Описание интерфейся интерпретатора SQL

1 После запуска выводится подсказка:

SQL>

2 Пользователь вводит одну из следующих команд:

**/?**- выводит список возможных команд на экран.

**/? >** <имя файла> - выводит список возможных команд в файл <имя файл> (переназначение вывода).

<имя файла> - парные угловые скобки не являются частьб синтаксиса команд.

**EXIT**

Команда завершить работу интерпретатора

Команда закрывает открытую таблицу (файл) и завершает работу интепретатора.

**CREATE TABLE** <имя файла>

(<имя поля1> <тип> [(<ширина> [, <точность>])

[, <имя поля2> ... ]])

… - символ продолжения списка полей. Не является частью синтаксиса команды.

[] – отмечает необязательный элемент команды. Не является частью синтаксиса команды.

Команда CREATE TABLE создает файл формата DBF (смотри Приложение А).

Каждое поле таблицы характеризуется именем, типом, шириной и точностью. Эти характеристики должны содержаться в самой команде.

< имя файла > - имя создаваемого файла (таблицы), которое может содержать путь.

<имя поля1> и <имя поля2> - имена полей новой базы данных. Возможно до 255 полей.

<тип> - буква, означающая тип данных поля. Hекоторые типы данных требуют обязательного задания <ширины> поля и <точности> (количество символов после запятой).

<тип>, <ширина> и <точность> могут быть следующими:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип | Ширина | Точность | Описание |
| C | n | - | Строка символов <=n. Если длина строки <n, строка дополоняется пробелами. Обработка строк регистрозависима. |
| D | - | - | Дата в формате dd.mm.yyyy |
| L | - | - | Логический тип (.TRUE. или .FALSE.) |
| M | - | - | Мемо. Содержит ссылку на файл с именем <имя таблицы.mem> c произвольным текстом объёмf <=4 гигабайта. Таблица может содержать только одно поле типа M. |
| N | n | d | Число длиной n и точность d |

ПРИМЕР

CREATE TABLE employee (name C(20), addr C(30), city C(30),;

zip C(5), salary N(8,2), comments M)

**INSERT** INTO имя\_таблицы (список\_полей\_через\_запятую) VALUE (список значений\_через\_запятую)

Команда INSERT добавляет к таблице строку. Строковые значения берутся в кавычки “ ”. Если список полей не полон, значения неуказанных полей считаются неопределёнными.

Поле типа MEMO может заполняться также, как и символные поля, но при большом объёме текста используется загрузка из файла, в имено которого может присутствовать путь к файлу.

ПРИМЕР

INSERT INTO employee (name, city, salary, comments) VALUE (“Петров”, “Новосибирск”, 20000, @имя\_загружаемого\_файла)

**UPDATE** имя\_таблицы SET список\_пар\_имя\_поля=значение через\_запятую [WHERE логическое\_выражение]

Команда UPDATE заменяет значения указанных полей на укзанные значения в записях, удовлетворяющих логическому выражению. Если фраза WHERE не указана, замена происходит во всех записях.

Логическое выражение – элементарное\_выражение[ЛСэлементарное\_выражение[ЛС……]]

Элементарное\_выражение - имя\_поляКОПзначение

КОП – обозначение операции сравнения (=, <>,>=,<=)

ЛС – логическая связка (.AND., .OR.,.XOR.)

Возможно применение операции отрицания .NOT.

Строковые значения берутся в кавычки “ ”. Сравнение строк производится с учётом регистра.

Для изменения порядка вычисления используются скобки ()

ПРИМЕР

UPDATE employee SET salary=25000 WHERE name=“ Петров”

**DELETE** FROM TABLE [WHERE логическое\_выражение]

Команда логически удаляет их таблицы записи (помечает ка удалённые), удовлетворяющие логическому выражению. Если фраза WHERE не указана, удаляются все записи.

Логическое выражение – элементарное\_выражение[ЛСэлементарное\_выражение[ЛС……]]

Элементарное\_выражение - имя\_поляКОПзначение

КОП – обозначение операции сравнения (=, <>,>=,<=)

ЛС – логическая связка (.AND., .OR.,.XOR.)

Возможно применение операции отрицания .NOT.

Строковые значения берутся в кавычки “ ”. Сравнение строк производится с учётом регистра.

Для изменения порядка вычисления используются скобки ()

ПРИМЕР

DELETE FROM employee WHERE name=“ Петров”

**SELECT** \* / список полей через запятую FROM имя\_таблицы [WHERE логическое\_выражение]

Команда выбирает из указанной таблицы записи, удовлетворяющие логическому выражению. Если фраза WHERE не указана, выбираются все записи.

\* - означает все поля таблицы.

Результат команды выводится в виде таблицы. Первая строка содержит имена выбранных или всех полей (названия столбцов),

Остальные строки содержат значения выбранных полей. Ширина ячейки определяется размером поля.

Логическое выражение – элементарное\_выражение[ЛСэлементарное\_выражение[ЛС……]]

Элементарное\_выражение - имя\_поляКОПзначение

КОП – обозначение операции сравнения (=, <>,>=,<=)

ЛС – логическая связка (.AND., .OR.,.XOR.)

Возможно применение операции отрицания .NOT.

Строковые значения берутся в кавычки “ ”. Сравнение строк производится с учётом регистра.

Для изменения порядка вычисления используются скобки ()

ПРИМЕР

SELECT name, salare FROM employee WHERE name=“ Петров”

**TRUNCATE** имя\_таблицы

Команда физически удаляет из таблицы с указанным именем записи, отмеченные командой DELETE, сжимая файл так, освободившееся пространство смещается в конец файла.

ПРИМЕР

TRUNCATE employee

**DROP** TABLE имя\_таблицы

Команда удаляет таблицу (файл) с указанным именем. Имя может включать путь к файлу

3 После успешного выполнения команды выводится подсказка

SQL>

4 Контролируемые ошибки:

Неверная команда

Нет такой таблицы (файла)

Нет такого поля

Несоответствие значения типу поля

Ошибка в выражении

Не могу открыть таблицу (файл)

Не могу создать таблицу (файл)