## **Методические рекомендации к проведению лабораторных работ по основам баз данных и системам управления базами данных с использованием языка структурированных запросов в среде MS Access и консоли IntеrBase/Firebird на основе реализации клиент-серверных технологий**

*Щепакина Т.Е., к.п.н.*

*Институт информатизации*

*образования РАО*

Одно из центральных мест в курсе «Информатика и информационно-коммуникационные технологии» занимают темы, связанные с обучением школьников основам баз данных и систем управления базами данных (СУБД). При изучении этих тем у школьников формируется целостное видение основных понятий информатики, принципов функционирования информационных систем, основных принципов обработки информации с использованием языка структурированных запросов, взаимосвязи разнообразных тем школьного курса информатики.

Определим *содержание обучения* базам данных и СУБД на основе клиент-серверных технологий, состоящее из двух частей: инвариантной и вариативной составляющих. *Инвариантная* часть содержания обучения может быть ориентирована на формирование у учащихся знаний в области: основных понятий баз данных и СУБД; проектирования реляционных баз данных; структурирования информации в реляционных базах данных; работы в среде СУБД с графическим интерфейсом (на примере MS Access). В *вариативной* части содержания обучения базам данных и СУБД на основе клиент-серверных технологий могут рассматриваться: языковые средства СУБД, вопросы проектирования запросов к базам данных, программирование в среде СУБД, построенной на командном интерфейсе и т.д. (на примере консоли СУБД InterBase). При этом реализация вариативной части обучения предполагает существование условий профильной дифференциации обучения учащихся. Особенностью вариативной части обучения является работа с клиент-серверными технологиями и привлечение учеников к проектной деятельности, результатом которой может стать в достаточной мере функциональная база данных с прикладным содержанием.

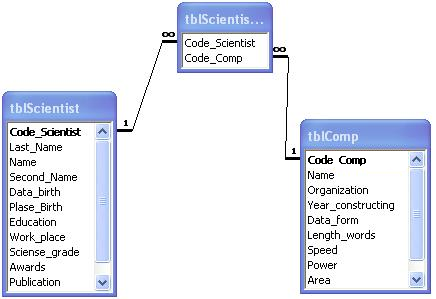
*Инвариантная часть* содержания обучения может содержать 8 лабораторных работ, рассчитанных на 12 учебных часов. Каждая из лабораторных работ имеет единую структуру: тему, цель, план работы, теоретические сведения к изучению данной темы, перечень заданий для практического выполнения, контрольные вопросы. Среди предложенных практических заданий есть задания, требующие выполнения пошаговых алгоритмов, а также задания для самостоятельного выполнения, целью которых является закрепление знаний и приобретение навыков работы с базами данных в предложенных СУБД (MS Access, InterBase).

Во время выполнения лабораторных работ ученики должны самостоятельно (посредством СУБД MS Access) моделировать, создавать функциональную базу данных по определенной тематике (предлагается наполнение, связанное с историей отечественной вычислительной техники или др.) и уметь работать с ней в дальнейшем. Приведем структуру обучения основам баз данных и СУБД на примере базы данных «ComputingHistory», содержащую сведения об отечественных ЭВМ XX столетия.

В лабораторной работе № 1 *«Понятие об объектах и их представлении в виде моделей. Представление модели в форме совокупности данных и связей между ними. Понятие базы данных и системы управления базами данных (СУБД). Интерфейс СУБД MS Access»* ученикам предлагается ознакомиться с основными понятиями, относящимися к базам данных, ознакомиться со средой СУБД MS Access, рассмотреть объекты баз данных (на примере учебных баз данных «Борей» и Crafts), осознать необходимость применения на практике баз данных.

В лабораторной работе №2 *«Создание структуры реляционной базы данных. Ввод и работа с данными в режиме «Таблица»»* предлагается пошаговое создание базы данных с тремя таблицами. Для этого ученикам необходимо: во-первых, ознакомиться с предложенными в теоретической части понятиями систем баз данных, их предназначением, основными понятиями системы управления базами данных; во-вторых, научиться создавать таблицы баз данных в режиме конструктора, усвоить переход из режима конструктора к режиму таблицы; в-третьих, усвоить основные приемы заполнения и редактирования таблиц баз данных, а также научиться пользоваться средствами поиска, сортировки данных, устанавливать фильтры.

Важным при создании полнофункциональной учебной базы данных является наличие связей между таблицами в базах данных. Целью лабораторной работы №3 *«Связи между таблицами как отображение связей между параметрами реального мира (его модели). Создание связей между таблицами в СУБД MS Access»* является изучение вопроса установления связей между таблицами, осознание необходимости установления подобных связей, обретения знаний относительно возможности и необходимости обеспечения целостности данных. В базе данных «ComputingHistory» ученикам предлагается установить связи «многие-ко-многим» между таблицами Comp и Scientist за счет создания дополнительной связывающей таблицы Scientist\_Comp (см. рис. 1).



*Рис. 1. Схема данных*

В лабораторной работе № 4 *«Конструирование форм. Редактирование данных в форме*» учитель предлагает ученикам ознакомиться с новым объектом СУБД – формой, научиться создавать формы в режиме мастера, а также редактировать данные в режиме формы, научиться работать с графическими объектами. Работа по оформлению внешнего вида форм и подготовке графических объектов, которые будут на ней размещаться, требует творческого подхода от учеников. В процессе работы ученики имеют возможность проявить свои индивидуальные способности.

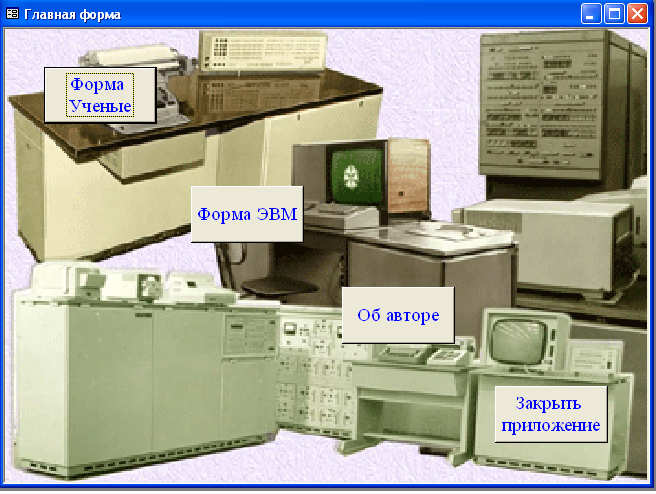
Для удобства работы с формами ученикам предлагается добавить на форме элементы управления (лабораторная работа №5 *«Работа по отладке форм. Создание элементов управления на форме*»): поле со списком (для быстрого выбора и перехода по записям в базе данных); управляющую кнопку с названием «Закрыть форму». Тематика лабораторных работ не является традиционной и их выполнение позволяет ученикам сконструировать более функциональную, практически применимую базу данных и будет способствовать активизации их творческой и самостоятельной деятельности.

Учителю следует уделить особое внимание работе по созданию и использованию запросов. В лабораторной работе №6 *«Создание запросов и работа с ними в СУБД»* ученики создают простые запросы (например, на отображение всех записей из таблицы Comp с определенными полями); создают сложные запросы (например, на выборку фамилий и кодов ученых, принимавших участие в создании определенной ЭВМ); проводят работу по ознакомлению с построением более сложных запросов с помощью структурированного языка запросов SQL (например, по созданию запроса для выборки записей об ЭВМ, в разработке которых принимали участие определенные ученые и установления ограничения на повторяемые записи таблицы разработок Scientist\_Comp).

Целью лабораторной работы №7 *«Работа с элементами управления. Подчиненная форма»* является формирование навыков по созданию сложной формы (с размещением на основной форме еще и подчиненной формы), ознакомление с возможностью обработки событий с помощью VBA (Visual Basic for Application), формирование навыков установления связей между записями разных таблиц с использованием таких элементов управления, как поле со списком. Обучение основам баз данных и СУБД расширяется за счет размещения на основной форме подчиненной формы и связывания ее с запросами, сформулированными в процессе лабораторной работы №6.

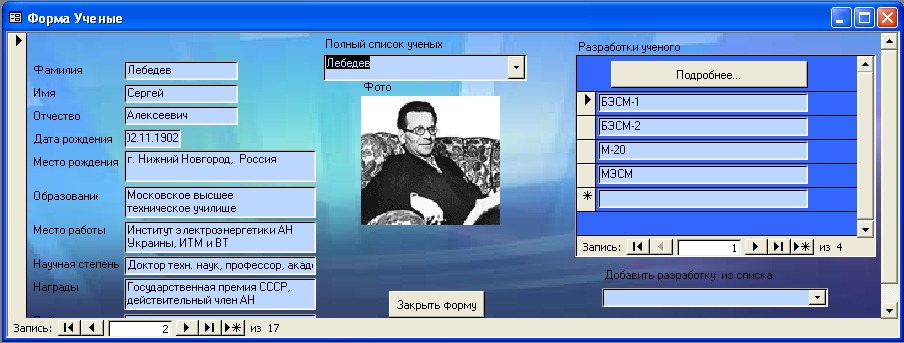
В лабораторной работе №8 *«Создание и отладка главной формы базы данных»* ученикам предлагается самостоятельно создать и настроить главную форму базы данных для перехода к формам Scientist, Comp или Author. На главной форме также должна размещаться кнопка, позволяющая закрыть приложение MS Access. Форма Author может быть создана по собственному желанию и содержать информацию об авторе-разработчике созданной базы данных. Таким образом, ученики самостоятельно закрепляют навыки по созданию форм, созданию управляющих кнопок на форме, используя инструментарий MS Access.

На последнем занятии ученики должны продемонстрировать учителю результат работы по созданию базы данных, показать работу элементов управления, возможность перехода от формы с данными одной таблицы к другой форме. На рисунке 2 представлен вариант результата работы по созданию главной формы.



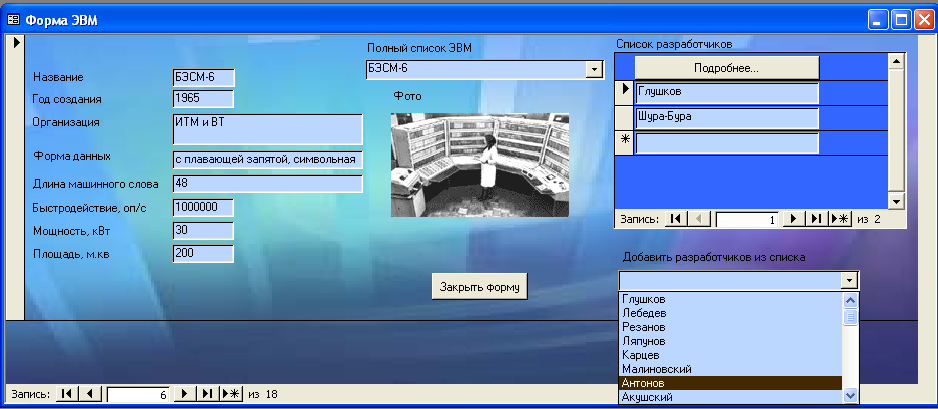
*Рис. 2. Главная форма базы данных «ComputingHistory»*

На рисунке 3 представлен вариант результата работы по созданию формы со сведениями об ученых.



*Рис. 3. Форма со сведениями об ученых*

На рисунке 4 представлен вариант результата работы по созданию формы со сведениями об отечественных ЭВМ.



*Рис. 4. Пример формы со сведениями об отечественных ЭВМ*

Одной из главных особенностей предложенной инвариантной части обучения базам данных является акцентирование внимания учащихся на содержательном наполнении баз данных (сведения о разработках отечественной вычислительной техники ХХ века и биографические сведения выдающихся деятелей науки того времени). Лабораторные работы могут быть полезны не только для осуществления учениками информационных процессов во время работы с базами данных, но и для открытия новых сведений, стимулирования познавательной деятельности.

Для формирования у учеников необходимых знаний, умений и навыков работы с СУБД, отличных от MS Access, учителю необходимо сосредоточить внимание учеников не только на изучении особенностей интерфейса конкретной СУБД, но и на формирование глубокого понимания сущности процессов обработки данных, что отражено в вариативной части обучения. Работа с вариативной частью содержания обучения возможна за счет использования резервных часов для осуществления профильного обучения или для изучения во время факультативных занятий.

*Вариативная часть***,** состоящая из 6 лабораторных работ, является продолжением работы по созданию базы данных «ComputingHistory» и рассчитана на 12 учебных часов. Расширение инвариантной части осуществляется благодаря возможностям использования клиент-серверных технологий при работе с СУБД InterBase/FireBird.

Вариативная часть обучения начинается с ознакомления с предназначением и возможностями консоли СУБД InterBase и самостоятельного установления на рабочий компьютер InterBase 6 Server, а также с создания пустой базы данных.

В лабораторной работе №9 *«Работа с системой баз данных InterBase. Создание структуры базы данных. Ознакомление с интерфейсом InterBase Console»* ученикам предлагается создать пустую базу данных. Для создания структуры базы данных ученики должны предварительно ознакомиться с синтаксисом языка определения данных ЯОД или DDL (Data Definition Language).

Перечислим SQL-операторы *языка определения данных**(ЯОД): CREATE TABLE (с*оздать таблицу), *DROP TABLE (*удалить таблицу*), ALTER TABLE* (изменить структуру таблицы); *CREATE INDEX* (создать индекс); *DROP INDEX (*удалить индекс); *ADD PRIMERY KEY* (добавить первичный ключ); *ADD FOREIGN KEY (*установить связь между таблицами).

Рассмотрим некоторые примеры использования возможностей операторов*языка определения данных*, указанных в приведенном выше опорном конспекте. Обучаемому следует начать работу с базой данных с создания структуры таблицы.

*Пример 1.* Оператор создания таблицы *Scientist*, имеющей поля: *Code\_Scientist* – идентификатор ученого, *Last\_Name* – фамилия ученого, *Name* – имя ученого, *Second\_Name* – отчество, *Data\_Birth* – дата рождения, *Place\_Birth –* место рождения*; Education* − образование, *Awards* – отличия и вознаграждения ученого, *Publications –* количество публикаций, *Photo -* фото может иметь вид:

|  |  |
| --- | --- |
| /\* Table: Scientist\*/ |  |
| *CREATE TABLE Scientist* | /\*Создать таблицу с именем *Scientist*, содержащую следующие поля: \*/ |
| *(* |  |
| *Code\_Scientist INTEGER NOT NULL,* | /\**Code\_Scientist*, тип целый, не может быть пустым (не иметь никакого значения)\*/ |
| *Last\_Name VARCHAR(25),* | /\* *Last\_Name* буквенного типа длиной 25 символов \*/ |
| *Name VARCHAR(20),* | /\* *Name* буквенного типа длиной 20 символов \*/ |
| *Second\_Name VARCHAR(20),* | /\* *Second\_Name* буквенного типа длиной 20 символов \*/ |
| *Data\_Birth TIMESTAMP,* | /\* *Data\_Birth* типа дата/время\*/ |
| *Place\_Birth VARCHAR(100),* | /\**Place\_Birth* буквенного типа длиной 100 символов\*/ |
| *Education VARCHAR(100),* | /\* *Education* буквенного типа длиной 20 символов \*/ |
| *Scientist\_Grade VARCHAR(60),* | /\* *Scientist\_Grade* буквенного типа длиной 20 символов \*/ |
| *Awards VARCHAR(100 ,* | /\* *Awards* буквенного типа длиной 20 символов \*/ |
| *Publications VARCHAR(50),* | /\* *Publications* буквенного типа длиной 20 символов \*/ |
| *Photo BLOB* | /\* *Photo*, тип – двоичный объект (Binary Large Object)\*/ |
| *PRIMARY KEY (Code\_Scientist)* | /\*создать первичный ключ по полю *Code\_Scientist* \*/ |
| *);* |  |

*Пример 2.* Рассмотрим использование оператора ALTER для изменения структуры отношений, в данном случае, установление внешнего ключа для таблицы SCIENTIST\_COMP:

*ALTER TABLE Scientist\_Comp* /\*Изменить таблицу *Scientist\_Comp* \*/

*ADD Foreign KEY (Code\_Scientist)* /\*добавить внешний ключ по полю CODE\_SCIENTIST \*/

*REFERENCES SCIENTIST (Code\_Scientist*) /\*значения в данном поле могут быть только такими, которые содержат поле *Code\_Scientist* таблицы *Scientist* \*/ ;

*Пример 3.* Пусть в созданной прежде таблице *Scientist* следует добавить поле *Awards*, предназначенное для сохранения данных о поощрениях и вознаграждениях данного ученого. Для этого следует записать оператор вида:

*ALTER TABLE Scientist*  /\*Изменить таблицу *Scientist* *\*/*

*(ADD Awards VARCHAR(100)) /\**добавить поле с именем *Awards буквенного типа, максимальный размер до 100\*/.*

*Пример 4.* Оператор создание индекса *main\_index* для таблицы *SCIENTIST,* которыйбудет сортировать фамилии в алфавитном порядке и по хронологии рождения, может иметь вид:

*CREATE INDEX main\_index* /\*Создать индекс *main\_index \*/*

*ON Scientist* *(LAST\_NAME, DATA\_BIRTH DESC)* /\*для таблицы *SCIENTIST* по указанным полям в порядке убывания*\*/.*

В лабораторной работе №10 *«Заполнение данными пустой базы, созданной InterBase, с использованием скриптов»* ученикам предлагается выполнение конвертирования данных с помощью использования предварительно подготовленных учителем скриптов, содержащих структуру каждой из трех таблиц базы данных. Ученикам необходимо зарегистрироваться на удаленном сервере, что в дальнейшем обеспечит возможность работы с клиент-серверными технологиями. Следует обязательно проверить наличие первичных и внешних ключей с целью правильного установления связей между таблицами базы данных «ComputingHistory».

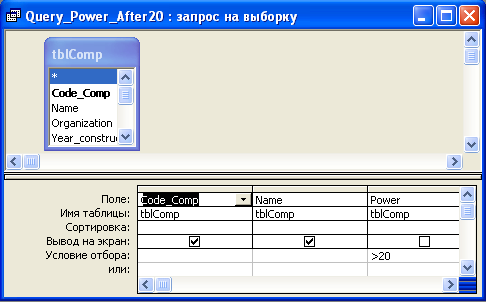
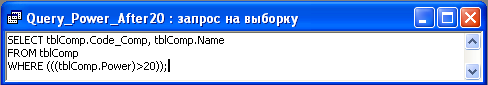
Целью лабораторной работы №11 *«Работа с реляционными базами данных. Реляционные объекты данных. Основные операции реляционной алгебры»* является закрепление знаний о работе с реляционными базами данных на примере таблиц Comp и Scientist, ознакомление с основными операциями реляционной алгебры. Это обусловлено необходимостью осознания внутренних механизмов работы базы данных, фундаментом которых являются операции реляционной алгебры. В данной лабораторной работе на самостоятельное выполнение предлагается большинство заданий из инвариантной части методики.

В лабораторной работе №12 *«Язык структурированных запросов SQL как основа работы с реляционными базами данных. Обработка данных с помощью операции выборки SELECT»* предоставляются теоретические сведения, позволяющие ученикам ознакомиться с историей возникновения, предназначением и возможностями структурированного языка запросов SQL. Целью выполнения лабораторной работы является формирование у школьников навыков работы с базой данных посредством использования операции SELECT, а также инструкций FROM, WHERE, GROUP BY, HAVING, которые входят в состав языка манипулирования данными ЯМД или DML (Data Manipulation Language). Ученикам предлагаются примеры построения некоторых SQL-запросов и достаточное количество заданий для самостоятельного решения средствами InterBase за счет построения простых, сложных запросов, использования подзапросов.

Одной из проблем обучения основам баз данных в школьном курсе информатики является формальный подход к изучению данной. В большинстве существующих СУБД имеются встроенные интерфейсы, в которых пользователь явным образом не использует операции структурированного языка запросов, например, операции SELECT, являющейся основой любой выборки данных.

Для работы с базой данных пользователь может выбрать необходимые команды (подпункты) меню или заполнять поля в формах. По результатам обучения работе с базами данных средствами СУБД MS Access большинство учеников не имеет представления о существовании языка структурированных запросов SQL (Structured Query Language) − стандартного средства доступа к удаленным базам данных, обеспечивающего управление структурой баз данных и манипулирование данными.

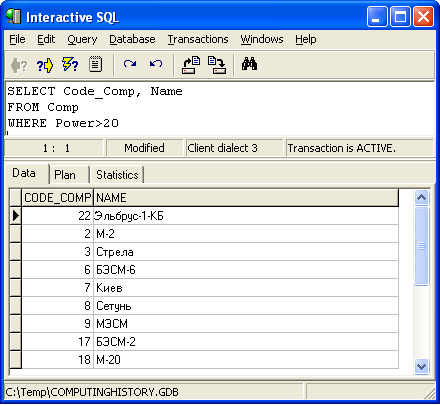
Учитель может предложить ученикам сравнить составление SQL-запроса средствами СУБД MS Access и СУБД InterBase. На рисунке 5 приведено составление простого запроса посредством «Конструктора запросов» и его структуры в виде SQL с помощью графического интерфейса СУБД MS Access.



*Рис. 5. Пример реализации запросов в MS Access (запрос в режиме Конструктора и SQL)*

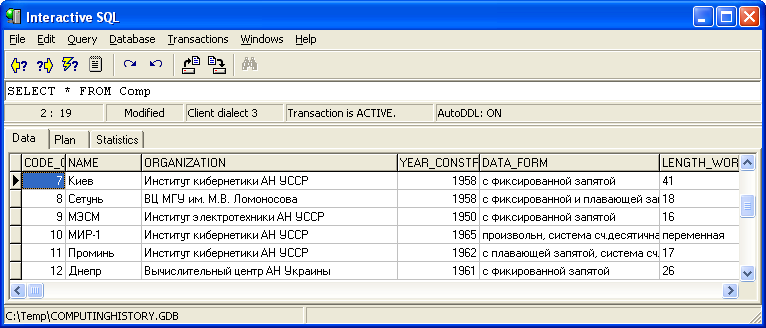
На рисунке 6 приведен тот же запрос, организованный посредством командного интерфейса в СУБД InterBase.

*Рис.6. Построение SQL-запроса в InterBase Console*



Таким образом, учитель на сравнительном примере может продемонстрировать ученикам, что синтаксис запроса в режиме SQL достаточно сложный, содержит избыточные квадратные скобки и некоторые лишние параметры, что доказывает невысокую эффективность работы SQL в MS Access. MS Access имеет интерфейс, основанный на меню и формах, что обусловлено необходимостью перестрахования всех неквалифицированных действий пользователей.

Учителю следует подкреплять обучение основам языка структурированных запросов достаточным количеством примеров на применение *операции выборки* (SELECT). Операция выборки в SQL − это табличное выражение, которое может быть достаточно сложным. Выражение выборки может содержать несколько компонентов: инструкции SELECT, FROM, WHERE, Group By, HAVING. Учитель может привести пример выборки данных на получение полной информации обо всех ЭВМ средствами SQL и проиллюстрировать выполнение запроса в консоли InterBase.



*Рис. 7. Результат выполнения предложенного запроса в консоли InterBase*

Понимание работы с базами данных формируется на этапе создания запросов с вложенными подзапросами, что отображает нелинейный принцип обработки информации, когда выборка выполняется по нескольким таблицам. Неподдельный интерес у учеников могут вызвать задания на нахождение неочевидного и непредсказуемого заранее ответа. Целесообразным будет приведение учителем примеров составления запросов с использованием подзапросов.

*Пример 1.* Для получения имен ученых, принимавших участие в разработке БЭСМ-6 с уникальным номером 6 можно составить запрос:

*SELECT DISTINCT Last\_Name*

*FROM Science*

*WHERE Code\_Science IN*

*(SELECT Code\_Science*

*FROM Science\_Comp*

*WHERE Code\_Comp=6)*

При этом подзапрос *SELECT Code\_Scientist FROM Scientist\_Comp WHERE Code\_Comp=6* используется для представления множества значений, поиск которых осуществляется с помощью инструкции *IN Condition* (*Condition* – «условие») система вычисляет полностью запрос, выполнив вычисление подзапроса. Предложенное задание можно решить также с помощью операции соединения:

*SELECT s.Last\_name*

*FROM Scientist s, Science\_comp sc*

*WHERE s.code\_Scientist=sc.code\_Scientist*

*AND sc.code\_Comp=6*

*Пример 2.* Запрос на получение фамилий всех ученых, которые не принимали участие в разработке БЭСМ-6, занесенной в базы данных ComputingHistory под уникальным номером 6, будет иметь следующую структуру:

*SELECT DISTINCT Last\_Name*

*FROM Scientist*

*WHERE Code\_Scientist NOT IN*

*(SELECT Code\_Scientist*

*FROM Scientist\_Comp*

*WHERE Code\_Comp=6)*

*Пример 3.* Для получения названия ЭВМ, разработкой которых руководил ученый с уникальным номером 1, можно сформулировать следующий запрос:

*SELECT c.Name*

*FROM Comp c, Scientist\_Comp sc*

*WHERE c.Code\_Comp=sc.Code\_Comp AND*

*Sc.Code\_Science=1*

Для получения названия ЭВМ, разработкой которых руководил Глушков В.М., запрос в консоли InterBase будет иметь синтаксис, представленный на рисунке 8.



*Рис. 8. SQL-запрос и результат его выполнения в консоли InterBase*

*Пример 4.* Получить номера всех ЭВМ, которые имеют быстродействие больше 5 000 операций в секунду или были разработаны под руководством И.С. Брука (в таблице SCIENTIST занесен под уникальным номером 12), или те и другие. Для получения данной информации можно составить такой запрос:

*SELECT Code\_Comp*

*FROM Comp*

*WHERE Speed>5000*

*UNION*

*SELECT Code\_Comp*

*FROM Scientist\_Comp*

*WHERE Code\_Scientist=12*

Следует заметить, что лишние повторяемые строки всегда выключаются из результата безусловных операторов UNION, INTERSECT или EXCEPT (оператор EXCEPT в языке SQL есть аналогом операции MINUS реляционной алгебры).

Работа с языком SQL определяется моделированием анализа и обработки данных, получения новой информации на основе запросов, не очевидной заранее. В качестве примера приведем SQL-запрос на получение номеров для всех ЭВМ, разработкой которых руководили более чем один ученый из приведенных в таблице ученых (Scientist).

*Пример 5.* Получитьномера для всех ЭВМ, разработкой которых руководили более чем один ученый из приведенных в таблице ученых (Scientist).

*SELECT Code\_Comp; /\** выбери значение *Code\_Comp* \*/

*FROM Scientist\_Comp; /\**изтаблицы «*Scientist\_Comp*»\*/

*GROUP BY Code\_Comp;* /\*сгруппированные по совпадающим значениям поля Code\_ *Scientist* \*/

*HAVING COUNT (Code\_Scientist*)>1/\*которые встречаются в таблице чаще 1 раза \*/

Результатом данной выборки будет перечень номеров ЭВМ, имеющих более одного из авторов, перечисленных в таблице *Scientist*. Эта информация не была доступна в явном виде даже на этапе заполнения базы. Для каждого оператора в предложенном и в последующих примерах запишем в качестве примечания команду на русском языке.

*Пример 6.* Получить номера ЭВМ, ученые-разработчики которых обучались в Московском энергетическом институте.

*SELECT Code\_Comp /\**Выбрать значение в поле *Code\_Comp \*/*

*FROM Scientist \_Comp /\**из таблицы Scientist \_Comp *\*/*

*WHERE code\_Scientist in* /\*где значения поля code\_Scientist находятся в следующем множестве значений: \*/

*(SELECT Code\_Scientist /\**Выбрать значение поля C*ode\_Scientist \*/*

*FROM Scientist /\**из таблицы *Scientist \*/*

*WHERE Education*=’ Московский энергетический институт ’)/\*для записей, в которых поле Education равняется указанному значению \*/.

В лабораторной работе №13 *«Ознакомление с клиент-серверными системами. Система баз данных InterBase. Работа по редактированию базы данных, расположенной на удаленном сервере»* ученикам предлагается согласовать структуру, понятия и возможности, которые открывает работа с клиент-серверными системами.

При выполнении данной работы ученики должны работать с базой данных, расположенной на сервере, дополняя ее новыми сведениями. Ученики самостоятельно выбирают источники информации (Интернет, периодическая литература, справочные материалы и др.) и, не ограничиваясь временными границами урока, могут вносить соответствующие сведения в базу данных «ComputingHistory», расположенную на сервере, работая на любом подключенном к локальной сети компьютере. Таким образом, подобная работа не только формирует знания и навыки использования операций структурированного языка запросов SQL в клиент-серверных системах, но и способствует формированию как внешней, так и внутренней мотивации, эмоциональной заинтересованности обучением во время выполнения проектной деятельности.

При работе с клиент-серверными технологиями учителям необходимо ознакомить обучаемых со стандартами системы «клиент-сервер», уделить внимание *операциям языка манипулирования данных**(ЯМД)* DML, не использующим курсор, среди которых можно выделить *UPDATE* (изменить запись); *INSERT* (добавить новую запись); *DELETE* (удалить запись). Учителю следует подкреплять обучение основам языка структурированных запросов достаточным количеством примеров на применение операций языка обработки данных при работе с клиент-серверными технологиями.

*Пример 1.* Вставьте строку в таблицу *Comp*, содержащую информацию про *ЭВМ МИР-2.*

Для того чтобы вставить строку в таблицу *Comp*, содержащую информацию про *ЭВМ МИР-2,* необходимо составить запрос:

*INSERT* /\*Вставить\*/

*INTO COMP (Code\_Comp, Name, Organization, Year\_Constructing, Data\_Form, Speed, Power,Squares)* /\*в таблицу *Comp* новую запись с  указанием порядка следования полей ... \*/

*VALUES (20, ‘МИР-2’, ‘Институт кибернетики АН УССР’, 1969, ‘с плавающей запятой’, 20 000, 5, 20)* /\*с перечнем значений полей для новой записи ...\*/.

*Пример 2.* Вставьте несколько строк в дополнительно созданную таблицу TEMP, которые будут содержать данные об уникальном номере, названии и быстродействии для ЭВМ, площадь которых, необходимая для установки, меньше чем 80 кв. м.

Для того чтобы вставить необходимые строки в дополнительно созданную таблицу TEMP, составим следующий SQL-запрос:

*INSERT* /\*Вставить\*/

*INTO TEMP (Code\_Comp, Name, Power) /\**во временно созданную таблицу TEMP со следующими полями ....*\*/*

*SELECT Code\_Comp, Name, Power /\**с выборкой следующих полей ... *\*/*

*FROM Comp /\** из таблицы *Comp*\*/

*WHERE Squares<80/ \**при условии, что площадь, необходимая для установки, меньше 80 кв.м *\*/*

*Пример 3.* Замените значение поля «Форма данных» для ЭВМ «МЭСМ» на значение «с фиксированной запятой».

Запрос на замену значения формы данных для ЭВМ МЭСМ на значение «с фиксированной запятой» может быть следующим:

*UPDATE COMP*  /\*Обновить данные в таблице *Comp* \*/

*SET Data\_Form=’с фиксированной запятой’ /\**Установить в поле *DATA\_FORM* указанное значение ....*\*/*

*WHERE Name=’МЭСМ’ /\**Для тех записей, где поле *Name* содержит значение «МЭСМ»*\*/*

*Пример 4.* Удалите все строки таблицы *Scientist\_Comp*, относящиеся к ЭВМ с уникальным номером 9.

SQL-запрос в данном случае будет иметь вид:

*DELETE /\**Изъять *\*/*

*FROM Scientist\_Comp /\**из таблицы *Scientist\_Comp* записи *\*/*

*WHERE Code\_Comp=9 /\**в которых поле *Code\_Comp* содержит значение, равное «9»*\*/*

Следует объяснить учащимся, что удалять, не проверив, нужны ли записи, сведения из базы данных, не желательно без предварительного согласования целесообразности совершения данного действия. Вышеперечисленные примеры на составление SQL-запросов подтверждают, что общепринятый термин «язык запросов» не совсем достаточно отображает рассматриваемые понятия, поскольку слово «запрос» подразумевает под собой лишь выборку, в то время как с применением этого языка выполняются также операции обновления, вставки и удаления, а также множество других.

Лабораторная работа №14 *«Работа по редактированию базы данных, расположенной на сервере»* является последней и итоговой при изучении основ баз данных. Ученики должны приобрести навыки работы с базой данных, расположенной на удаленном сервере, закрепить навыки работы со структурированным языком запросов SQL, приобрести умения работы с клиент-серверным технологиям посредством СУБД InterBase. Следует подвести итоги выполнения проекта, просмотреть базы данных, разработанные каждым учеником, сравнить по содержательному наполнению базы данных, расположенные на сервере.

Рассмотренное выше содержание обучения теме «Базы данных. СУБД» на основе клиент-серверных технологий характеризуется необходимостью использования достаточного количества заданий, в ходе решения которых ученики осуществляют основные информационные процессы сбора информации по определенной тематике базы данных из разных источников, обработки сведений и классификации их по смыслу (структурирование в табличной форме). Ученики также осуществляют дальнейшее практическое применение базы данных; передачу сведений с использованием клиент-серверных технологий и языка запросов SQL.

В соответствии с содержанием обучения сформулированы *требования* к знаниям, умениям и навыкам в области баз данных и СУБД на основе клиент-серверных технологий. Обучаемые *должны* *знать*: основные понятия баз данных и СУБД; этапы проектирования реляционных баз данных; основныеэтапы моделирования информационных процессов при решении задач средствами СУБД; *должны* *уметь*: представлять, передавать, сохранять и обрабатывать данные с помощью современных информационных систем; устанавливать отношения и связи между таблицами в среде СУБД; структурировать информацию в реляционных базах данных; проектировать, разрабатывать и использовать базы данных прикладного содержания; настраивать элементы управления и использовать возможности VBA в среде СУБД, использовать возможности языка структурированных запросов SQL для обеспечения управления структурой БД и манипулирования данными; *должны* *иметь навыки*: создания таблиц, форм, запросов (в том числе сложных) средствами, предоставленными изучаемой СУБД; работы в среде СУБД с графическим и командным интерфейсом.

**Литература**

1. Дейт К. Дж. Введение в системы баз данных, 7-е издание: Пер. с англ.. – М.; СПб.: Издательский дом «Вильямс», 2002. – 1072 с.: ил.
2. Щепакина Т.Е. Методика обучения СУБД Access в процессе моделирования, создания и работы над базой данных «История отечественных ЭВМ» // Компьютерные инструменты в образовании.– СПб. – 2003. − №5. С.42-49.
3. Щепакина Т.Е. Методические аспекты SQL-ориентированного изучения баз данных с использованием системы клиент-сервер // Ученые записки ИИО РАО. – 2005. − Выпуск 18. − С. 224-233.

**-----------------------------------------------------------------------------------------------------------------------**

[Р](http://rpio.ru/)[оссийский портал информатизации образования](http://portalsga.ru/) [содержит: законодательные и нормативные правовые акты государственного регулирования информатизации образования, федеральные и региональные программы информатизации сферы образования, понятийный аппарат информатизации образования, библиографию по проблемам информатизации образования, по учебникам дисциплин цикла Информатика, научно-популярные, документальные видео материалы и фильмы, периодические издания по информатизации образования и многое другое.](http://portalsga.ru)

