

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Романчук Иван Сергеевич

Должность: Ректор

Дата подписания: 30.03.2023 10:42:57

Уникальный программный ключ:

6319edc2b582ffda6ea443ff01d5779368d0957ac34f5cd074d81181530452479

**РОССИЙСКАЯ ФЕДЕРАЦИЯ МИНИСТЕРСТВО ОБРАЗОВАНИЯ И
НАУКИ ФГАОУ ВО ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ
НАУК**

КАФЕДРА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Нестерова О. А.

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ
ЛАБОРАТОРНЫХ РАБОТ**

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	2
Тема 1. ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАЦИОННЫХ СИСТЕМ	3
Тема 2. ПОЛЬЗОВАТЕЛИ ИНФОРМАЦИОННЫХ СИСТЕМ; ПРЕИМУЩЕСТВА ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ ДАННЫМИ;	8
АДМИНИСТРАТОР БАЗЫ ДАННЫХ	8
Тема 3. ОСНОВНЫЕ ОПЕРАЦИИ НАД ДАННЫМИ, СТРУКТУРИРОВАННЫЙ ЯЗЫК ЗАПРОСОВ – SQL Transacr-SQL в MS SQL Server	14
Тема 4. ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ. СОХРАННОСТЬ И ЗАЩИТА БАЗ ДАННЫХ	20
Тема 5. ТЕХНОЛОГИИ И МОДЕЛИ АРХИТЕКТУРЫ КЛИЕНТ/СЕРВЕР	24
Тема 6. РАБОТА ПРИЛОЖЕНИЙ С БАЗАМИ ДАННЫХ. ТЕНДЕНЦИИ РАЗВИТИЯ ИНФОРМАЦИОННЫХ СИСТЕМ	29
Тема 7. РАСПРЕДЕЛЕННЫЕ БАЗЫ ДАННЫХ	30
СПИСОК ЛИТЕРАТУРЫ	36

ВВЕДЕНИЕ

Целью дисциплины «Системы управления базами данных» является формирование у студентов набора профессиональных качеств, обеспечивающих решение задач, связанных с разработкой информационных приложений на базе промышленных систем управления базами данных (СУБД). Студент должен знать характеристики и типы систем баз данных,

области применения систем управления базами данных, этапы проектирования баз данных, физическую организацию баз данных.

Задачей курса является:

- изучение основных понятий и принципов разработки и эксплуатации информационных систем и баз данных;
 - освоение технологий построения приложений на базе промышленных СУБД;
 - изучение программных средств, используемых для создания баз данных;
 - формирование практических навыков использования СУБД.
- Изучение курса основано на следующих дисциплинах: «Базы данных», «Информационные технологии», «Языки программирования», «Технологии и методы программирования».

В результате изучения дисциплины студенты должны знать:

- особенности основных типов промышленных СУБД;
- основные принципы работы с распределенными базами данных;
- назначение и основные компоненты баз данных; уметь:
- формировать модель предметной области (инфологический дизайн);
- сформировать логическую модель данных (datalogical design);
- создать схему базы данных для конкретной СУБД (физический дизайн);
- выбрать необходимые инструменты для разработки информационных систем на базе промышленных СУБД;

Тема 1. ОСНОВНЫЕ ПОНЯТИЯ ИНФОРМАЦИОННЫХ СИСТЕМ.

Система — это образующая единое целое совокупность материальных и нематериальных объектов, объединенных некоторыми общими признаками, назначениями, свойствами, условиями существования, жизнедеятельности, функционирования и т. д.

Функционирование системы - процесс переработки входной информации в выходную, носящий последовательный характер во времени.

Подсистема - часть любой системы.

Свойства системы (в т. ч. ИС):

- **сложность** - система зависит от множества входящих в нее компонентов, их структурного взаимодействия, а также сложности внутренних и внешних связей;
- **делимость** - система состоит из ряда подсистем или элементов, выделенных по определенным признакам и отвечающих конкретным целям и задачам;
- **целостность системы** - означает то, что все элементы системы функционируют как единое целое;
- **многообразие элементов системы и различие их природы** - свойство связано с функционированием элементов, их спецификой и автономностью;
- **структурность** - определяет наличие установленных связей и отношений между элементами внутри системы, распределение элементов системы по уровням и иерархиям;
- **адаптивность системы** - означает приспособляемость системы к условиям конкретной предметной области;
- **интегрируемость** - означает возможность взаимодействия

системы с вновь подключаемыми компонентами или подсистемами.

Информационная система — это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических,

технологических и других средств, а также персонала, предназначенная для сбора, обработки, хранения и выдачи экономической информации и принятия управленческих решений.

Свойства информационных систем:

- любая ИС может быть подвергнута анализу, построена и управляема на основе общих принципов построения сложных систем;
- при построении ИС необходимо использовать системный подход;
- ИС является динамичной и развивающейся системой;
- ИС следует воспринимать как систему обработки информации, состоящую из компьютерных и телекоммуникационных устройств, реализованную на базе современных технологий;
- выходной продукцией ИС является информация, на основе которой принимаются решения или производятся автоматическое выполнение рутинных операций;
- участие человека зависит от сложности системы, типов и наборов данных, степени формализации решаемых задач.

Вопросы для изучения

1. Этапы развития информационных систем.
2. Роль и место системы управления базой данных (СУБД) в информационной системе.
3. Обзор промышленных СУБД.

Лабораторная работа №1

Сценарий сдачи лабораторной работы: скрипты на изменение/вставку/удаление данных, демонстрирующие работу созданных ограничений и триггеров.

При отсутствии в БД полей соответствующего типа создайте их.

I. Создание ограничений

При создании таблиц Вы уже используете такие виды ограничений, как определение типа значений столбца, первичный ключ (ключевой столбец таблицы), внешний ключ (ссылка на ключевой столбец другой таблицы), разрешение нулевых значений, столбец-идентификатор (счетчик). Кроме этого, существуют и другие виды ограничений. Ограничения для таблицы можно создавать как при создании таблицы, так и при изменении структуры существующей таблицы.

Создайте файл запроса. Напишите команды, устанавливающие следующие ограничения.

1. Создайте тип [vc10] varchar(10) NOT NULL;
2. Установите значение по умолчанию (текущая дата) для поля типа datetime.
3. Установите проверку условия: одно поле типа дата должно быть меньше другого поля типа дата, либо иметь нулевое значение.
4. Установите проверку значения поля, содержащего номер телефона (текстового типа, например, varchar, а не юникод, например, nvarchar!). Значение должно соответствовать шаблону, например, “33-33-33”, либо иметь нулевое значение.
5. Установите проверку уникальности для любого (не ключевого) поля.
6. Установите для ключевого поля хотя бы одной из таблиц свойство “идентификатор”.
7. Создайте диаграмму базы данных. Установите соответствующие связи между таблицами. В обозревателе свойств выберите действия.
 - Таблица1 – Таблица2 - каскадное изменение/удаление;
 - Таблица1 – Таблица3 - установить значение в NULL;

- Таблица1 – Таблица4 - установить значение по умолчанию 0.

II. Проверка ограничений

1. Создайте новый файл запроса.

- Напишите несколько команд вставки/удаления/изменения/просмотра данных в таблицах, в том числе и заведомо неправильные запросы в соответствии с ограничениями, которые были созданы.
- Нажмите на панели инструментов кнопку Включить статистику клиента, а затем кнопку Выполнить запрос.
- Перейдите на вкладку Статистика клиента и просмотрите приведенные на ней данные.

2. Создайте еще один файл запроса.

- В окне запроса введите команды для включения/отключения, если это возможно, или удаления ограничений.
- В том числе создать запрос для включения/отключения свойства идентификатора.

3. Повторите запросы изменения/вставки/удаления.

- Убедитесь, что запросы, не соответствующие условиям ограничений, теперь выполняются.

4. Включите или создайте снова все ограничения.

III. Создание ограничений с помощью триггеров

Выберите одну из таблиц БД. Допустим, Students. Создайте таблицу StudArc – архив, куда будут переноситься данные в случае изменения или удаления записи в некоторой таблице вашей БД. Структура таблицы StudArc аналогична структуре выбранной таблицы с дополнительными полями: Action bit NOT NULL – действие, 0 – изменение, 1 – удаление; dtAct – текущая дата;

userName varchar(10) NOT NULL – имя пользователя.

1. Создайте триггер [students_Del] на удаление данных в таблице Students.

В StudArc необходимо переносить все данные удаляемой записи, кроме этого, записывать значение 1 в поле Action, текущую дату и системного пользователя в соответствующие поля.

2. Создайте триггер [students_Upd] на изменение таблицы Students, который записывает в StudArc две записи.

Одна запись должна содержать данные до изменения, вторая запись – после изменения, даже если меняется только одно поле.

3. Создайте триггер [students_Ins], который не позволяет добавлять более 10 записей в таблицу.

4. Протестируйте работу триггеров.

Создайте новый файл запроса. Напишите команды включения/отключения триггеров.

Тема 2. ПОЛЬЗОВАТЕЛИ ИНФОРМАЦИОННЫХ СИСТЕМ. ПРЕИМУЩЕСТВА ЦЕНТРАЛИЗОВАННОГО УПРАВЛЕНИЯ

ДАНЫМИ. АДМИНИСТРАТОР БАЗЫ ДАННЫХ

Пользователи базы данных делятся на три большие группы:

- **Прикладные программисты.** Это сотрудники компании, которые создают прикладное программное обеспечение для решения конкретных задач компании. Программное обеспечение может быть написано на стандартном языке программирования или же на языке (обычно

называемом языке четвертого поколения), входящем в комплект системы управления базой данных;

- **Конечные пользователи.** Это те люди, которым информация базы данных требуется для выполнения их прямых служебных обязанностей;
- **Администраторы.** Выделяют администратора сервера баз данных. Он ведает установкой, конфигурированием сервера, регистрацией пользователей, групп, ролей и т. п. Прямо или косвенно он обладает всеми привилегиями, которые имеют или могут иметь другие пользователи. Кроме администратора сервера может быть и администратор базы данных. К этой категории относится любой пользователь, создавший базу данных, и, следовательно, являющийся ее владельцем. Он может предоставлять другим пользователям доступ к базе и к содержащимся в ней объектам. Администратор базы отвечает за ее сохранение и восстановление. В принципе в организации может быть много администраторов баз данных.

Преимущества системы баз данных по сравнению с бумажными методами сохранения записей, следующие:

- компактность;
- скорость;
- низкие трудозатраты;

Преимущества баз данных, связанные с централизованным управлением:

- возможность сокращения избыточности;
- возможность устранения противоречивости;
- возможность общего доступа к данным;
- соблюдение стандартов;
- введение ограничений для обеспечения безопасности;

- обеспечение целостности данных;
- возможность сбалансировать противоречивые требования.

Основная цель систем баз данных – обеспечение независимости данных. Независимость данных можно определить как иммунитет к изменениям в структуре хранения данных и в методах доступа к ним.

Вопросы для изучения

1. Пользователи базы данных.
2. Управления доступом к базам данных и разрешениями на их объекты.

Лабораторная работа №2 Создание

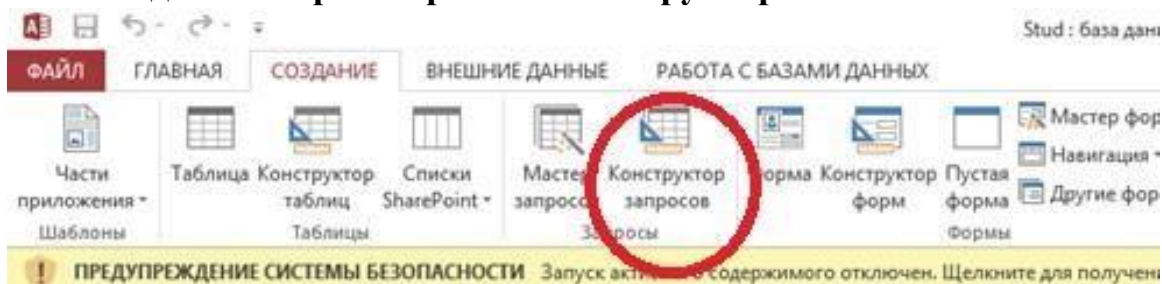
запросов в MS Access.

Параллельно с выполнением заданий рекомендуется ознакомление с материалами **Интерактивного учебника по SQL**: <http://www.sql-tutorial.ru/>

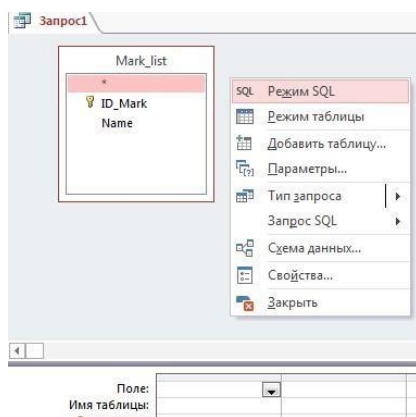
В учебнике имеются возможность практического обучения владения языком SQL. Прохождение обучения будет оцениваться дополнительными баллами.

Первые три задания выполняются в режиме конструктора, остальные в режиме SQL (допустимо переключение в режим конструктора, если невозможно сразу сформулировать команду запроса)

Создание запроса в режиме конструктора:



Создание запроса в режиме SQL:



Общий вид запроса на выборку данных: SELECT

```
[DISTINCT | ALL]{*
| [<выражение для столбца> [[AS]
<псевдоним>]] [,...]} FROM <имя таблицы>
[[AS] <псевдоним>] [,... ]
[WHERE <предикат>]
[[GROUP BY <список столбцов>] [HAVING
<условие на агрегатные значения>] ]
[ORDER BY <список столбцов>]
```

I. Задание

1. Выбрать студентов с 1997 года рождения
2. Выбрать иногородних студентов
3. Выбрать должников
4. Выбрать студентов, экзамен по предметам, начинающихся на 'С'
5. Выбрать студентов мужского пола, слушающих предметы, фамилия преподавателей которых начинается на 'Ива'
6. Выбрать предметы, специальностей, код которых заканчивается на '5'
7. Выбрать студентов, которые еще учатся (добавить поле Дата выпуска)
8. Выбрать выпускников
9. Выбрать студентов с неуказанным адресом

10. Выбрать студентов, выпустившихся в 2007, 2008, 2009 годах
11. Выбрать преподавателей, читающих 'СУБД', 'ЯП', 'Мат. анализ'
12. Выбрать студентов, сдавших хоть один предмет
13. Выбрать преподавателей, которые принимали экзамен у Иванова и Петрова
14. Найти минимальную, максимальную, среднюю оценку
15. Найти сумму заработанных баллов каждым студентом
16. Найти кол-во отличников по каждому предмету
17. Найти кол-во преподавателей по предмету
18. Выбрать студентов с количеством долгов больше 2, но меньше 10
19. Выбрать студентов, сдавших больше, чем 5 раз экзаменов, но меньше, чем 8
20. Выбрать строки из ведомости с оценками больше, чем 2, но меньше, чем 5

Общие запросы:

21. Для каждого города (город, откуда студент приехал) указать количество студентов и средний балл
22. Для каждого преподавателя указать сколько раз он принимал экзамен (группировка по дате сдачи)
23. Проверить есть ли предметы, у которых дата сдачи студентом указана больше, чем дата выпуска
24. Проверить есть ли студенты, у которых указана дата сдачи, но оценка не поставлена
25. Выбрать предметы, которые сдавали студенты мужского пола, старше 20 лет
26. Найти все студентов, которые не сдали ни одной одного предмета до 01.09.2012

Тема 3. ОСНОВНЫЕ ОПЕРАЦИИ НАД ДАННЫМИ, СТРУКТУРИРОВАННЫЙ ЯЗЫК ЗАПРОСОВ – SQL Transact-SQL в MS SQL Server.

В архитектуре «клиент-сервер» язык SQL занимает очень важное место. Именно он используется как язык общения клиентского программного обеспечения с серверной СУБД, расположенной на удаленном компьютере. Так, клиент посылает серверу запрос на языке SQL, а сервер разбирает его, интерпретирует, выбирает план выполнения, выполняет запрос и отправляет клиенту результат.

Разработан стандарт языка SQL, который является совместной разработкой *ANSI (American National Standards Institute)* и *ISO (International Organization for Standardization)*.

Достоинства SQL:

1. Наличие международных стандартов.
2. Независимость от конкретной СУБД. Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своем тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую.
3. Поддержка архитектуры клиент-сервер.
4. Распространенность.
5. Быстрое обучение.
6. Декларативность. С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. Каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако программисту полезно представлять, как СУБД будет разбирать текст его запроса. Чем сложнее сконструирован

запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных.

Недостатки SQL:

1. Неполное соответствие реляционной модели данных (наличие дубликатов, необязательность первичного ключа, возможность упорядочения результатов).
2. Недостаточно продуманный механизм неопределенных значений.
3. Сложность формулировок и громоздкость.

Команды языка определения данных (*DDL – Data Definition Language*)

Команда	Смысл	Действие
<i>CREATE TABLE</i>	Создать таблицу	Создает новую таблицу в БД
<i>DROP TABLE</i>	Удалить таблицу	Удаляет таблицу из БД
<i>TRUNCATE TABLE</i>	Удалить данные из таблицы	Уничтожает данные в таблице, но сохраняет ее структуру и индексы
<i>ALTER TABLE</i>	Изменить таблицу	Изменяет структуру существующей таблицы или ограничения целостности, задаваемые для данной таблицы
<i>CREATE VIEW</i>	Создать представление	Создает виртуальную таблицу, соответствующую некоторому SQL-запросу
<i>ALTER VIEW</i>	Изменить представление	Изменяет ранее созданное представление
<i>DROP VIEW</i>	Удалить представление	Удаляет ранее созданное представление
<i>CREATE INDEX</i>	Создать индекс	Создает индекс для некоторой таблицы для обеспечения быстрого доступа по атрибутам, входящим в индекс
<i>DROP INDEX</i>	Удалить индекс	Удаляет ранее созданный индекс

Команды языка манипулирования данными (*DML – Data Manipulation Language*)

Команда	Смысл	Действие
1	2	3
<i>DELETE</i>	Удалить строки	Удаляет одну или несколько строк, соответствующих условиям фильтрации, из базовой таблицы. Применение команды согласуется с принципами поддержки целостности, поэтому эта команда не всегда может быть выполнена корректно, даже если синтаксически она записана правильно
<i>INSERT</i>	Вставить строку	Вставляет одну строку в базовую таблицу. Допустимы модификации команды, при которых сразу несколько строк могут быть перенесены из одной таблицы или запроса в базовую таблицу
<i>UPDATE</i>	Обновить строку	Обновляет значения одного или нескольких столбцов в одной или нескольких строках, соответствующих условиям фильтрации

Язык запросов (*DQL – Data Query Language*)

Команда	Смысл	Действие
<i>SELECT</i>	Выбрать строки	Команда, заменяющая все операторы реляционной алгебры и позволяющая сформировать результирующее отношение, соответствующее запросу

Средства управления транзакциями

Команда	Смысл	Действие
<i>COMMIT</i>	Завершить транзакцию	Завершить комплексную взаимосвязанную обработку информации, объединенную в транзакцию
<i>ROLLBACK</i>	Откатить транзакцию	Отменить изменения, проведенные в ходе выполнения транзакции
<i>SAVEPOINT</i>	Сохранить промежуточную точку выполнения транзакции	Сохранить промежуточное состояние БД, пометить его для того, чтобы можно было в дальнейшем к нему вернуться

Средства администрирования БД

Команда	Смысл	Действие
---------	-------	----------

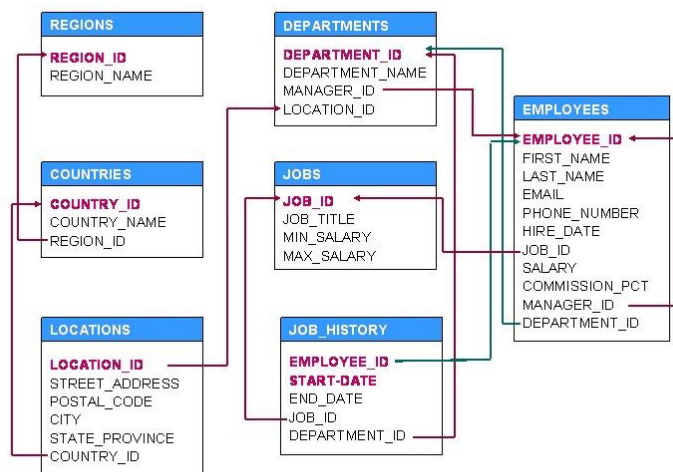
1	2	3
А) Управление БД		
<i>ALTER DATABASE</i>	Изменить БД	Изменить набор основных объектов в базе данных, ограничений, касающихся всей базы данных
<i>CREATE DATABASE</i>	Создать БД	Создать новую базу данных, определив основные параметры для нее
<i>DROP DATABASE</i>	Удалить БД	Удалить существующую базу данных (только в том случае, если вы имеете право выполнить это действие)
Б) Управление доступом		
<i>GRANT</i>	Предоставить права	Предоставить права доступа на ряд действий над некоторым объектом БД
<i>REVOKE</i>	Лишить прав	Лишить прав доступа к некоторому объекту или некоторым действиям над объектом
<i>DENY</i>	Запретить доступ	Запретить доступ к объектам базы данных

Вопросы для изучения

1. Языковые средства манипулирования данными в реляционных СУБД.
2. Языковые средства описания данных реляционных СУБД.
3. Хранимые процедуры и функции.
4. Работа с триггерами.

Лабораторная работа №3

Создайте базу данных, имеющую следующую структуру:



Примерный перечень запросов: Restricting and Sorting Data

1. Таблица Employees. Получить список с информацией обо всех сотрудниках
2. Таблица Employees. Получить список всех сотрудников с именем 'David'
3. Таблица Employees. Получить список всех сотрудников с job_id равным 'IT_PROG'
4. Таблица Employees. Получить список всех сотрудников из 50го отдела (department_id) с зарплатой(salary), большей 4000
5. Таблица Employees. Получить список всех сотрудников из 20го и из 30го отдела (department_id)
6. Таблица Employees. Получить список всех сотрудников, у которых последняя буква в имени равна 'a'
7. Таблица Employees. Получить список всех сотрудников из 50го и из 80го отдела (department_id) у которых есть бонус (значение в колонке commission_pct не пустое)
8. Таблица Employees. Получить список всех сотрудников, у которых в имени содержатся минимум 2 буквы 'n'
9. Таблица Employees. Получить список всех сотрудников, у которых длина имени больше 4 букв

- 10.** Таблица Employees. Получить список всех сотрудников, у которых зарплата находится в промежутке от 8000 до 9000 (включительно)
- 11.** Таблица Employees. Получить список всех сотрудников, у которых в имени содержится символ '%'
- 12.** Таблица Employees. Получить список всех ID менеджеров
- 13.** Таблица Employees. Получить список работников с их позициями в формате: Donald(sh_clerk)

Тема 4. ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ. СОХРАННОСТЬ И ЗАЩИТА БАЗ ДАННЫХ

Существуют следующие виды ограничений:

1. Ограничители значений (domain constraints);
2. Ограничители ключей (key constraints);
3. Ограничители записи (entity constraints);
4. Ссылочная целостность (Declarative Referential Integrity, DRI).

Ограничители ключей задаются посредством запрета неопределенных Null значений для ключевых полей (по неопределенному значению невозможно произвести идентификацию записи) и требованием уникальности значения ключевого поля. При сравнении неопределенных значений не действуют стандартные правила сравнения: одно неопределенное значение никогда не считается равным другому неопределенному значению. В SQL во фразе WHERE для выявления равенства значения некоторого атрибута неопределенному значению применяют специальные предикаты:

<имя атрибута> IS NULL и <имя атрибута> IS NOT NULL

Уникальность значений любого поля задается с помощью оператора UNIQUE. Для объявления уникальности совокупности полей применяется конструкция вида:

UNIQUE (<поле1>, <поле2>, ...)

Для объявления первичного ключа используется оператор PRIMARY KEY. Дополнительно, с помощью специфичных для каждой СУБД операторов, может устанавливаться автоматическая нумерация ключевого поля.

Ограничители записи также задаются посредством запрета неопределенных значений, но уже не для ключевых, но важных в семантическом смысле полей. Тогда, по крайней мере, в отношении, имеющим хотя бы одно обязательное к заполнению поле, не будет полностью пустых кортежей.

Ссылочная целостность обеспечивает поддержку непротиворечивого состояния БД (согласованного состояния внешних ключей) в процессе модификации данных, а также при выполнении операций добавления или удаления записей. В SQL для этого вводится оператор FOREIGN KEY

```
FOREIGN KEY(<список полей>) REFERENCES <имя  
таблицы>(<список полей>) ON DELETE CASCADE ON UPDATE  
CASCADE.
```

```
Например: FOREIGN KEY(client_id) REFERENCES clients(id) ON  
DELETE CASCADE
```

Ограничители ключей, записи и ссылочная целостность определяют правила работы СУБД с реляционными структурами данных. Но, с другой стороны, эти аспекты никак не касаются содержания базы данных. Для определения некоторых ограничений, которые связаны с содержанием базы данных, вводятся ограничители значений. Ограничители значений задаются путем определения типа поля (домена), задания условия на значение, задания списка возможных значений и определения значения по умолчанию. Значение по умолчанию задается с помощью оператора

```
DEFAULT(<значение или выражение>) {FOR <имя поля>}
```

А ограничители значений вводятся с помощью оператора CHECK(<условие>). Внутри CHECK могут использоваться операторы сравнения, функции IN, BETWEEN, LIKE и другие функции SQL.

Например:

```
DEFAULT(GetDate()) for date
CHECK(price between 0 and 100000)
CHECK(passport like 'MP%20')
CHECK(город in('Минск','Москва','Киев')) или
CHECK(город in(select capital from capitals))
```

Все рассмотренные выше ограничители целостности можно напрямую задать в инструкции определения таблицы CREATE TABLE

Вопросы для изучения

1. Создание ограничений и управление транзакциями.
2. Резервное копирования и восстановления баз данных.

Лабораторная работа №4

Примерный перечень запросов: Using Single-Row Functions to Customize Output

1. Таблица Employees. Получить список всех сотрудников, у которых длина имени больше 10 букв
2. Таблица Employees. Получить список всех сотрудников, у которых в имени есть буква 'b' (без учета регистра)
3. Таблица Employees. Получить список всех сотрудников, у которых в имени содержатся минимум 2 буквы 'a'
4. Таблица Employees. Получить список всех сотрудников, зарплата которых кратна 1000

5. Таблица Employees. Получить первое 3х значное число телефонного номера сотрудника если его номер в формате XXX.XXX.XXXX
6. Таблица Departments. Получить первое слово из имени департамента для тех, у кого в названии больше одного слова
7. Таблица Employees. Получить имена сотрудников без первой и последней буквы в имени
8. Таблица Employees. Получить список всех сотрудников, у которых последняя буква в имени равна 'm' и длиной имени большей 5ти
9. Таблица Dual. Получить дату следующей пятницы
10. Таблица Employees. Получить список всех сотрудников, которые работают в компании больше 17 лет
11. Таблица Employees. Получить список всех сотрудников, у которых последняя цифра телефонного номера нечетная и состоит из 3ех чисел, разделенных точкой
12. Таблица Employees. Получить список всех сотрудников, у которых в значении job_id после знака '_' как минимум 3 символа, но при этом это значение после '_' не равно 'CLERK'
13. Таблица Employees. Получить список всех сотрудников заменив в значении PHONE_NUMBER все '.' на '-'

Тема 5. ТЕХНОЛОГИИ И МОДЕЛИ АРХИТЕКТУРЫ КЛИЕНТ/СЕРВЕР

Основной принцип технологии «клиент—сервер» применительно к технологии баз данных заключается в разделении функций стандартного интерактивного приложения на 5 групп, имеющих различную природу:

- функции ввода и отображения данных (Presentation Logic);
- прикладные функции, определяющие основные алгоритмы решения задач приложения (Business Logic);
- функции обработки данных внутри приложения (Database Logic),
- функции управления информационными ресурсами (Database Manager System);
- служебные функции, играющие роль связок между функциями первых четырех групп.

Презентационная логика (Presentation Logic) как часть приложения определяется тем, что пользователь видит на своем экране, когда работает приложение. Сюда относятся все интерфейсные экранные формы, которые пользователь видит или заполняет в ходе работы приложения, к этой же части относится все то, что выводится пользователю на экран как результаты решения некоторых промежуточных задач либо как справочная информация. Поэтому основными задачами презентационной логики являются:

- формирование экранных изображений;
- чтение и запись в экранные формы информации;
- управление экраном;
- обработка движений мыши и нажатие клавиш клавиатуры.

Некоторые возможности для организации презентационной логики приложений предоставляет знако-ориентированный пользовательский интерфейс, задаваемый моделями CICS (Customer Control Information System) и IMS/DC фирмы IBM и моделью TSO (Time Sharing Option) для централизованной main-фреймовой архитектуры. Модель GUI — графического пользовательского интерфейса, поддерживается в операционных средах Microsoft's Windows, Windows NT, в OS/2 Presentation Manager, X-Windows и OSF/Motif.

Бизнес-логика (Business processing Logic), — это часть кода приложения, которая определяет собственно алгоритмы решения конкретных задач приложения. Обычно этот код пишется с использованием различных языков программирования, таких как C, C++, Cobol, SmallTalk, Visual-Basic.

Логика обработки данных (Data manipulation Logic) — это часть кода приложения, которая связана с обработкой данных внутри приложения. Данными управляет собственно СУБД (DBMS). Для обеспечения доступа к данным используются язык запросов и средства манипулирования данными стандартного языка SQL

Обычно операторы языка SQL встраиваются в языки 3-го или 4-го поколения (3GL, 4GL), которые используются для написания кода приложения.

Процессор управления данными (Database Manager System Processing) — это собственно СУБД, которая обеспечивает хранение и управление базами данных. В идеале функции СУБД должны быть скрыты от бизнес-логики приложения, однако для рассмотрения архитектуры приложения нам надо их выделить в отдельную часть приложения.

В **централизованной архитектуре** (Host-based processing) эти части приложения располагаются в единой среде и комбинируются внутри одной исполняемой программы.

В **децентрализованной архитектуре** эти задачи могут быть поразному распределены между серверным и клиентским процессами. В зависимости от характера распределения можно выделить следующие **модели распределений**:

- распределенная презентация (Distribution presentation, DP);
- удаленная презентация (Remote Presentation, RP);
- распределенная бизнес-логика (Remote business logic, RBL);
- распределенное управление данными (Distributed data management, DDM);
- удаленное управление данными (Remote data management, RDA).

Вопросы для изучения

1. Анализ предметной области: определение требований к БД, сбор и анализ требований пользователей.
2. Проектирование архитектуры информационного приложения.

Лабораторная работа №5

Сценарий сдачи лабораторной работы: скрипты создания изменения и просмотра, демонстрирующие работу созданных представлений.

При отсутствии в БД полей соответствующего типа создайте их.

I. Задание

1. Заполнить БД осмысленными данными, позволяющими получать непустые «датасеты» в результате обращения к представлению
2. Создать представления по примеру для своей БД

II. Пример на БД институт

1. Объекты информационной системы:

- а) Сотрудники: ФИО, возраст, пол, адрес, телефон, паспортные данные, должность;
- б) «Должности»: наименование должности, оклад, обязанности, требования;
- в) «Кафедры»: наименование, описание, номер кабинета;
- г) «Направления»: наименование, код, описание;
- д) «Дисциплины»: наименование, преподаватель, кафедра, семестр, описание;
- е) «Студенты»: ФИО, дата рождения, пол, адрес, телефон, паспортные данные;
- ж) «Экзамены»: студент, дата сдачи, дата пересдачи, оценка, преподаватель.

2. Создать представления:

- а) «Список преподавателей»: ФИО преподавателя, наименование должности, количество дисциплин;
- б) «Список дисциплин»: наименование дисциплины, преподаватель, наименование направления, наименование кафедры, дата экзамена, ФИО студента, который сдал экзамен последним;
- в) «Список студентов-должников»: ФИО студента, адрес, телефон, количество долгов, среднее количество дней долга;
- г) «Список из 10 дисциплин, которые чаще всего не сдают в срок»: наименование дисциплины, преподаватель, количество долгов.

- 3. Отобразить горизонтальную и вертикальную проекцию каждого представления или агрегирующий запрос к представлению:**

Например, вывести ФИО всех доцентов.

- 4. В представлении “Список студентов” добавить отображение еще одного столбца “оклад”.**

- 5. Удалить представление “Список студентов-должников”.**

- 6. Допустим, кто-то из студентов уже работает в институте.**

Создать представление: (ФИО, возраст) как:

а) единый список всех людей, кто учится или работает в институте (и студентов и сотрудников без дублирования);

б) список сотрудников, которые учатся в институте;

в) список студентов, которые не работают в институте.

В п.6 необходимо реализовать запросы как операции над множествами.

Вопросы для изучения

- 1.** Создание Windows-форм и отчетов для приложений, использующих базы данных, средствами MS Visual Studio и MS SQL Server.

Лабораторная работа №6

Сценарий сдачи лабораторной работы: скрипты создания изменения и просмотра, демонстрирующие работу созданных представлений.

При отсутствии в БД полей соответствующего типа создайте их.

I. Задание

1. Создать заданные функции и процедуры;
2. Прописать комментарии в тексте; Комментарии должны быть написаны простым языком и не подтверждать очевидное. Например, «объявление массива» или «вызов функции» и т. д. Из комментария должно быть понятно, что именно делает код.
3. Прописать проверочные скрипты для демонстрации работы функций и процедур.

II. Задание

1. Необходимые функции:

- а) скалярная функция, возвращающая количество проведенных экзаменов преподавателя, за определенный период;
- б) табличная функция, возвращающая информацию о должниках, которые должны были сдать экзамен до заданной даты.

ФИО	Номер зачетки	Количество долгов	Список долгов через запятую	Дата первого долга	Дата последнего долга
Абсурдов	1735786	2	СУБД, РЗиМП, ТМП	01.07.2020	02.07.2020
...

2. Необходимые процедуры:

а) процедура записи сдачи экзамена:

- ✦ в качестве параметров передается ФИО, номер зачетки, предмет, дата и оценка
- ✦ значения заносятся в соответствующие таблицы;
- ✦ в сводную ведомость заносятся значения внешних ключей;

б) процедура, возвращающая информацию о долгах заданной группы и по заданному предмету в следующем формате:

Дисциплина	Преподаватель	Количество студентовдолжников	Последний (по дате должник)
СУБД	Желудкова О.В.	5	Абсурдов А.С.
...
итого: дисциплин 3		студентов 48	

Тема 7. РАСПРЕДЕЛЕННЫЕ БАЗЫ ДАННЫХ

Распределенная база данных — это набор отношений, хранящихся в разных узлах компьютерной сети и логически связанных таким образом, чтобы составлять единую совокупность данных.

Распределенная база данных предполагает хранение данных на нескольких узлах сети, обработку данных и их передачу между этими узлами в процессе выполнения запросов. Разбиение данных в распределенной базе данных может достигаться путем хранения различных таблиц на разных компьютерах или хранения разных фрагментов одной таблицы на разных компьютерах. Для пользователя (или прикладной программы) не должно иметь значения, каким образом распределены данные между компьютерами. Работа с распределенной базой данных должна осуществляться так же, как и с централизованной.

В основе распределенных ИС лежат две основные идеи:

- работа множества пользователей с общей БД;
- объединение распределенных данных на логическом и физическом уровнях в общей БД.

Перечислим основные принципы создания и функционирования распределенных БД:

- прозрачность размещения данных для пользователя (пользователю распределенная БД должна представляться точно так же, как и нераспределенная);
- изолированность пользователей друг от друга (на работу одного пользователя с БД не должна влиять работа других пользователей с ней);
- синхронизация БД и непротиворечивость состояния данных в любой момент времени.

Вопросы для изучения

1. Распределенные базы данных.
2. Система распределенных баз данных.

3. Узлы.
4. Распределенная система управления базами данных (РСУБД).
5. Однородность.
6. Преимущества распределенных хранилищ данных.
7. Примеры распределенных систем.
8. Основной принцип распределенных систем.

Лабораторная работа №7

Сценарий сдачи лабораторной работы: скрипты создания изменения и просмотра, демонстрирующие работу созданных представлений.

При отсутствии в БД полей соответствующего типа создайте их.

I. Задание

1. Создать заданные триггеры;
2. Прописать комментарии в тексте; Комментарии должны быть написаны простым языком и не подтверждать очевидное. Например, «объявление массива» или «вызов функции» и т. д. Из комментария должно быть понятно, что именно делает код.
3. Прописать проверочные скрипты для демонстрации работы триггеров.

II. Задание

1. Необходимые триггеры:

- a) триггер, автоматически выставляющий дату проведения консультации и дату проведения экзамена;
- b) триггер, запрещающий удалять предмет, по которому имеются должники;
- c) в таблице Студент есть поле “Номер зачетки”, а в таблице Ведомость поле “Последние 4 символа номера зачетки”; создать триггер, который при

изменении номера зачетки студента записывает последние три символа номера зачетки в таблицу ведомость;

d) триггер, проверяющий, что в поле «Адрес» обязательно содержится

название улицы;

e) триггер, фиксирующий операции создания, изменения структуры и удаления таблиц в базе данных. В отдельной таблице должны записываться тип операции, дата, системный пользователь;

f) в таблице Студент есть поле “Номер зачетки”, а в таблице Ведомость поле “Последние 4 символа номера зачетки”; создать триггер, который при изменении поля “Последние 4 символа номера зачетки” в ведомости перезаписывает их в поле “Номер зачетки” в таблице Студент.

Лабораторная работа №8

Сценарий сдачи лабораторной работы: Демонстрация ER-модели базы данных. Демонстрация интерфейса. Внесение данных. Изменение данных. Вывод данных

При отсутствии в БД полей соответствующего типа создайте их.

I. Задание

Необходимо создать клиент-серверное приложение, реализующее следующий функционал:

- 1) Пользовательский интерфейс ввода\вывода данных
- 2) Пользовательский интерфейс настройки прав доступа к системе
- 3) Обработка попытки ввода некорректных данных

Лабораторная работа №9

В рамках реализации функционала лабораторной работы №8 необходимо чтобы:

- 1) Все операции с данными в приложении реализовывались через хранимые процедуры (вставка, удаление, изменение)
- 2) Все операции по выводу данных на экран реализовывались через функции или представления MS SQL (или иной СУБД)
- 3) Через триггеры должен быть реализован функционал хранения истории операций с указанием пользователей

СПИСОК ЛИТЕРАТУРЫ

- 1. Стасышин, В.М. Проектирование информационных систем и баз данных:** учебное пособие / В.М. Стасышин. - Новосибирск: НГТУ, 2012. - 100 с. - ISBN 978–5-7782-2121-5; то же [Электронный ресурс]. - URL: <http://znanium.com/catalog/product/548234>. (дата обращения: 15.05.2020).
- 2. Шаньгин, В.Ф. Комплексная защита информации в корпоративных системах:** учеб. пособие / В. Ф. Шаньгин. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018. — 592 с.; [Электронный ресурс]. - URL: <http://znanium.com/catalog/product/937502>. (дата обращения: 15.05.2020).
- 3. Разработка и эксплуатация автоматизированных информационных систем:** учеб. пособие / Л. Г. Гагарина. — М.: ИД «ФОРУМ»: ИНФРА-М, 2018.; [Электронный ресурс]. - URL: <http://znanium.com/catalog/product/942717>. (дата обращения: 15.05.2020).

Интернет-ресурсы

1. Вузовские электронно-библиотечные системы учебной литературы;
2. Вузовские электронно-библиотечные системы учебной литературы;
3. База научно-технической информации ВИНТИ РАН;
4. Доступ к открытым базам цитирования, в т.ч. springer.com, scholar.google.com, mathnet.ru;
5. Среды разработки на языках C#, C++, MS Visual Studio;
6. Системы управления базами данных: MS SQL Server, InterBase/FireBird, MySQL, PostgreSQL;
7. Средство моделирования MS Office Visio.