

## Terminal

Терминал поддерживает следующие команды:

**save [ path\_to\_file ]** - сохранение текущей сети в файл. Если путь к файлу не указан, то создается файл *semnet.db* в текущей директории.

**exit** — выход из терминала.

**size** — показывает количество узлов в сети

**import view\_node for view\_parent\_node [ -L locale\_code ] [ paramName=paramValue ... ]** - внедрение в сеть нового узла, где **view\_node** — это представление нового узла, **view\_parent\_node** — представление родительского понятия, **locale\_code** — код языка для которого введено представление узла и в конце может быть сколько угодно параметров с их значением. Пары имя=значение необходимо разделять пробелами.

**select value by [ id ] | [ param paramName ]** - выборка узлов происходит либо по *id* либо по заданному значению параметра, имя которого указывается в соответствующем разделе.

**restore [ path\_to\_file ]** - восстановление семантической сети из файла.

**find value [ in where ]** - нечеткий поиск значения. Если раздел **in** не задан то ищем в представлениях узлов.

### **Примеры использования.**

Для запуска необходимо перейти в директорию с проектом и ввести в командной строке:

```
ant console
```

В директории с проектом находится файл с тестовой базой данных. Для ее загрузки введите в запущенном терминале команду

```
restore phys.db;
```

Все команды терминала необходимо заканчивать точкой с запятой (исключение — **exit**).

После загрузки сети из файла попробуем сделать следующее:

1. Посмотреть сколько узлов сейчас в сети а затем сразу сохранить сеть в другой файл. Для этого можем одной строкой ввести:

```
size; save /home/user/networks/otherFileName.example;
```

после чего на экран будет выведено сообщение с количеством узлов и сообщении о состоянии сохранения файла.

2. Сейчас у нас загружена семантическая сеть элементарных частиц стандартной модели. Давайте найдем все частицы, которые относятся ко второму поколению. Для этого введем:

```
select 2 by param поколение;
```

т. е. мы делаем запрос на выборку узлов в которых значение 2 принимает параметр с именем «поколение».

3. Далее попробуем посмотреть все информацию об узле «мезон»:

```
select мезон by param display;
```

В выводе терминала можно посмотреть, что к параметра понятия «мезон» отнесена «область» применения — физика, а так же, что у данного понятия есть более общее родительское понятие, id узла которого равен 1. Ссылки на дочерние узлы так же указаны. Попробуем выбрать один из дочерних узлов по его id:

```
select 33 by id;
```

Вывод показывает, что это понятие «каон».

4. Далее найдем все калибрующие бозоны (их имена имеют вид: ?-бозон, где ? - латинская буква). Для этого введем следующее

```
find x-бозон;
```

Вывод данной команды позволяет отследить все дерево понятий до его корня.

5. Попробуем добавить новую частицу в существующую сеть. Пусть это будет позитрон. Для этого введем:

```
import позитрон for лептон область=физика заряд=+1 поколение=1  
тип=античастица;
```

для проверки, выберем все античастицы:

```
select античастица by param тип;
```

Еще несколько строк для примеров:

*select -1/3e by param заряд;* - выборка всех частиц с зарядом  $-1/3 \cdot e$

*find +x/3e in заряд;* - все частицы, заряд которых исчисляется в третьих долях от заряда  $e$ .

*select 2 by param спин;* - частицы, имеющие спин равный двум.