**Лабораторное занятие №7.**

**Выборка данных оператором SELECT.**

**Лабораторное (практическое) занятие** рассчитано на 2 часа, относится к теме « »

**Формируемые компетенции:** ОК.1-9 ПК 2.1 - 2.4

**Цель:** Изучить используемый в реляционных СУБД оператор извлечения данных из таблиц. Получить навыки работы с оператором SELECT в программе "SQLiteStudio".

**Методическое и техническое обеспечение:** IBM PC-совместимый компьютер, SQLiteStudio.

## Исходные данные

Студент получает индивидуальный вариант исходных данных с кратким описанием предметной области, который используется при выполнении всех лабораторных работ. При этом каждая очередная лабораторная работа является продолжением выполненной ранее и поэтому они должны обязательно выполняться последовательно. Вариант задания приведен в лабораторной работе №1.

**Теоретические сведения**

В SQL имеется единственный оператор, который предназначен для выборки данных из базы данных. Как и операторы INSERT, DELETE и UPDATE он относится к подмножеству DML.

Ниже приведен почти полный синтаксис оператора **SELECT**.

**SELECT** [**DISTINCT | ALL**]

**{\*** | <величина> [,<величина> ...]**}**

[**INTO** :Переменная [, :Переменная ...]]

**FROM** <tableref> [, <tableref> ...]

[**WHERE** <условие поиска>]

[**GROUP BY** Колонка [, Колонка ...]]

[**HAVING** <условие поиска>]

[**UNION** [**ALL**] <select\_expr>]

[**ORDER BY** <список сортировки>];

<величина> = **{**Колонка | :Переменная | <константа> | <выражение> | <функция>

| udf ([<величина> [, <величина> ...]])

| **NULL** | **USER}** [**AS** Псевдоним]

<константа> = Число | 'Строка'

< выражение > = SQL выражение, возвращающее единичное значение

< функция> =

**COUNT** **(\*** | [**ALL**] <величина> | **DISTINCT** <величина>**)**

| **SUM**  **(**[**ALL**] <величина> | **DISTINCT** <величина>**)**

| **AVG** **(**[**ALL**] <величина> | **DISTINCT** <величина>**)**

| **MAX** **(**[**ALL**] <величина> | **DISTINCT** <величина>**)**

| **MIN** **(**[**ALL**] <величина> | **DISTINCT** <величина>**)**

| **CAST(**<величина> **AS** <тип данных>**)**

| **UPPER(**<величина>**)**

| **GEN\_ID(**Имя\_Генератора, <величина>**)**

<tableref> = **{**<joined\_table> table view

| procedure[**(**<величина> [, <величина> ...]**)**]**}**

[ Псевдоним]

<joined\_table> = <tableref> <join\_type> **JOIN** <tableref> **ON** <условие поиска>

| **(**<joined\_table>**)**

<join\_type> = [**INNER**] | **{LEFT | RIGHT | FULL}** [**OUTER**]

< условие поиска> =

< величина> < оператор сравнения>

**{**<величина> | **(**<select\_one>**)}**

| <величина> [**NOT**] **BETWEEN** <величина> **AND** <величина>

| <величина> [**NOT**] **LIKE** <величина>

| <величина> [**NOT**] **IN (**<величина> [, <величина> ... ] | <select\_list>**)**

| <величина> **IS** [**NOT**] **NULL**

| <величина> **{** >= | <=**}** <величина>

| < величина> [**NOT**] **{** = | < | > **}** < величина>

| **{ALL | SOME | ANY} (**<select\_list>**)**

| **EXISTS (**<select\_expr>**)**

| **SINGULAR (**<select\_expr>**)**

| <величина> [**NOT**] **CONTAINING** <величина>

| <величина> [**NOT**] **STARTING** [**WITH**] <величина>

| ( <условие поиска>)

| **NOT** < условие поиска>

| <условие поиска> **OR** <условие поиска>

| <условие поиска> **AND** <условие поиска>

<оператор сравнения> =

**{** = | < | > | <= | >= | !< | !> | <> | != **}**

<select\_one> = оператор SELECT, выбирающий одну колонку и возвращающий ровно одно значение

<select\_list> = оператор SELECT, выбирающий одну колонку, возвращающий ноль или много значений

<select\_expr> = оператор SELECT, выбирающий несколько величин и возвращающий ноль или много значений

< список сортировки> = **{**Колонка | Номер**}** [**ASC** | **DESC**]

[, < список сортировки > ... ]

Некоторые параметры, входящие в этот оператор, описаны в табл. 1.

Таблица 1 Описание параметров оператора SELECT

|  |  |
| --- | --- |
| **Параметр** | **Описание** |
| DISTINCT | ALL | DISTINCT - предотвращает дублирование данных, которые будут извлечены.  ALL (по умолчанию) - приведет к извле­чению всех данных |
| {\*  |< величина> [,< величина >...]} | Звездочка (**\***) означает, что надо извлекать все колонки из указанных таблиц.  <величина> [,<величина> ...] -извлекает список указанных колонок, переменных или выражений |
| INTO : Переменная [,  : Переменная ... ] | Используется только в триггерах и храни­мых процедурах для операторов SELECT, возвращающих не более одной строки. Указывается список переменных, в которые извлекаются величины |
| FROM <tableref> [,<tableref>... ] | Указывает список таблиц, просмотров и хранимых процедур, из которых извлекаются данные. Список может включать соединения и соединения могут быть вложенными |
| table | Имя существующей в базе данных табли­цы |
| view | Имя существующего базе данных про­смотра |
| procedure | Имя существующей хранимой процедуры, предназначенной для использования в операторе SELECT |
| Псевдоним | Короткое альтернативное имя для табли­цы, просмотра или колонки. После описа­ния в <tableref>, псевдоним может ис­пользоваться для ссылок на таблицу или просмотр |
| join\_type | Задает тип соединения, которое может быть внутренним или внешним |
| Параметр | Описание |
| WHERE <условие поиска> | Указывает условие, которое ограничивает количество извлекаемых строк |
| GROUP BY Колонка [,  Колонка ...] | Разбивает результат запроса на группы, содержащие все строки с идентичными значениями указанными в списке колонок |
| HAVING  <условие поиска> | Использует совместно с GROUP BY. Зада­ет условие, которое ограничивает количе­ство возвращаемых групп |
| UNION [ALL] | Объединяет результаты нескольких запро­сов. Все запросы должны извлекать оди­наковое количество столбцов, тип данных каждого столба первого запроса должен совпадать с типом данных других запро­сов, имена столбцов в разных запросах мо­гут отличаться. Необязательный параметр ALL указывает, что надо выводить одина­ковые строки |
| ORDER BY  <список сортировки> | Указывает колонки, по которым будет производиться сортировка извлекаемых строк. Можно указывать либо имена коло­нок, либо их порядковые номера в списке извлекаемых колонок. Если указать **ASC**, то строки будут выдаваться в порядке возрастания значений сортируемых полей, если **DESC** - в порядке убывания значений |

Как видно из синтаксиса оператора SELECT, обязательными являются только предложение **SELECT** с перечнем выдаваемых колонок и предложение **FROM**.

Пример простейшего оператора SELECT:

*-- Выдать перечень всех служащих:*

**SELECT \* FROM** Employee;

Пример. Подсчитаем количество регионов (штатов) в стране США:

**SELECT COUNT(\*) FROM** REFREG **WHERE** CODCTR = 'USA';

**--***Выбрать список сотрудников только одной организации*

**SELECT** CODPEOPLE **AS** "Код сотрудника",

DUTIES **AS** "Должность",

SALARY **AS** "Оклад"

**FROM** STAFF **WHERE** CODORG = 11;

*--Находит среднюю зарплату сотрудников этой организации и месячный фонд заработной платы.*

**SELECT AVG(**SALARY**), SUM(**SALARY**)**

**FROM** STAFF

*-- Выберем из этого списка только тех сотрудников, чей оклад находится в диапазоне от 12 000 до 23 700:*

**SELECT** CODPEOPLE **AS** "Код сотрудника",

DUTIES **AS** "Должность",

SALARY **AS** "Оклад"

**FROM** STAFF

**WHERE** CODORG =11 **AND**

SALARY **BETWEEN** 12000 **AND** 23700;

*--Вывести сотрудников фамилия которых должна заканчиваться на "ОВ".*

**SELECT** COD **AS** "Код",

NAME3 **AS** "Фамилия",

NAME1 **AS** "Имя",

NAME2 **AS** "Отчество",

BIRTHDAY **AS** "Дата рождения"

**FROM** PEOPLE

**WHERE** NAME3 **LIKE** **'**%OB**'**

**ORDER BY** NAME3;

Ниже приведено несколько упрощенных вариантов синтаксиса оператора SELECT, помогающих научиться составлять простые запросы. Упрощенный синтаксис внутреннего соединения (стандарт SQL-92):

**SELECT** Колонка [, Колонка ...] | **\***

**FROM** <tableref\_left> [**INNER**] **JOIN** <tableref\_right>

[**ON** <условие поиска>]

[**WHERE** <условие поиска>];

Упрощенный синтаксис внешнего соединения:

**SELECT** Колонка [, Колонка ...] | **\***

**FROM** <tableref\_left>

**{LEFT | RIGHT | FULL} [OUTER] JOIN**

<tableref\_right>

[**ON** <условие поиска>]

[**WHERE** <условие поиска>];

Упрощенный синтаксис использования подзапроса:

**SELECT** [**DISTINCT**] Колонка [, Колонка ...]

**FROM** <tableref> [, <tableref> ...]

**WHERE**

**{**expression **{**[**NOT**] **IN** | <оператор сравнения>**}**

| [**NOT**] **EXISTS**

**}**

**(SELECT** [**DISTINCT**] Колонка [, Колонка ...]

**FROM** <tableref> [, <tableref> ...]

**WHERE** <условие поиска>

**) ;**

***Ход работы***

Для выполнения запросов в программе " SQLiteStudio " необходимо выполнить следующие действия:

1. Подключиться к базе данных и выполнить команду "Tools > Open SQL Editor". В результате откроется окно "SQL Редактор" (рис. 1).

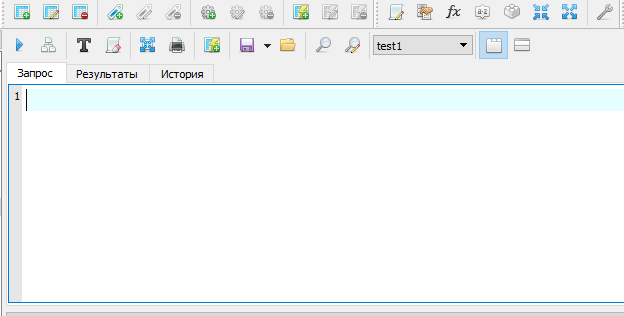


Рис. 1: Окно выполнения запросов

1. Ввести в поле на вкладке "**Запрос**" текст запроса.
2. Нажать на панели инструментов кнопку  [Выполнить] (F9).
3. Если запрос правильный, то в результате произойдет его выполнение и ре­зультат будет отображен на вкладке "Результаты" (рис. 2).

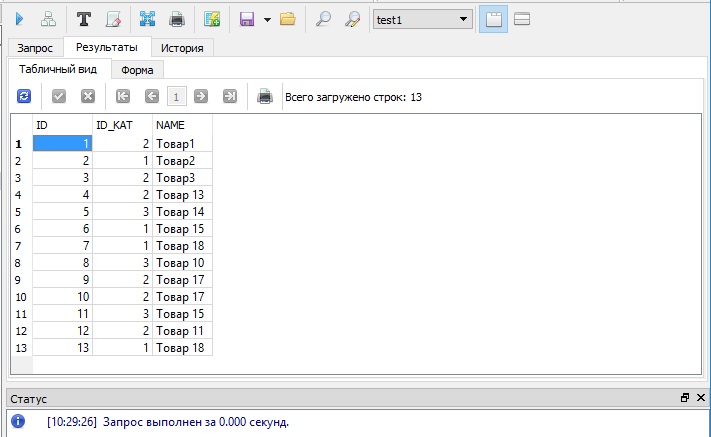


Рис. 2: Окно с результатом выполнения запроса

Если же в запросе будет ошибка, то программа выдаст соответствующее сообщение, как показано на рисунке 3.

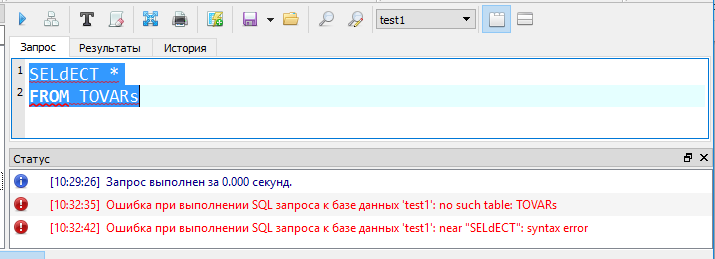


Рис. 3: Окно с результатом выполнения запроса

***Отчет о выполнении работы***

Отчет о выполнении необходимо оформить на листах формата A4.

Отчет должен содержать описание и результаты работы, представляемые в следующей последовательности:

1. Перечень запросов к базе данных, выполненных при выполнении этой лабораторной работы с краткими пояснениями и данными о количестве строк, выданных при выполнении каждого запроса.

2. Словесное описание двадцати запросов к разработанной базе данных.

3. Распечатка всех запросов с краткими комментариями и указанием количества возвращенных строк.