

Министерство образования Республики Беларусь
Учреждение образования «Витебский государственный
университет имени П.М. Машерова»
Кафедра информатики и информационных технологий

Н.Д. Адаменко

Модели данных и СУБД

Методические рекомендации

В 3 частях

ЧАСТЬ 3

*Витебск
ВГУ имени П.М. Машерова
2015*

УДК 004.65(075.8)
ББК 32.972.32я73
А28

Печатается по решению научно-методического совета учреждения образования «Витебский государственный университет имени П.М. Машерова». Протокол № 1 от 23.10.2015 г.

Автор: доцент кафедры информатики и информационных технологий ВГУ имени П.М. Машерова, кандидат педагогических наук
Н.Д. Адаменко

Р е ц е н з е н т ы :

кафедра математики и информационных технологий УО «ВГТУ»;
доцент кафедры математики и информационных технологий
УО «ВГТУ», кандидат физико-математических наук *Т.В. Никонова*

Адаменко, Н.Д.

А28 Модели данных и СУБД : методические рекомендации : в 3 ч. /
Н.Д. Адаменко. – Витебск : ВГУ имени П.М. Машерова,
2015. – Ч. 3. – 54 с.

Учебное издание содержит описание технологии проектирования баз данных с помощью СУБД MS SQL Server, а также методические материалы для проведения лабораторных занятий, последовательно формирующих основные умения, необходимые для эффективной работы с СУБД MS SQL Server. Материалы предлагаемого издания могут найти применение при изучении дисциплин, связанных с освоением способов разработки многопользовательских систем баз данных: «Модели данных и СУБД» для специальностей «Программное обеспечение компьютерных систем», «Прикладная информатика (веб-программирование и компьютерный дизайн)», «Компьютерная безопасность (радиофизические методы и программно-технические средства)», «Программное обеспечение информационных технологий», «Информационные системы» для специальности «Математика и информатика».

УДК 004.65(075.8)
ББК 32.972.32я73

© Адаменко Н.Д., 2015
© ВГУ имени П.М. Машерова, 2015

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА	
Тема: Администрирование баз данных. Резервное копирование	4
ЛИТЕРАТУРА	11
ПРИЛОЖЕНИЯ	12
Приложение 1. Варианты индивидуальных заданий на проектирование и разработку БД	12
Приложение 2. Типы данных, используемые в SQL-сервере	41
Приложение 3. Типы ограничений	42
Приложение 4. Данные для заполнения таблиц	43
Приложение 5. Краткое определение основных терминов	46
Приложение 6. Дополнительные параметры создаваемого индекса ...	48
Приложение 7. Список функций SQL Server	49

ЛАБОРАТОРНАЯ РАБОТА

ТЕМА: АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ. РЕЗЕРВНОЕ КОПИРОВАНИЕ

Цель работы: Ознакомиться с понятием привилегий, способами их создания. Изучить основные способы резервного копирования баз данных.

ОСНОВНЫЕ СВЕДЕНИЯ

Безопасность баз данных и привилегии

При хранении информации в СУБД одной из основных задач является обеспечение безопасности данных. В языке SQL используются следующие основные принципы защиты данных:

В БД действующими лицами являются пользователи. Манипуляции с данными происходят от имени конкретного пользователя;

В SQL используется система привилегий, т.е. прав пользователя на проведение тех или иных действий над определённым объектом базы данных.

Администратор БД создаёт пользователей и даёт им привилегии, пользователи, которые создают таблицы, сами имеют права на управление этими таблицами.

Каждый пользователь в среде SQL имеет специальное идентификационное имя. Команда, посланная в БД, ассоциируется с определённым пользователем. Каждый пользователь в SQL имеет набор привилегий. Эти привилегии могут изменяться со временем – новые добавляться, старые удаляться. При этом пользователь, создавший таблицу любого вида, является ею владельцем. Это означает, что такой пользователь имеет все привилегии в этой таблице и может передавать привилегии другим пользователям для данной таблицы.

Существуют следующие привилегии:

SELECT – пользователь может выполнять запросы к таблице;

INSERT – пользователь может выполнять вставку записей;

UPDATE – пользователь может выполнять корректировку данных;

DELETE – пользователь может выполнять удаления в таблице;

REFERENCES – пользователь может определять внешние ключи;

INDEX – пользователь может создавать индекс в таблице;

SYNONYM – пользователь может создавать синонимы для объекта;

ALTER – пользователь может выполнять команду ALTER TABLE.

В SQL привилегии даются и отменяются двумя операторами – GRANT (допуск) и REVOKE (отмена).

Синтаксис:

GRANT *привилегия1, привилегия2...*

ON *таблица*

TO *пользователь1, пользователь2...*;

Для оператора GRANT существует два аргумента, которые имеют специальное значение – это ALL PRIVILEGES (используется вместо имени привилегии, чтобы отдать все привилегии в таблице) и PUBLIC (используется вместо имени пользователя, чтобы отдать соответствующую привилегию всем пользователям).

Для того чтобы пользователи могли передавать свои привилегии другим пользователям, используется оператор WITH GRANT OPTION.

Синтаксис:

GRANT *привилегия1, привилегия2...*

ON *таблица*

TO *пользователь1, пользователь2...*

WITH GRANT OPTION;

С помощью этого оператора пользователь получает особые привилегии для данной таблицы, и может предоставить эту привилегию к той же таблице другому пользователю.

Для удаления привилегии используется оператор

Синтаксис:

REVOKE *привилегия*

ON *таблица*

FROM *пользователь*;

При этом привилегии отменяются пользователем, который их предоставил.

Управление привилегиями и правами доступа

В системе SQL-сервер организована двухуровневая настройка ограничения доступа к данным. На первом уровне в системе необходимо создать так называемое *Имя входа* пользователя, что позволяет ему подключиться к серверу. На втором уровне для каждой базы данных необходимо создать запись пользователя, т.е. с помощью Имен входа осуществляется подключение к SQL-серверу, после чего определяются уровни доступа этого пользователя для каждой базы данных в отдельности.

Для создания учетных записей пользователей в SQL Server необходимо:

Выбрать в группе объектов SQL-сервера *Имена входа* в списке *Безопасность*, после чего выполнить команду *Создать имя входа*. На экран будет выведено диалоговое окно настройки параметров создаваемой учетной записи пользователя.

На закладке *Общие* окна, представленного на рисунке 1 в поле *Имя входа* вводится имя учетной записи, а поле *Password* – пароль пользователя (при выборе проверки подлинности SQL Server). Здесь также можно установить базу данных, подключение к которой осуществляется по умолчанию при входе пользователя в систему (список <База данных по умолчанию>), а также используемый по умолчанию для данного пользователя язык.

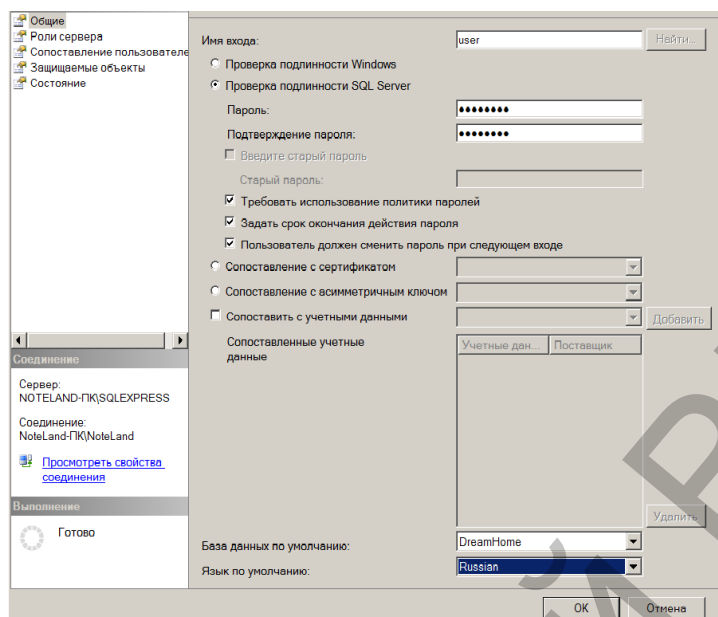


Рисунок 1

В закладке *Роли сервера* этого диалогового окна представлены опции, с помощью которых назначаются серверные роли для создаваемой учетной записи. Возможны следующие серверные роли:

Sysadmin – любые функции администрирования;

Securityadmin – управление доступом, возможность создания баз данных;

Serveradmin – настройка конфигураций и выполнение функций закрытия SQL-сервера;

Setupadmin – управление связями между серверами;

Processadmin – управление процессами, выполняющимися на сервере;

Diskadmin – управление файлами сервера;

Dbcreator – управление процессами создания и удаления баз данных.

С помощью закладки *Сопоставление пользователя* осуществляется настройка роли для базы данных. В верхнем списке осуществляется выбор требуемой базы данных (базы, к которой будет разрешен доступ). А в нижнем списке отображается перечень ролей доступа к объектам базы данных. Возможны следующие роли:

public – нет специальных ролей;

db_owner – полный доступ;

db_accessadmin – возможность добавления и удаления пользователей базы данных;

db_securityadmin – управление всеми процессами доступа пользователей к объектам базы данных;

db_ddladmin – выполнение основных команд, кроме GRANT, REVOKE;

db_backupoperator – функции запуска процедуры резервного копирования;

db_datareader – возможность чтения всех данных из любых таблиц базы данных;

db_datawriter – возможность изменения всех данных из любых таблиц базы данных.

Дальнейшее изменение настроек учетной записи пользователя можно осуществлять с помощью выбора необходимого пользователя в списке *Имя входа* и выбора пункта *Свойства*. Учетные записи доступа к SQL-серверу отображаются в группе объектов *Безопасность/Имена входа* сервера, а права доступа к базам данных в группе *Безопасность/Имена входа* баз данных.

Для создания пользователя в базе данных необходимо:

Выполнить команду *Безопасность/Пользователи/Создать пользователя* нужной базы данных.

На вкладке *Общие* ввести Имя пользователя и выбрать ранее созданное Имя входа в окне *Выбор имени входа*.

Для настройки прав пользователя в базе данных необходимо:

Выбрать *имя входа* в группе *Безопасность* нужной базы данных.

Выбрав в контекстном меню нужного пользователя *Свойства*, настроить параметры пользователя:

На вкладке *Общие* указать *Схемы, принадлежащие данному пользователю* и *Членство в роли базы данных*

На вкладке *Защищаемые объекты* щелкнуть *Найти*, окне *Добавление объектов* выбрать *Определенные объекты*, в окне *Выбор объектов* указать *Типы объектов* (например, *Таблицы*).

Указать, к каким таблицам определен доступ, и определить, какие манипуляции с выбранными объектами будут разрешены и какие – запрещены пользователю.

Резервное копирование

При разработке систем баз данных особое внимание уделяется вопросам резервного копирования. В системе SQL-сервер имеются инструменты для создания резервных копий и последующего восстановления баз данных. При этом необходимо разработать стратегию создания резервных копий – периодичность создания, размещение файлов, перечень баз данных, требующих копирования, время запуска процедуры и т.д.

Процесс создания и восстановления резервных копий баз данных осуществляется с помощью среды SQL Server Management Studio. Перед началом процесса резервирования баз данных можно настроить новое хранилище данных – устройство, используемое для хранения резервных копий.

Для создания хранилища данных надо:

Выбрать в списке объектов SQL-сервера *Устройства резервного копирования*, группу *Объекты сервера*.

Выполнить команду *Создать устройство резервного копирования*, после чего на экране отобразится диалоговое окно настройки параметров нового

хранилища данных.

В поле *Имя устройства* необходимо указать имя создаваемого устройства. После сохранения параметров устройства его имя отобразится в перечне объектов.

В поле *Файл* необходимо указать путь к файлу, в котором будет храниться резервная копия.

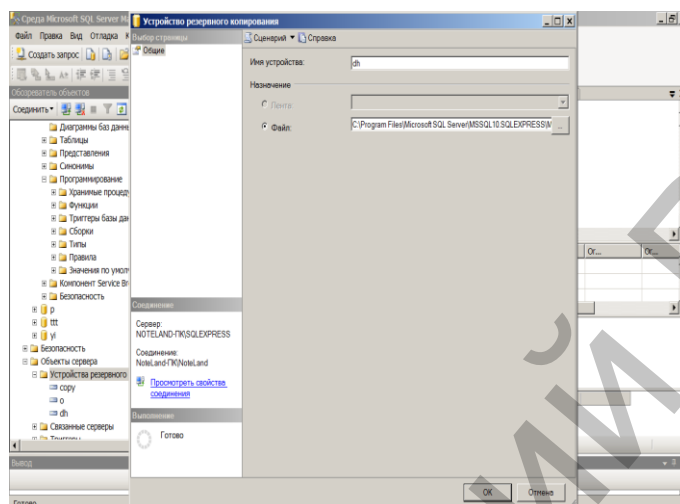


Рисунок 2

Выбрать в списке объектов SQL-сервера базу данных, резервное копирование которой необходимо осуществить, после чего выполнить команду *Задачи/Создать резервную копию*. При выполнении этого действия на экране отобразится диалоговое окно настройки процесса резервного копирования.

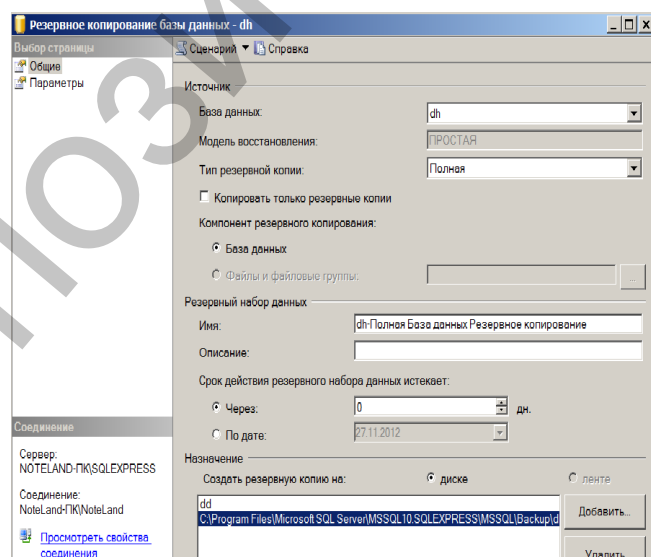


Рисунок 3

В поле *База данных* отображается имя базы данных, резервное копирование которой надо выполнить. В поле *Описание* можно определить краткое описание создаваемого процесса.

Указать, когда истекает срок действия резервного набора данных.

В поле *Создать резервную копию на* выбрать нужное хранилище либо щелкнуть на кнопке *Добавить*, и в окне *Выбор места расположения резервной копии* выбрать либо *Имя файла* (и задать путь к файлу, в котором будет храниться копия), либо *Устройство резервного копирования*. (По умолчанию выбирается путь на жестком диске компьютера в папку Backup основного расположения баз SQL-сервера. Для того чтобы изменить место размещения копии, сначала нажать «Удалить» (*Remove*), чтобы удалить существующее назначение, а затем «Добавить» (*Add...*) для добавления нового).

Способы восстановления данных из резервной копии:

Выбрать в списке объектов SQL-сервера базу данных, которую надо восстановить или установить курсор на группе.

Выполнить команду *Задачи/Восстановить/База данных*, что приведет к открытию диалогового окна восстановления базы данных.

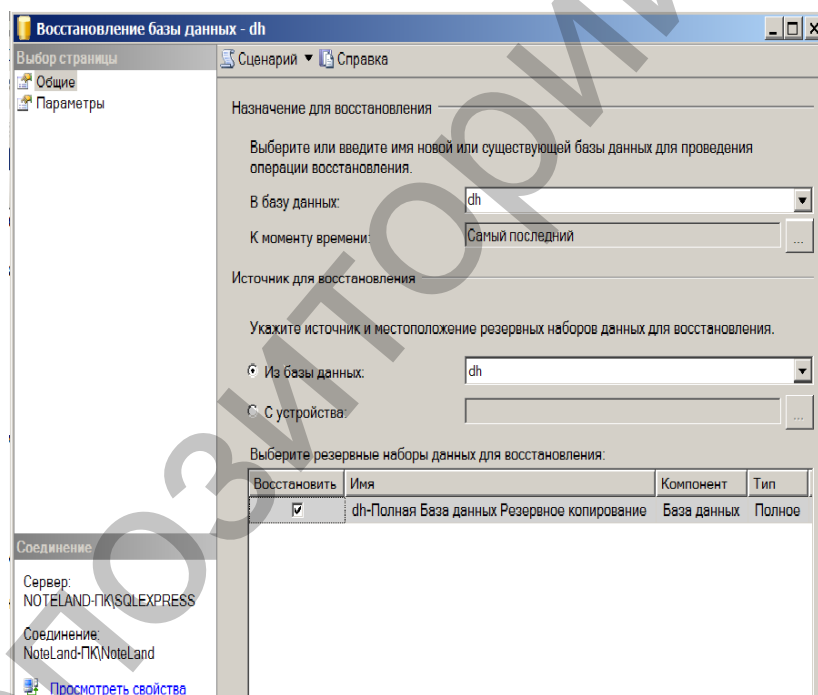


Рисунок 4

В поле «В базу данных» следует ввести имя для восстанавливаемой базы (если будет указано имя существующей базы, это эквивалентно тому, что сначала полностью удалить существующую базу и затем восстановить из резервной копии новую).

Затем установить переключатель «С устройства» и указать путь к файлу резервной копии, нажав кнопку «...», затем установить галочку «Восстановить» в нужной строке (их может быть несколько, если один файл *.bak содержит несколько резервных копий базы).

На странице «Параметры» установить галочку «Перезаписать существующую базу данных» и **проверить пути** в списке «Восстановить файлы базы данных как» (пути должны указывать на существующую папку на SQL-сервере, к которой предоставлены права на запись – пути по умолчанию обычно должны заканчиваться папкой DATA).

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие реализуются в основные принципы защиты данных в языке SQL?
2. Какие привилегии могут быть созданы в SQL?
3. С помощью каких команд в SQL даются и отменяются привилегии?
4. Что представляет собой двухуровневая настройка ограничения доступа к данным?
5. Как изменить настройки созданной учетной записи?
6. Для чего нужны системные таблицы?
7. Что такое системный каталог?
8. Что необходимо сделать перед началом процесса резервного копирования баз данных?

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. С помощью обозревателя MS SQL Server Management Studio создайте новую роль в своей базе данных и определите для нее набор привилегий для работы с таблицами STAFF и PROPERTY (чтение данных из таблиц, обновления, вставки данных, просмотр определения).
2. Создайте имя входа, задав для него проверку подлинности SQL Server.
3. С помощью команды CREATE создайте нового пользователя в своей БД, и, затем, определите для него созданную Вами роль с помощью системной процедуры sp_addrolemember.
4. С помощью команды GRANT обеспечьте для этого пользователя две привилегии, дублирующие привилегии роли, которой он относится. Аннулируйте одну из привилегий с помощью команды REVOKE, а вторую - с помощью команды DENY. Проверьте, сохранились ли у пользователя аннулированные привилегии.
5. Определите набор полномочий для роли GUEST, проверьте возможность выполнения пользователем GUEST разрешённых ему действий.
6. Создайте новое хранилище данных для дальнейшего резервного копирования.
7. Настройте параметры вновь созданного хранилища данных.
8. Проверьте записанную информацию в хранилище данных.
9. Удалите некоторые записи из любой таблицы, а затем восстановите базу данных из созданной резервной копии.

ЛИТЕРАТУРА

1. Базы данных: модели, разработка, реализация / Т.С. Карпова. СПб., 2001.
2. Программирование баз данных MS SQL Server 2005/ Базовый курс. / Роберт Виейра. – М. Диалектика. 2007. – 831 с.
3. Конолли, Томас, Берг, Каролин, Страчан, Анна. Базы даны: проектирование, реализация и сопровождение. Теория и практика: учеб. пос. – 2-е изд. / Пер с англ. – М., 2000.
4. Астахова И. Ф. SQL в примерах и задачах: учеб. пособие для студ. высш. учеб. заведений. – Минск: Новое знание, 2002. – 176 с.
5. Хомоненко А.Д. Базы данных: учеб. для высш. учеб. заведений / под ред. А.Д. Хомоненко. – 4-е изд., доп. и перераб. – СПб.: КОРОНА принт, 2004. – 736 с.
6. Марков А. С. Базы данных. Введение в теорию и методологию : учеб. для студ., обуч. по спец. «Прикладная математика и информатика», а также при подготовке бакалавров и магистров по этому напр. – М.: Финансы и статистика, 2004. – 512 с.
7. Кузин А.В. Базы данных: учеб. пособие для студ. высш. учеб. заведений, обучающихся по напр. подготовки дипломированных спец. 654600 «Информатика и вычислительная техника». – М.: Академия, 2005. – 315 с.
8. Полякова Л.Н. Основы SQL: курс лекций: учеб. пособие для студ. вузов, обуч. по спец. 351400 «Прикладная информатика». – М.: Интернет-Университет Информационных Технологий, 2004. – 368 с.

ПРИЛОЖЕНИЯ

Приложение 1

Варианты индивидуальных заданий на проектирование и разработку БД

Разработка включает следующие этапы:

- проектирование БД;
- анализ предметной области в соответствии с вариантом индивидуального задания;
- выявление сущностей предметной области (не менее 5) и их атрибутов (минимально необходимый перечень атрибутов приведен и может быть расширен)
- определение ключевых атрибутов сущностей;
- определение связей между сущностями;
- разработку инфологической модели;
- преобразование инфологической модели в реляционную модель;
- определение правил, действующих в предметной области, определение возможных пользователей и решаемых ими задач.
- Создание БД в среде MS SQ Server (SQL код создания таблиц с определением всех видов ограничений, создание диаграммы).
- Ввод данных в таблицы. Объем введенных данных должен обеспечивать получение результатов всех запросов, проверку работы триггеров и хранимых процедур.
- Создание представлений пользователей, триггеров (не менее 3), запросов (не менее 10), хранимых процедур, реализующих задачи пользователей.
- Создание приложения для каждого пользователя. Приложение должно поддерживать решение задач пользователей по работе с информационной системой: обеспечивать ввод информации в таблицы созданной базы данных с помощью соответствующих форм, поиск необходимой информации, запуск хранимых процедур.

Описание предметной области и примерный состав атрибутов каждого варианта:

Вариант 1 БД "Отель"

Описание предметной области:

В БД хранятся сведения об отелях, принадлежащих одной компании. Отели находятся в разных городах. Цены на номера одного типа во всех отелях одинаковы и зависят от типа номера и количества мест. Номер может быть забронирован, занят или свободен. При заезде в отель постояльцы проходят регистрацию. Информация о регистрации

постояльцев отеля (выехавших из отеля) хранится в течение года и 1 января удаляется в архив.

БД должна содержать следующий минимальный набор сведения:

- Адрес отеля.
- Название отеля.
- Номер комнаты.
- Тип комнаты.
- Количество мест.
- Цена комнаты за сутки проживания.
- Имя постояльца.
- Фамилия постояльца.
- Отчество постояльца.
- Адрес постоянного проживания.
- Дата заезда.
- Дата отъезда.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

В поле *Тип комнаты* должно помещаться одно из следующих значений “люкс”, “стандартный”. Значение в поле *Цена* должно находиться в диапазоне от 100 т.р. до 500 т.р.

Значение в поле *Номер комнаты* должно находиться в пределах от 10 до 100.

Значения, помещаемые в поля “Дата заезда” и “Дата отъезда” должны быть по умолчанию равны текущей дате.

2. Создайте запросы:

- Составить список всех 2-местных номеров отелей, с ценой менее 200 т.р., упорядочив данные в порядке уменьшения стоимости.
- Выбрать все записи регистрации постояльцев, которые выехали из отелей в течение двух последних недель.
- Чему равен общий суточный доход каждого отеля за последний месяц?
- Составить список свободных номеров одного из отелей на текущий день.
- Найти общие потери от незанятых номеров за текущий день.
- Определить в каком отеле имеется наибольшее количество незанятых номеров на текущие сутки.
- Создать таблицу со структурой аналогичной структуре таблицы регистрации для хранения архивных записей. Скопируйте в нее все записи, созданные до 1 января 2014 года. Удалите из основной таблицы регистрации все записи, занесенные в архив.

3. Создайте представления:

- Для турагентов (поиск свободных номеров в отелях).
- Для владельца компании (информация о доходах каждого отеля за прошедший месяц).

4. Создайте хранимые процедуры:

- для увеличения цены всех номеров на 5 %, если в отеле нет свободных номеров.
- для получения информации о свободных одноместных номерах отеля на завтрашний день. Если таких номеров нет, то выдать соответствующее сообщение.
- бронирования двухместного номера в гостинице на заданную дату и количество дней проживания.

5. Создайте необходимые триггеры.

Вариант 2

БД “Сессия”

Описание предметной области:

БД содержит сведения о сдаче сессии студентами. Номер зачетной книжки однозначно идентифицирует студента. БД должна содержать следующий минимальный набор сведений:

- Номер зачетной книжки.
- Фамилия студента.
- Имя студента.
- Отчество студента.
- Курс.
- Группа.
- Код дисциплины.
- Название дисциплины.
- Оценка.
- Фамилия преподавателя.
- Имя преподавателя.
- Отчество преподавателя.
- Кафедра.
- Дата сдачи экзамена.
- Аудитория.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

Значение в поле Курс должно находиться в диапазоне от 1 до 5.

Значение в поле Оценка должно находиться в пределах от 2 до 10.

2. Создайте запросы:

- Составить список дисциплин, которые должны быть сданы каждой

группой с указанием дат сдачи и фамилий преподавателей.

- Вывести список студентов, получивших более двух двоек.
 - Вывести список студентов, получивших двойки с указанием фамилии преподавателя, которым они должны пересдать экзамен.
 - Вывести фамилии студентов, получивших оценки по дисциплине, которые выше среднего балла по этой дисциплине
 - Создать рейтинговый список групп по результатам сдачи сессии, упорядочить его по убыванию.
 - Создайте списки студентов, упорядоченные по группам и фамилиям студентов, содержащие данные о средних баллах и размерах стипендии. Формулу для вычисления стипендии (стипендия зависит от среднего балла) задайте самостоятельно.
 - Вывести список студентов, сдавших все положенные экзамены.
 - Вывести список студентов, получивших максимальный средний балл в своей группе.
 - Составить список на отчисление (отчисляются студенты, сдавшие три и более экзаменов на оценку ниже 4-х баллов).
 - Составьте запрос для назначения повышенной стипендии студентам, сдавшим все экзамены на оценку не ниже 9 баллов.
 - Рассчитать количество оценок “8”, “9”, “10”, полученных студентами.
3. Создайте представление для учебного отдела, содержащее данные о результатах сдачи сессии.
4. Создайте хранимые процедуры:
- Для повышения стипендии отличникам на 10%.
 - Для перевода студентов на следующий курс.
 - Для отчисления студента, получившего более одной двойки в сессию.
 - Для изменения оценки при успешной пересдаче экзамена.
5. Создать пользовательскую функцию для вычисления среднего балла заданного студента.
6. Создайте триггер для занесения данных о студенте в таблицу ”К отчислению” при получении им третьей оценки ниже 4-х баллов.

Вариант 3

БД “Библиотека”

Описание предметной области:

Каждая книга может храниться в нескольких экземплярах.

Для каждого экземпляра известно место его хранения (комната, стеллаж, полка).

Читателю не может быть выдано более 3-х книг одновременно.

Книги выдаются читателям на срок не более 10 дней.

БД должна содержать следующий минимальный набор сведений:

- Автор (фамилия и имя (инициалы) или псевдоним автора издания).

- Название (заглавие) издания.
- Номер тома (части, книги, выпуска).
- Составитель (фамилия и имена (инициалы) каждого из составителей издания).
- Язык, с которого выполнен перевод издания.
- Вид издания (сборник, справочник, монография ...).
- Область знания.
- Переводчик (фамилия и инициалы переводчика).
- Место издания (город).
- Издательство (название издательства).
- Год выпуска издания.
- Библиотечный шифр (например, ББК 32.973).
- Количество книг.
- Номер (инвентарный номер) экземпляра.
- Номер комнаты (помещения для хранения экземпляров).
- Номер стеллажа в комнате.
- Номер полки на стеллаже.
- Цена конкретного экземпляра.
- Дата изъятия экземпляра с установленного места.
- Номер читательского билета (формуляра).
- Фамилия читателя.
- Имя читателя.
- Отчество читателя.
- Адрес читателя.
- Телефон читателя.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

- В библиотеке хранятся книги, выпущенные не позднее 1970 года.
- В библиотеке имеется 10 комнат для хранения книг, в каждой комнате 30 стеллажей, каждый стеллаж состоит из 50 полок.
- Дата выдачи экземпляра по умолчанию равна текущей дате.
- Возраст читателей должен быть не меньше 16 лет.

2. Создайте запросы:

- Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.
- Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
- Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
- Вывести список книг, которые находятся в библиотеке в

единственном экземпляре.

- Подсчитать количество читателей, которые не обращались в библиотеку в течение года.
- Исключить из библиотеки читателей, которые не обращались в библиотеку в течение года и которые сдали все книги.
- Вывести список книг по программированию на C#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

3. Создать представления для администрации библиотеки, содержащие сведения о должниках.

сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

4. Создать хранимые процедуры:

- Для проверки наличия экземпляров заданной книги в библиотеке (процедура должна возвращать количество экземпляров книги).
- Для ввода в базу данных новой книги.
- Для ввода нового читателя (необходимо проверить наличие читателя в картотеке, чтобы не назначить ему номер вторично).

5. Создать необходимые триггеры.

Вариант 4.

БД "Учет выполнения заданий"

Описание предметной области:

Сотрудники организации выполняют проекты. Проекты состоят из нескольких заданий. Каждый сотрудник может участвовать в одном или нескольких проектах, или временно не участвовать ни в каких проектах.

Над каждым проектом может работать несколько сотрудников отделов, или временно проект может быть приостановлен, тогда над ним не работает ни один сотрудник. Над каждым заданием в проекте работает ровно один сотрудник. Каждый сотрудник числится в одном отделе.

БД должна содержать следующий минимальный набор сведений:

- Номер сотрудника.
- Фамилия сотрудника.
- Имя сотрудника.
- Отчество сотрудника.
- Должность сотрудника.
- Оклад сотрудника.
- Название организации.
- Номер организации.
- Адрес организации.
- Номер телефона отдела.
- Номер отдела.

- Название отдела.
- Код проекта.
- Название проекта.
- Номер задания.
- Дата начала выполнения задания.
- Срок выполнения задания.
- Отметка о выполнении задания.
- Дата контроля выполнения задания.
- Причина невыполнения задания.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

Оклад сотрудника должен находиться в пределах от 200\$ до 500\$.

Срок выполнения задания не должен превышать 30 дней.

Дата начала выполнения задания и дата контроля выполнения задания по умолчанию равны текущей дате.

Поле причина невыполнения задания может содержать 2 значения, имеющих следующий смысл: “уважительная”, “неуважительная”.

2. Создайте запросы:

- Составить список всех заданий каждого проекта с указанием организаций, отделов и исполнителей, занятых в его выполнении.
- Составить список проектов, работа над которыми была начата больше месяца назад.
- Вывести список сотрудников, оклад которых превышает средний оклад сотрудников своего отдела.
- Найти отдел, работающий над максимальным количеством проектов.
- Составить список сотрудников, проектов, заданий, в выполнении которых они участвуют и дат предполагаемого выполнения ими заданий.
- Составить список сотрудников, не выполнивших задания в срок с указанием проектов и заданий, которые они должны были выполнить и количества дней просрочки выполнения заданий.
- Составить список проектов, в выполнении которого участвует более трех человек.
- Составить список проектов, срок выполнения которых истекает сегодня, и которые включают больше трех невыполненных заданий.
- Повысить оклад сотрудников, участвующих в выполнении более чем одного проекта, у которых оклад не превышает средний оклад сотрудников его должности.

3. Создать представление для руководителей проектов, содержащее сведения об исполнителях, отделах, сроках выполнения заданий, включенных в проект.

4. Создать хранимые процедуры:

- Для повышения оклада сотрудников, выполнивших задания с трехдневным опережением графика на заданный процент.
 - Для печати предупреждения сотруднику, не сдавшему задание в срок по неуважительной причине.
 - Для поиска номера телефона сотрудника (телефон установлен в каждом отделе).
5. Создать пользовательскую функцию для вычисления количества заданий, в выполнении которых участвует сотрудник.
1. Создать триггер для запрета удаления данных о сотруднике в случае, если он не завершил выполнение всех своих заданий

Вариант 5

БД "Издательство компьютерной литературы"

Описание предметной области:

Издательство занимается выпуском литературы по различным областям информатики. Покупатели книг приобретают книги на базе издательства. Когда на базе заканчиваются книги, издается дополнительный тираж.

БД должна содержать следующий минимальный набор сведений:

- Фамилия автора.
- Имя автора.
- Отчество автора.
- Код автора.
- E-mail автора.
- Код ISBN.
- Название книги.
- Код категории книги.
- Категория книги.
- Количество страниц.
- Год начала издания.
- Розничная цена книги.
- Тираж.
- Количество экземпляров на базе издательства.
- Код заказчика.
- Фамилия заказчика.
- Имя заказчика.
- Отчество заказчика.
- Адрес заказчика.
- Телефон заказчика.
- Код заказа.
- Количество экземпляров книги в заказе.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

Количество страниц книги находится в интервале от 50 до 2000.

Год начала издания по умолчанию равен текущему году.

Розничная цена книги находится в диапазоне от 500 до 40000 рублей.

Тираж не превышает 10000 штук.

2. Создать запросы:

- Список книг, изданных в текущем году и относящихся к категории "Базы данных".
- Список покупателей, заказавших книг на сумму, превышающую среднюю сумму заказа.
- Список книг, которые не заказывались в течение последних двух кварталов.
- Список авторов, не написавших ни одной книги, относящейся к категории "Базы данных".
- Список книг, в названиях которых содержится слово "проектирование" и которые присутствуют на базе в количестве, превышающем 50 экземпляров.
- Покупателя, сделавшего заказ на максимальную сумму.
- Список книг, не попавших ни в один из заказов.

3. Создать представление, содержащее сведения о количестве заказанных экземпляров каждой книги, изданной в текущем году.

4. Создать хранимые процедуры:

- Для снижения цен на книги, которые находятся на базе в количестве, превышающем 1000 штук.
- Для ввода новой книги.
- Для оформления заказа.
- Для поиска книг заданного автора.

1. Создать триггеры для увеличения на 1 % стоимость книги, если число проданных экземпляров превышает 5 штук.

Вариант 6

БД "Пассажир"

Описание предметной области:

Информационная система служит для продажи железнодорожных билетов. Билеты могут продаваться на текущие сутки или предварительно (не более чем за 45 суток). Цена билета при предварительной продаже снижается на 5 %.

БД должна содержать следующий минимальный набор сведений:

- Номер поезда.

- Название поезда.
- Тип поезда.
- Пункт назначения.
- Пункт назначения для проданного билета.
- Тип вагона.
- Количество мест в вагоне.
- Цена билета.
- Дата отправления.
- Время отправления.
- Номер вагона.
- Номер билета.
- Место.
- Фамилия пассажира.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

Дата отправления по умолчанию равна текущей дате.

Количество вагонов в поезде не может быть меньше 3 и больше 30.

Задайте ограничения на типы вагонов и типы поездов.

2. Создать запросы:

- Свободные места на все поезда, отправляющиеся с вокзала в течение следующих суток.
- Список пассажиров, отправившихся из Витебска в Москву всеми рейсами за прошедшие сутки.
- Номера поездов, на которые проданы все билеты на следующие сутки.
- Свободные места в купейные вагоны всех рейсов до Москвы на текущие сутки.
- Выручка от продажи билетов на все поезда за прошедшие сутки.
- Общее количество билетов, проданных по всем направлениям в вагоны типа "СВ".
- Количество непроданных билетов на все поезда, формирующиеся в Витебске, за прошедшие сутки (номер поезда, тип вагона, количество).
- Номера и названия поездов, формирующихся в Витебске все вагоны которых были заполнены менее чем наполовину за прошедшие сутки.

3. Создать представление для пассажиров о наличии свободных мест.

4. Создать хранимые процедуры:

- Для повышения цен в пригородные поезда на 20%.
- Для поиска и продажи билета по запросу пассажира (направление, дата, количество билетов, время отправления).

Вариант 7

БД "Курсы"

Описание предметной области:

Подразделение занимается организацией внебюджетного образования. Имеется несколько типов краткосрочных курсов, предназначенных для определенных специальностей, связанных с программным обеспечением ИТ. Каждый тип курсов имеет определенную длительность и свой перечень изучаемых дисциплин. На каждую специальность может быть набрано несколько групп. По каждой дисциплине могут проводиться лекционные и лабораторные занятия. Подразделение обеспечивает следующие ресурсы: учебные классы, лекционные аудитории и преподавателей. Необходимо составить расписание занятий.

БД должна содержать следующий минимальный набор сведений:

- Фамилия слушателя.
- Имя слушателя.
- Специальность.
- Номер группы.
- Количество человек в группе.
- Название дисциплины.
- Количество часов.
- День недели.
- Номер пары.
- Номер аудитории.
- Вид занятий (лекционные или практические).
- Фамилия преподавателя.
- Должность преподавателя.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

- Количество слушателей в группе не может быть больше 15.
- Количество часов, отводимых на изучение дисциплины, находится в диапазоне от 10 до 100.

2. Создать запросы:

- Вывести все номера группы и специальности, где количество слушателей меньше 10.
- Вывести перечень изучаемых дисциплин по тем специальностям, где количество слушателей меньше 10.
- Вывести список преподавателей, которые не проводят занятия на третьей паре ни в один из дней недели.
- вывести список свободных лекционных аудиторий на понедельник.
- вычислить общее количество учебных часов по каждой специальности.

3. Создать представление для потенциальных слушателей, содержащее перечень специальностей, изучаемых на них дисциплин и количество часов.

4. Создать хранимые процедуры:

- Для получения расписания занятий для группы на определенный день недели.
- Записи на курс слушателя.
- Получения перечня свободных лекционных аудиторий (с указанием времени) на любой день недели. Если свободных аудиторий не имеется, то выдать соответствующее сообщение.

5. Создать триггер для удаления данных о преподавателе и передаче его занятий другому преподавателю.

Вариант 8.

БД Аэропорт

Описание предметной области:

Необходимо обеспечить продажу билетов на нужный рейс, при отсутствии билетов (необходимого количества билетов) предложить билет на ближайший рейс.

БД должна содержать следующий минимальный набор сведений:

- Бортовой номер самолета.
- Тип самолета.
- Количество мест.
- Грузоподъемность.
- Скорость.
- Дата выпуска.
- Налёт в часах.
- Дата последнего ремонта.
- Назначение самолета.
- Расход топлива.
- Код экипажа.
- Паспортные данные членов экипажа.
- Номер рейса,
- Дата вылета,
- Время вылета.
- Аэропорт вылета,
- Аэропорт назначения.
- Расстояние.
- ФИО пассажира.
- Паспортные данные.
- Номер места.
- Тип места.
- Цена билета.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

а) Самолеты, выпущенные ранее 10 лет назад должны проходить профилактический ремонт ежегодно.

б) Билеты продаются не более чем за 30 дней до вылета и не менее чем за 1 час до вылета.

2. Создать запросы:

- Определить расчетное время полета по всем маршрутам.
- Определить расход топлива по всем маршрутам.
- Вывести данные экипажа, совершившего максимальное количество полетов за прошедшую неделю.
- Вывести данные о том, сколько свободных мест оставалось в самолетах, совершавших полет по одному из рейсов за вчерашний день.
- Рассчитать убытки компании за счет непроданных билетов за вчерашний день.
- Вывести список самолетов, которые не ремонтировались в течение более чем 3 лет.
- Определить, каким количеством самолетов каждого типа владеет компания.
- Определить, какой тип самолетов чаще всего летал в заданный аэропорт назначения
- Вывести список самолетов “возраст” которых превышает средний “возраст” самолетов этого типа,
- Определить тип самолетов, летающих во все аэропорты назначения

3. Создать представление для пассажиров авиакомпании.

4. Создать хранимые процедуры:

Для поиска и продажи билетов на нужный рейс.

5. Создать необходимые триггеры.

Вариант 9

БД ” Оптовая база ”

Описание предметной области:

Оптовая база закупает товары у компаний-поставщиков и поставляет их компаниям – покупателя. Доход оптовой базы составляет 5 % от стоимости товара проданного компании – покупателю. Один и тот же товар может доставляться несколькими поставщиками и один и тот же поставщик может доставлять несколько видов товаров. Цены товара у разных поставщиков могут отличаться.

БД должна содержать следующий минимальный набор сведений:

- Код сотрудника.

- Паспортные данные сотрудника.
- Код товара.
- Название товара.
- Единица измерения товара.
- Количество товара.
- Минимальный запас товара.
- Стоимость единицы товара.
- Примечание – описание товара.
- Код поставщика.
- Название компании поставщика.
- Адрес поставщик.
- Дата поставки.
- Количество товара в партии.
- Номер счета.
- Код организации – покупателя.
- Название компании покупателя.
- Адрес покупателя.
- Дата вывоза.
- Количество товара в партии.
- Продажная цена товара.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных:
2. Создайте запросы:
 - Вывести список поставщиков, которые поставляют все товары.
 - Определить поставщика, который поставляет каждый из товаров по самой низкой цене.
 - Вывести названия товаров, цены на которые у всех поставщиков одинаковы.
 - Чему равен общий суточный доход оптового склада за прошедший день?
 - Вычислить стоимость каждого вида товара, находящегося на базе.
 - В какой день было вывезено минимальное количество товара?
 - Сколько различных видов товара имеется на базе?
 - Вывести список товаров, у которых в коде присутствует символ “_”
 - Создать таблицу со структурой аналогичной структуре таблицы регистрации для хранения архивных записей. Скопируйте в нее все записи, созданные до 1 января 2011года. Удалите из основной таблицы регистрации все записи, занесённые в архив.
3. Создайте представления:
 - Для компаний - покупателей (поиск нужного товара).

4. Создайте хранимую процедуру:

- для снижения цены на заданный процент для товаров, у которых срок пребывания на складе превысил заданный норматив.

5. Создайте пользовательскую функцию для расчёта стоимости закупки заданного вида товара за прошедшие сутки.

5. Создайте триггер удаления, запуск которого происходит при отпуске товара покупателю. Предусмотреть вывод сообщения при недостаточном количестве товара на базе.

Вариант 10

БД "Автовокзал"

Описание предметной области:

По одному и тому же маршруту отправляется несколько рейсов ежедневно. Номер рейса определяется маршрутом и временем отправления. Билеты могут продаваться предварительно, но не ранее чем за 10 суток. Места в билете не указываются. На каждый рейс может продаваться не более 10 билетов без места, цена на которые снижается на 10%.

БД должна содержать следующий минимальный набор сведений:

- Номер рейса.
- Номер водителя.
- Номер автобуса.
- Паспортные данные водителя.
- Пункт отправления.
- Пункт назначения.
- Дата отправления.
- Время отправления.
- Время в пути.
- Тип автобуса.
- Количество мест в автобусе.
- Номер билета.
- Цена билета.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно)

2. Создать запросы:

- Вывести количество автобусов каждого типа, отправляющихся с автовокзала.
- Вывести фамилии водителей и номера автобусов, отправившиеся в рейсы до 12 часов текущего дня.
- Рассчитать выручку от продажи билетов за прошедший день.
- Вывести список водителей, которые не выполнили ни одного рейса за прошедший день.

- Вывести сумму убытков из-за непроданных мест в автобусе за прошедшую неделю.
 - Сколько рейсов выполнил каждый водитель.
 - Вывести тип автобуса, который используется на всех рейсах.
 - Вывести данные водителя, который провел максимальное время в пути за прошедшую неделю.
3. Создать представление для пассажиров (количество свободных мест на все рейсы).
4. Создать хранимые процедуры:
- Продажи билета.
 - Возврата билета.
 - Добавления нового рейса.
5. Создать триггер для занесения стоимости каждого проданного билета во временную таблицу «выручка за текущий день».

Вариант 11

БД Автомастерская

Описание предметной области:

Автомастерская осуществляет ремонт автомашин, используя для этих целей штат мастеров и свои мастерские. Стоимость ремонта включает цену деталей и стоимость работы. Заработная плата мастеров составляет 50 % стоимости работы.

БД должна содержать следующий минимальный набор сведений:

- Табельный номер мастера.
- ФИО мастера.
- Разряд мастера.
- Адрес.
- Дата заказа.
- Гос. Номер автомобиля.
- Марка.
- Мощность автомобиля.
- Год выпуска.
- Цвет автомобиля.
- Дата принятия в ремонт.
- Плановая дата окончания ремонта.
- Фактическая дата окончания ремонта.
- Вид ремонта.
- Стоимость ремонта.
- Название детали.
- Цена детали.
- Марка автомобиля.
- ФИО владельца.
- Номер телефона владельца.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно)
2. Создать запросы:
 - Выбрать фамилию того механика, который чаще всех работает с автомобилями марки "Тойота".
 - Определить тех владельцев автомобилей, которых всегда обслуживает один и тот же механик. Вывести фамилии механика и его постоянного клиента.
 - Вывести фамилии механиков, которые не выполняли работы в срок и количество дней просрочки выполнения заказа.
 - Вывести данные владельца самого старого автомобиля.
 - Сколько автомобилей отремонтировал каждый механик.
 - Вывести данные механика, который выполнял все виды ремонта за прошедшую неделю.
 - Сколько заработал каждый водитель за прошедший месяц?
 - Вывести данные владельцев автомобилей, которые обращались в ремонт больше одного раза.
 - За каждый день просрочки выполнения заказа механику назначается штраф в размере 5%. Рассчитать штраф каждого механика за прошедший месяц.
3. Создать представление для заказчиков (фамилию механика и модель автомобиля, которую он ремонтирует чаще всего).
4. Создать хранимые процедуры:
 - Повышения цены деталей для автомобиля "Ford" на 10 %.
 - Создайте процедуру для повышения разряда тех мастеров, которые отремонтировали больше 3 автомобилей.
5. Создать триггер для занесения стоимости каждого выполненного заказа во временную таблицу «выручка водителя за текущий день».

Вариант 12.

БД Прокат автомобилей

Описание предметной области:

Компания предоставляет прокат автомобилей. В пункт проката обращаются клиенты, данные о которых регистрируют в базе. Цена проката зависит от марки автомобиля, технических характеристик и года выпуска. За каждый час просрочки возврата автомобиля начисляется штраф. При каждом обращении фиксируется дата выдачи автомобиля и дата возврата, номер. Если клиент не вернул автомобиль в срок и не оформил продление, ему назначается штраф и автомобиль больше не выдается. Постоянным клиентам предоставляются скидки.

БД должна содержать следующий минимальный набор сведений:

- ФИО.
- Паспортные данные.
- Код должности.
- Наименование должности.
- Оклад.
- Обязанности.
- Код марки.
- Наименование,
- Технические характеристики,
- Описание.
- Код автомобиля.
- Регистрационный номер.
- Номер кузова.
- Номер двигателя.
- Год выпуска.
- Пробег,
- Цена автомобиля.
- Цена проката.
- Дата последнего ТО.
- Код сотрудника-механика.
- Специальные отметки.
- Отметка о возврате.
- Код клиента.
- ФИО.
- Адрес.
- Телефон.
- Паспортные данные.
- Дата и время выдачи автомобиля.
- На сколько часов.
- Дата и время возврата автомобиля.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно)
2. Создать запросы:
 - Какой автомобиль находился в прокате максимальное количество часов?
 - Какой автомобиль ни разу не был в прокате?
 - Автомобили какой марки чаще всего брались в прокат?
 - Определить убытки от простоя автомобилей за вчерашний день.
 - Вывести данные автомобиля, имеющего максимальный пробег.
 - Вывести данные клиента обратившегося в прокат больше двух раз.

- Сколько часов находился в прокате каждый автомобиль.
- Какой автомобиль находился в прокате дольше всех
- Вывести данные клиентов, не вернувших автомобиль вовремя.
- Определить, каким количеством каждого типа владеет компания.

Определить средний “возраст” автомобилей компании.

3. Создать представление для клиентов компании.

4. Создать хранимые процедуры:

- Выполнить списание автомобилей, выпущенных ранее заданного года.
- Выдачи автомобиля и расчёта стоимости с учетом скидки постоянным клиентам.

5. Создать пользовательскую функцию для вычисления количества автомобилей заданной как параметр марки.

6. Создать триггер, который помещает в поле *специальные отметки* признак, который характеризует постоянного клиента (если клиент обращается в прокат третий раз).

Вариант 13

БД “Ресторан”

Описание предметной области:

Сотрудники ресторана – повара и официанты. За каждым официантом закреплены определенные столы. Каждый повар готовит определенный набор блюд. Запас продуктов на складе не должен быть ниже заданного значения. Цена заказа складывается из стоимости ингредиентов и наценки, которая составляет 40 % стоимости ингредиентов.

БД должна содержать следующий минимальный набор сведений:

- ФИО сотрудника.
- Паспортные данные сотрудника.
- Категория сотрудника.
- Должность сотрудника.
- Оклад сотрудника.
- Наименование ингредиента
- Код ингредиента.
- Дата закупки.
- Объём закупки.
- Количество продукта на складе.
- Необходимый запас продукта.
- Срок годности.
- Цена ингредиента.
- Поставщик.
- Наименование блюда.
- Код блюда.

- Объём ингредиента.
- Номер стола.
- Дата заказа.
- Код заказа.
- Количество.
- Название блюда.
- Ингредиенты, входящие в блюдо

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно)
2. Создать запросы:
 - Вывести данные официанта, принявшего максимальное число заказов.
 - Вывести данные официанта, принявшего заказы на максимальную сумму.
 - Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа).
 - Подсчитать, сколько ингредиентов содержит каждое блюдо.
 - Вывести название блюда, содержащее максимальное число ингредиентов.
 - Какой повар может приготовить максимальное число блюд?
 - Какой из ингредиентов используется во всех блюдах?
 - Какой из ингредиентов используется в максимальном количестве блюд.
3. Создать представление для посетителей ресторана, содержащее сведения обо всех блюдах и их ценах.
4. Создать хранимые процедуры:
 - Вывести сведения о заказах заданного официанта на заданную дату.
 - Формирования заказа. При этом необходимо выполнить расчет стоимости заказа по заданному названию блюда и количеству единиц в заказе, изменить значение количества продукта на складе, и если количество продукта на складе достигло критического значения – вывести сообщение и занести название продукта и дату во временную таблицу.
 - Повышения оклада заданного сотрудника на 30 % при повышении его категории.
5. Создать пользовательскую функцию для расчета стоимости ингредиентов для заданного блюда
6. Создать триггер для удаления данных из связанных таблиц при удалении блюда из БД.

Вариант 14

БД Таксопарк

Описание предметной области:

Система должна фиксировать все вызовы такси. Каждому водителю ежедневно начисляется заработная плата в зависимости от количества вызовов и их тарифа (50% от заработанной им суммы).

- Код сотрудника.
- ФИО сотрудника.
- Адрес сотрудника.
- № телефона сотрудника.
- Паспортные данные сотрудника.
- Должность сотрудника.
- Категория сотрудника.
- Наименование марки автомобиля.
- Технические характеристики.
- Стоимость.
- Код тарифа.
- Наименование тарифа.
- Цена за километр.
- Код автомобиля.
- Код марки.
- Год выпуска.
- Пробег.
- Дата последнего ТО.
- Дата вызова.
- Время посадки пассажира.
- Время высадки пассажира.
- Номер телефона пассажира.
- Откуда.
- Куда.
- Расстояние

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно)

2. Создать запросы:

- Вывести данные о водителе, который чаще всех доставляет пассажиров на улицу Чкалова.
- Вывести данные об автомобилях, которые имеют пробег более 250 тысяч километров и которые не проходили ТО в текущем году.
- Сколько раз каждый пассажир воспользовался услугами таксопарка?
- Вывести данные пассажира, который воспользовался услугами

- таксопарка максимальное число раз.
- Вывести данные о водителе, который ездит на самом дорогом автомобиле.
 - Вывести данные пассажира, который всегда ездит с одним и тем же водителем.
 - Какие автомобили имеют пробег больше среднего пробега для своей марки.
3. Создать представление, содержащее сведения о незанятых на данный момент водителях
4. Создать хранимые процедуры:
- Вывести данные о зарплате заданного водителя за прошедшие сутки.
 - Для вывода данных о пассажирах, которые заказывали такси в заданном, как параметр, временном интервале.
 - Вывести сведения о том, куда был доставлен пассажир по заданному, как параметр, номеру телефона пассажира.
 - Для вычисления суммарного дохода таксопарка за прошедший месяц.
5. Создать пользовательскую функцию для вычисления заработка водителя за прошедшие сутки. Аргумент функции – код сотрудника.
6. Создать триггер для фиксации в БД заработанной водителем суммы и начисления ему заработной платы (после высадки пассажира).

Вариант 15

БД "Распределение аудиторного фонда"

Описание предметной области:

БД содержит сведения об аудиториях и расписании проводимых в них занятий. Время начала и окончания занятия по дням недели фиксировано.

База данных используется для получения справок о наличии свободных аудиторий в указанное время, о месте и времени проведения определенных занятий.

БД должна содержать следующий минимальный набор сведений:

- Номер аудитории.
- Количество мест,
- Тип аудитории.
- Код дисциплины.
- Название дисциплины.
- Вид занятия.
- ФИО преподавателя.
- Номер студенческой группы.
- Максимально возможное количество студентов, посещающих занятие.
- Дата.

- День недели.
- Время начала занятия.

Задания

1. Создать таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно)
2. Создать запросы:
 - Вывести список преподавателей, не имеющих занятий в понедельник
 - Найти недельную нагрузку студентов каждой группы
 - Вывести список свободных лекционных аудиторий в заданное время.
 - Вывести количество аудиторий каждого типа.
 - Вывести еженедельное количество часов занятий для каждой группы.
 - Найти номера аудиторий каждого типа, имеющих максимальное количество мест.
 - Вывести фамилии преподавателей, которые всегда проводят практические занятия в одной и той же аудитории.
3. Создать представление, содержащее данные о расписании на каждый день
4. Создать хранимые процедуры:
 - Вывести список свободных аудиторий для проведения практических занятий заданной группы в заданное время.
 - Вывести расписание занятий для заданного преподавателя.
 - Вывести список аудиторий, в которых может разместиться заданная, как параметр группа.
5. Создать необходимые триггеры.

Вариант 16

БД "Спортивный клуб".

Описание предметной области:

БД должна осуществлять: ведение списков спортсменов и тренеров. Тренеры разделены по категориям. При достижении спортсменами определенного рейтинга категория тренера повышается; учёт проводимых соревнований (с ведением их архива); учёт травм, полученных спортсменами.

Предусмотреть: возможность перехода спортсмена от одного тренера к другому; составление рейтингов спортсменов; составление рейтингов тренеров; выдачу информации по соревнованиям; выдачу информации по конкретному спортсмену; подбор возможных кандидатур на участие в соревнованиях (соответствующего уровня мастерства, возраста и без травм)

БД должна содержать следующий минимальный набор сведений:

- ФИО тренера.
- № телефона тренера.

- Паспортные данные тренера.
- Категория тренера.
- Оклад тренера.
- Вид соревнования.
- Категория соревнования.
- Место проведения соревнования.
- Дата проведения соревнования.
- Фамилия спортсмена.
- Имя спортсмена.
- Результат спортсмена.
- Отчество спортсмена.
- Место, которое занял спортсмен.
- Количество баллов спортсмены за место.
- Количество баллов тренера за место.
- Дата рождения спортсмена.
- Категория спортсмена.
- Рейтинг спортсмена.
- Вид травмы.

Задания

1. Создать таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно).
2. Создать запросы:
 - с каким количеством спортсменов работает каждый тренер,
 - найти тренеров, чьи спортсмены не имеют травм.
 - Найти тренера, получающего минимальную зарплату.
 - Определить количество соревнований каждой категории.
 - Найти тренера, работающего с самыми молодыми спортсменами (средний возраст спортсменов минимален),
 - сколько спортсменов участвует в соревнованиях каждой категории.
3. Создать представление, содержащее сведения обо всех тренерах, соревнованиях, в которых участвовали их спортсмены и местах которые они заняли.
4. Создать хранимые процедуры:
 - Для вывода данных о результатах заданного спортсмена за прошедший год.
 - Для ввода данных о соревнованиях, проводимых в первом квартале текущего года.
5. Создать триггер для повышения рейтинга спортсмена, рейтинга и оклада тренера после участия в соревновании

Вариант 17

БД "Телефонная станция"

Описание предметной области:

Описание предметной области:

Информационная система служит для хранения информации об абонентах телефонной станции и для учета оплаты всех видов услуг абонентами. В системе должны храниться сведения о продолжительности разговоров каждого абонента, о стоимости внутренних и междугородных переговоров, о задолженности абонента. Цена минуты в ночное время снижается на 20%.

БД должна содержать следующий минимальный набор сведений:

- ФИО абонента
- Номер телефона.
- Адрес абонента.
- Город.
- Зона (город, республика, СНГ, далее зарубежье)
- Продолжительность.
- Дата звонка.
- Время звонка.
- Код зоны.
- Цена минуты.
- Сумма оплаты.
- Дата оплаты.

Задания

1. Создать таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно).
2. Создать запросы:
 - Вывести суммарное время переговоров каждого абонента.
 - Найти среднюю продолжительность разговора абонента АТС.
 - Вывести количество междугородных переговоров каждого абонента.
 - Вывести список абонентов, не внесших оплату за прошедший месяц.
 - Сколько звонков было сделано в каждый из следующих городов: в Москву, Лондон, Париж.
 - Вывести список абонентов, звонивших только в ночное время.
 - Вывести список абонентов, время разговоров которых превышает среднее для этой же зоны.
3. Создать представление, содержащее сведения обо всех абонентах и их переговорах за прошедший месяц
4. Создать хранимые процедуры:
 - вывести список всех звонков заданного абонента.
 - вывести задолженность по оплате для заданного абонента.
 - рассчитать сумму, которую должен внести каждый абонент
5. Создать необходимые триггеры.

Вариант 18

БД "ГАИ"

Описание предметной области:

Гаи производит регистрацию автомобилей и следит за безопасностью движения. БД служит для ведения статистики нарушений правил движения. БД должна содержать следующий минимальный набор сведений: номер водительского удостоверения.

БД должна содержать следующий минимальный набор сведений:

- ФИО водителя.
- Адрес.
- Номер телефона.
- Номер автомобиля.
- Марка автомобиля.
- Модель автомобиля.
- Год выпуска.
- Дата регистрации в ГАИ.
- Код нарушения.
- Вид нарушения.
- Сумма штрафа.
- Срок лишения прав управления автомобилем.
- Дата нарушения.
- Время нарушения.
- Район нарушения.
- Личный номер инспектора.
- ФИО инспектора.

Задания

1. Создать таблицы, используя необходимые средства поддержки целостности данных. (Ограничения задать самостоятельно).

2. Создать запросы:

- Вывести данные водителей многократно (более одного раза) нарушивших правила движения.
- В каком районе чаще нарушают правила движения.
- Вывести данные водителей, который были лишены прав управления автомобилем в текущем месяце.
- Вывести данные водителей, которые нарушили правила движения в ночное время.
- Вывести данные инспектора, оштрафовавшего максимальное число водителей.
- Вывести данные водителей, заплативших штраф одному и тому же инспектору более одного раза.
- Водители автомобилей какой марки реже всего подвергаются штрафу.

- Сколько водителей было лишено прав за прошедшую неделю.
 - За какое нарушение чаще всего штрафуются водители
3. Создать представление, содержащее следующие данные: вид нарушения, время нарушения, номер водительского удостоверения, сумма штрафа.
4. Создать хранимые процедуры:
- Вывести все сведения о владельце автомобиля по заданному, как параметр номеру автомобиля.
 - Вывести данные инспектора, оштрафовавшего одного и того же водителя более одного раза.
 - Вывести количество нарушений, повлекших лишение прав в заданном, как параметр районе.
5. Создать триггер удаления автомобиля из базы.

Вариант 19.

БД Банк

Описание предметной области:

Клиенты банка имеют вклады различных видов. Сотрудники банка заключают договоры с клиентами. Ежемесячно начисляется процент по вкладу, и полученная сумма добавляется к сумме вклада заказчика. Вкладчик имеет право снимать проценты по вкладу или всю сумму вклада с процентами по истечении срока вклада. При снятии денег до истечения срока вклада процент за текущий месяц не начисляется.

БД должна содержать следующий минимальный набор сведений:

- ФИО сотрудника.
- Возраст сотрудника.
- Адрес сотрудника.
- № телефона сотрудник.
- Паспортные данные сотрудника.
- Должность сотрудника.
- Оклад сотрудника (зависит от категории).
- Наименование вклада.
- Минимальный срок вклада.
- Минимальная сумма вклада.
- Процент по вкладу.
- Срок вклада.
- Код валюты.
- Процентная ставка.
- Код валюты.
- Наименование валюты.
- ФИО вкладчика.
- Адрес вкладчика.
- Телефон вкладчика.

- Паспортные данные.
- Номер договора.
- Дата вклада.
- Дата возврата.
- Сумма вклада.
- Сумма возврата.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных для реализации следующих требований:

Сумма вклада должна быть не меньше минимальной суммы по данному вкладу.

Срок вклада по умолчанию равен 90 дням. Дата вклада по умолчанию равна текущей дате.

2. Создайте запросы:

- Найти вкладчика, имеющего несколько вкладов.
- Найти вкладчика, имеющего вклады во всех видах валюты.
- Вывести данные вкладчика, имеющего максимальный вклад в белорусских рублях.
- Какой из вкладов пользуется наибольшей популярностью.
- Кто из сотрудников заключил максимальное число договоров.
- Вывести список вкладчиков, у которых срок вклада истекает завтра и суммы начислений, которые могут быть ими востребованы.
- Вывести список сотрудников, заключивших договоры на максимальную сумму за последний месяц.

3. Создать представление, содержащее сведения обо всех сотрудниках банка и заключенных ими договорах за прошедший месяц.

4. Создать хранимые процедуры:

- о текущей сумме вклада и сумме начисленного за месяц процента для заданного клиента.

5. Создать триггер для запрета выдачи денег клиенту, если остаток на счете оказывается меньше минимальной суммы вклада.

Вариант 20

БД Автозаправки

Фирмы – поставщики автомобильного топлива " имеют сеть заправочных автозаправок. На автозаправках реализуется автомобильное топливо всех видов.

Топливо продается за безналичный расчет с помощью специальных пластиковых карт. База данных предназначена для анализа продаж автомобильного топлива клиентам по видам топлива в сети заправок конкретной фирмы-производителя, спроса на автомобильное топливо и т.д. Каждая фирма имеет несколько автозаправок. Каждый вид топлива предоставляется несколькими фирмами;

БД должна содержать следующий минимальный набор сведений:

- Карт-счёт клиента.
- Сумма на счете клиента.
- Ф.И.О. клиента.
- Адрес клиента.
- Телефон клиента.
- Код автозаправки.
- Адрес автозаправки.
- Название фирмы.
- Юридический адрес.
- Телефон.
- Код топлива.
- Вид топлива.
- Единица измерения.
- Цена (руб.) за литр.
- Дата продажи топлива.
- Количество топлива.

Задания

1. Создайте таблицы, используя необходимые средства поддержки целостности данных:
2. Создайте запросы:
 - Сколько раз заправлял автомобиль каждый из клиентов
 - Кто из клиентов не приобрел топливо в июле текущего года?
 - Найти клиента, купившего наибольший объем топлива.
 - Вывести данные клиента, купившего топлива на наибольшую сумму
 - Какое топливо пользуется наибольшим спросом?
 - Сколько топлива каждого вида было продано за прошедший месяц.
 - Какая из заправок продала топлива на наибольшую сумму?.
 - Вывести список сотрудников, заключивших договоры на максимальную сумму за последний месяц.
3. Создать представление, содержащее сведения обо всех заправках и всех видах топлива, которые они продают.
4. Создать хранимые процедуры:
 - Вывести сведения обо всех покупках одного из клиентов за заданную дату (данные клиента, дата, объем топлива, уплаченная сумма). Параметры процедуры: карт-счёт клиента, дата.e
5. Создать триггер для уменьшения суммы на счете клиента при каждой покупке.

Типы данных, используемые в SQL-сервере

Текстовые типы данных:

Название	Описание
Char	Размерность до 8000 символов.
<i>Nchar</i>	Размерность до 4000 символов.
<i>Varchar</i>	Используется для хранения текстовой информации переменной длины. Размерность до 8000 символов.
<i>Nvarchar</i>	Используется для хранения текстовой информации переменной длины. Размерность до 4000 символов.

Числовые типы данных:

Название	Описание
<i>Int</i>	Положительные и отрицательные целые числа. Диапазон: от -2^{31} до $+2^{31}$
<i>Smallint</i>	Диапазон: от -32768 до 32767
<i>Tinyint</i>	Положительные целые числа. Диапазон: от 0 до 255
<i>Decimal</i>	Точный числовой, 5-17 байт
<i>Real</i>	Положительные и отрицательные числа с плавающей точкой с точностью до 7 цифр. Диапазон: от $-3,4E-38$ до $+3,4E+38$
<i>Float</i>	Положительные и отрицательные числа с плавающей точкой с точностью до 15 цифр. Диапазон: от $-1,7E-308$ до $+1,7E+308$
<i>Money</i>	Для хранения денежных значений. Диапазон от -922 337 203 685 477,5808 до + 922 337 203 685 477,5807
<i>Smallmoney</i>	от -214 748,3648 до 214 748,3647

Типы данных даты и времени:

Название	Описание
<i>Datetime</i>	Хранение комбинации даты и времени. Диапазон: от 01.01.1753 до 31.12.9999
<i>Smalldatetime</i>	Хранение комбинации даты и времени. Диапазон: от 01.01.1900 до 06.06.2079

Типы данных для хранения больших объемов информации:

Название	Описание
<i>Text</i>	Большие объемы текстовой информации. Размерность: от 1 до 214 783 647 байт.
<i>Ntext</i>	Большие объемы текстовой информации в Unicode. Размерность: от 1 до 1 073 741 823 байт.
<i>Image</i>	Длинные цепочки двоичных данных, что позволяет, например, записывать в таблицу рисунки, фотографии.

Типы данных специального назначения:

Название	Описание
<i>Bit</i>	Информация, принимающая только одно значение. Диапазон: от 0 до 1.
<i>Binary</i>	Хранение битовых цепочек. Размерность: до 80000 байт.
<i>Varbinary</i>	Хранение битовых цепочек варьируемой длины. Размерность: до 80000 байт.
<i>Timestamp</i>	Автоматически размещает значение счетчика каждый раз при вставке новой записи.

Типы ограничений

Тип целостности	Тип ограничения	Краткое название	Описание
Целостность полей	DEFAULT	DF	Определяет значение поля по умолчанию (на случай, если значение не было явно указано в операторе INSERT)
	CHECK	СК	Указывает правило достоверности для значений поля
	FOREIGN KEY	FK	Значения поля (полей) внешнего ключа должны соответствовать значениям поля (полей) первичного ключа таблицы, на которую ссылаются
Целостность сущностей	PRIMARY KEY	PK	Уникально идентифицирует каждую запись. Это гарантирует отсутствие в таблице повторяющихся значений и позволяет создать индекс для повышения производительности. Значения NULL не допускаются
	UNIQUE	U	Предотвращает дублирование непервичных ключей, что позволяет создать индекс для повышения производительности. Допускаются значения NULL
Ссылочная целостность	FOREIGN KEY	FK	Определяет поле или набор полей, значения которых соответствуют первичному ключу связанной таблицы
Целостность, определяемая пользователем	CHECK	СК	Определяет правило допустимости значений данных для поля

Данные для заполнения таблиц

Таблица BRANCH

<i>Branch_no</i>	Postcode	City	Street	House	Btel_no	Fax-no
1	210009	Витебск	Замковая	4/офис404	8(02122)37-73-56	37-73-58
2	210033	Витебск	Суворова	9/11	8(02122)36-01-80	33-25-23
3	211440	Новополоцк	Молодёжная	18	8(02144)57-31-29	57-25-30
4	211460	Россоны	Ленина	3	8(02159)25-55-20	

Таблица OWNER

<i>Owner_no</i>	FName	LName	City	Street	House	Flat	Otel_no
1	Жерносек	Юрий	Витебск	Терешковой	28/1	7	62-07-94
2	Панкратова	Инна	Новополоцк	Парковая	26	12	57-12-48
3	Амбражевич	Сергей	Новополоцк	Двинская	5	18	52-14-89
4	Поскрёбышева	Елена	Витебск	П.Бровки	26/1	40	23-00-72(а6978)
5	Титов	Николай	Витебск	Интернационалистов	35	187	8(029)633-76-68
6	Скребкова	Алла	Новополоцк	Молодёжная	1	22	8(029)688-84-46
7	Николаев	Влад	Витебск	Фрунзе	33	214	8(029)673-07-30
8	Цалко	Сергей	Лепель	Ленина	14/2	4	25-17-90
9	Цыркунова	Наталья	Россоны	Цветочная	90	15	26-32-48
10	Яковлев	Андрей	Витебск	Лазо	65		21-72-25

Таблица BUYER

Buyer_no	FName	LName	City	Street	House	Flat	Htel_no	Wtel_no	Prof_Rooms	Max_Price	Branch_no
1	Невердасов	Виктор	Витебск	Московский пр-т	16/4	117	62-08-19	36-40-80	2	110000	2
2	Кассап	Светлана	Новополоцк	Гайдара	17а	4	57-12-48		1	78000	3
3	Орлов	Александр	Минск	Либнехта	93	15	8(017)286-13-21		3	14500	1
4	Сафронова	Светлана	Витебск	пр-т Победы	3/1	324	8(029)661-07-30	22-67-94	2	60000	4
5	Окорков	Вадим	Минск	Лермонтова	35	187		8(017)224-84-24	5	300000	1
6	Семёнов	Вячеслав	Витебск	Замковая	4	13	23-00-72(аб964)		2	10500	1
7	Краснова	Жанна	Витебск	Клиническая	104			23-50-70	1	90000	2
8	Будда	Елена	Лепель	Ленина	3	5	25-17-90		4	200000	3
9	Боровая	Наталья	Орша	Смоленская	12/4	26	26-32-48		2	140000	2
10	Алипов	Игорь	Витебск	Московский пр-т	22	4	21-72-25		3	150000	2

44

Таблица STAFF

Staff_no	FName	LName	DOB	Sex	City	Street	House	Flat	Stel_no	Date_Joined	Position	Salary	Branch_no
ВМ0550260	Батуркин	Александр	17.10.68	М	Новополоцк	Парковая	5	13	23-79-77	1.01.2001	менеджер	2500000	3
ВМ0550261	Чубаро	Наталья	25.05.72	Ж	Витебск	Чкалова	21/1	12	8(029)662-47-32	10.02.2002	торговый агент	1800000	1
ВМ0550262	Коваленко	Светлана	1.02.70	Ж	Витебск	Чкалова	24	49	62-51-23	15.01.2001	менеджер	2500000	3
ВМ0550263	Логинов	Вадим	9.09.67	м	Витебск	Чкалова	41/2	96	23-06-73	2.09.1999	директор	3000000	1
ВМ0550264	Суворов	Виталий	11.07.65	М	Новополоцк	Школьная	47/1	157	22-29-03	1.02.1999	торговый агент	1800000	3
ВМ0550265	Жарков	Герман	6.03.68	М	Витебск	Гагарина	31	28	63-43-98	2.10.2001	менеджер	2800000	2
ВМ0550266	Ганущенко	Галина	5.11.69	ж	Витебск	Лазо	90/1	54	62-43-64	3.11.2004	торговый агент	1800000	2
ВМ0550267	Сотникова	Ольга	7.12.67	Ж	Полоцк	Вокзальная	7			2.02.2002	маклер	1000000	3
ВМ0550268	Янчиленко	Сергей	28.02.70	М	Россоны	Ленина	28/2	25		17.05.2005	торговый агент	1500000	4

Таблица PROPERTY

<i>Property_no</i>	<i>Data_registration</i>	<i>Postcode</i>	<i>City</i>	<i>Street</i>	<i>House</i>	<i>Flat</i>	<i>Floor_Type</i>	<i>Floor</i>	<i>Rooms</i>	<i>The_area</i>	<i>Balcony</i>	<i>Ptel</i>	<i>Selling_Price</i>	<i>Branch_no</i>	<i>Staff_no</i>	<i>Owner_no</i>
3000	16.12.11	210033	Витебск	Смоленская	11	57	5П	3	1	31/18/10	Б	T	60000	1	BMO550262	1
3001	17.12.11	210035	Витебск	Бровки	11	49	9К	9	1	37/21/7	Бз	-	70000	2	BMO550262	5
3002	18.12.11	210029	Витебск	Строителей	23/2	214	12П	2	2	81/29/9	Лз	T	92000	2	BMO550266	7
3003	19.12.11	210005	Витебск	Лазо	11	4	3К	3	2	65/40/7,6	2Бз	-	15000	1	BMO550261	5
3004	15.12.11	211460	Россоны	Ленина	32	20	5П	5	3	68,4/44,3/9,8 1	2Бз	T	100000	4	BMO550268	7
3005	11.12.11	211440	Новопол	Школьная	11	56	9П	3	1	36/18/8,2	Б	T	75000	3	BMO550260	6
3006	14.12.11	211440	Новопол	Молодёжная	5	14	9П	2	2	46/27/6,8	Лз	T	60000	3	BMO550264	3
3007	20.12.11	211180	Полоцк	Вокзальная	8	15	5К	5	3	65/38/7	Б	T	80000	3	BMO550267	2
3008	16.01.11	211460	Россоны	Советская	17	1	5К	1	3	65/38/7	-	T	47500	4	BMO550268	7

45

Таблица VIEWING

<i>Date_View</i>	<i>Comments</i>	<i>Property_no</i>	<i>Buyer_no</i>
17.01.12	согласен	3002	1
17.01.12	согласен	3003	1
18.01.12	не согласен	3002	4
19.01.12	согласен	3005	2
25.03.12	требует ремонта	3001	7

Краткое определение основных терминов

База Данных (БД) – некоторый набор таблиц, связанных между собой определенной связью для выполнения различного рода задач.

Блокировка – временно накладываемое ограничение на выполнение некоторых операций обработки данных.

Внешние ключи – это поля таблицы, которые соответствуют первичным ключам из других таблиц.

Данные – это информация, зафиксированная в определенной (структурированной) форме, пригодной для последующей обработки, хранения и передачи.

Запрос – специальным образом описанное требование, определяющее состав производимых над базой данных операций по выборке или модификации хранимых данных.

Импорт данных – это процесс, в результате которого данные извлекаются из внешних источников и после соответствующей обработки вставляются в таблицы базы данных SQL Server.

Индекс – упорядоченный список полей или групп полей в таблице, используемый для ускорения операции поиска записей в таблице, а также выполнения других операций, использующих поиск.

Нормализация – удаление избыточных данных из каждой таблицы в базе данных, с целью устранения дублирования и обеспечения непротиворечивости хранимых данных.

Первичный ключ – это уникальное поле (или несколько полей), однозначно определяющее записи таблицы базы данных.

Представление – некоторое подобие таблицы, содержание которого выбирается из других таблиц с помощью выполнения запроса, причем, при изменении значений в таблицах, данные автоматически меняются и в представлении.

Привилегии – права пользователя на проведение тех или иных действий над определенным объектом базы данных.

Резервное копирование – создание резервной копии базы данных.

Репликация – это копирование данных, расположенных на одном сервере, на один или несколько других серверов.

Система Управления Базами Данных (СУБД) представляет собой комплекс инструментальных средств (программных и языковых) реализующих централизованное управление БД и обеспечивающих доступ к данным (изменения, добавления, удаления, резервного копирования и т.д.).

Стандартное значение (умолчание) – это значение, которое будет присвоено колонке таблицы при вставке строки, если в команде явно не указано значение для этой колонки.

Транзакция – последовательность операций над базой данных, отслеживаемая системой управления базами данных от начала до завершения как единое целое.

Триггеры – это специальный класс хранимых процедур, автоматически запускаемых при добавлении, изменении и удалении данных из таблицы.

Хранимая процедура – программа (процедура) обработки данных, хранящаяся и выполняемая на компьютере-сервере.

Экспорт данных – это процесс, в результате которого данные из таблицы SQL Server трансформируются в указанный формат.

Языковые средства – языки, с помощью которых описывается структура данных (DDL) и языки манипулирование данными (DML).

РЕПОЗИТОРИЙ ВГУ

Дополнительные параметры создаваемого индекса

Unique values – ввод в определенное поле только уникальных значений. Позволяет осуществлять автоматическую проверку уникальности при каждом добавлении новой записи. Рекомендуется установить обязательный ввод значений в поле, для которого планируется создание уникального индекса;

Clustered index – в системе SQL-сервер имеется возможность физического индексирования данных. Другими словами, использование индексов приводит к созданию отдельной структуры, которая связывается с физическим расположением данных в таблице. Использование этой опции позволяет произвести так называемое кластерное индексирование, в результате чего будут отсортированы данные в самой таблице согласно порядку этого индекса, и вся добавляемая информация будет приводить к изменению физического порядка данных. При этом нужно учитывать, что в таблице может быть определён только один кластерный индекс;

Ignore duplicate values – игнорирование ввода повторяющихся значений в проиндексированных полях. Использование данного параметра совместно с *Unique values* позволяет игнорировать уникальность значений для этого поля. Обычно использование этого параметра имеет значение при ориентации на распределение данных в разрабатываемых структурах;

Do not recompute statistics (not recommended) – определяет функцию автоматического обновления статистики для таблицы. Устанавливать не рекомендуется.

File group – осуществляет выбор файловой группы, в которой будет находиться создаваемый индекс. Использование индекса из другой файловой группы повышает производительность некластерных индексов в связи с параллельностью выполнения процессов ввода/вывода и работы с самим индексом. При выборе данного параметра активизируется список файловых групп, позволяющий определить необходимую группу для размещения индекса;

Fill factor – осуществляется настройка разбиения индекса на страницы. Коэффициент *fillfactor* определяет в процентном соотношении размер создаваемых индексных страниц. Если планируется частое изменение, удаление и добавление информации в таблице базы данных, то коэффициент следует установить как можно меньше, например, 20. А установка коэффициенту значения 100 рекомендуется при использовании больших таблиц, обращение к которым обычно происходит только для чтения;

Pad index – определяет заполнение внутреннего пространства индекса и используется совместно с параметром *fillfactor*;

Drop existing – определяет повторное создание кластерного индекса, что позволяет предотвратить нежелательное обновление кластерных индексов.

Список функций SQL Server

Компоненты даты

<i>Компонент</i>	<i>Сокращение</i>	<i>Допустимые значения</i>
<i>Год (Year)</i>	<i>Yy</i>	<i>1753-9999</i>
<i>Месяц (Month)</i>	<i>Mm</i>	<i>1-12</i>
<i>День года (Dayofyear) Dy</i>		<i>1-366</i>
<i>День месяца (Day)</i>	<i>Dd</i>	<i>1-31</i>
<i>День недели (Weekday)</i>	<i>Dw</i>	<i>1-7 (1 = воскресенье)</i>

Некоторые глобальные переменные

<i>Глобальная переменная</i>	<i>Описание</i>
<i>@@CONNECTIONS</i>	<i>Количество успешных подключений плюс количество неудачных попыток подключения</i>
<i>@@CURSOR_ROWS</i>	<i>Количество записей в последнем открытом курсоре</i>
<i>@@ERROR</i>	<i>Номер ошибки для последней команды TSQL</i>
<i>@@IDENTITY</i>	<i>Последнее вставленное значение счётчика</i>
<i>@@LANGUAGE</i>	<i>Название текущего языка</i>
<i>@@LANGID</i>	<i>Идентификатор текущего языка</i>
<i>@@MAX_CONNECTIONS</i>	<i>Максимальное количество одновременных подключений, разрешенное SQL Server</i>
<i>@@ROWCOUNT</i>	<i>Количество записей, участвовавших в выполнении последней команды</i>
<i>@@TRANCOUNT</i>	<i>Количество активных транзакций</i>
<i>@@VERSION</i>	<i>Дата, версия SQL Server и тип процессора</i>

Список функций

<i>Функция</i>	<i>Описание</i>
<i>ABS(числовое_выражение)</i>	<i>Модуль числа</i>
<i>ACOS(вещественное_выражение)</i>	<i>Арккосинус (в радианах)</i>
<i>ASCII(символьное_выражение)</i>	<i>ASCII-код первого символа в строке</i>
<i>ASIN(вещественное_выражение)</i>	<i>Арксинус (в радианах)</i>
<i>ATAN(вещественное_выражение)</i>	<i>Арктангенс (в радианах)</i>
<i>CASE</i>	<i>Вычисление результата по списку выражений</i>
<i>CAST</i>	<i>Преобразование типа данных</i>
<i>CEILING</i>	<i>Наименьшее целое, большее либо равное заданной величине</i>
<i>CHAR(целое_выражение)</i>	<i>Преобразование целого числа в ASCII-символ</i>
<i>CHARINDEX(символьное_выражение, выражение, начало)</i>	<i>Начальная позиция подстроки в выражении (0, если подстрока не найдена)</i>
<i>COALESCE(выражение1, выражение2)</i>	<i>Первое выражение, отличное от NULL</i>
<i>CONVERT(тип_данных, выражение [, стиль])</i>	<i>Преобразование типа данных</i>
<i>COS(вещественное_выражение)</i>	<i>Косинус угла</i>
<i>COT(вещественное_выражение)</i>	<i>Котангенс угла</i>
<i>CURRENT_USER</i>	<i>Эквивалент USER_NAME()</i>
<i>DATALENGTH(символьное_выражение)</i>	<i>Целое число символов в выражении, не считая завершающих пробелов</i>
<i>ROUND(числовое_выражение, целое_выражение[, функция])</i>	<i>Округление числового выражения до позиции, задаваемой целым выражением. Если третий аргумент отличен от 0, ROUND не округляет, а отбрасывает цифры</i>

Функция	Описание
<i>DATENAME</i> (компонент, выражение_дата)	<i>Компонент даты, выраженный в виде строки. По возможности преобразуется в имя (например, June)</i>
<i>DATEPART</i> (компонент, выражение_дата)	<i>Заданный компонент даты в виде целого числа</i>
<i>DATEDIFF</i> (компонент, выражение_дата1, выражение_дата2)	<i>Разность дат, выраженная в заданных компонентах</i>
<i>DAY</i> (дата)	<i>Целое число, определяющее день месяца</i>
<i>EXP</i> (вещественное_выражение)	<i>Экспонента заданного числа</i>
<i>FLOOR</i> (числовое_выражение)	<i>Наибольшее целое, меньшее либо равное заданной величине</i>
<i>GETDATE</i> ()	<i>Текущие системная дата и время</i>
<i>GETANSINULL</i> (имя БД)	<i>Выводит допустимо или недопустимо использовать в БД значение NULL</i>
<i>IDENT_SEED</i> (таблица)	<i>Выводит начальное значение счетчиков в таблице</i>
<i>IDENT_INCR</i> ({имя_таблицы имя_представления})	<i>Величина приращения столбца счётчика</i>
<i>IS_MEMBER</i> ({группа / роль})	<i>1, если текущий пользователь является членом группы NT или роли SQL Server, 0 — если не является и NULL, если группа/роль не определена</i>
<i>IS_SRVROLEMEMBER</i> (роль{, имя})	<i>1, если имя пользователя входит в роль SQL Server</i>
<i>ISDATE</i> (выражение)	<i>1, если выражение определяет допустимую дату</i>
<i>ISNUMERIC</i> (выражение)	<i>1, если выражение является числовым</i>
<i>LEFT</i> (символьное_выражение, целое выражение)	<i>Заданное число символов от начала строки</i>

Функция	Описание
<i>LEN</i> (символьное_выражение)	Длина строки в символах без учета завершающих пробелов
<i>LOG10</i> (вещественное_выражение)	Десятичный логарифм
<i>LTRIM</i> (символьное_выражение)	Удаление начальных пробелов
<i>LOWER</i> (символьное_выражение)	Преобразование строки в нижний регистр
<i>MONTH</i> (дата)	Целый номер месяца
<i>PI</i> ()	Константа 3,1415926...
<i>POWER</i> (основание, степень)	Возведение в степень
<i>QUOTENAME</i> (символьное_выраже	Преобразование строки в идентификатор.
<i>RADIANS</i> ()	Преобразование градусов в радианы
<i>RAND</i> ([целое_выражение])	Получение случайного вещественного числа в интервале от 0 до 1 (необязательное целое • выражение используется для раскрутки генератора случайных чисел)
<i>REPLACE</i> (символьное_выражение1, символьное_выражение2,	Замена всех экземпляров строки 2 в строке 1 на строку 3
<i>REVERSE</i> (символьное_выражение)	Обратная перестановка строки
<i>RIGHT</i> (символьное_выражение, целое_выражение)	Заданное число символов в конце строки
<i>ROUND</i> (числовое_выражение, целое_выражение [, функция])	Округление числового выражения до позиции, задаваемой целым выражением. Если третий аргумент отличен от 0, <i>ROUND</i> не округляет, а отбрасывает цифры
<i>RTRIM</i> (символьное_выражение)	Удаление завершающих пробелов
<i>SIGN</i> (целое_выражение)	+1 для положительных чисел, -1 для отрицательных и 0 для нуля
<i>SIN</i> (вещественное_выражение)	Синус угла
<i>SPACE</i> (целое_выражение)	Построение строки из заданного количества
<i>SQRT</i> (вещественное_выражение)	Извлечение квадратного корня
<i>SQUARE</i> (вещественное_выражение)	Возведение в квадрат
<i>STR</i> (вещественное_выражение	Преобразование числа в строку
<i>STUFF</i> (символьное_выражение1,	<i>n</i> символов строки 1, начиная с заданной

Функция	Описание
<i>SUBSTRING</i> (выражение, начало,	<i>Получение подстроки</i>
<i>SUSER_ID</i> ([имя_пользователя])	<i>Идентификатор пользователя SQL Server</i>
<i>SYSTEM_USER</i>	<i>Имя текущего пользователя SQL Server</i>
<i>TAN</i> (вещественное_выражение)	<i>Тангенс угла</i>
<i>UPPER</i> (символьное_выражение)	<i>Преобразование строки в верхний регистр</i>
<i>USER_ID</i> (имя_в_базе)	<i>Идентификатор пользователя в базе данных</i>
<i>USER_NAME</i> ([идентификатор_пользователя]) или <i>SESSION_USER</i>	<i>Имя пользователя в базе данных</i>
<i>YEAR</i> (выражение_дата)	<i>Год в виде четырёх цифр</i>
<i>SUM</i> ([all distinct] выражение)	<i>Вычисляет общее количество (разных) значений в числовом столбце</i>
<i>AVG</i> ([all distinct] выражение)	<i>Вычисляет среднее арифметическое</i>
<i>COUNT</i> ([all distinct] выражение)	<i>Вычисляет количество (разных) значений в числовом столбце, отличных от NULL</i>
<i>COUNT</i> (*)	<i>Вычисляет количество выбранных записей. Единственная агрегатная функция, учитывающая значения NULL</i>
<i>MAX</i> (выражение)	<i>Определяет максимум среди выбранных значений</i>
<i>MIN</i> (выражение)	<i>Определяет минимум среди выбранных значений</i>
<i>STDEV</i> (выражение)	<i>Вычисляет среднее квадратичное отклонение</i>
<i>STDEVP</i> (выражение)	<i>Вычисляет среднее квадратичное отклонение для выборки</i>
<i>VAR</i> (выражение)	<i>Вычисляет статистическую дисперсию</i>
<i>VARP</i> (выражение)	<i>Вычисляет статистическую дисперсию</i>

Учебное издание

АДАМЕНКО Наталья Дмитриевна

МОДЕЛИ ДАННЫХ И СУБД

Методические рекомендации

В 3 частях

Часть 3

Технический редактор

Г.В. Разбоева

Компьютерный дизайн

Т.Е. Сафранкова

Подписано в печать .2015. Формат 60x84¹/₁₆. Бумага офсетная.

Усл. печ. л. 3,13. Уч.-изд. л. 2,07. Тираж экз. Заказ .

Издатель и полиграфическое исполнение – учреждение образования

«Витебский государственный университет имени П.М. Машерова».

Свидетельство о государственной регистрации в качестве издателя,

изготовителя, распространителя печатных изданий

№ 1/255 от 31.03.2014 г.

Отпечатано на ризографе учреждения образования

«Витебский государственный университет имени П.М. Машерова».

210038, г. Витебск, Московский проспект, 33.