Создание запросов в СУБД Access средствами SQL

Запрос — объект базы данных, используемый для выборки или модификации хранимых данных.

В режиме конструктора можно открывать различные запросы: запрос на выборку, перекрестный запрос и запрос на изменение.

Запрос на выборку и перекрестный запрос также можно открыть в режиме таблицы для просмотра результатов.

Запросы на выборку и их использование

Запрос на выборку является наиболее часто используемым типом запроса. Запросы этого типа выбирает данные из одной или нескольких таблиц и отображают их в виде таблицы, записи в которой можно обновлять (с некоторыми ограничениями). Запросы на выборку можно также использовать для группировки записей и вычисления сумм, средних значений, подсчета записей и нахождения других типов итоговых значений.

Оператор SELECT

Основой SQL является инструкция SELECT, используемая для создания запросов на выборку.

Синтаксис инструкции:

SELECT [ALL | DISTINCT | DISTINCTROW]
список_выбора
FROM имена таблиц
[WHERE критерий поиска]
[GROUP BY имя столбца, имя столбца,...]
[HAVING условие поиска]
[ORDER BY критерий столбца [ASC | DESC]];

SELECT — выбрать (директива) данные из указанных столбцов и (если необходимо) выполнить перед выводом их преобразование в соответствии с указанными выражениями и (или) функциями

FROM — из (условие) перечисленных таблиц, в которых расположены эти столбцы

WHERE — где (условие) строки из указанных таблиц должны удовлетворять указанному перечню условий отбора строк

GROUP BY — группируя по (условие) указанному перечню столбцов с тем, чтобы получить для каждой группы единственное агрегированное значение, используя во фразе SELECT SQL – функции: SUM (сумма), COUNT (количество), MIN (минимум), MAX (максимум), AVG (среднее значение)

HAVING — имея в результате лишь те группы, которые удовлетворяют указанному перечню условий отбора групп (условие)

ORDER BY — спецификация сортировки (условие) определяет порядок сортировки: ASC – сортировка по возрастанию, DESC - сортировка по убыванию.

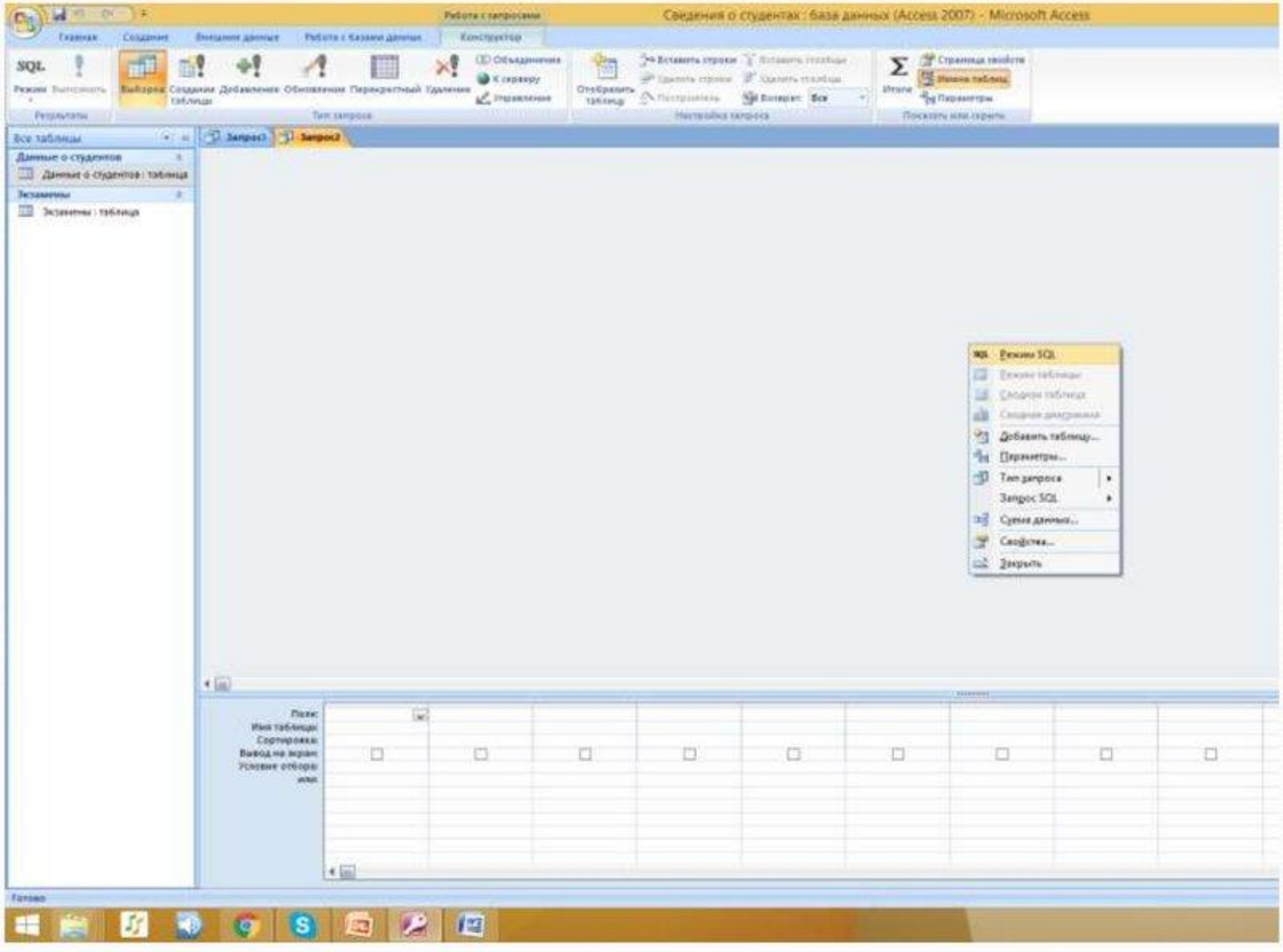
Запросы с использованием единственной таблицы

Все запросы на получение практически любого количества данных из одной или нескольких таблиц выполняются с помощью единственного предложения SELECT. В общем случае результатом реализации предложения SELECT является другая таблица. К этой новой (рабочей) таблице может быть снова применена операция SELECT и т.д., т.е. такие операции могут быть вложены друг в друга. Представляет исторический интерес тот факт, что именно возможность включения одного предложения SELECT внутрь другого послужила мотивировкой использования прилагательного "структуризированный" в названии языка SQL.

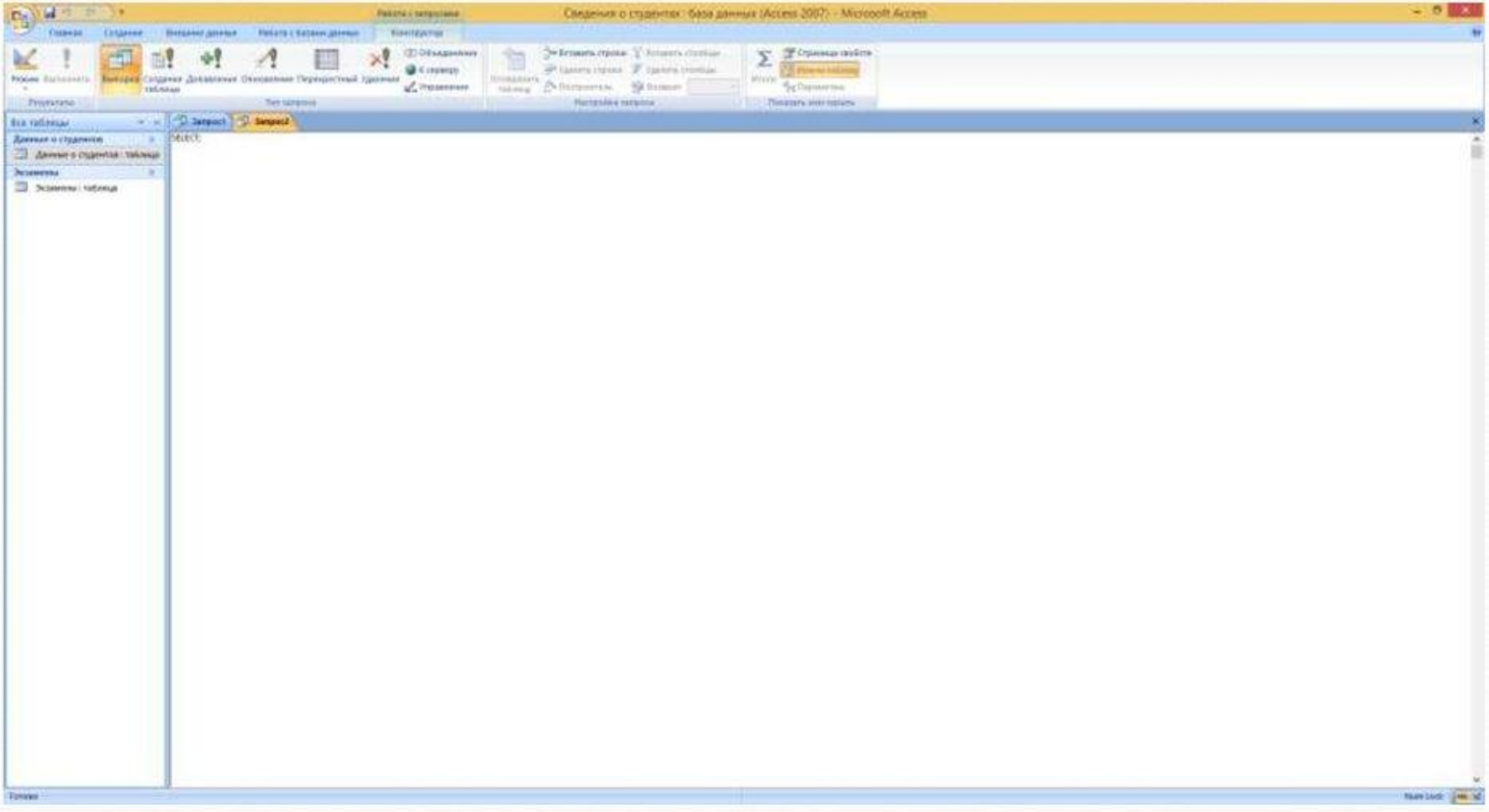
ПРЕДИКАТЫ

1. Сравнения =, <>, >=, <, <=
2. В интервале - “между” BETWEEN a1 and a2
3. Входит в множество IN (=[Предмет] IN (“История”,“Информатика”)
4. Подобие < имя > Like < образец >
   (что) (с чем сравнивать)

Режим SQL в MS Access



Окно SQL

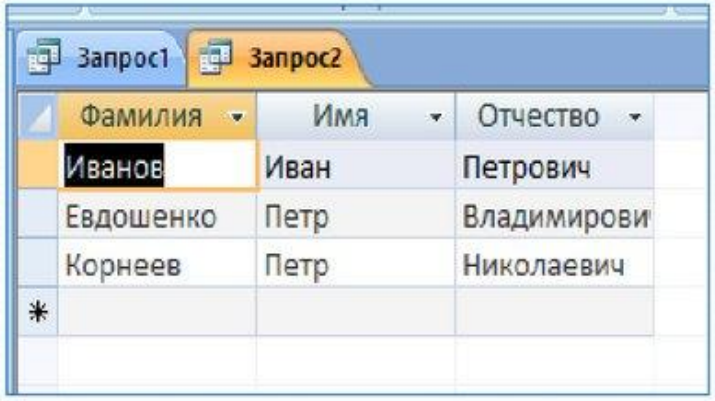

Сведения о студентах : база данных (Access 2007) - Microsoft Access
Все таблицы
Данные о студентах
Данные о студентах : таблица
Экзамены
Экзамены : таблица
Запрос
Запрос2
SELECT;

Рассмотрим синтаксис запросов на выборку:

1. Запрос на выборку фамилии, имени и даты рождения студента

```
SELECT      Фамилия, Имя, Отчество
FROM Данные;
```

Результат:



При необходимости получения полной информации о Студенте, можно было бы дать запрос

SELECT Фамилия, Имя, Отчество, Город, Адрес, Телефон (и т.д.)

FROM Данные

или использовать его более короткую нотацию:

SELECT * (Звездочка (*) может применяться для вывода полного списка столбцов)

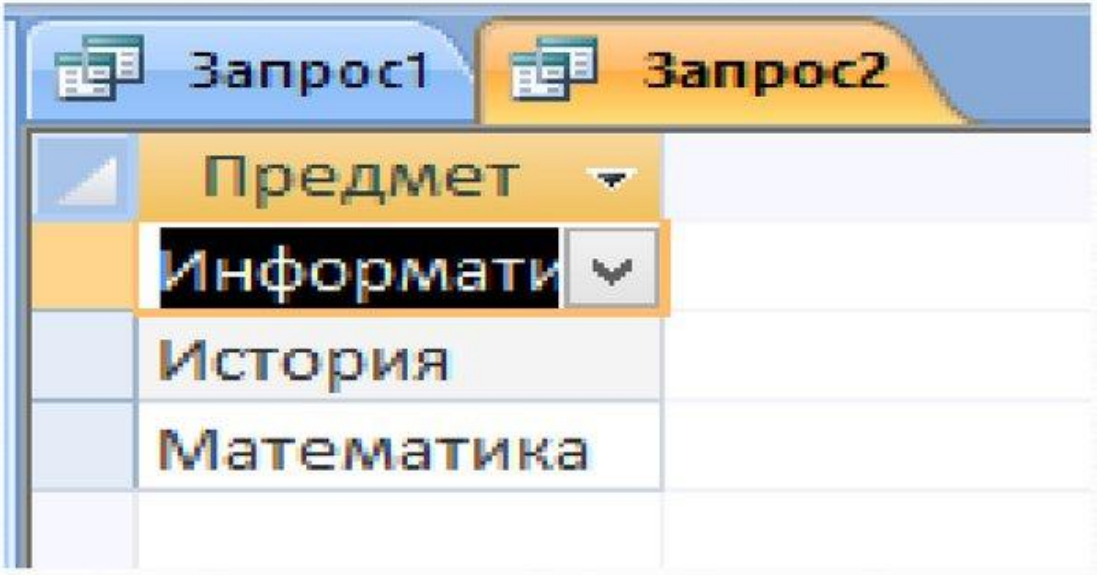
FROM Данные

Результат:

Для исключения дубликатов и одновременного упорядочения перечня необходимо дополнить запрос ключевым словом DISTINCT (различный, различные), как показано в следующем примере:

```
SELECT DISTINCT Предмет;
FROM Экзамены;
```

Результат:

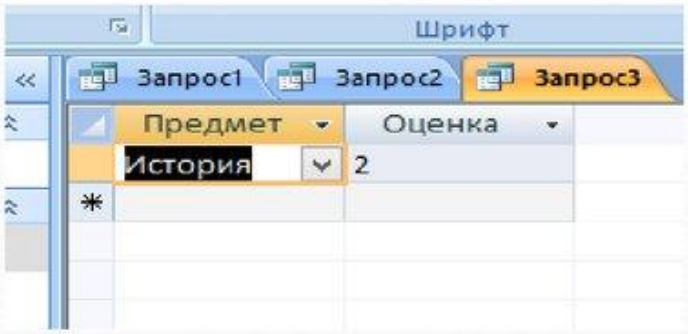


В синтаксисе фразы WHERE показано, что для отбора нужных строк таблицы можно использовать операторы сравнения = (равно), <> (не равно), < (меньше), <= (меньше или равно), > (больше), >= (больше или равно), которые могут предваряться оператором NOT, создавая, например, отношения "не меньше" и "не больше".

Так, для получения перечня предметов, по которым были получены 2, можно сформировать запрос

```
SELECT Экзамены.Предмет, Экзамены.Оценка
FROM Экзамены
WHERE (((Экзамены.Оценка)="2"));
```

Результат:



Создайте запросы:

1. На получения предметов, по которым были получены 5 или 4.
2. На получение списка студентов, проживающих в г.Оренбурге.
3. Список студентов, получающих стипендию более 1600.

Оператор INSERT

```
INSERT INTO <имя_таблицы> [(<имя столбца_1> [, <имя
  столбца_1> ...])] {VALUES (<значение_1> [,
  <значение_2> ...]) | <выражение SELECT>};
```

Так, например, чтобы ввести строку в таблицу Продавцов, вы можете использовать следующее условие:

1. `INSERT INTO Salespeople VALUES (1001, 'Peel', 'London', .12);`
2. `INSERT INTO Customers (city, cname, cnum) VALUES ('London', 'Honman', 2001);`

Оператор DELETE

Вы можете удалять строки из таблицы командой модификации - DELETE. Она может удалять только введенные строки, а не индивидуальные значения полей.

```
DELETE FROM <имя_таблицы> [WHERE <условие>];
```

Например

```
1. DELETE FROM Salespeople WHERE snum = 1003;
2. DELETE FROM Salespeople WHERE city = 'London';
```

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Википедия – режим доступа: ru./wiki/SQL
2. Вопросы практического програмирования – режим доступа: mstu.edu/education/materials/zelenkov/ch_4_7.html
3. Введение в структурированный язык запросов SQL – режим доступа: intuit/department/database/sql/1/
4. Всё про Sql – режим доступа: sql/
5. Введение в стандарты языка баз данных SQL – режим доступа: citforum/database/sqlbook/index.shtml