**JDBC**

Цель работы

Получить общее представление о прикладном программном интерфейсе *JDBC* для соединения с СУБД.

Указания к работе

*JDBC* (*Java DataBase Connectivity*) – стандартный прикладной интерфейс (*API*) языка *Java* для организации взаимодействия между приложением и СУБД. Это взаимодействие осуществляется с помощью драйверов *JDBC*, обеспечивающих реализацию общих интерфейсов для конкретных СУБД и конкретных прото- колов. В *JDBC* определяются четыре типа драйверов:

* Драйвер, использующий другой прикладной интерфейс взаимодействия с СУБД, в частности *ODBC* (так называемый *JDBC*-*ODBC* – мост). Стандартный драйвер первого типа *sun*.*jdbc*.*odbc*.*JdbcOdbcDriver* входит в *JDK*.
* Драйвер, работающий через внешние (*native*) библиотеки (т.е. клиента СУБД).
* Драйвер, работающий по сетевому и независимому от СУБД протоколу с промежуточным *Java*-сервером, который, в свою очередь, подключается к нужной СУБД.
* Сетевой драйвер, работающий напрямую с нужной СУБД и не требующий установки *native*-библиотек.
* При запуске программы производится проверка существования таблиц, необходимых для работы программы, если таких таблиц не найдено – программа создает их и заполняет таблицы базовым количеством данных (данные для запронения таблицы брать и заранее созданного текстового файла).
* Все настройки для работы с базой данных хранятся в отдельном файле (.java, .xml или .property)
* После успешного запуска программа должна производить выборку данных из созданных таблиц(ы) и выводить полученный результат пользователю в виде таблицы
* В создаваемых таблицах должен быть уникальный идентификатор (id) с поддержкой автоматического инкремента.

**Последовательность действий при подключении:**

1.Загрузка класса драйвера базы данных при отсутствии экземпляра этого класса.

2.Установка соединения с БД. Для установки соединения с БД вызывается статический метод ***getConnection*()** класса ***DriverManager***.

3. Создание объекта для передачи запросов. После создания объекта *Connection* и установки соединения можно начинать работу с БД с помощью операторов *SQL*. Для выполнения запросов применяется объект ***Statement***, создаваемый вызовом метода ***createStatement*()** класса *Connection*.

4.Выполнение запроса. Результаты выполнения запроса помещаются в объект ***ResultSet***

5. Обработка результатов выполнения запроса производится методами интерфейса ***ResultSet***, где самыми распространенными являются ***next*()** и ***getString*(*int pos*)** а также аналогичные методы.

6.Закрытие соединения

*cn*.*close*();

Задания

В каждом из заданий необходимо выполнить следующие действия:

* Организацию соединения с базой данных вынести в отдельный класс, метод которого возвращает соединение.
* Создать БД. Привести таблицы к одной из нормированных форм.
* Создать класс для выполнения запросов на извлечение информации из БД с использованием компилированных запросов.
* Создать класс на добавление информации.
* Создать *HTML*документ с полями для формирования запроса.
* Результаты выполнения запроса передать клиенту в виде *HTML* документа.

**1.Файловая система**. В БД хранится информация о дереве каталогов файловой системы каталоги, подкаталоги, файлы. Для каталогов необходимо хранить:

* родительский каталог;
* название.

Для файлов необходимо хранить:

* родительский каталог;
* название;
* место, занимаемое на диске.

Определить полный путь заданного файла (каталога).

Подсчитать количество файлов в заданном каталоге, включая вложенные файлы и каталоги.

Подсчитать место, занимаемое на диске содержимым заданного каталога.

Найти в базе файлы по заданной маске с выдачей полного пути.

Переместить файлы и подкаталоги из одного каталога в другой.

Удалить файлы и каталоги заданного каталога.

**2.Видеотека.** В БД хранится информация о домашней видеотеке - фильмы, актеры, режиссеры. Для фильмов необходимо хранить:

* название;
* имена актеров;
* дату выхода;
* страну, в которой выпущен фильм.

Для актеров и режиссеров необходимо хранить:

* ФИО;
* дату рождения.

Найти все фильмы, вышедшие на экран в текущем и прошлом году.

Вывести информацию об актерах, снимавшихся в заданном фильме.

Вывести информацию об актерах, снимавшихся как минимум в *N* фильмах.

Вывести информацию об актерах, которые были режиссерами хотя бы одного из фильмов.

Удалить все фильмы, дата выхода которых была более заданного числа лет назад.

**3.Расписание занятий.** В БД хранится информация о преподавателях БГУИР и проводимых ими занятиях. Для предметов необходимо хранить:

* название;
* время проведения (день недели);
* аудитории, в которых проводятся занятия.

Для преподавателей необходимо хранить:

* ФИО;
* предметы, которые он ведет;
* количество пар в неделю по каждому предмету;
* количество студентов, занимающихся на каждой паре.

Вывести информацию о преподавателях, работающих в заданный день недели в заданной аудитории.

Вывести информацию о преподавателях, которые не ведут занятия в заданный день недели.

Вывести дни недели, в которых проводится заданное количество занятий.

Вывести дни недели, в которых занято заданное количество аудиторий.

Перенести первые занятия заданных дней недели на последнее место.

**4.Письма.** В БД хранится информация о письмах и отправляющих их людях. Для людей необходимо хранить:

* ФИО;
* дату рождения.

Для писем необходимо хранить:

* отправителя;
* получателя;
* тему письма;
* текст письма;
* дату отправки.

Найти пользователя, длина писем которого наименьшая.

Вывести информацию о пользователях, а также количестве полученных и отправленных ими письмах.

Вывести информацию о пользователях, которые получили хотя бы одно сообщение с заданной темой.

Вывести информацию о пользователях, которые не получали сообщения с заданной темой.

Направить письмо заданного человека с заданной темой всем адресатам.

**5.Сувениры.** В БД хранится информация о сувенирах и их производителях. Для сувениров необходимо хранить:

* название;
* реквизиты производителя;
* дату выпуска;
* цену.

Для производителей необходимо хранить:

* название;
* страну.

Вывести информацию о сувенирах заданного производителя.

Вывести информацию о сувенирах, произведенных в заданной стране.

Вывести информацию о производителях, чьи цены на сувениры меньше заданной.

Вывести информацию о производителях заданного сувенира, произведенного в заданном году.

Удалить заданного производителя и его сувениры.

**6.Заказ.** В БД хранится информация о заказах магазина и товарах в них. Для заказа необходимо хранить:

* номер заказа;
* товары в заказе;
* дату поступления.

Для товаров в заказе необходимо хранить:

* товар;
* количество.

Для товара необходимо хранить:

* название;
* описание;
* цену.

Вывести полную информацию о заданном заказе.

Вывести номера заказов, сумма которых не превосходит заданную, и количество различных товаров равно заданному.

Вывести номера заказов, содержащих заданный товар.

Вывести номера заказов, не содержащих заданный товар и поступивших в течение текущего дня.

Сформировать новый заказ, состоящий из товаров, заказанных в текущий день.

Удалить все заказы, в которых присутствует заданное количество заданного товара.

**7.Продукция.** В БД хранится информация о продукции компании. Для продукции необходимо хранить:

* название;
* группу продукции (телефоны, телевизоры и др.);
* описание;
* дату выпуска;
* значения параметров.

Для групп продукции необходимо хранить:

* название;
* перечень групп параметров (размеры и др.).

Для групп параметров необходимо хранить:

* название;
* перечень параметров.

Для параметров необходимо хранить:

* название;
* единицу измерения.

Вывести перечень параметров для заданной группы продукции.

Вывести перечень продукции, не содержащий заданного параметра.

Вывести информацию о продукции для заданной группы.

Вывести информацию о продукции и всех ее параметрах со значениями.

Удалить из базы продукцию, содержащую заданные параметры.

Переместить группу параметров из одной группы товаров в другую.

**8.Погода.** В БД хранится информация о погоде в различных регионах. Для погоды необходимо хранить:

* регион;
* дату;
* температуру;
* осадки.

Для регионов необходимо хранить:

* название;
* площадь;
* тип жителей.

Для типов жителей необходимо хранить:

* название;
* язык общения.

Вывести сведения о погоде в заданном регионе.

Вывести даты, когда в заданном регионе шел снег и температура была ниже заданной отрицательной.

Вывести информацию о погоде за прошедшую неделю в регионах, жители которых общаются на заданном языке.

Вывести среднюю температуру за прошедшую неделю в регионах с площадью больше заданной.

**9.Магазин часов.** В БД хранится информация о часах, продающихся в магазина. Для часов необходимо хранить:

* марку;
* тип (кварцевые, механические);
* цену;
* количество;
* реквизиты производителя.

Для производителей необходимо хранить:

* название;
* страну.

Вывести марки заданного типа часов.

Вывести информацию о механических часах, цена на которые не превышает заданную.

Вывести марки часов, изготовленных в заданной стране.

Вывести производителей, общая сумма часов которых в магазине не превышает заданную.

**10.Города.** В БД хранится информация о городах и их жителях. Для городов необходимо хранить:

* название;
* год основания;
* площадь;
* количество населения для каждого типа жителей.

Для типов жителей необходимо хранить:

* город проживания;
* название;
* язык общения.

Вывести информацию обо всех жителях заданного города, разговаривающих на заданном языке.

Вывести информацию обо всех городах, в которых проживают жители выбранного типа.

Вывести информацию о городе с заданным количеством населения и всех типах жителей, в нем проживающих.

Вывести информацию о самом древнем типе жителей.

**11. Планеты.** В БД хранится информация о планетах, их спутниках и галактиках. Для планет необходимо хранить:

* название;
* радиус;
* температуру ядра;
* наличие атмосферы;
* наличие жизни;
* спутники.

Для спутников необходимо хранить:

* название;
* радиус;
* расстояние до планеты.

Для галактик необходимо хранить:

* название;
* планеты.

Вывести информацию обо всех планетах, на которых присутствует жизнь, и их спутниках в заданной галактике.

Вывести информацию о планетах и их спутниках, имеющих наименьший радиус и наибольшее количество спутников.

Вывести информацию о планете, галактике, в которой она находится, и ее спутниках, имеющей максимальное количество спутников, но с наименьшим общим объемом этих спутников.

Найти галактику, сумма ядерных температур планет которой наибольшая.

**12.Точки.** В БД хранится некоторое конечное множество точек с их координатами.

Вывести точку из множества, наиболее приближенную к заданной.

Вывести точку из множества, наиболее удаленную от заданной.

Вывести точки из множества, лежащие на одной прямой с заданной прямой.

**13.Треугольники.** В БД хранятся треугольники и координаты их точек на плоскости.

Вывести треугольник, площадь которого наиболее приближена к заданной.

Вывести треугольники, сумма площадей которых наиболее приближена к заданной.

Вывести треугольники, которые помещаются в окружность заданного радиуса.

**14.Словарь.** В БД хранится англорусский словарь, в котором для одного английского слова может быть указано несколько его значений и наоборот. Со стороны клиента вводятся последовательно английские (русские) слова. Для каждого из них вывести на консоль все русские (английские) значения слова.

**15.Словари.** В двух различных базах данных хранятся два словаря: русско-белорусский и белорусско-русский. Клиент вводит слово и выбирает язык. Вывести перевод этого слова.

**16.Стихотворения.** В БД хранятся несколько стихотворений с указанием автора и года создания. Для хранения стихотворений использовать объекты типа *Blob*. Клиент выбирает автора и критерий поиска.

В каком из стихотворений больше всего восклицательных предложений?

В каком из стихотворений меньше всего повествовательных предложений?

Есть ли среди стихотворений сонеты и сколько их?

**17.Четырехугольники.** В БД хранятся координаты вершин выпуклых четырехугольников на плоскости.

Вывести координаты вершин параллелограммов.

Вывести координаты вершин трапеций.

**18.Треугольники.** В БД хранятся координаты вершин треугольников на плоскости.

Вывести все равнобедренные треугольники.

Вывести все равносторонние треугольники.

Вывести все прямоугольные треугольники.

Вывести все тупоугольные треугольники с площадью больше заданной.