Лабораторная работа №1

Тема: Управление базами данных с помощью SQL. Выборка, чтение данных, получения итоговых значений. Однотаблични запросы. Внесение изменений в базы данных.

Цель: Применяя SQL-операторы, и специальные средства научиться выполнять сложную обработку данных с помощью запросов и совершенствовать их вывода. Научиться выполнять обобщенную групповую обработку значений полей с помощью агрегатных функций в SQL запросах. Научиться применять средства, которые управляют значениями в таблицах. Усвоить команды DML. Научиться с помощью SQL - запросов модифицировать, удалять, вносить информацию в БД.

Содержание работы по варианту индивидуального задания:

1. С помощью SQL- запроса вывести все данные с определенной таблице. В инструкции применить необязательно сокращения в виде символа «звездочка» (\*).

2. Вывести с помощью SQL- запроса информацию из таблицы только по двум полям.

3. Вывести данные поля вашей таблицы таким образом, чтобы результат не имел дубликатов (создать SQL- запрос).

4. По таблице данным получите информацию обо всех объектах (поставщики, клиенты, больные, студенты и т.д.), которым в поле, например, код или номер соответствует определенное значение (например 2003), а в другом по- значение больше или равно определенному значению (например> = 3).

5. С помощью агрегатной функции в SQL- запросе подсчитайте количество записей в таблице вашей БД, не считать пустые значения, но учитывать дубликаты.

6. Необходимо найти максимальное значение увеличенного вдвое поля, например стипендии или зарплаты, или цены товара, или другого поля со своей БД с помощью SQL- запроса.

7. Получите информацию из таблицы с данными о студентах (или другие объекты), упорядочивая их по размеру стипендии (или другое числовое поле) в порядке убывания, а для студентов, имеющих одинаковый размер стипендии в алфавитном порядке их фамилий.

8. С помощью DML в окне SQL построить запрос, благодаря которому можно было бы вносить новые записи в вашей таблице. Два поля таблицы должны заполняться определенной информацией, третье поле должно иметь значение NULL, а последние-за умолчанию.

9. Удалите из специально для этого созданной таблицы все записи с помощью SQL.

10. Удалите из таблицы те записи, которые в конкретном поле имеют определенное значение с помощью DML.

11. С помощью DML измените где-либо поля на «пусто», если поле дата имеет значение 01.01.2010.

12. Уменьшить с помощью DML значение числового поля на 25% в таблице, если его значение, например, равна 32500.

13. В режиме SQL Вид создайте инструкцию, с помощью которой создается архивная копия некоторой таблицы.

теоретические сведения

Запрос SQL- это запрос, который создается с помощью инструкций SQL. Инструкция (строка) SQL это выражение, определяет команду SQL (SELECT, UPDATE или DELETE), и включает строки (например WHERE или ORDER BY). Инструкции SQL обычно применяются в запросах и в статистических функциях. Язык SQL (Structured Query Language) применяется при создании запросов, а также для обновления и управления реляционными базами данных, такими как базы данных Microsoft Access.

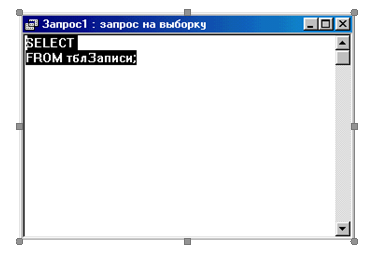
SQL - это самая распространенный язык управления базами данных в системах клиент / сервер. Основное преимущество SQL заключается в том, что ее можно использовать для разработки запросов в любой системе управления данными, совместимой с SQL. SQL - запрос состоит из последовательности SQL - инструкций (SQL-statement), которые указывают, что нужно сделать с входным набором данных (таблицей или запросом) для генерации исходного набора. Путем аргументов (параметров) этих инструкций (clause) конкретизируются действия, которые выполняются, то есть задают имена полей, имена таблиц, условия, отношение и т.п. Извлеченная из базы информация (выходной набор) обрабатывается с помощью специальных статистических функций (aggregate functions). При такой обработке можно определить, например, минимальное и максимальное значения, сумму и среднее значение.

Инструкции SQL можно применять в Microsoft Access в тех местах, где возможно ввести имя таблицы, запроса или поля.

В некоторых ситуациях инструкция SQL создается автоматически. Например, применяя мастер для создания формы или отчета, которые получают данные из нескольких таблиц, автоматически создается инструкция SQL, которая будет значением свойства Источник строк (RowSource) формы или отчета. Создавая список или поля со списком с помощью мастера также создается инструкция SQL, которая становится значением свойства Источник строк (RowSource) списка или поля со списком.

Если мастер не применялся, возможно создать инструкцию SQL в ячейках свойств Источник записей (RecordSource) или Источник строк (RowSource), применяя построитель рядом с соответствующей ячейкой и создав запрос в режиме конструктора запроса.

С помощью языка структурированных запросов SQL, реализованых в Access, можно составить любое количество сложных запросов. Этот язык позволяет также управлять обработкой запросов, SQL-запрос представляет собой последовательность инструкций, в которую могут входить выражения и статистические функции SQL.

рисунок 1

Когда в режиме конструктора пользователь создает спецификацию запроса, Access одновременно строит соответствующий SQL-запрос. Изменения в SQL-запросе автоматически отображаются и в спецификации QBE - запроса. Чтобы отобразить на экране или исправить SQL-запрос, нужно вызвать команду SQL Вид меню Вид (в режиме конструктора запроса) рисунок 1.

В окне SQL вводятся инструкции, составляющие SQL-запрос. При вводе текста в этом окне слишком длинные строки разрываются. Для повышения наглядности инструкции каждый командную строку SQL можно начинать с новой строки, используя комбинацию клавиш (Ctrl + Enter). К примеру:

SELECT SNUM, SFAM, SIMA, SOTCH, STIP

FROM STUDENTS;

SELECT- ключевое слово, которое сообщает БД, эта команда является запросом.

SNUM, SFAM, SIMA, SOTCH, STIP-список полей из таблицы, которые выбираются запросом. Запрос не влияет на информацию в таблицах, он только показывает данные.

FROM STUDENTS - ключевое слово, которое сопровождается именем таблицы, которую используем в качестве источника информации.

Точка с запятой «;» - применяется во всех интерактивных командах SQL для сообщения БД что команда готова для выполнения.

"\*" -если Необходимо получить каждое поле таблицы. К примеру:

SELECT \* FROM STUDENTS;

что приведет к тому же результату, что и предыдущая команда. Если необходимо вывести только некоторые поля из таблицы, просто необходимо исключить из списка ненужные поля.

Если есть необходимость в удалении лишних данных используется DISTINCT-аргумент, который обеспечивает возможность изымать значение повторяющиеся из предложения SELECT. К примеру:

SELECT DISTINCT SNUM FROM USP;

эту инструкцию применяют для получения списка результатов без дубликатов.

Если применить ALL, то это будет иметь обратный эффект.

WHERE - инструкция команды SELECT, которая позволяет устанавливать предикат , условие которых возможно будет выполняться либо не исполнять любой записи таблицы. Команда получает только те записи из таблицы, для которых такое утверждение будет верным. К примеру:

SELECT SFAM, STIP

FROM STUDENTS

WHERE STIP = 25.50;

выведутся только те фамилии и размер стипендии студентов, для которых в поле STIP будет значение 25.50.

Связывая предикатив с булевськимы операторами (AND, OR, NOT) и реляционными операторами (=,>, <,> =, <=, <>), намного увеличивается возможность выборки нужных данных. Также можно применять специальные операторы IN, BETWEEN, LIKE, IS NULL.

Запросы способны выполнять обобщенную групповую обработку значений полей, реализуемой с помощью агрегатных функций. Агрегатные функции получают одиночное значение для всей группы таблицы. В SQL допускают следующие функции:

COUNT, SUM, AVG, MAX, MIN. Чтобы найти сумму всей полученной стипендии в таблице о студентах имеем:

SELECT SUM (STIP)

FROM STUDENTS;

Функция COUNT применяется для подсчета количества значений в столбце. К примеру:

SELECT COUNT (SNUM)

FROM USP;

Если с COUNT применять ALL (используется по умолчанию) - нельзя подсчитать значение NULL, но учитываются дубликаты, а COUNT с «\*» включает записи с NULL и дубликаты.

Для упорядочения вывода полей таблиц SQL имеет команду ORDER BY, что позволяет сортировать вывода запроса в соответствии со значениями среди количества полей. Если указать несколько полей, то столбцы вывода упорядочиваются один внутри другого, при этом надо определить роста (ASC) или убывания (DESC) для каждого столбца. К примеру:

SELECT \*

FROM USP

ORDER BY SFAM ASC;

выведет таблицу с информацией о студентах в алфавитном порядке фамилий.

Однотабличные запросы - это запросы, которые в качестве источника данных используют только одну таблицу.

Для сохранения набранной в SQL окне последовательности инструкций необходимо нажать кнопку ОК. Если в SQL-запросе нет ошибок, Access сформирует эквивалентный QBE-запрос и заполните поля QBE-области. Теперь запрос можно выполнить, вызвав из меню Запрос Выполнить или активизировав кнопку с восклицательным знаком на панели инструментов.

С помощью языка структурированных запросов SQL, реализованого в Access, можно составить любое число сложных запросов. Этот язык позволяет также управлять обработкой запросов, SQL-запрос представляет собой последовательность инструкций, в которую могут входить выражения и статистические функции SQL.

Примерами SQL-запросов есть запросы на объединение, запросы к серверу, управляющие и подчиненные запросы. Некоторые SQL-запросы невозможно создать в бланке запроса, для определения запроса в режиме конструктора.

Запрос к серверу-запрос SQL, применяемый для передачи команд непосредственно на сервер базы данных ODBC. Запрос к серверу позволяет напрямую работать с таблицами на сервере вместо обработки их данных с помощью ядра базы данных Microsoft Access.

Управляющий запрос- запрос SQL, содержащий инструкции языка описания данных DDL. Такие инструкции позволяют создавать или изменять объекты в БД.

Запрос на объединение-запрос, в котором с помощью оператора UNION объединяются результаты двух или нескольких запросов на выборку.

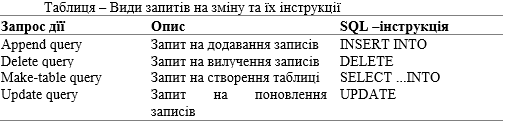
Подчиненные запросы-запросы, в которых инструкция SQL SELECT, помещается внутри другого запроса на выборку или на смену.

Разработка запросов на объединение, запросов к серверу и управляющих запросов осуществляется только в окне запроса в режиме SQL, а для подчиненных запросов пользователь может ввести инструкцию SQL в строку Поле или Условие отбора в бланке запроса.

**Запросы на изменение с помощью SQL**

Во время работы с SQL важно уметь пользоваться средствами управления информацией в таблицах. Как вы уже знаете, запросы на изменение используются для добавления, удаления и обновления записей, а также для сохранения результирующего набора записей запроса в виде таблицы. Значения могут вставляться, модифицироваться и удаляться с полей с помощью трех команд DML (Язык манипулирования данными) -INSERT, UPDATE, DELETE.

Существует четыре подвида запросов на изменение, каждому из которых соответствует определенная SQL инструкция.

Таблица - Виды запросов на изменение и их инструкции

**Добавление информации в БД**

В SQL все записи в таблицу вводятся с помощью команды модификации INSERT. Но следует помнить, что имя таблицы, в которую происходит вставка, должно быть предварительно определено, а каждое значение, вставляется должно совпадать с типом данных столбца, в который оно вставляется. Например, для добавления записи в таблицу преподавателей TEACHERS, можно воспользоваться следующим выражением:

INSERT INTO TEACHERS

VALUES (4006, "Федченко", "Светлана", "Геннадиевна", # 01/09/1999 #)

Можно указать столбцы, в которые необходимо осуществить вставку значений. К примеру:

INSERT INTO TEACHERS (TDATE, TFAM, TIMA)

VALUES (# 01/09/1999 #, "Федченко", "Светлана")

Для полей, которые не указаны в запросе автоматически устанавливаются значения по умолчанию.

Есть возможность с помощью команды INSERT получать или выбирать значения из одной таблицы и помещать их в другую вместе с запросом.

Для полей, которые не указаны в запросе автоматически устанавливаются значения по умолчанию.

Есть возможность с помощью команды INSERT получать или выбирать значения из одной таблицы и помещать их в другую вместе с запросом.

**удаление данных**

Удаление строк из таблицы можно осуществить командой модификации DELETE. Надо учитывать, то, что команда может удалять только целые записи таблицы, а не индивидуальные значения где-либо полей. Для удаления всего содержимого таблицы STUDENTS возможно выполнить следующее:

DELETE FROM STUDENTS;

Для определения строк, которые нужно удалить при условии, применяют предикат. Например, для удаления информации, касающейся студента Нагорного можно использовать следующую команду:

DELETE FROM STUDENTS

WHERE SNUM = 3416;

В качестве предикат используют номер студенческого билета.

**Изменение существующих данных**

Возможность изменения всех или некоторых значений в таблице реализуется с помощью команды UPDATE. В ней указывается имя таблицы, используемой и слово SET, которое определяет изменение которая состоится для нужного поля таблицы. К примеру:

UPDATE USP

SET OCENKA = 5;

Приведет к изменению в таблице USP всех оценок на «5».

Для изменения единого значения можно применять предикат. К примеру:

UPDATE USP

SET OCENKA = 5;

WHERE PNUM = 2003;

С помощью UPDATE можно модифицировать данные с нескольких полив- SET состоянии определить любое количество полей, разделенных запятыми.

Модифицировать сразу несколько таблиц одной командой UPDATE не позволено, поэтому нельзя применять название таблицы через точку в именах полей

В строке SET команды UPDATE можно применять выражения, располагая их в списке для того поля, которое необходимо изменить. К примеру:

UPDATE STUDENTS

SET STIP = STIP \* 2;

Команда увеличивает значение стипендии в 2 раза.

К недостаткам команды UPDATE можно отнести невозможность ссылаться на таблицу, которая задиена в любом подзапросе из команды модификации. Например, невозможно одной командой выполнить такое действие, как модификация оценок для студентов, у которых оценки ниже середньои.Для этого необходимо выполнить один запрос:

SELECT AVG (OCENKA)

FROM USP;

а затем результат этого запроса применить для модификации:

UPDATE USP

SET OCENKA = OCENKA-1

WHERE OCENKA <4.2;

**Запрос на создание таблицы**

С помощью SQL-инструкции можно создавать таблицы на основе уже существующих таблиц. К примеру:

SELECT \* INTO VIPUSK

FROM STUDENT;

С помощью этих инструкций будет создана таблица VIPUSK, которая содержит все данные из таблицы STUDENT.

**Преобразование QBE - запроса в SQL-запрос**

Запросы, созданные в окне конструктора запроса, Access автоматически превращает в SQL запросы. Для этого необходимо выполнить следующую последовательность операций:

\* В окне проектирования QBE-запроса создайте QBE-запрос, необходимо преобразовать в SQL запрос.

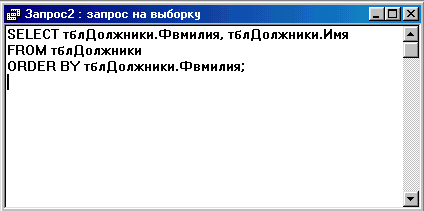
\* В меню Вид выберите команду SQL Изд. На экране появится окно SQL.

рисунок 2

В окне SQL появляется SQL-запрос, соответствующий QBE-запроса, составленном в окне конструктора (рисунок 2).

SQL-запрос можно отредактировать в окне SQL. После закрытия окна изменения автоматически отображаются и в QBE-запросе.

**Контрольные вопросы к лабораторной работе 1:**

1. Какие запросы называют SQL-запросами?

2. Какое главное назначение SQL-запросов?

3. С помощью какой инструкции в запросах осуществляется выборка данных?

4. С помощью какого символа в инструкции SQL-запросов можно вывести все данные из таблицы?

5. Какое средство предотвращает дублирование информации в SQL-запросах?

6. В каких ситуациях инструкция SQL создается автоматически в Microsoft Access?

7. С помощью какого средства в SQL-запросах в результате сохраняется дублирование строк вывода?

8. Что значит инструкция SELECT ... FROM ...?

9. Для чего применяется команда ORDER BY?

10. Какая инструкция команды SELECT, позволяет устанавливать предикатив?

11. Какая функция применяется для подсчета количества значений в столбце таблицы в SQL-запросе?

12. Какие особенности языка SQL?

13. Какие типы SQL-запросов вы знаете?

14. Что представляет собой запрос на объединение?

15. Что представляет собой запрос к серверу?

16. Что представляет собой управляющий запрос?

17. Что представляет собой подчиненный запрос?

18. Возможно ли превращать SQL запросы в QBE и наоборот?

19. Какие запросы на изменение вы знаете?

20. С помощью какой инструкции можно создать запрос на добавление записей в таблицу?

21. Что означает команда UPDATE в запросах?

22. Какую инструкцию SQL должен иметь запрос на изъятие записей?

23. Что означает инструкция SELECT ... INTO