, в отличие от скалярных функций, оперируют значениями столбцов множества строк. К агрегатным функциям относятся:

- SUM–вычисляет итог;
- MAX–возвращает наибольшее значение;
- MIN–возвращает наименьшее значение;
- AVG–вычисляет среднее значение;
- COUNT–вычисляет количество значений в столбце.

Аргументам и функций выступают поля таблицы или результаты выражений над ними.

Вложенность не допускается.

Из агрегатных функций можно составлять любые выражения.

Для функцийSUMиAVGстолбец должен содержать числовые значения.

Для функцийCOUNT()можно указать аргумент*для подсчета всех строк без исключения.

По умолчанию вышеперечисленные пять функций учитывают все строки выборки для вычисления результата.

Но выборка может содержать повторяющиеся значения. Если необходимо выполнить вычисления только над уникальными значениями, исключив из набора значений повторяющиеся данные, то для этого применяется оператор DISTINCT (кроме COUNT(*)). По умолчанию вместо DISTINCT применяется оператор ALL, который выбирает все строки. Так как этот оператор неявно подразумевается при отсутствии DISTINCT, то его можно не указывать.

Агрегатные функции можно применить не только на всю таблицу, но также на группу значений. Для этого применяется команда GROUPBY, которая пишется после WHERE. После команды GROUPBY перечисляется название столбцов, по которым следует группировать данные. Предложение GROUPBY указывает, что результаты запроса следует разделить

на группы, применить агрегатную функцию по отдельности к каждой группе и получить для каждой группы одну строку результатов.

В качестве элемента группировки должен выступать любой возвращаемый элемент, указанный в предложении SELECT, кроме значений агрегатных функций.

Если столбец, по которому производится группировка, содержит значение NULL, то строки со значением NULL составят отдельную группу.

Команда HAVING<условие> применяется для фильтрации строк, возвращаемых при использовании предложения GROUPBY. HAVING пишется после GROUPBY, имеет такой формат, как WHERE, но в качестве значения используется значение, возвращаемое агрегатными функциями.

Практическая часть

Дана таблица Страны:

Название	Столица	Площадь	Население	Континент
Австрия	Вена	83858	8741753	Европа
Азербайджан	Баку	86600	9705600	Азия
Албания	Тирана	28748	2866026	Европа
Алжир	Алжир	2381740	39813722	Африка
Ангола	Луанда	1246700	25831000	Африка
Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
Афганистан	Кабул	647500	29822848	Азия
Бангладеш	Дакка	144000	160221000	Азия
Бахрейн	Манама	701	1397000	Азия
Белиз	Бельмопан	22966	377968	Северная Америка
Белоруссия	Минск	207595	9498400	Европа
Бельгия	Брюссель	30528	11250585	Европа
Бенин	Порто-Ново	112620	11167000	Африка
Болгария	София	110910	7153784	Европа
Боливия	Сукре	1098580	10985059	Южная Америка
Ботсвана	Габороне	600370	2209208	Африка
Бразилия	Бразилиа	8511965	206081432	Южная Америка
Буркина-Фасо	Уагадугу	274200	19034397	Африка
Бутан	Тхимпху	47000	784000	Азия
Великобритания	Лондон	244820	65341183	Европа
Венгрия	Будапешт	93030	9830485	Европа
Венесуэла	Каракас	912050	31028637	Южная Америка
Восточный Тимор	Дили	14874	1167242	Азия
Вьетнам	Ханой	329560	91713300	Азия

Пример 1: Вывести максимальную площадь стран: **SELECT**

MAX(Площадь)**AS**Макс_площадь
FROM
Страны

Пример 2: Вывести наименьшее население стран в Африке:**SELECT**
MIN(Население)**AS**Мин_население
FROM
Страны
WHERE
Континент='Африка'

Пример 3: Вывести суммарное население стран Северной и Южной Америки:**SELECT**
SUM(Население)**AS**Суммарное_население
FROM
Страны
WHERE
Континент = 'Северная Америка' OR Континент='Южная Америка'

Пример 4: Вывести среднее население стран, кроме европейских.
Результат округлить до двух знаков:
SELECT
ROUND(**AVG**(**CAST**(Население**AS****FLOAT**)),2)**AS** Среднее_население
FROM
Страны
WHERE
Континент!='Европа'

Пример 5: Вывести количество стран, название которых начинается с буквы «С»:
SELECT
COUNT(*)**AS**Количество
FROM
Страны
WHERE
LEFT(Название,1)='С'

Пример 6: Вывести количество континентов, где есть страны:
`SELECT
COUNT(DISTINCT Континент) AS Количество_Континентов
FROM
Страны`

Пример 7: Вывести разницу населения между странами с наибольшим и наименьшим количеством граждан:
`SELECT
MAX(Население) - MIN(Население) AS Разница
FROM
Страны`

Пример 8: Вывести количество стран на каждом континенте. Результат отсортировать по количеству стран по убыванию:
`SELECT
Континент
, COUNT(Название) AS Количество_Стран FROM
Страны GROUP BY
Континент ORDER BY
Количество_Стран DESC`

Пример 9: Вывести количество стран по первым буквам в названии. Результат отсортировать в алфавитном порядке:
`SELECT
LEFT(Название, 1) AS Первая_буква
, COUNT(Название) AS Количество_Стран
FROM
Страны
GROUP BY
LEFT(Название, 1) ORDER BY
Первая_буква`

Пример 10: Вывести список континентов, где плотность населения больше, чем 100 чел. на кв. км:
`SELECT
Континент
, AVG(CAST(Население AS FLOAT) / Площадь) AS Сред_Плотность`

FROM

Страны

GROUPBY

КонтинентHAVING

AVG(CAST(НаселениеASFLOAT)/Площадь)>100

Пример 11: Ожидается, что через 25 лет население Европы и Азии вырастет на 20%,Северной Америки и Африки на 50%, а остальных частей мира – на 70%. Вывести список континентов с прогнозируемым населением:

SELECT

Континент

,CASE

WHENКонтинентIN('Европа','Азия')THENFLOOR(SUM(Население)*1.2)

FROM

WHENКонтинентIN('СевернаяАмерика','Африка')THENFLOOR(SUM(Население)*1.5)

-ELSEFLOOR(SUM(Население) * 1.7)ENDASСуммарное_Население
Страны

GROUPBY

Континент

Пример 12: Вывести список континентов, где разница по населению между наибольшими и наименьшими странами не более в1000раз:

SELECT

Континент

FROM

Страны

GROUPBY

Континент

HAVINGMAX(Население)<=1000*MIN(Население)

Пример 13: Вывести количество стран, у которых нет столицы (не введена в базу):

SELECT

```
COUNT(*)ASКоличество
FROM
Страны
WHERE
СтолицаISNULL
```

Пример 14: Вывести количество символов в самых длинных и коротких названиях стран и столиц:

```
SELECT
MAX(LEN(Название))ASДл_Название
,MAX(LEN(Столица))ASДл_Столица
,MIN(LEN(Название))ASKр_Название
,MIN(LEN(Столица))ASKр_Столица
FROM
Страны
```

Пример 15: Вывести список континентов, у которых средняя плотность среди стран с площадью более 1млн.кв.км больше, чем 30 чел. на кв. км. Результат отсортировать по плотности по убыванию:

```
SELECT
Континент
,AVG(CAST(НаселениеASFLOAT)/Площадь)ASПлотность
FROM
Страны
WHERE
Площадь>1000000GROUPBY
КонтинентHAVING
AVG(CAST(НаселениеAS FLOAT) / Площадь) >30ORDERBY
ПлотностьDESC
```

Задание

1. Вывести минимальную площадь стран.
2. Вывести наибольшую по населению страну в Северной и Южной Америке.
3. Вывести среднее население стран. Результат округлить до одного знака.

4. Вывести количество стран, у которых название заканчивается на «ан», кроме стран, у которых название заканчивается на «стан».

5. Вывести количество континентов, где есть страны, название которых начинается с буквы «Р».

6. Сколько раз страна с наибольшей площадью больше, чем страна с наименьшей площадью?

7. Вывести количество стран с населением больше, чем 100 млн. чел. на каждом континенте. Результат отсортировать по количеству стран по возрастанию.

8. Вывести количество стран по количеству букв в названии. Результат отсортировать по убыванию.

9. Ожидается, что через 20 лет население мира вырастет на 10%. Вывести список континентов с прогнозируемым населением:

10. Вывести список континентов, где разница по площади между наибольшими и наименьшими странами не более 10000 раз:

11. Вывести среднюю длину названий Африканских стран.

12. Вывести список континентов, у которых средняя плотность среди стран с населением более 1 млн. чел. больше, чем 30 чел. на кв. км.

Контрольные вопросы:

1. Что такое измерения и для чего они нужны? Какие типы измерений Вы использовали в этой лабораторной работе?

2. Что такое источник данных, представление источника данных и куб?

3. Что такое пользовательские иерархии и для чего они нужны?

Лабораторная работа № 6. Изменение мер, атрибутов и иерархий

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Данные часто хранятся в несколько связанных таблицах. Для выбора данных используются разные методы соединения таблиц.

Когда выбор данных осуществляется из нескольких таблиц, в конструкции **SELECT** для каждого поля указывается таблица в виде **<таблица>.<поле>**. Если название поля уникальное, то можно его указать без таблицы, иначе это обязательно, чтобы избежать коллизий. Чтобы не повторить длинные названия таблиц, можно использовать псевдоним для таблиц. Псевдоним указывается в конструкции как **FROM**.

При неявном соединении таблиц формат конструкций **FROM** и **WHERE** имеет следующий вид:

FROM<таблица1> [псевдоним1], <таблица2> [псевдоним2]...
[**WHERE**<условие_соединения>[**AND**<условие_поиска>]...]

Таким образом, если более одной таблицы присутствует в конструкции **FROM**, то их разделяют запятой.

Если условие соединения не указывать, тогда результат будет декартовым произведением, то есть для каждой строки одной из таблиц берутся все возможные сочетания строк из других таблиц.

Для явного соединения таблиц используется команда **JOIN**. У явного соединения есть следующие разновидности:

– Внутреннее соединение – осуществляется с помощью команды **INNER JOIN**. Из двух таблиц берутся только связанные строки. **INNER JOIN** имеет следующий формат записи:

<таблица1>**INNER JOIN**<таблица2>**ON**<таблица1>.<связующее_поле>=<таб-лица2>.<связующее_поле>.

При внутреннем соединении слово **INNER** можно пропустить.

– Внешнее соединение – осуществляется с помощью команды **OUTER JOIN**. **OUTER JOIN** имеет следующий формат записи:

<таблица1>**LEFT** | **RIGHT** |
FULL OUTER JOIN<таблица2>**ON**<таблица1>.<связую-

щее_поле>=<таблица2>.<связующее_поле>.

У внешнего соединения есть три разновидности:

–Левое внешнее соединение LEFT OUTER JOIN – из таблицы, название которой является левым операндом команды JOIN, выбираются все строки, из второй таблицы – только те записи, которые имеют связь с первой таблицей.

–Правое внешнее соединение RIGHT OUTER JOIN – из таблицы, название которой является правым операндом команды JOIN, выбираются все строки, из первой таблицы только те записи, которые имеют связь со второй таблицей.

–Полное внешнее соединение FULL OUTER JOIN – из обеих таблиц выбираются все строки.

При использовании внешних соединений, слово OUTER можно пропустить.

–Перекрестное соединение CROSS JOIN – декартово произведение двух таблиц.

CROSS JOIN имеет следующий формат записи:

<таблица1>CROSS JOIN <таблица2>.

При выборе данных из трех и более таблиц, с помощью явного соединения, результат зависит от порядка соединения.

Используя соединение, можно связать таблицу с собой. При таком соединении псевдоним обязателен.

Набор данных, полученных из нескольких таблиц, не отличается от набора, полученного из одной таблицы. Ему тоже можно применить группировки и т.д.

Практическая часть

Даны следующие таблицы:

Таблица 1. Факультет

Аббревиатура	Название
Ен	Естественные науки
Гн	Гуманитарные науки
Ит	Информационные технологии
Фм	Физико-математический

Таблица 2. Кафедра

Шифр	Название	Факультет
------	----------	-----------

вм	Высшаяматематика	ен
ис	Информационныесистемы	ит
мм	Математическоемоделирование	фм
оф	Общаяфизика	ен
пи	Прикладнаяинформатика	ит
эф	Экспериментальнаяфизика	фм

Таблица 3. *Сотрудник*

Таб_номер	Шифр	Фамилия	Должность	Зарплата	Шеф
101	пи	ПрохоровП.П.	зав.кафедрой	35000,00р.	101
102	пи	СеменовС.С.	преподаватель	25000,00р.	101
105	пи	ПетровП.П.	преподаватель	25000,00р.	101
153	пи	СидороваС.С.	инженер	15000,00р.	102
201	ис	АндреевА.А.	зав.кафедрой	35000,00р.	201
202	ис	БорисовБ.Б.	преподаватель	25000,00р.	201
241	ис	ГлуховГ.Г.	инженер	20000,00р.	201
242	ис	ЧерновЧ.Ч.	инженер	15000,00р.	202
301	мм	БасовБ.Б.	зав.кафедрой	35000,00р.	301
302	мм	СергееваС.С.	преподаватель	25000,00р.	301
401	оф	ВолковВ.В.	зав.кафедрой	35000,00р.	401
402	оф	ЗайцевЗ.З.	преподаватель	25000,00р.	401
403	оф	СмирновС.С.	преподаватель	15000,00р.	401
435	оф	ЛисинЛ.Л.	инженер	20000,00р.	402
501	вм	КузнецовК.К.	зав.кафедрой	35000,00р.	501
502	вм	РоманцевР.Р.	преподаватель	25000,00р.	501
503	вм	СоловьевС.С.	преподаватель	25000,00р.	501
601	эф	ЗверевЗ.З.	зав.кафедрой	35000,00р.	601
602	эф	СорокинаС.С.	преподаватель	25000,00р.	601
614	эф	ГригорьевГ.Г.	инженер	20000,00р.	602

Таблица 4. *Специальность*

Номер	Направление	Шифр
01.03.04	Прикладная математика	мм
09.03.02	Информационные системы и технологии	ис
09.03.03	Прикладная информатика	пи
14.03.02	Ядерные физика и технологии	эф
38.03.05	Бизнес-информатика	ис

Таблица 5. *Дисциплина*

Код	Объем	Название	Исполнитель
101	320	Математика	вм
102	160	Информатика	пи
103	160	Физика	оф

202	120	Базыданных	ис
204	160	Электроника	эф
205	80	Программирование	пи
209	80	Моделирование	мм

Таблица 6.Заявка

Номер	01.03.04				09.03.02					09.03.03					14.03.02				38.03.05				
Код	101	205	209		101	102	103	202	205	209	101	102	103	202	205	101	102	103	204	101	103	202	209

Таблица 7.Зав_кафедрой

Таб_номер	101	201	301	401	501	601
Стаж	15	18	20	10	18	8

Таблица 8. Инженер

Таб_номер	153	241	242	435	614
Специальность	электроник	электроник	программист	электроник	программист

Таблица 9. Преподаватель

Таб_номер	Звание	Степень
101	профессор	д. т.н.
102	доцент	к.ф.-м.н.
105	доцент	к.т.н.
201	профессор	д.ф.-м.н.
202	доцент	к.ф.-м.н.
301	профессор	д. т.н.
302	доцент	к.т.н.
401	профессор	д. т.н.
402	доцент	к.т.н.
403	ассистент	—
501	профессор	д.ф.-м.н.
502	профессор	д.ф.-м.н.
503	доцент	к.ф.-м.н.
601	профессор	д.ф.-м.н.

Таблица 10.Студент

Рег_номер	Номер	Фамилия
10101	09.03.03	НиколаеваН.Н.
10102	09.03.03	ИвановИ.И.

10103	09.03.03	КрюковК. К.
20101	09.03.02	АндреевА.А.
20102	09.03.02	ФедоровФ.Ф.
30101	14.03.02	БондаренкоБ.Б.
30102	14.03.02	ЦветковК.К.
30103	14.03.02	ПетровП.П.
50101	01.03.04	СергеевС. С.
50102	01.03.04	КудрявцевК.К.
80101	38.03.05	МакаровМ.М.
80102	38.03.05	ЯковлевЯ.Я.

Таблица 11. Экзамен

Дата	Код	Рег_номер	Таб_номер	Аудитория	Оценка
05.06.2015	102	10101	102	т505	4
05.06.2015	102	10102	102	т505	4
05.06.2015	202	20101	202	т506	4
05.06.2015	202	20102	202	т506	3
07.06.2015	102	30101	105	ф419	3
07.06.2015	102	30102	101	т506	4
07.06.2015	102	80101	102	м425	5
09.06.2015	205	80102	402	м424	4
09.06.2015	209	20101	302	ф333	3
10.06.2015	101	10101	501	т506	4
10.06.2015	101	10102	501	т506	4
10.06.2015	204	30102	601	ф349	5
10.06.2015	209	80101	301	э105	5
10.06.2015	209	80102	301	э105	4
12.06.2015	101	80101	502	с324	4
15.06.2015	101	30101	503	ф417	4
15.06.2015	101	50101	501	ф201	5
15.06.2015	101	50102	501	ф201	3
15.06.2015	103	10101	403	ф414	4
17.06.2015	102	10101	102	т505	5

Пример 1: Выбрать факультет и кафедры, используя неявное соединение.

Результат отсортировать по алфавиту:

SELECT

Ф.Название**AS**Факультет

,К.Название**AS**Кафедра

FROM

ФакультетФ,КафедраК

WHERE

Ф.Аббревиатура = К.Факультет
ORDERBY
Факультет,Кафедра

Пример 2:Выбрать факультет и кафедры, используя явное соединение.
Результат отсортировать по алфавиту:

SELECT

Ф.НазваниеASФакультет
,К.НазваниеASКафедра

FROM

ФакультетФ

INNERJOINКафедраKONФ.Аббревиатура=К.Факультет

ORDERBY

Факультет,Кафедра

Пример 3:Выбрать все факультеты и их кафедры, если существуют.
Результат отсортировать по алфавиту:

SELECT

Ф.НазваниеASФакультет
,К.НазваниеASКафедра

FROM

ФакультетФ

LEFTOUTERJOINКафедраKONФ.Аббревиатура =К.Факультет

ORDERBY

Факультет,Кафедра

Пример 4:Вывести из таблиц «Кафедра», «Специальность» и «Студент»
данные о студентах:

SELECT

С.Фамилия
,П.Направление
,К.НазваниеASКафедра

FROM

Студент С

INNERJOINСпециальность П ONС.Номер = П.НомерINNERJOINКафедра

KONП.Шифр =К.Шифр

Пример 5: Вывести для каждого сотрудника фамилию, должность, зарплату и фамилию его непосредственного руководителя:

SELECT

С.Фамилия

,С.Должность

,С.Зарплата

,П.Фамилия AS Руководитель

FROM

СотрудникС

INNER JOIN СотрудникП ON С.Шеф=П.Таб_номер

Пример 6: Вывести список студентов, сдавших хотя бы один экзамен. По правилам соединения, студенты, не сдававшие экзамены, в выборке представлены не будут:

SELECT

С.Фамилия

FROM

СтудентС

INNER JOIN ЭкзаменЭ ON С.Рег_номер=Э.Рег_номер

GROUP BY

С.Фамилия

Пример 7: Вывести из таблиц «Студент» и «Экзамен» учетные номера и фамилии студентов, а так же количество сданных экзаменов и средний балл для каждого студента:

SELECT

С.Фамилия

,COUNT(Э.Оценка) AS [Количество экзаменов]

,AVG(Э.Оценка) AS [Средний балл]

FROM

СтудентС

INNER JOIN ЭкзаменЭ ON С.Рег_номер=Э.Рег_номер

GROUP BY

С.Фамилия

Пример 8: Вывести список заведующих кафедрами и их зарплаты, и стаж работы:

SELECT

С.Фамилия
,С.Зарплата
,З.Стаж
FROM
СотрудникС
INNERJOINЗав_кафедройЗ**ON**С.Таб_номер=З.Таб_номер

Пример 9: Вывести список кандидатов и докторов физико-математических наук:

SELECT
С.Фамилия
,П.Степень
FROM
СотрудникС
INNERJOINПреподавательП**ON**С.Таб_номер=П.Таб_номер
WHERE
П.СтепеньIN('к.ф.-м.н.','д.ф.-м.н.')

Пример 10: Вывести название дисциплины, фамилию, должность и степень преподавателя, дату и место проведения экзаменов в хронологическом порядке:

SELECTDISTINCT
Д.Название**AS**Дисциплина
,С.Фамилия
,С.Должность
,П.Степень
,Э.Дата
,Э.Аудитория
FROM
ЭкзаменЭ
INNERJOINДисциплинаД**ON**Э.Код=Д.Код
INNER JOIN Сотрудник**CON** Э.Таб_номер=
С.Таб_номерINNERJOINПреподавательП**ON**Э.Таб_номер=П.Таб_номер
ORDERBY
Э.Дата

Пример 11: Вывести фамилию преподавателей и количество их экзаменов:**SELECT**

С.Фамилия

,**COUNT**(Э.Дата)**AS**[Количествоэкзаменов]

FROM

ЭкзаменЭ

INNERJOINСотрудник**CON**Э.Таб_номер=С.Таб_номер

GROUPBY

С.Фамилия

Пример 12: Вывести список студентов, не сдавших ни одного экзамена:**SELECT**

С.Фамилия

FROM

СтудентС

LEFTOUTERJOINЭкзаменЭ**ON**С.Рег_номер=Э.Рег_номер

WHERE

Э.Рег_номер**ISNULL**

Задание

1. Вывести из таблиц «Кафедра», «Специальность» и «Студент» данные о студентах, которые обучаются на данном факультете (например, «ИТ»).
2. Вывести из таблиц «Кафедра», «Специальность» и «Сотрудник» данные о выпускающих кафедрах (факультет, шифр, название, фамилию заведующего).
3. Выпускающей считается та кафедра, на которую есть ссылки в таблице «Специальность».
4. Вывести в запросе для каждого сотрудника номер и фамилию его непосредственного руководителя.
5. Для заведующих кафедрами поле руководителя оставить пустым.
6. Вывести список студентов, сдавших минимум два экзамена.
7. Вывести список инженеров с зарплатой, меньшей 20000руб.
8. Вывести список студентов, сдавших экзамены в заданной аудитории.
9. Вывести из таблиц «Студент» и «Экзамен» учетные номера и фамилии студентов, а также количество сданных экзаменов и средний балл для каждого студента только для тех студентов, у которых средний балл не меньше заданного (например, 4).

10. Вывести список заведующих кафедрами и их зарплаты, и степень.
11. Вывести список профессоров.
12. Вывести название дисциплины, фамилию, должность и степень преподавателя, дату и место проведения экзаменов в хронологическом порядке в заданном интервале даты.
13. Вывести фамилию преподавателей, принявших более трех экзаменов.
14. Вывести список студентов, не сдавших ни одного экзамена в указанной дате.

Контрольные вопросы:

1. Что такое измерения и для чего они нужны? Какие типы измерений Вы использовали в этой лабораторной работе?
2. Опишите процесс изменения меры.
3. Что такое атрибут, какие свойства могут быть у атрибута?

Лабораторная работа № 7.

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

В зависимости от контекста запрос SELECT может вернуть результат в одном из трех видов:

- таблица–запрос возвращает набор строк и столбцов;
- список значений–запрос возвращает значения только одного столбца, но, возможно, в нескольких строках;
- скалярное значение – запрос возвращает значение одного столбца в одной строке. Результат запроса можно использовать в других запросах. Место использования запроса зависит от вида возвращаемого значения.

В предложении SELECT может использоваться только скалярный подзапрос, который возвращает одно значение.

Подзапросы пишутся в скобках.

Подзапросы, которые используются в предложении FROM, должны иметь название для всех столбцов.

Подзапросы, которые используются в предложении FROM, должны иметь псевдоним. Подзапросы, которые используются в предложении WHERE для сравнения со столбцом, должны возвращать значения соответствующего типа.

Подзапросы, которые используются в предложении WHERE для сравнения со столбцом, должны быть вторым операндом оператора сравнения.

Каждый вложенный запрос, в свою очередь, может содержать один или более вложенный запрос. В инструкцию можно вложить любое количество запросов (в практике, до 32). Подзапросы выполняются, начиная с самого глубокого.

Подзапросы бывают коррелирующими и некоррелирующими. В некоррелирующих подзапросах команды выполняются один раз, то есть результат подзапроса не зависит от строк, выбранных в основном запросе. Такой подзапрос выполняется один раз для всего внешнего запроса.

Также существуют коррелирующие подзапросы, результаты которых зависят от строк, выбранных в основном запросе. Коррелирующие подзапросы имеют связь с внешним запросом выполняется столько раз, сколько строк в основном запросе.

Выполнение коррелирующих подзапросов сильно влияет на эффективность выполнения, поэтому их необходимо использовать только в крайних случаях.

Команду IN можно применить к результатам подзапросов, возвращающих список значений.

В предложении WHERE значение столбца можно сравнить со списками значений, возвращаемых под запросом. Для этого используются

операторы ALL и ANY|SOME.

При использовании оператора ALL условие в операции сравнения должно быть верно для всех значений, возвращаемых под запросом.

При использовании оператора ANY|SOME условие в операции сравнения должно быть верно для всех значений, возвращаемых под запросом.

Оператор EXISTS проверяет, возвращает ли подзапрос какое-либо значение. Здесь важны не данные, а их существование.

Практическая часть

Дана таблица **Страны**:

Название	Столица	Площадь	Население	Континент
Австрия	Вена	83858	8741753	Европа
Азербайджан	Баку	86600	9705600	Азия
Албания	Тирана	28748	2866026	Европа
Алжир	Алжир	2381740	39813722	Африка
Ангола	Луанда	1246700	25831000	Африка
Аргентина	Буэнос-Айрес	2766890	43847000	ЮжнаяАмерика
Афганистан	Кабул	647500	29822848	Азия
Бангладеш	Дакка	144000	160221000	Азия
Бахрейн	Манама	701	1397000	Азия
Белиз	Бельмопан	22966	377968	Северная Америка
Белоруссия	Минск	207595	9498400	Европа
Бельгия	Брюссель	30528	11250585	Европа
Бенин	Порто-Ново	112620	11167000	Африка
Болгария	София	110910	7153784	Европа
Боливия	Сукре	1098580	10985059	ЮжнаяАмерика
Ботсвана	Габороне	600370	2209208	Африка
Бразилия	Бразилиа	8511965	206081432	ЮжнаяАмерика
Буркина-Фасо	Уагадугу	274200	19034397	Африка
Бутан	Тхимпху	47000	784000	Азия
Великобритания	Лондон	244820	65341183	Европа
Венгрия	Будапешт	93030	9830485	Европа
Венесуэла	Каракас	912050	31028637	ЮжнаяАмерика
ВосточныйТимор	Дили	14874	1167242	Азия
Вьетнам	Ханой	329560	91713300	Азия

Пример 1: Вывести список стран и процентное соотношение их населения к суммарному населению мира:

SELECT

Название

,Столица

,Площадь

```

,Население
,Континент
,ROUND(CAST(НаселениеAS
FLOAT) * 100 /(
SELECT
SUM(Население)
FROM
Страны
FROM),
3)ASПроцент

```

```

Страны
ORDERBY
ПроцентDESC

```

Пример 2::Вывести список стран мира, население которых больше, чем среднее население всех стран мира:

```

SELECT
Название
,Столица
,Площадь
,Население
,Контине
нтFROM
Страны

```

```

WHERE
Население>(
SELECT
AVG(Население)
FROM
Страны)

```

Пример 3: С помощью подзапроса вывести список африканских стран, население которых больше 50 млн. чел.:

```

SELECT
Название
,Столица
,Площадь
,Население

```

```

        ,Контин
ентFROM
(SELECT
Т
Название
,Столица
,Площадь
,Население
,КонтинентFROM
        Стр
аныWHE
RE
Континент='Африка')A
WHERE
Население>50000000

```

Пример 4: Вывести список стран и процентное соотношение их населения к суммарному населению к той части мира, где они находятся:

```

SELECT
Название
,Столица
,Площадь
,Население
,Континент
,ROUND(CAST(НаселениеAS
FLOAT) * 100 /(
SELECT
SUM(Население)
FROM
Страны Б
WHERE
А.Континент=Б.Континент
),3)ASПроцент
FROM
Страны А

```

ORDERBY

ПроцентDESC

Пример 5: Вывести список стран мира, население которых больше, чем среднее население стран в той части света, где они находятся:

SELECT

Название

,Столица

,Площадь

,Население

,КонтинентFROM

Страны АWHERE

Население>(

SELECT

AVG(Население)

FROM

Страны Б

WHERE

Б.Континент=А.Континент

)

Пример 6: Вывести список стран мира, которые находятся в тех частях света, среднее население которых больше, чем общемировое:

SELECT

Название

,Столица

,Площадь

,Население

,Континент

FROM

Страны

WHERE

КонтинентIN(

SELECT

Континент

FROM

Страны GROUP BY

Континент H

AVING

AVG(Население) >

(SELECT

AVG(Население)

FROM)

Страны))

Пример 7: Вывести список азиатских стран, население которых больше, чем в любой европейской стране:

SELECT

Название

, Столица

, Площадь

, Население

, Континент FR

OM

Страны WH

ERE

Континент =

'Азия' AND

Население > ALL(

SELECT

Население

FROM

Страны

WHERE

Континент = 'Европа'

)

Пример 8: Вывести список европейских стран, население которых больше, чем население хотя бы одной южноамериканской страны:

SELECT

Название

, Столица

, Площадь

```

,Население
,КонтинентFROM
    Страны
WHERE
Континент =
'Европа'AND
Население>ANY(
SELECT
Население
FROM
Страны
WHERE
Континент='ЮжнаяАмерика'
)

```

Пример 9: Если в Африке есть хотя бы одна страна, население которой больше 100 млн. чел., вывести список всех африканских стран:

```

SELECT
Название
,Столица
,Площадь
,Население
,КонтинентFROM
    СтраныWHERE
Континент =
'Африка'AND
EXISTS(
SELECT
*
FROM
Страны
WHERE
Континент = 'Африка'AND
Население>100000000
)

```

Пример 10: Вывести список стран в той части света, где находится страна

«Науру»:SELECT
 Название
 ,Столица
 ,Площадь
 ,Население
 ,КонтинентFROM
 СтраныWHERE
 Континент=(
 SELECT
 Континент
 FROM
 Страны
 WHERE
 Название='Науру'
)

Пример11:Вывести список стран, население которых не превышает населения страны «Гондурас»:

SELECT
 Название
 ,Столица
 ,Площадь
 ,Население
 ,КонтинентFROM
 СтраныWHERE
 Население!>(
 SELECT
 Население
 FROM
 Страны
 WHERE
 Название='Гондурас'
)

Пример 12:Вывести название страны с наибольшим населением среди стран с наименьшим населением на каждом континенте:

SELECT

Название

,Столица

,Площадь

,Население

,КонтинентFROM

СтраныWHERE

Население=(

SELECT

MAX(Мин_Нас)

FROM(

SELECT

MIN(Население)ASМин_НасFROM Страны)

GROUPLY

Континент

)A

Задание

1. Вывести список стран и процентное соотношение площади каждой из них к общей площади всех стран мира.
2. Вывести список стран мира, плотность населения которых больше, чем средняя плотность населения всех стран мира.
3. С помощью подзапроса вывести список европейских стран, население которых меньше 5 млн. чел.
4. Вывести список стран и процентное соотношение их площади к суммарной площади той части мира, где они находятся.
5. Вывести список стран мира, площадь которых больше, чем средняя площадь стран той части света, где они находятся.
6. Вывести список стран мира, которые находятся в тех частях света, средняя плотность населения которых превышает общемировую.
7. Вывести список южноамериканских стран, в которых живет больше людей, чем в любой африканской стране.
8. Вывести список африканских стран, в которых живет больше людей, чем хотя бы в одной южноамериканской стране.
9. Если в Африке есть хотя бы одна страна, площадь которой больше 2млн.кв.км, вывести список всех африканских стран.

10. Вывести список стран той части света, где находится страна «Фиджи».
11. Вывести список стран, население которых не превышает население страны «Фиджи».
12. Вывести название страны с наибольшим населением среди стран с наименьшей площадью на каждом континенте.

Контрольные вопросы:

- 1) Что такое измерения и для чего они нужны? Какие типы измерений Вы использовали в этой лабораторной работе?
- 2) Охарактеризуйте процесс развертывания куба.
- 3) Какая информация доступна при просмотре куба?

Лабораторная работа № 8. Определение расширенных свойств атрибутов и измерений

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Использование измененной версии проекта Analysis Services Tutorial

Оставшиеся занятия основаны на улучшенной версии проекта служб Службы Analysis Services Tutorial, с которым велась работа на первых трех занятиях.

Загрузка и извлечение файла проекта

1. [Щелкните здесь](#), чтобы перейти на страницу загрузки, где представлены образцы проектов. Образцы проектов включены в **adventure-works многомерных учебник projects.zip** загрузки.

2. Нажмите кнопку **adventure-works многомерных учебник projects.zip** для загрузки пакета, содержащий проекты для этого руководства.

По умолчанию этот ZIP-файл сохраняется в папке «Загрузки». Этот ZIP-файл необходимо перенести в расположение с более коротким путем (например, создайте папку C:\Tutorials для хранения таких файлов). После этого можно извлечь файлы из ZIP-файла. При попытке распаковать файлы из папки «Загрузки», путь к которой значительно больше, возникнет ошибка из занятия 1.

3. Создайте папку на уровне корневого диска или с неглубоким вложением, например C:\Tutorial.

4. Переместить **adventure-works многомерных учебник projects.zip** файл во вложенную папку.

5. Щелкните файл правой кнопкой мыши и выберите команду **Извлечь все**.

6. Перейдите в папку **Lesson 4 Start**, где находится файл **Analysis Services Tutorial.sln**.

Загрузка и обработка улучшенной версии проекта

1. В SQL Server Data Tools (SSDT) на **файл** меню, щелкните **заккрыть решение** закрыть файлы, которые не будут использоваться.

2. В меню **Файл** выберите пункт **Открыть**, затем выберите **Проект/Решение**.

3. Перейдите в папку, где находятся распакованные файлы проекта из учебника.

Найдите папку с именем **Lesson 4 Start**, а затем дважды щелкните файл **Analysis Services Tutorial.sln**.

4.Разверните улучшенную версию проекта Службы Analysis Services Tutorial на локальном экземпляре служб Службы Analysis Servicesили другом экземпляре и убедитесь, что обработка завершена успешно.

Представление источника данных

Представление источника данных расширенной версии проекта содержит дополнительную таблицу фактов и четыре дополнительные таблицы измерений из базы данных **AdventureWorksDW2012**.

Обратите внимание, что, поскольку представление источника данных содержит уже десять таблиц, диаграмма выглядит переполненной. Это затрудняет нахождение нужных таблиц и понимание связей между ними. Для разрешения этой проблемы таблицы поделены между двумя логическими диаграммами: **Продажи через Интернет** и **Товарооборот посредников**. Каждая из диаграмм построена на основе одной таблицы фактов. Создание логических диаграмм позволяет просматривать определенные подмножества таблиц в представлении источника данных и работать с ними, вместо того чтобы всегда просматривать все таблицы и их связи в единой схеме.

Диаграмма Internet Sales

Диаграмма **Продажи через Интернет** содержит таблицы, которые относятся к продаже продуктов Adventure Works напрямую покупателям через Интернет. В этой диаграмме содержатся четыре таблицы измерений и одна таблица фактов, которые были добавлены в представление источника данных **Adventure Works DW 2012** на занятии 1. Ниже приведены эти таблицы.

- **Geography**
- **Customer**
- **Дата**
- **Продукт**
- **InternetSales**

Диаграмма Reseller Sales

Диаграмма **Товарооборот посредников** содержит таблицы, относящиеся к продаже продуктов компании Adventure Works при помощи посредников. Эта диаграмма содержит следующие семь таблиц измерений и одну таблицу фактов из базы данных **AdventureWorksDW2012** :

- **Торговый посредник**
- **Promotion**
- **SalesTerritory**
- **Geography**
- **Дата**
- **Продукт**
- **Сотрудник**
- **ResellerSales**

Обратите внимание, что таблицы **DimGeography**, **DimDate** и **DimProduct** используются как в диаграмме **Продажи через Интернет**, так

и в диаграмме **Товарооборот посредников**. Таблицы измерений могут быть связаны с несколькими таблицами фактов.

Определение свойств родительского атрибута в иерархии родители-потомки

Иерархия типа «родители-потомки» представляет собой иерархию в измерении, которая основана на двух столбцах таблицы. Вместе эти столбцы определяют иерархическую связь между элементами измерения. Первый столбец, называемый *ключевым столбцом элемента*, идентифицирует каждый элемент измерения. Второй столбец, называемый *родительским столбцом*, идентифицирует родителя для каждого из элементов измерения. Свойство **NamingTemplate** родительского атрибута определяет имя каждого уровня в иерархии типа "родители-потомки", а свойство **MembersWithData**— будут ли отображаться данные для родительских элементов.

Примечание

Если измерение создается с помощью мастера измерений, мастер распознает таблицы, которые имеют связи типа «родители-потомки», и автоматически определяет иерархию типа «родители-потомки».

При выполнении задач этого раздела будет создан шаблон именования, определяющий имя каждого из уровней в иерархии типа "родители-потомки" в измерении **Сотрудник**. Затем этот родительский атрибут будет настроен таким образом, чтобы все данные о родителях были скрыты и отображались только данные по продажам для элементов конечного уровня.

Просмотр измерения Employee

1. В обозревателе решений дважды щелкните файл **Employee.dim** в папке **Измерения**, чтобы открыть конструктор измерений для измерения "Сотрудник".

2. Перейдите на вкладку **Браузер**, убедитесь в том, что в списке **Иерархия** выбран элемент **Сотрудники**, а затем разверните элемент **Все сотрудники**.

Обратите внимание, что сотрудник **Кен Дж. Санчес**— менеджер высшего уровня в этой иерархии типа "родители-потомки".

3. Выберите элемент **Кен Дж. Санчес**.

Обратите внимание, что имя уровня для этого элемента—**Уровень 02**. (Имя уровня указано сразу после значения **Текущий уровень**: над элементом **Все сотрудники**.) В следующей задаче будут определены более понятные имена для каждого уровня.

4. Разверните элемент **Кен Дж. Санчес**, чтобы увидеть имена всех сотрудников, подчиняющихся этому менеджеру, а затем выберите элемент **Брайан С. Уэлкер**, чтобы узнать имя этого уровня.

Обратите внимание, что имя уровня для этого элемента —**Уровень 03**.

5. В обозревателе решений дважды щелкните файл **Analysis Services Tutorial.cube** в папке **Кубы**, чтобы открыть конструктор кубов для куба "Учебник по Службы Analysis Services".

6. Перейдите на вкладку **Браузер**.

7. Щелкните значок Excel, а затем выберите **Включить** при появлении запроса на включение подключений.

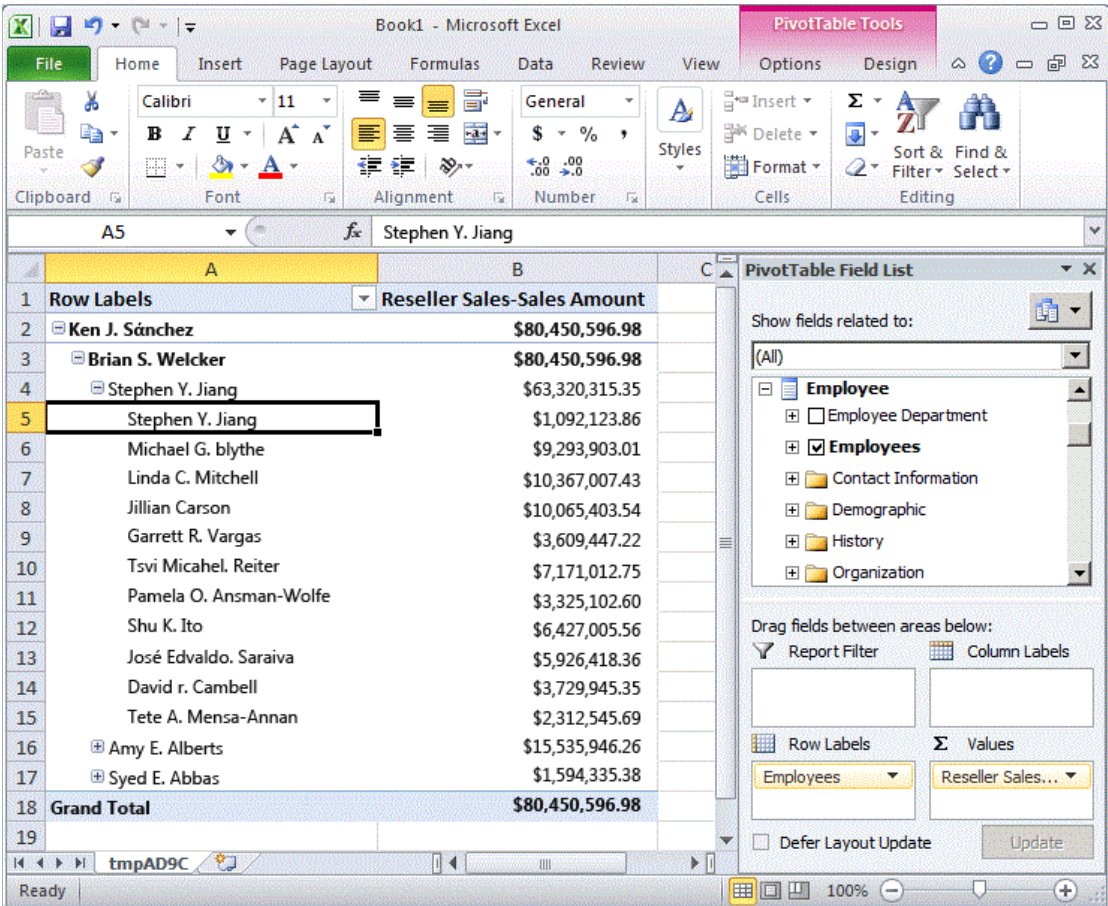
8. В списке полей сводной таблицы разверните узел **Продажи через торговых посредников**. Перетащите меру **Товарооборот посредников—объем продаж** в область значений.

9. В списке полей сводной таблицы разверните узел **Сотрудник**, а затем перетащите иерархию **Сотрудники** в область **Строки**.

Все элементы иерархии Employees будут добавлены в столбец A в отчете сводной таблицы.

На следующем рисунке показана развернутая иерархия Employees.

10.



Row Labels	Reseller Sales-Sales Amount
Ken J. Sánchez	\$80,450,596.98
Brian S. Welcker	\$80,450,596.98
Stephen Y. Jiang	\$63,320,315.35
Stephen Y. Jiang	\$1,092,123.86
Michael G. blythe	\$9,293,903.01
Linda C. Mitchell	\$10,367,007.43
Jillian Carson	\$10,065,403.54
Garrett R. Vargas	\$3,609,447.22
Tsvi Micahel. Reiter	\$7,171,012.75
Pamela O. Ansman-Wolfe	\$3,325,102.60
Shu K. Ito	\$6,427,005.56
José Edvaldo. Saraiva	\$5,926,418.36
David r. Cambell	\$3,729,945.35
Tete A. Mensa-Annan	\$2,312,545.69
Amy E. Alberts	\$15,535,946.26
Syed E. Abbas	\$1,594,335.38
Grand Total	\$80,450,596.98

Обратите внимание, что продажи, совершенные каждым менеджером уровня 03, отображаются и на уровне 04. Это связано с тем, что каждый менеджер является также подчиненным другого менеджера. В следующей задаче предстоит скрыть эти суммы продаж.

Изменение свойств родительского атрибута в измерении Employee

1. В конструкторе измерений откройте измерение **Сотрудник**.

2. Перейдите на вкладку **Структура измерения**, а затем выберите иерархию атрибута **Сотрудники** на панели **Атрибуты**.

Обратите внимание на уникальный значок атрибута. Этот значок указывает, что атрибут является ключом родителя в иерархии типа «родители-потомки». Обратите внимание, что в окне свойств свойство **Использование** для этого атрибута определено как **Родитель**. Это свойство устанавливается мастером измерений во время создания измерения. Мастер автоматически выявляет связи типа «родители-потомки».

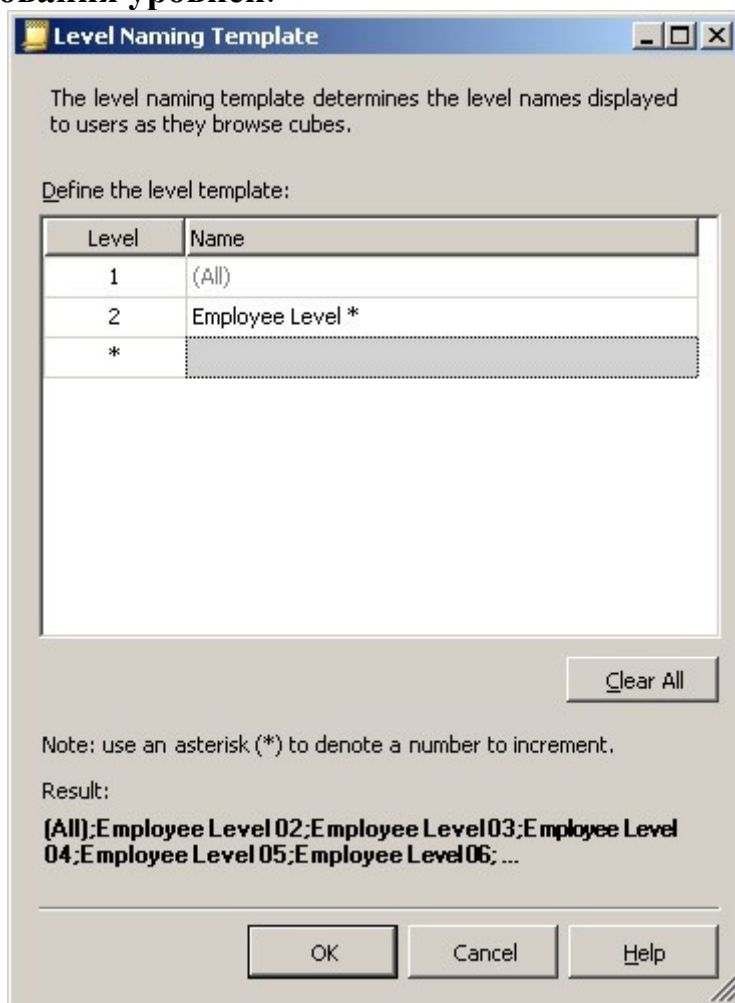
3. В окне свойств в ячейке свойств **NamingTemplate** нажмите кнопку с многоточием (...).

В диалоговом окне **Шаблон именования уровней** необходимо задать шаблон именования уровней, который определяет имена уровней в иерархии типа "родители-потомки", отображаемых при просмотре кубов.

4. Во второй строке * в столбце **Имя** введите **Уровень сотрудника ***, а затем щелкните третью строку.

Обратите внимание, что в поле **Результат** каждый уровень теперь называется "Уровень сотрудника", за которым следует последовательно увеличивающееся число.

На рисунке ниже показаны изменения в диалоговом окне **Шаблон именования уровней**.



5. Нажмите кнопку **ОК**.

6. В окне "Свойства" для атрибута **Сотрудники** в ячейке свойств **MembersWithData** выберите **NonLeafDataHidden**, чтобы изменить это значение для атрибута **Сотрудники**.

Данные в иерархии типа «родители-потомки», которые относятся к элементам уровня, не являющегося конечными, будут скрыты.

Просмотр измерения **Employee** с измененными атрибутами

1. В меню **Построение** среды SQL Server Data Tools (SSDT) выберите команду **Развернуть Analysis Services Tutorial**.

2. После успешного завершения развертывания перейдите в конструктор кубов и откройте куб "Учебник по Службы Analysis Services". Затем нажмите кнопку **Повторное соединение** на панели инструментов вкладки **Браузер**.

3. Щелкните ярлык **Excel**, а затем выберите **Включить**.

4. Перетащите меру **Товарооборот посредников — объем продаж** в область значений.

5. Перетащите иерархию **Сотрудники** в область "Метки строк".

На следующем рисунке показаны изменения, внесенные в иерархию **Employees**. Обратите внимание на то, что **Stephen Y. Jiang** больше не отображается как сотрудник самого себя.

Row Labels	Reseller Sales-Sales Amount
Ken J. Sánchez	\$80,450,596.98
Brian S. Welcker	\$80,450,596.98
Stephen Y. Jiang	\$63,320,315.35
Michael G. Blythe	\$9,293,903.01
Linda C. Mitchell	\$10,367,007.43
Jillian Carson	\$10,065,803.54
Garrett R. Vargas	\$3,609,447.22
Tsvi Michael. Reiter	\$7,171,012.75
Pamela O. Ansman-Wolfe	\$3,325,102.60
Shu K. Ito	\$6,427,005.56
José Edvaldo. Saraiva	\$5,926,418.36
David R. Campbell	\$3,729,945.35
Tete A. Mensa-Annan	\$2,312,545.69
Amy E. Alberts	\$15,535,946.26
Syed E. Abbas	\$1,594,335.38
Grand Total	\$80,450,596.98

Задания:

1. Использование измененной версии проекта Analysis Services Tutorial
2. Определение свойств родительского атрибута в иерархии "родители-потомки"

Контрольные вопросы:

1. Как просмотреть список измерений?
2. Опишите процесс определения свойств атрибутов.
3. Как изменить атрибуты и свойства в таблицах?

Лабораторная работа № 9. Определение расширенных свойств атрибутов и измерений

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Анализ данных и обоснование решений	ПК-4

Теоретическая часть

Автоматическое группирование элементов атрибута

При просмотре куба обычно выполняется распределение по измерениям элементов одной иерархии атрибута на основе элементов другой иерархии атрибута. Например, можно сгруппировать продажи по таким признакам, как город, приобретенный товар или пол. Однако при работе с некоторыми типами атрибутов лучше группировать их элементы автоматически с использованием служб Microsoft Службы Analysis Services, основываясь на распределении элементов в рамках иерархии атрибута. Например, с помощью служб Службы Analysis Services можно объединить заказчиков в группы на основе их годового дохода. При этом пользователь, просматривающий иерархию атрибута, будет видеть названия и значения групп вместо самих элементов групп. Данный подход ограничивает количество отображаемых уровней, что упрощает анализ данных.

Свойство **DiscretizationMethod** определяет, создают ли службы Службы Analysis Services группирование, а также определяет тип выполняемого группирования. По умолчанию в службах Службы Analysis Services группирование не выполняется. При включении автоматического группирования можно настроить параметры служб Службы Analysis Services таким образом, чтобы выполнялся автоматический выбор оптимального метода группирования, основанного на структуре данного атрибута, либо выбрать один из алгоритмов группирования в следующем списке:

EqualAreas

В службах Службы Analysis Services диапазоны групп создаются таким образом, что элементы измерения распределяются по группам пропорционально.

Clusters

В службах Службы Analysis Services группы создаются путем проведения одномерной кластеризации входных значений, используя метод кластеризации К-средних с нормальным распределением. Эта функция применима только в отношении числовых столбцов.

После выбора метода группирования необходимо указать количество групп, используя свойство **DiscretizationBucketCount**.

В ходе выполнения задач данного раздела будут опробованы следующие критерии группирования: значение годового дохода в измерении **Заказчик**, количество часов отсутствия по болезни в измерении

Сотрудники, данные о количестве часов отпуска сотрудников в измерении **Сотрудники**. Затем предстоит выполнить обработку куба учебника по службам Службы Analysis Services и просмотреть результаты группирования элементов. На конечной стадии будут изменены свойства групп элементов, чтобы выявить эффект от изменения типа группирования.

Группирование элементов иерархии атрибута в измерении «Заказчик»

1. В обозревателе решений дважды щелкните элемент **Заказчик** в папке **Измерения**. Откроется конструктор измерений для измерения «Заказчик».

2. На панели **Представление источника данных** щелкните правой кнопкой мыши таблицу **Заказчик** и выберите пункт **Просмотр данных**.

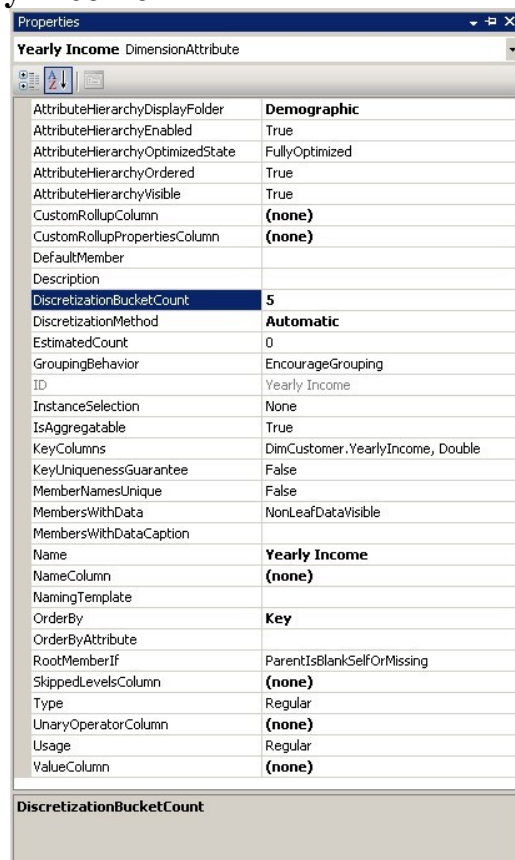
Обратите внимание на диапазон данных в столбце **YearlyIncome**. Указанные значения используются для заполнения иерархии атрибута **Yearly Income**, если не включено группирование элементов.

3. Закройте вкладку **Просмотр таблицы Customer**.

4. На панели **Атрибуты** выберите атрибут **Yearly Income**.

5. В окне свойств измените значение свойства **DiscretizationMethod** на **Automatic**, а значение свойства **DiscretizationBucketCount**— на **5**.

На следующем рисунке показаны измененные свойства атрибута **Yearly Income**.



Properties	
Yearly Income DimensionAttribute	
AttributeHierarchyDisplayFolder	Demographic
AttributeHierarchyEnabled	True
AttributeHierarchyOptimizedState	FullyOptimized
AttributeHierarchyOrdered	True
AttributeHierarchyVisible	True
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
DefaultMember	
Description	
DiscretizationBucketCount	5
DiscretizationMethod	Automatic
EstimatedCount	0
GroupingBehavior	EncourageGrouping
ID	Yearly Income
InstanceSelection	None
IsAggregatable	True
KeyColumns	DimCustomer.YearlyIncome, Double
KeyUniquenessGuarantee	False
MemberNamesUnique	False
MembersWithData	NonLeafDataVisible
MembersWithDataCaption	
Name	Yearly Income
NameColumn	(none)
NamingTemplate	
OrderBy	Key
OrderByAttribute	
RootMemberIf	ParentIsBlankSelfOrMissing
SkippedLevelsColumn	(none)
Type	Regular
UnaryOperatorColumn	(none)
Usage	Regular
ValueColumn	(none)

Группирование элементов иерархии атрибута в измерении Employee

1. В конструкторе измерений откройте измерение «Сотрудник».

2. На панели **Представление источника данных** щелкните правой кнопкой мыши таблицу **Сотрудник** и выберите пункт **Просмотр данных**.

Обратите внимание на значения столбцов **SickLeaveHours** и **VacationHours**.

3. Закройте вкладку **Просмотр таблицы Employee**.

4. На панели **Атрибуты** выберите атрибут **Sick Leave Hours**.

5. В окне свойств измените значение свойства **DiscretizationMethod** на **Clusters**, а значение свойства **DiscretizationBucketCount** — на **5**.

6. На панели **Атрибуты** выберите атрибут **Vacation Hours**.

7. В окне свойств измените значение свойства **DiscretizationMethod** на **Равные области**, а значение свойства **DiscretizationBucketCount** на **5**.

Просмотр измененных иерархий атрибутов

1. В меню **Построение** среды SQL Server Data Tools (SSDT) выберите команду **Развернуть Analysis Services Tutorial**.

2. После завершения развертывания переключитесь в конструктор кубов на куб по службам Службы Analysis Services Tutorial, а затем нажмите кнопку **Повторное соединение** на вкладке **Обозреватель**.

3. Щелкните ярлык **Excel**, а затем выберите **Включить**.

4. Перетащите меру **Продажи через Интернет** — сумма продаж в область значений списка полей сводной таблицы.

5. В списке полей разверните измерение **Продукт**, а затем перетащите пользовательскую иерархию **Линии моделей товаров** в область **Метки строк** списка полей.

6. В списке полей разверните измерение **Заказчик**, разверните папку отображения **Демография** и перетащите иерархию атрибута **Годовой доход** в область **Метки столбцов**.

Теперь элементы иерархии атрибута **Годовой доход** сгруппированы в шесть сегментов, один из которых содержит данные о клиентах с неизвестным годовым доходом. Отображаются не все сегменты.

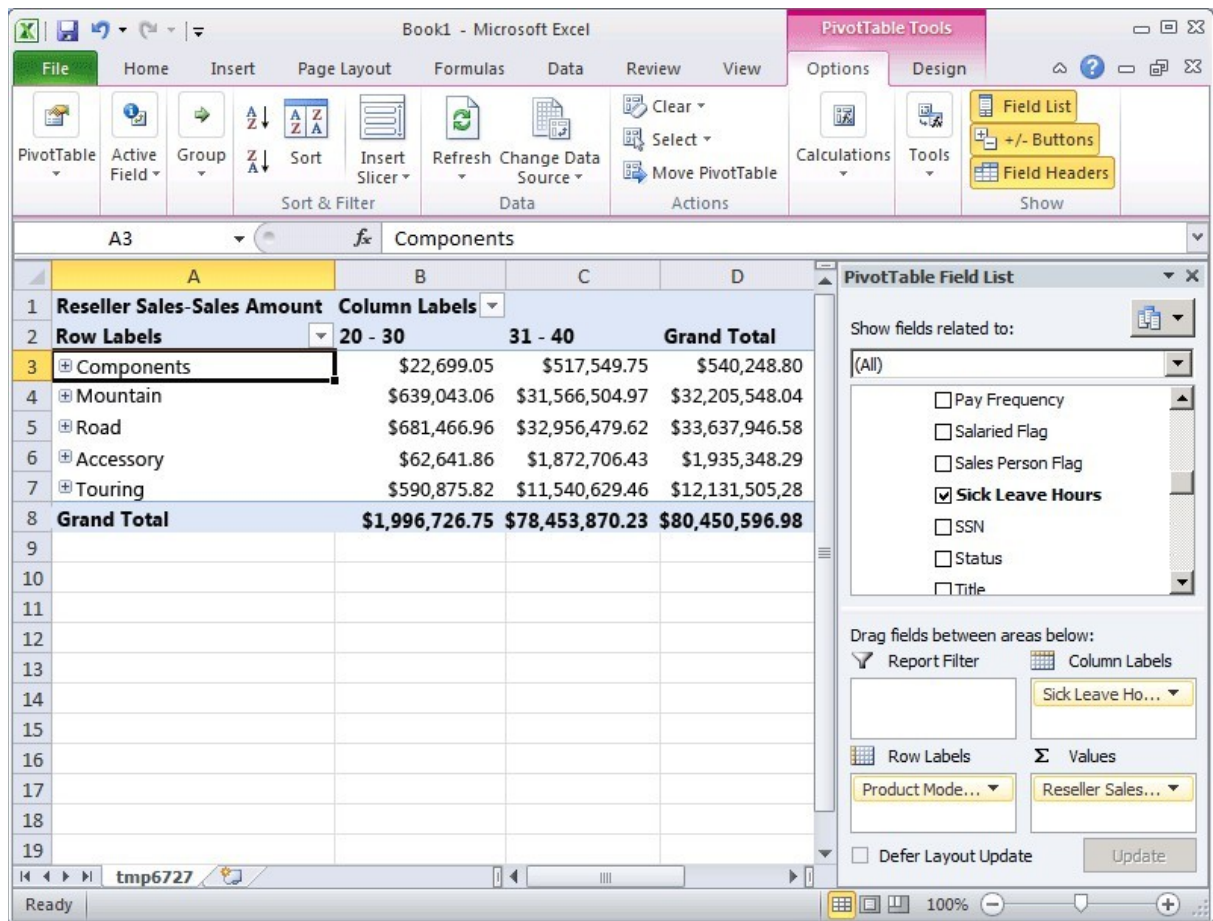
7. Удалите иерархию атрибута **Годовой доход** из области столбцов и удалите меру **Продажи через Интернет** — сумма продаж из области **Значения**.

8. Добавьте меру **Товaroоборот посредников** — сумма продаж в область данных.

9. В списке полей разверните измерение **Сотрудник**, затем узел **Организация**, а затем перетащите **Часы отсутствия по болезни** в область **Метки столбцов**.

Обратите внимание, что все продажи были сделаны сотрудниками, включенными в одну из двух групп. Кроме того, обратите внимание, что сотрудники, отсутствовавшие по болезни от 32 до 42 часов, сделали существенно больше продаж, чем сотрудники, отсутствовавшие от 20 часов до 31 часа.

На приведенном ниже рисунке приведена зависимость объемов продаж от количества часов отпуска по болезни.



10. Удалите иерархию атрибута **Sick Leave Hours** из раздела столбцов панели **Данные**.

11. Добавьте столбец **Vacation Hours** к разделу столбцов панели **Данные**.

Обратите внимание, что отображаются две группы, созданные с помощью метода группирования по равным областям. Остальные три группы не показаны, так как не содержат значений.

Изменение свойств группирования и просмотр результата изменений

1. Откройте в конструкторе измерений измерение **Сотрудник** и выберите на панели **Атрибуты** атрибут **Часы отпуска**.

2. В окне свойств задайте для свойства **DiscretizationBucketCount** значение **10**.

3. В меню **Построение** среды SQL Server Data Tools (SSDT) выберите команду **Развернуть Analysis Services Tutorial**.

4. После завершения развертывания переключитесь обратно в конструктор кубов на куб по службам Службы Analysis Services Tutorial.

5. На вкладке **Обозреватель** нажмите кнопку **Повторное подключение**, щелкните значок Excel и снова создайте сводную таблицу, чтобы просмотреть результаты изменения метода группирования.

а. Перетащите меру «Товарооборот посредников — сумма продаж» в область значений

б. Перетащите атрибут «Часы отпуска» (в папке «Организация сотрудников») в область столбцов

с. Перетащите атрибут «Линии моделей товаров» в область строк

Обратите внимание, что созданы три группы элементов атрибута **Часы отпуска**, в которых содержатся значения по продажам товаров. В остальных 7 группах содержатся элементы без данных о продажах.

Скрытие и отключение иерархий атрибутов

По умолчанию иерархия атрибута создается для каждого атрибута в измерении и каждая иерархия доступна для разделения данных фактов по измерениям. Эта иерархия состоит из уровня «Все» и уровня подробностей, содержащего все элементы иерархии. Ранее рассматривался вопрос о том, что атрибуты можно организовывать в пользовательские иерархии для предоставления путей перемещения в кубе. В определенных случаях может потребоваться отключить или скрыть некоторые атрибуты и их иерархии. Например, определенные атрибуты, такие как номера социального страхования или номера национальной принадлежности, ставки заработной платы, даты рождения и имена входа в систему не являются атрибутами, по которым пользователь будет организовывать измерения куба. Эти сведения обычно просматриваются только для справки по конкретному члену атрибута. Данные иерархии атрибутов может потребоваться скрыть, оставляя видимыми только сами атрибуты как свойства члена конкретного атрибута. Кроме того, может возникнуть необходимость сделать элементы других атрибутов, такие как имена заказчиков или почтовые индексы, видимыми только при просмотре через пользовательскую иерархию, а не независимо через иерархию атрибута. Одной из причин для этого может быть большое число различающихся элементов в иерархии атрибута. Наконец, чтобы увеличить производительность обработки, следует выключить иерархии атрибутов, которыми не будут пользоваться пользователи.

Значение свойства **AttributeHierarchyEnabled** определяет, создана ли иерархия атрибута. Если значением свойства является **False**, иерархия атрибутов не создана и атрибут нельзя использовать в качестве уровня пользовательской иерархии, то есть иерархия атрибутов существует только как свойство элемента. Однако отключенную иерархию атрибута можно использовать для сортировки элементов другого атрибута. Если значением свойства **AttributeHierarchyEnabled** является **True**, значение свойства **AttributeHierarchyVisible** определяет, является ли иерархия атрибута видимой независимо от пользовательской иерархии.

Если иерархия атрибута включена, может возникнуть необходимость указать значения следующих трех дополнительных свойств.

- **IsAggregatable**

По умолчанию для всех иерархий атрибутов задается уровень «(Все)». Чтобы отключить уровень "(Все)" для включенной иерархии атрибутов, задайте для этого свойства значение **False**.

Примечание

Атрибут, свойство **IsAggregatable** которого установлено в значение **false**, может использоваться только в качестве корневого элемента многоуровневой иерархии, и для него должен быть указан элемент по умолчанию (в противном случае элемент по умолчанию будет выбран ядром служб Службы Analysis Services).

- **AttributeHierarchyOrdered**

По умолчанию в службах Службы Analysis Services элементы включенных иерархий атрибутов сортируются при обработке, а затем сохраняются в зависимости от значения свойства **OrderBy**, например «Имя» или «Ключ». Если сортировка не важна, производительность обработки можно повысить, указав для этого свойства значение **False**.

- **AttributeHierarchyOptimizedState**

По умолчанию в службах Службы Analysis Services создается индекс для каждой включенной иерархии атрибута при обработке, чтобы повысить производительность выполнения запросов. Если не планируется использовать иерархию атрибута для просмотра, можно повысить производительность обработки, задав в качестве значения этого свойства **NotOptimized**. Тем не менее, если скрытая иерархия используется в качестве ключевого атрибута измерения, создание индекса элементов атрибута позволит повысить производительность.

Эти свойства неприменимы, если иерархия атрибута отключена.

В задачах этого раздела необходимо отключить номера социального страхования и другие атрибуты в измерении «Сотрудники», которые не будут использоваться для просмотра. Затем предстоит скрыть иерархии атрибутов имени заказчика и почтового кода в измерении «Заказчик». Если у этих иерархий много элементов атрибутов, их просмотр будет достаточно медленным вне зависимости от пользовательской иерархии.

Определение свойств иерархии атрибута в измерении «Сотрудники»

1. Перейдите в конструктор измерений на измерение «Сотрудники» и откройте вкладку **Обозреватель**.

2. Проверьте, включены ли в список **Иерархия** следующие иерархии атрибутов:

- **Базовая ставка**
- **Дата рождения**
- **Идентификатор входа**
- **ИНН менеджера**
- **ИНН**

Перейдите на вкладку **Структура измерения**, а затем выберите следующие атрибуты на панели **Атрибуты**. Можно выбрать несколько мер. Для этого щелкните каждую из них, удерживая нажатой клавишу CTRL.

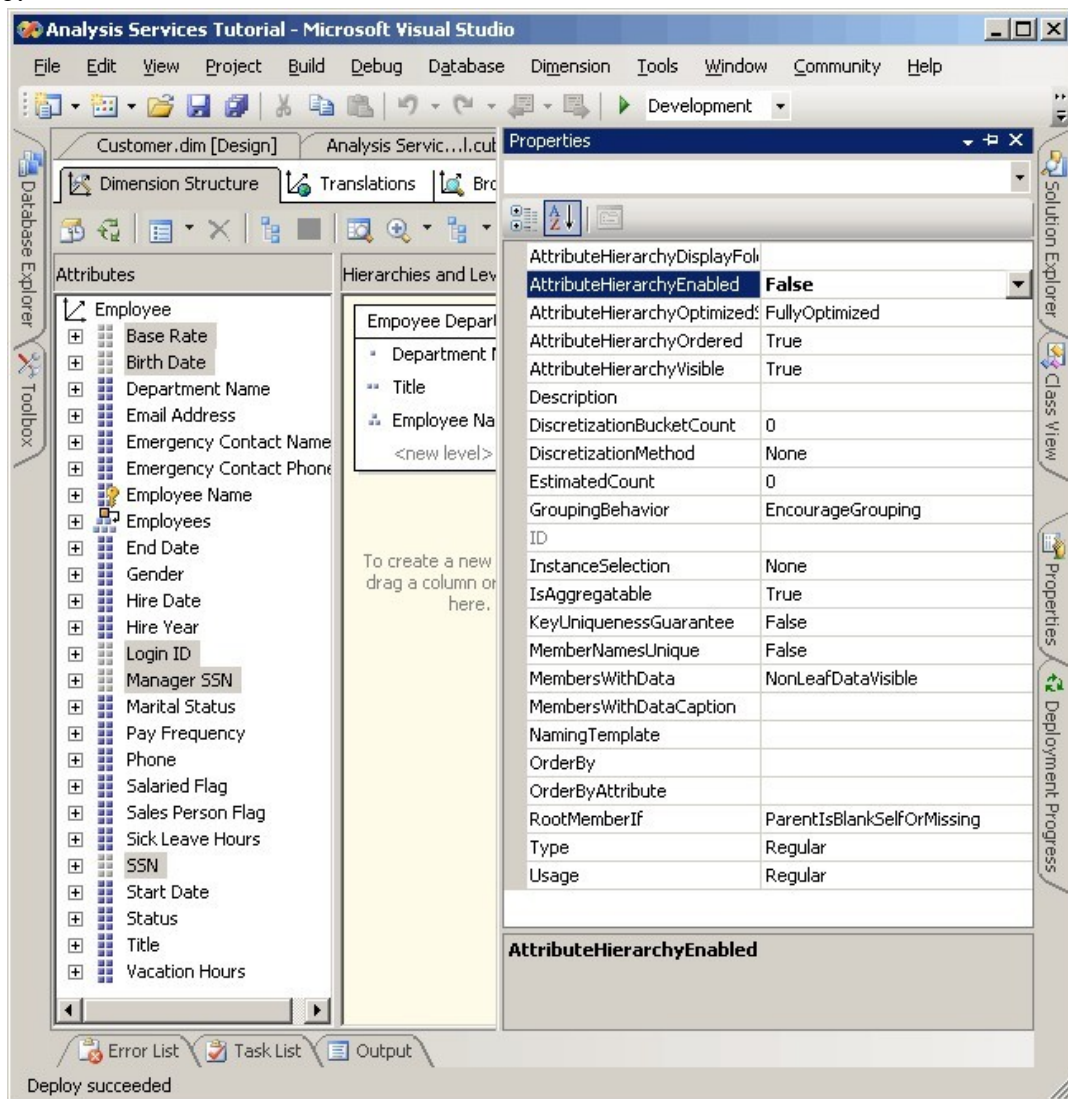
- **Базовая ставка**
- **Дата рождения**

- **Идентификатор входа**
- **ИНН менеджера**
- **ИНН**

В окне свойств установите для свойства **AttributeHierarchyEnabled** выбранных атрибутов значение **False**.

Обратите внимание, что на панели **Атрибуты** значок каждого из атрибута изменился и указывает, что этот атрибут отключен.

На рисунке ниже показано, что для свойства **AttributeHierarchyEnabled** выбранных атрибутов установлено значение **False**.



В меню **Сборка** выберите команду **Развернуть Analysis Services Tutorial**.

После успешного окончания обработки перейдите на вкладку **Обзор**, нажмите кнопку **Повторное соединение** затем попытайтесь просмотреть измененные иерархии атрибутов.

Обратите внимание, что элементы измененных атрибутов недоступны для просмотра в качестве иерархий атрибутов в списке **Иерархия**. При попытке добавления отключенной иерархии атрибута в качестве уровня пользовательской иерархии будет выведено сообщение об

ошибке, уведомляющее о том, что для включения в пользовательскую иерархию необходимо активировать иерархию атрибута.

Настройка свойств иерархии атрибута в измерении «Заказчик»

1. Перейдите в конструктор измерений на измерение «Заказчик» и откройте вкладку **Обозреватель**.

2. Проверьте, включены ли в список **Иерархия** следующие иерархии атрибутов:

- **Полное имя**
- **Почтовый индекс**

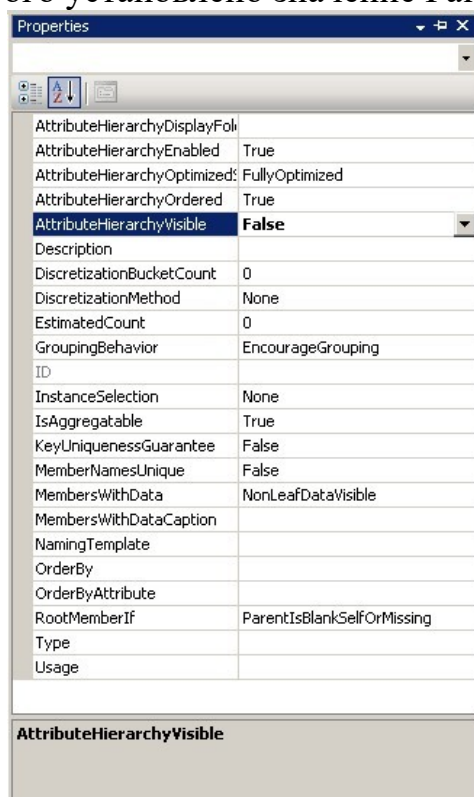
Перейдите на вкладку **Структура измерения** и выберите следующие атрибуты на панели **Атрибуты**, удерживая нажатой клавишу CTRL, чтобы выделить несколько атрибутов одновременно:

- **Полное имя**
- **Почтовый индекс**

В окне свойств установите для свойства **AttributeHierarchyEnabled** выбранных атрибутов значение **False**.

Поскольку элементы этих иерархий атрибутов будут использоваться для разделения данных фактов по измерениям, их сортировка и оптимизация повысит производительность. Таким образом, свойства этих атрибутов изменять не следует.

На рисунке ниже показано свойство **AttributeHierarchyVisible**, для которого установлено значение False.



Перетащите атрибут **Почтовый индекс** с панели **Атрибуты** в пользовательскую иерархию **География заказчика** панели **Иерархии и уровни** непосредственно под уровень **Город**.

Обратите внимание, что скрытый атрибут по-прежнему может быть уровнем пользовательской иерархии.

В меню **Сборка** выберите команду **Развернуть Analysis Services Tutorial**.

После успешного окончания развертывания перейдите на вкладку **Браузер** измерения «Заказчик» и нажмите кнопку **Повторное соединение**.

Попробуйте выбрать одну из измененных иерархий атрибутов в списке **Иерархия**.

Обратите внимание, что ни одна из измененных иерархий атрибутов не отображается в списке **Иерархия**.

В списке **Иерархия** выберите значение **География заказчика** и просмотрите каждый из уровней на панели браузера.

Обратите внимание на то, что скрытые уровни **Почтовый индекс** и **Полное имя** видны в пользовательской иерархии.

Задания:

1. Автоматическое группирование элементов атрибута
2. Скрытие и отключение иерархий атрибутов

Контрольные вопросы:

1. Охарактеризуйте процесс автоматической группировки элементов атрибута
2. Охарактеризуйте процесс скрывания и отключения иерархий атрибутов

Список рекомендуемой литературы

Основная литература:

1. Нестеров, С. А. Интеллектуальный анализ данных средствами MS SQL Server 2008 / С.А. Нестеров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 338 с
2. Симчера, В. М. Методы многомерного анализа статистических данных : Учебное пособие / Симчера В. М. - Москва : Финансы и статистика, 2014. - 400 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-279-03184-9

Дополнительная литература:

1. Астапчук В. А., Терещенко П. В. - КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ: ТРЕБОВАНИЯ ПРИ ПРОЕКТИРОВАНИИ 2-е изд., испр. и доп. Учебное пособие для вузов - М.:Издательство Юрайт - 2019 - 113с. - ISBN: 978-5-534-08546-4 - Текст электронный // ЭБС ЮРАЙТ - URL: <https://urait.ru/book/korporativnyye-informacionnye-sistemy-trebovaniya-pri-proektirovanii-425572>

2. Компьютерные технологии анализа данных в эконометрике : монография / Д.М. Дайитбегов. - 3-е изд., доп. - М. : ИНФРА-М : Вузовский учебник, 2013. - 587 с. : ил., табл. - (Научная книга). - Библиогр.: с. 558-569. - Др. работы авт. на 4-й с. обл. - ISBN 978-5-9558-0275-6. - ISBN 978-5-16-006145-0

3. Уэс, Маккинли. Python и анализ данных / Маккинли Уэс : практическое пособие ; пер. А.А. Слинкин Электронный ресурс. - Python и анализ данных, 2019-04-19 : Профобразование ; Саратов, 2017. - 482 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-4488-0046-

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для обучающихся по организации и проведению самостоятельной
работы
по дисциплине «Анализ данных в бизнес-системах»
направление подготовки 38.03.05 Бизнес-информатика
направленность (профиль) «Информационная бизнес-аналитика и цифровые
технологии»

ВВЕДЕНИЕ

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

Целью данной дисциплины является формирование набора (ПК-4) компетенций будущего бакалавра по направлению подготовки 38.03.05 «Бизнес-информатика».

Задачами дисциплины «Анализ данных в бизнес-системах» являются: систематизация и закрепление полученных теоретических знаний и практических умений студентов; углубление и расширение теоретических знаний; формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу; развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности; формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации; развитие исследовательских умений; использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

Дисциплина «Анализ данных в бизнес-системах» входит в часть, формируемой участниками образовательных отношений учебного плана 2025 года по направлению подготовки 38.03.05 «Бизнес информатика». Ее освоение происходит в 7 семестре.

Общая характеристика самостоятельной работы студента при изучении дисциплины «Анализ данных в бизнес-системах»

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СР являются:

систематизация и закрепление полученных теоретических знаний и практических умений студентов;

углубление и расширение теоретических знаний;

формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;

развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

развитие исследовательских умений;

использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

План-график выполнения самостоятельной работы

Коды реализуемы х копетенций	Вид деятельности студентов	Итоговый продукт самостоятельно й работы	Средства и технологии оценки	Объем часов, в том числе (астр)		
				СРС	Контактная работа с преподавателе м	Всего
7 семестр						
ИД-2 ПК-4 ИД-3 ПК-4	Подготовка к отчету лабораторной работе		Собеседование	41.4	4.6	46
ИД-2 ПК-4 ИД-3 ПК-4	Подготовка к собеседованию	собеседование	Вопросы к собеседованию	45.0	5.0	50
ИД-2 ПК-4 ИД-3 ПК-4	Выполнение курсовой работы	Защита курсовой работы	Задания для курсовой работы	43.2	4.8	48
Итого за семестр				129.6	14.4	144
Итого				129.6	14.4	144

Работа по подготовке к лабораторным занятиям и активное в них участие — одна из форм изучения программного материала курса. Она направлена на подготовку высококвалифицированных магистров. Подготовку к занятиям следует начинать с внимательного изучения соответствующих разделов учебных пособий и учебников, далее — следует изучать специальную литературу и источники, работать с картами, таблицами, схемами, наконец, написать доклад, если студент получил такое задание. Готовясь к занятиям и принимая активное участие в их работе, студент проходит школу работы над источниками и литературой, получает навыки самостоятельной работы над письменным и устным сообщением (докладом), учится участвовать в дискуссиях, отстаивать свою точку зрения, формулировать и аргументировать выводы. Форма практических занятий во многом определяется его темой. Практика показывает, что основные формы занятий следующие: беседа на основе составленного преподавателем плана (она наиболее приемлема при обсуждении одного из теоретических вопросов по проблемам темы или монографии), коллоквиум по разделу учебника или одной из монографий (коллоквиум предполагает прежде всего проверку знаний по определенной теме, источникам, разделу курса); подготовка письменного доклада студентом, его устный доклад и обсуждение его на занятии.

В планы лабораторных занятий включены основные вопросы общего курса. В ходе занятий возможна их конкретизация и корректировка. При подготовке сообщений и докладов следует широко использовать опубликованные источники, исследовательскую литературу. Учебники и учебные пособия студент использует по своему выбору. Каждому студенту в течение года следует прочитать не менее двух монографий (или одну монографию и одну работу мемуарного характера), которые указаны в списке литературы или рекомендовано преподавателем из числа новых публикаций, составить краткий доклад и быть готовым к беседе по ним с преподавателем.

Методические рекомендации по изучению теоретического материала

Работа с книгой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия.

Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Различают два вида чтения; первичное и вторичное. *Первичное* - это внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятого слова. Содержание не всегда может быть понятно после первичного чтения.

Задача *вторичного* чтения - полное усвоение смысла целого (по сути это чтение может быть и не вторым, а третьим или четвертым).

Правила самостоятельной работы с литературой

Как уже отмечалось, самостоятельная работа с учебниками и книгами (а также самостоятельное теоретическое исследование проблем, обозначенных преподавателем на лекциях) – это важнейшее условие формирования у себя научного способа познания. Основные советы здесь можно свести к следующим:

- Составить перечень книг, с которыми Вам следует познакомиться;
- Сам такой перечень должен быть систематизированным.
- Обязательно выписывать все выходные данные по каждой книге (при написании курсовых и дипломных работ это позволит очень сэкономить время).
- Разобраться для себя, какие книги (или какие главы книг) следует прочитать более внимательно, а какие – просто просмотреть.

• При составлении перечней литературы следует посоветоваться с преподавателями и научными руководителями (или даже с более подготовленными и эрудированными сокурсниками), которые помогут Вам лучше сориентироваться, на что стоит обратить большее внимание, а на что вообще не стоит тратить время...

• Естественно, все прочитанные книги, учебники и статьи следует конспектировать, но это не означает, что надо конспектировать «все подряд»: можно выписывать кратко основные идеи автора и иногда приводить наиболее яркие и показательные цитаты (с указанием страниц).

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того, насколько осознанно читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста**:

1. информационно-поисковый (задача – найти, выделить искомую информацию)
2. усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)
3. аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)
4. творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

1. Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;
2. Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;
3. Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;
4. Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

5. Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по написанию доклада

Доклад – публичное сообщение, представляющее собой развернутое изложение на определенную тему, вид самостоятельной работы, который используется в учебных и внеаудиторных занятиях и способствует формированию навыков исследовательской работы, расширяет познавательные интересы, приучает критически мыслить.

Чтобы выступление было удачным, оно должно хорошо восприниматься на слух, быть интересным для слушателей. При выступлении приветствуется активное использование мультимедийного сопровождения доклада (презентация, видеоролики, аудиозаписи).

Общая структура доклада. Построение доклада включает три части: вступление, основную часть и заключение.

Вступление.

Формулировка темы доклада (она должна быть не только актуальной, но и оригинальной, интересной по содержанию).

Актуальность выбранной темы (чем она интересна, в чем заключается ее важность, почему учащимся выбрана именно эта тема).

Анализ литературных источников (рекомендуется использовать данные за последние 5 лет)

Основная часть. Состоит из нескольких разделов, постепенно раскрывающих тему. Возможно использование иллюстрации (графики, диаграммы, фотографии, карты, рисунки). Если необходимо, для обоснования темы используется ссылка на источники с доказательствами, взятыми из литературы (цитирование авторов, указание цифр, фактов, определений). Изложение материала должно быть связным, последовательным, доказательным.

Способ изложения материала для выступления должен носить конспективный или тезисный характер.

Заключение.

Подводятся итоги, формулируются главные выводы, подчеркивается значение рассмотренной проблемы, предлагаются самые важные практические рекомендации.

Требования к оформлению доклада.

Объем машинописного текста доклада должен быть рассчитан на произнесение доклада в течение 7 -10 минут (3-5 машинописных листа текста с докладом). Поэтому при подборе необходимого материала для доклада отбирается самое главное. В докладе должны быть кратко отражены главные моменты из введения, основной части и заключения. При подготовке конспекта доклада необходимо составить не только текст доклада, но и необходимый иллюстративный материал, сопровождающий доклад (основные тезисы, формулы, схемы, чертежи, таблицы, графики и диаграммы, фотографии и т.п.).

Список рекомендуемой литературы

Основная литература:

1. Нестеров, С. А. Интеллектуальный анализ данных средствами MS SQLServer 2008 / С.А. Нестеров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 338 с

2. Симчера, В. М. Методы многомерного анализа статистических данных : Учебное пособие / Симчера В. М. - Москва : Финансы и статистика, 2014. - 400 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-279-03184-9

Дополнительная литература:

1. Астапчук В. А., Терещенко П. В. - КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ: ТРЕБОВАНИЯ ПРИ ПРОЕКТИРОВАНИИ 2-е изд., испр. и доп. Учебное пособие для вузов - М.:Издательство Юрайт - 2019 - 113с. - ISBN: 978-5-534-08546-4 - Текст электронный // ЭБС ЮРАЙТ - URL: <https://urait.ru/book/korporativnye-informacionnye-sistemy-trebovaniya-pri-proektirovanii-425572>

2. Компьютерные технологии анализа данных в эконометрике : монография / Д.М. Дайитбегов. - 3-е изд., доп. - М. : ИНФРА-М : Вузовский учебник, 2013. - 587 с. : ил., табл. - (Научная книга). - Библиогр.: с. 558-569. - Др. работы авт. на 4-й с. обл. - ISBN 978-5-9558-0275-6. - ISBN 978-5-16-006145-0

3. Уэс, Маккинли. Python и анализ данных / Маккинли Уэс : практическое пособие ; пер. А.А. Слинкин Электронный ресурс. - Python и анализ данных, 2019-04-19 : Профобразование ; Саратов, 2017. - 482 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-4488-0046-7

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для студентов по выполнению курсовой работы
по дисциплине «Анализ данных в бизнес-системах»
для студентов направления 38.03.05 «Бизнес-информатика»
(Направленность (профиль) «Информационная бизнес-аналитика и цифровые
технологии»)

Содержание

Формулировка задания	77
Тематика курсовых работ	78
Структура работы	79
Общие требования к написанию и оформлению работы	79
Требования к оформлению курсовой работы	79
Последовательность выполнения задания	84
Критерии оценивания	84
Порядок защиты курсовой работы	86

Цель, задачи и реализуемые компетенции

Курсовая работа является самостоятельной исследовательской работой студента и представляет собой логически завершенное и оформленное научное исследование.

Цель — формирование у студента навыков научно-исследовательской работы, повышение уровня его профессиональной (теоретической и практической) подготовки, углубление знаний по учебной дисциплине, развитие интереса и навыков самостоятельной работы с научной и справочной литературой.

Выполнение курсовой работы имеет целью расширение знаний студентов, обучение методам теоретического анализа явлений и закономерностей науки, отработку навыков самостоятельного применения теоретических знаний к комплексному решению профессиональных задач, использования справочной литературы, методов математической обработки экспериментальных данных, компьютерных технологий. Системой курсовых работ (проектов) студент подготавливается к выполнению выпускной квалификационной работы.

В процессе выполнения курсовой работы студентом должны решаться следующие задачи:

- приобретение новых теоретических знаний в соответствии с темой работы и заданием руководителя;
- умение систематизировать, обобщать и логично излагать концепции, альтернативные точки зрения по исследуемой проблеме;
- развитие учебно-исследовательских и методических навыков, необходимых для системного научного анализа изучаемого явления;
- совершенствование профессиональной подготовки.

Основной целью курсовой работы по курсу «Анализ данных в бизнес-системах» является систематизация, углубление и активное применение знаний по проектированию баз данных, закрепление знаний, полученных в лекционном курсе, а также на лабораторных занятиях.

Компетенции обучающегося, формируемые в результате написания курсовой работы – ПК-4

Формулировка задания

Темы курсовых работ утверждаются на заседании кафедры, ведущей дисциплины, но которым учебными планами предусмотрены курсовые проекты, в течение 2-х недель после начала семестра. Выписка из протокола заседания кафедры передается в дирекцию института, где формируется распоряжение об утверждении тем курсовых работ. В течение 10 дней после выхода распоряжения об утверждении тем курсовых работ специалист по учебно-методической работе дирекции вносит темы работ в автоматизированную базу университета.

Основные содержательные и процессуальные аспекты, необходимые для выполнения работы, оформляются кафедрой в заданиях по курсовому проектированию (Приложение 1). В заданиях необходимо четко сформулировать тему работы (проекта) и требования, определяющие ее объем, содержание, а также исходные данные.

Цель курсовой работы состоит в изучении теоретических основ темы исследования и применении инструментальных средств проектирования и создания БД.

Задачи

- изучить информационные источники по теме исследования;
- дать характеристику основным терминам и определениям по теме курсовой работы, привести классификацию;
- изучить инструментарий и привести пример практического применения в конкретной предметной области.

Перечень подлежащих разработке вопросов

- а) по теоретической части рассматриваются основные понятия БД;

б) по проектной части рассматриваются проектирование и создание БД с помощью инструментальных средств.

Тематика курсовых работ

1. Основные сферы применения Data mining.
2. Статистические методы анализа данных. Описательная статистика.
3. Статистические методы анализа данных. Проверка гипотез.
4. Статистические методы анализа данных. Определение размера выборки.
5. Основные методы выявления вероятностной природы данных.
6. Выявление связей и закономерностей в данных.
7. Основные методы анализа временных рядов и их применение при обработке данных.
8. Методы построения деревьев решений и их применение для анализа данных.
9. Методы кластер-анализа и их применение для обработки данных.
10. Применение корреляционно-регрессионного анализа для решения экономических задач.
11. Применение факторного анализа для решения экономических задач.
12. Применение методов анализа временных рядов для решения экономических задач.
13. Применение методов построения деревьев решений в экономике.
14. Анализ данных с использованием баз и хранилищ данных, SQL и OLAP технологии.
15. Инструменты анализа данных в системе Statistica.
16. Инструменты анализа данных в системе Excel.
17. Применение методов дисперсионного анализа в экономике.
18. Применение ковариационного анализа для решения экономических задач.
19. Обзор и основные характеристики отечественных и зарубежных статистических пакетов.
20. Ряды распределения. Понятие, статистические характеристики ряда и способы построения.
21. Систематическая выборка. Выборка Пуассона
22. Основные причины применения выборочных обследований в условиях рыночных отношений в России
23. Основы проектирования инструментария выборочного обследования
24. Периодические статистические наблюдения малых предприятий
25. Выборочные обследования бюджетов семей (домашних хозяйств).
26. Использование выборочного метода в исследовании коммерческой деятельности.
27. Выборочное наблюдение как метод исследования маркетинговой деятельности
28. Выборочный метод как инструмент менеджмента качества.
29. Выборочный метод в статистике цен на потребительском рынке
30. Локальные выборочные обследования

Структура работы

Курсовая работа должна состоять из введения, теоретической части, эмпирической (практической, расчетно-графической) части, заключения, списка информационных источников и приложения. В отдельных случаях, в соответствии с тематикой работы (проекта), эмпирическая часть может отсутствовать.

Общие требования к написанию и оформлению работы

Курсовая работа должна содержать элементы новизны, наряду с фундаментальным аспектом должен быть проведен анализ современного состояния изучаемой проблемы, а

также включенность в региональную проблематику. Задание по курсовой работе (проекту) необходимо индивидуализировать с учетом интересов и способностей студентов.

Титульный лист оформляется в соответствии с Приложением 2.

Во введении обосновывается актуальность выбранной темы исследования; отражаются объект, предмет, цель, задачи, методы, новизна, теоретическая и практическая значимость курсовой работы.

Теоретическая часть должна содержать анализ состояния изучаемой проблемы на основе обзора научной, научно-информационной, справочной литературы. Представленный материал должен быть логически связан с целью исследования. В параграфах теоретической части необходимо отражать отдельные компоненты проблемы и завершать их выводами.

Проектная часть включает в себя: характеристику предметной области, обоснование создания базы данных, разработку инфологической модели, концептуальной модели, физической модели базы данных; создание необходимых форм, запросов и отчетов; выводы.

Заключение содержит выводы, подтверждающие или опровергающие первоначальные предположения (гипотезы), перспективы дальнейшего изучения проблемы, связь с практикой, анализ реализации целей и задач исследования.

Список информационных источников должен быть составлен в соответствии с требованиями ГОСТа к оформлению библиографии.

Приложение содержит весь фактический материал экспериментальных исследований (анкеты, опросники, схемы, чертежи, расчетные материалы, карты, рисунки, ответы респондентов и т.д.).

Требования к оформлению курсовой работы

Курсовую работу (проект) рекомендуется представлять в объеме 1-2 печатных листа.

Текст работы должен быть машинописным на одной стороне белой бумаги формата А4 (210х297 мм). При выполнении работы необходимо соблюдать равномерную плотность, контрастность и четкость изображения по всей работе. В работе должны быть четкие, не расплывшиеся линии, буквы, цифры и знаки.

При выполнении работы на компьютере (в текстовом редакторе WORD) необходимо установить следующие параметры:

- размер левого поля - 30 мм, правого - 15 мм, верхнего и нижнего полей - 20мм;
- шрифт Times New Roman, размер шрифта 14;
- выравнивание по ширине, первая строка равна отступ 1,25 см, межстрочный интервал – 1,5 (отступ справа/слева и интервал перед/после равны нулю, разбивка на страницы без запрета висячих строк).

Цвет шрифта - черный, подчеркивания и полужирный шрифт не допускаются. Разрешается использовать компьютерные возможности акцентирования внимания на определенных формулах, теоремах, применяя шрифты разной гарнитуры.

Все страницы курсовой работы, включая иллюстрации и приложения, нумеруются по порядку без пропусков и повторений. Первой страницей считается титульный лист, на котором номер страницы не ставится, на следующей странице ставится цифра «2» и т.д. В общую нумерацию входят список информационных источников и приложения.

Порядковый номер страницы проставляют в центре нижней части листа без точки.

Главы курсовой работы должны иметь порядковые номера в пределах всей работы, обозначенные арабскими цифрами и записанные с абзацного отступа (например, 1, 2, 3 и т.д.). Нумерация пунктов должна быть в пределах каждой главы, и номер пункта должен состоять из номеров главы и пункта в главе, разделенных точкой (например, 1.1, 1.2, 1.3 и т.д.).

Слово «Глава» не пишется.

Пункты, при необходимости, могут быть разбиты на подпункты, которые должны иметь порядковую нумерацию в пределах каждого пункта (например, 1.1.1, 1.1.2, 1.1.3 и т.д.).

В конце номера главы и пункта (подпункта) точка не ставится.

Каждую главу курсовой работы необходимо начинать с нового листа (страницы).

Заголовки глав следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая.

Заголовки пунктов и подпунктов следует печатать с абзацного отступа с прописной буквы без точки в конце, выравнивая по ширине.

Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой. Расстояние после заголовка главы и структурного элемента - 2 свободные строки, между текстом и заголовком пункта - 2 свободные строки, между заголовком пункта и текстом - 1 свободная строка.

Внутри пунктов или подпунктов могут быть приведены перечисления. Перед каждой позицией перечисления следует ставить дефис (использование других символов не допускается), запись производится с абзацного отступа.

При необходимости ссылки в тексте работы на один из элементов перечисления вместо дефиса ставятся строчные буквы в порядке русского алфавита, начиная с буквы а (за исключением букв ё, з, й, о, ч, ъ, ы, ь). Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, запись производится с абзацного отступа.

В тексте документа, за исключением формул, таблиц и рисунков, не допускается:

- применять математический знак минус (-) перед отрицательными значениями величин (следует писать слово «минус»);

- применять знак « Φ » для обозначения диаметра (следует писать слово «диаметр»). При указании размера или предельных отклонений диаметра на чертежах, помещенных в тексте документа, перед размерным числом следует писать знак « Φ »;

- применять без числовых значений математические знаки, например > (больше), < (меньше), = (равно), $\geq$ (больше или равно), $\leq$ (меньше или равно), $\neq$ (не равно), а также знаки № (номер), % (процент);

- применять индексы стандартов, технических условий и других документов без регистрационного номера.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должна быть оставлена одна свободная строка.

В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами. Пояснения символов и числовых коэффициентов, входящих в формулу, если они не пояснены ранее в тексте, должны быть приведены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться со слова «где» без двоеточия после него. Формулы, за исключением формул, помещаемых в приложении, должны нумероваться сквозной нумерацией арабскими цифрами, которые записывают на уровне формулы справа в круглых скобках. Одну формулу обозначают - (1). Допускается нумерация формул в пределах главы. В этом случае номер формулы состоит из номера главы и порядкового номера формулы, разделенных точкой, например (3.1). Формулы, помещаемые в приложениях, должны нумероваться отдельной нумерацией арабскими цифрами в пределах каждого приложения с добавлением перед каждой цифрой обозначения приложения, например формула (B.1).

Формулы, следующие одна за другой и не разделенные текстом, разделяют запятой.

Если уравнение не умещается в одну строку, то оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:) или других математических знаков, причем знак в начале следующей строки повторяют. При переносе формулы на знаке, символизирующем операцию умножения, применяют знак « $\times$ ».

Ссылки в тексте на порядковые номера формул дают в скобках, например, ... в формулах (5, 6).

Формулы в работе следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке.

$$p = \frac{(\text{ЧП}_0 - \overline{\text{ЧП}}) * \gamma(z) + \sqrt{D \text{ЧП}} * \beta(z)}{\overline{\text{ЧП}}} * 100, \quad (5)$$

$$z = \frac{(\text{ЧП}_0 - \overline{\text{ЧП}})}{\sqrt{D \text{ЧП}}}, \quad (6)$$

где p - риск инвестиционного проекта;

ЧП_0 - планируемая величина чистой прибыли, руб.;

$\overline{\text{ЧП}}$ - среднее значение чистой прибыли при различных вариантах, руб.;

$D \text{ЧП}$ - дисперсия чистой прибыли;

$\gamma(z)$ - функция Лапласа;

$\beta(z)$ - функция Гаусса.

Порядок изложения в документах математических уравнений такой же, как и формул.

Таблицы применяют для лучшей наглядности и удобства сравнения показателей. Название таблицы должно отражать ее содержание, быть точным, кратким.

Таблицы, за исключением таблиц приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Если в документе одна таблица, она должна быть обозначена «Таблица 1» или «Таблица В.1», если она приведена в приложении В.

Допускается нумеровать таблицы в пределах главы. В этом случае номер таблицы состоит из номера главы и порядкового номера таблицы в главе, разделенных точкой (Таблица 2.1, 2.2, 2.3 и т.д.).

Наименование таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире.

Таблица _____ – _____
номер Название таблицы с большой буквы (точка в конце не ставится)

1	2	3	4

На все таблицы работы должны быть приведены ссылки в ее тексте, при ссылке следует писать слово «таблица» с указанием ее номера.

Заголовки граф и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки граф - со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставят. Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк.

Таблицы слева, справа и снизу ограничивают линиями. Допускается применять размер шрифта в таблице меньший, чем в тексте.

Разделять заголовки и подзаголовки боковика и граф диагональными линиями не допускается. Горизонтальные и вертикальные линии, разграничивающие строки таблицы, допускается не проводить, если их отсутствие не затрудняет пользование таблицей.

Заголовки граф, как правило, записывают параллельно строкам таблицы. При необходимости допускается перпендикулярное расположение заголовков граф.

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а при необходимости, в приложении к документу.

Допускается помещать таблицу вдоль длинной стороны листа документа.

Таблицу с большим числом строк допускается переносить на другой лист (страницу). При переносе части таблицы на другой лист (страницу) слово «Таблица», ее номер и наименование указывают один раз слева (без абзацного отступа) над первой частью таблицы, а над другими частями также слева пишут слова «Продолжение таблицы» и указывают номер таблицы.

Таблицу с большим количеством граф допускается делить на части и помещать одну часть под другой в пределах одной страницы. Если строки и графы таблицы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае - боковик. При делении таблицы на части допускается ее головку или боковик заменять соответственно номером граф и строк. При этом нумеруют арабскими цифрами графы и (или) строки первой части таблицы. В рамках работы должно соблюдаться единообразие в оформлении переноса таблиц.

Если повторяющийся в разных строках графы таблицы текст состоит из одного слова, то его после первого написания допускается заменять кавычками; если из двух и более слов, то при первом повторении его заменяют словами «То же», а далее - кавычками. Ставить кавычки вместо повторяющихся цифр, марок, знаков, математических и химических символов не допускается.

Примечания приводят в документах, если необходимы пояснения или справочные данные к содержанию текста, таблиц или графического материала. Примечания следует помещать непосредственно после текстового, графического материала или в таблице, к которым относятся эти примечания.

Слово «Примечание» следует печатать с прописной буквы с абзаца.

Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается тоже с прописной буквы. Одно примечание не нумеруют. Несколько примечаний нумеруют по порядку арабскими цифрами без проставления точки. Примечание к таблице помещают в конце таблицы над линией, обозначающей окончание таблицы.

Иллюстрации (чертежи, графики, схемы, диаграммы и др.) следует располагать в работе непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные. На все иллюстрации должны быть даны ссылки в работе.

Чертежи, графики, диаграммы, схемы, иллюстрации, помещаемые в курсовую работу, должны соответствовать требованиям государственных стандартов Единой системы конструкторской документации (ЕСКД).

Допускается выполнение чертежей, графиков, диаграмм, схем посредством использования компьютерной печати.

Количество иллюстраций должно быть достаточным для пояснения излагаемого текста. Иллюстрации, за исключением иллюстраций приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается «Рисунок 1».

Допускается нумеровать иллюстрации в пределах главы. В этом случае номер иллюстрации состоит из номера главы и порядкового номера иллюстрации в главе, разделенных точкой.

Иллюстрации должны иметь наименование. Слово «Рисунок» и его наименование располагают под иллюстрацией посередине строки: Рисунок 1.1 – Название рисунка (с большой буквы, точку в конце не ставить).

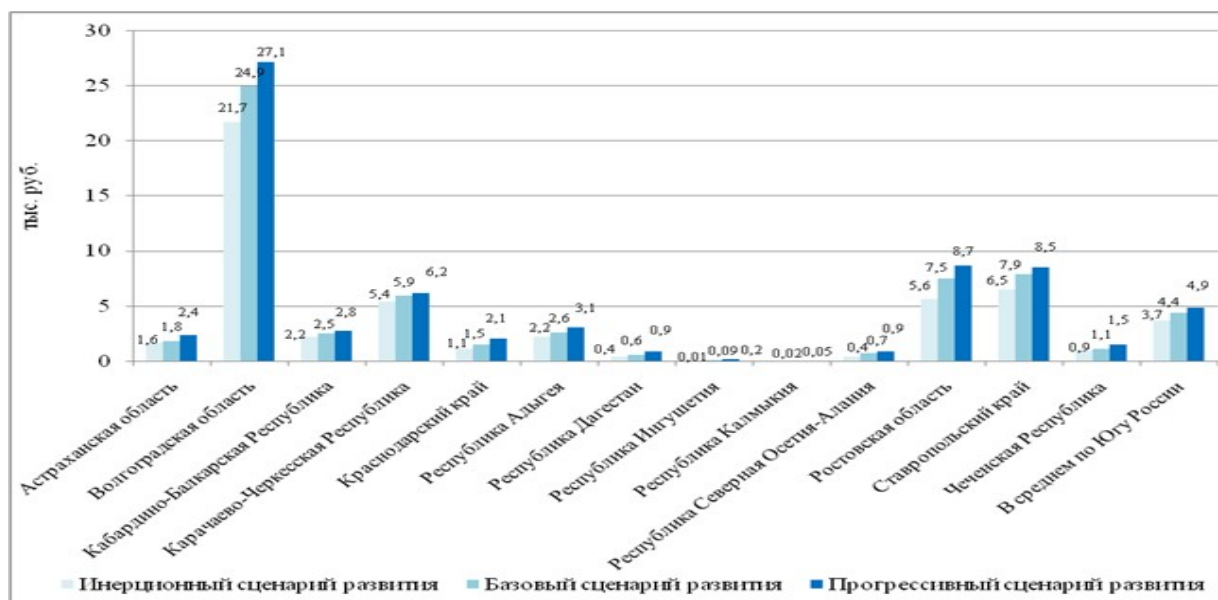


Рисунок 2.3 – Прогнозный выпуск инновационной продукции на душу населения в регионах Юга России в 2021 году, тыс. руб.

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например: Рисунок А.3.

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2» при сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах главы.

Список литературы оформляется в соответствии с требованиями ГОСТа к оформлению библиографии; в нем указываются все использованные в работе источники литературы и документации (приложение М).

Допускаются следующие способы группировки библиографических записей: алфавитный, систематический или в порядке появления ссылок на источники в тексте. При алфавитном способе группировки все библиографические записи располагают по алфавиту фамилий авторов или первых слов заглавий документов. Библиографические записи произведений авторов-однофамильцев располагают в алфавите их инициалов. При систематической (тематической) группировке материала библиографические записи располагают в определенной логической последовательности в соответствии с принятой системой классификации.

Приложение оформляют как продолжение данного документа на последующих его листах или выпускают в виде самостоятельного документа.

Приложения могут быть обязательными и информационными.

Информационные приложения могут быть рекомендуемого или справочного характера.

В тексте документа на все приложения должны быть даны ссылки. Степень обязательности приложений при ссылках не указывается. Приложения располагают в порядке ссылок на них в тексте документа. Каждое приложение следует начинать с новой страницы с указанием сверху посередине страницы слова «Приложение» и его обозначения. Приложение должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. После слова «Приложение» следует буква, обозначающая его последовательность. Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O.

В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами.

Если в документе одно приложение, оно обозначается «Приложение А».

Приложения, как правило, выполняют на листах формата А4.

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения. Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

Все приложения должны быть перечислены в содержании курсовой работы (при наличии) с указанием их обозначений и заголовков.

Последовательность выполнения задания

Содержание основных этапов подготовки курсовой работы представлено в таблице:

№ п/п	Содержание блока	Число %	Срок выполнения (номер недели)	Рекомендуемая литература
1	Получения задания. Литературный обзор. Теоретическая часть	15%	1-5	Основная литература: 1,2 Дополнительная литература: 1-3
2	Аналитический обзор предметной области.	15%	6	Основная литература: 1,2
3	Практическая часть	45%	7-8	Дополнительная литература: 1-3
5	Оформление отчета	15%	9	Основная литература: 1,2
6	Защита работы	10%	10-11	
	Итого:	100%		

Критерии оценивания

Оценка «отлично» выставляется студенту, если в курсовой работе глубоко и прочно усвоен программный материал, исчерпывающе, последовательно, четко и логически стройно построен, тесно увязаны теория с практикой, решены поставленные задачи, вопросы. В работе использована литература, правильно обосновывается принятое решение, раскрыты приемы выполнения практических задач. Компетенции ПК-4 освоены на повышенном уровне.

Оценка «хорошо» выставляется студенту, если в курсовой работе глубоко и прочно усвоен программный материал, исчерпывающе, последовательно, четко и логически стройно построен, тесно увязаны теория с практикой, решены поставленные задачи, вопросы. В работе правильно обосновывается принятое решение, раскрыты приемы выполнения практических задач, но существуют ошибки в выполнении задач, неправильно записаны запросы, построены и спроектированы задачи. Компетенции ПК-4 освоены полностью на базовом уровне.

Оценка «удовлетворительно» выставляется студенту, если в курсовой работе допускаются неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, неправильно выполнены практические работы. Если какая-либо часть курсовой работы является плагиатом, скомпилирована из фрагментов работ других авторов и носит несамостоятельный характер. Оформление работы совершенно не соответствует требованиям. Компетенции ПК-4 освоены частично на базовом уровне.

Оценка «неудовлетворительно» выставляется студенту, который не выполнил курсовую работу, либо выполнил с грубыми нарушениями требований, не раскрыл заявленную тему, не выполнил практической части работы. Содержание курсовой работы не соответствует ее теме. При написании работы не были использованы источники и литература. Компетенции ПК-4 не освоены.

Порядок защиты курсовой работы

Защита курсовой работы является обязательной формой проверки выполнения работы. Защита производится на заседании кафедры, научно-методического семинара кафедры, научной проблемной группы специальной комиссией, утверждаемой директором института, состоящей обычно из 3 преподавателей кафедры, при непосредственном участии руководителя, в присутствии студентов. Результаты наиболее интересных курсовых работ (проектов) могут быть доложены на научных конференциях. Публичная защита стимулирует научный интерес, творчество, ответственность студентов.

Защита состоит в коротком докладе студента по выполненной работе и в ответах на вопросы присутствующих на защите. Научный руководитель зачитывает отзыв на курсовую работу (проект) студента (Приложение 3).

Результаты защиты курсовой работы, согласно действующему Положению о текущем контроле и промежуточной аттестации в СКФУ, оцениваются дифференцированной отметкой по пятибалльной системе..

Студент, не представивший в установленный срок курсовую работу (проект) или не защитивший ее по неуважительной причине, считается имеющим академическую задолженность.

Список литературы

основная литература:

1. Нестеров, С. А. Интеллектуальный анализ данных средствами MS SQLServer 2008 / С.А. Нестеров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 338 с

2. Симчера, В. М. Методы многомерного анализа статистических данных : Учебное пособие / Симчера В. М. - Москва : Финансы и статистика, 2014. - 400 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-279-03184-9

дополнительная литература:

1. Астапчук В. А., Терещенко П. В. - КОРПОРАТИВНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ: ТРЕБОВАНИЯ ПРИ ПРОЕКТИРОВАНИИ 2-е изд., испр. и доп. Учебное пособие для вузов - М.:ИздательствоЮрайт - 2019 - 113с. - ISBN: 978-5-534-08546-4 - Текст электронный // ЭБС ЮРАЙТ - URL: <https://urait.ru/book/korporativnye-informacionnye-sistemy-trebovaniya-pri-proektirovanii-425572>

2. Компьютерные технологии анализа данных в эконометрике : монография / Д.М. Дайитбегов. - 3-е изд., доп. - М. : ИНФРА-М : Вузовский учебник, 2013. - 587 с. : ил., табл. - (Научная книга). - Библиогр.: с. 558-569. - Др. работы авт. на 4-й с. обл. - ISBN 978-5-9558-0275-6. - ISBN 978-5-16-006145-0

3. Уэс, Маккинли. Python и анализ данных / Маккинли Уэс : практическое пособие ; пер. А.А. Слинкин Электронный ресурс. - Python и анализ данных, 2019-04-19 : Профобразование ; Саратов, 2017. - 482 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-4488-0046-7

Приложение 1

УТВЕРЖДАЮ
Заведующий кафедрой
Цифровых бизнес-технологий и систем учета,
кандидат экономических наук, доцент

_____ А.А. Калашников

Институт экономики и управления
Кафедра цифровых бизнес-технологий и систем учета
Направление Бизнес-информатика

ЗАДАНИЕ на курсовую работу

студента Иванова Иван Ивановича
по дисциплине Анализ данных в бизнес-системах

1. Тема работы Автоматизация управления эффективностью бизнеса на основе системы сбалансированных показателей

2. Цель состоит в изучении аспектов автоматизации управления эффективностью бизнеса на основе системы сбалансированных показателей.

3. Задачи

3.1 выявить ключевые аспекты ССП как инструмента управления бизнеса;

3.2 рассмотреть основные принципы ССП;

3.3 охарактеризовать особенности ССП в части проектирование бизнеса, карты стратегии.

4. Перечень подлежащих разработке вопросов

а) по теоретической части рассматривается Сбалансированная система показателей как инструмент реализации стратегии

б) по проектной части рассматриваются особенности автоматизации управления эффективностью бизнеса.

5. Исходные данные

а) по литературным источникам Интернет-сайты, журналы, газеты

б) по вариантам, разработанным преподавателем

в) иное данные информационных агентств, рейтинговых агентств

6. Список рекомендуемой литературы

6.1 Д. Нортон, Р. Каплан. Сбалансированная система показателей. От стратегии к действию. - М.: Олимп-Бизнес, 2012.

6.2 Д. Нортон, Р. Каплан. Стратегическое единство. Создание синергии организации с помощью сбалансированной системы показателей - М.:

Вильямс, 2012.

6.3 Евченко А.В., Методы исследования систем управления [Текст]: Учеб.пособие. /А.В. Евченко, Э.Н. Кузьбожев/-Курск: Курск. Гос. техн. ун-т, 2001. 168 с.

6.4 Игнатьева А. В. Исследование систем управления [Текст] : учебное пособие для студентов вузов. - М. : ЮНИТИ-ДАНА, 2002. - 157 с.

6. Контрольные сроки предоставления отдельных разделов курсовой работы

25% - подготовительная часть «11» марта 2026 г.

50% - теоретическая часть «31» марта 2026 г.

75% - проектная часть «19» апреля 2026 г.

100% - курсовая работа к защите – «13» мая 2026 г.

7. Срок защиты курсовой работы «24» мая 2026 г.

Дата выдачи задания «16» февраля 2026 г.

Руководитель курсовой работы
кандидат экономических наук,
доцент

А.А. Калашников

Задание принял к исполнению студент очной формы обучения 2 курса группы
БИН-б-о-25-1

И.И. Иванов

(личная подпись)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЭКОНОМИКИ И УПРАВЛЕНИЯ
КАФЕДРА ЦИФРОВЫХ БИЗНЕС-ТЕХНОЛОГИЙ И СИСТЕМ УЧЕТА

КУРСОВАЯ РАБОТА

по дисциплине
«Анализ данных в бизнес-системах»

на тему
«Автоматизация управления эффективностью бизнеса на основе системы
сбалансированных показателей»

Выполнил:

Иванов Иван Иванович
студент 2 курса
группы БИН-б-о-19-1
очной формы обучения

Руководитель работы:

Калашников Александр
Александрович
кандидат экономических наук,
доцент кафедры цифровых
бизнес-технологий и систем
учета

Работа допущена к защите _____ 20.05.2026 г. _____

Работа выполнена и
защищена с оценкой _____ Дата защиты _____

Члены комиссии: _____ А.А. Калашников _____
(должность) (подпись) (И.О. Фамилия)

_____ Е.Ю. Калашникова
_____ Д.Г. Ловянников

Ставрополь, 2026 г.

Отзыв
на курсовую работу студента 2 курса
Иванова Иван Ивановича
«Автоматизация управления эффективностью бизнеса на основе системы
сбалансированных показателей»

Актуальность: курсовая работа посвящена изучению методик автоматизации управления эффективностью бизнеса на основе системы сбалансированных показателей. В настоящее время любая компания нуждается в бизнес-процессах, оптимально организованных для выживания в условиях глобальной конкуренции и динамичной бизнес среды. На сегодняшний день наиболее эффективным инструментом стратегического управления является Сбалансированная система показателей – ССП (Balanced Scorecard - BSC).

С этих позиций особую актуальность приобретает изучение теоретических основ автоматизации управления бизнеса.

В первой главе рассматриваются основы целевого управления, ССП, ключевые аспекты, показатели.

Вторая глава посвящена непосредственно этапам разработки проекта, формулированию карты стратегии, построению карт распределения ответственности.

Выводы, сделанные в Заключение, соответствуют целям, поставленным во Введении.

Проанализирован значительный объем литературы – 34 источника.

За время работы студент проявил себя как творческий и инициативный исследователь. В процессе подготовки и написания работы студент соблюдал сроки выполнения календарного графика и проявил отличные навыки в работе с теоретическими источниками.

Таким образом, работа выполнена на высоком теоретическом и практическом уровне, соответствует требованиям, предъявляемым к курсовым работам, и заслуживает высокой оценки.

Научный руководитель
кандидат экономических наук,
доцент

А.А. Калашников

«.....» 2026 г.