

Тема 4: Система за управление на бази от данни

Реализирайте проста СУБД, която поддържа работа с множество таблици. Всяка от тях се състои от произволен брой колони, всяка от които може да е от тип число, дата, или символен низ. Всяка колона има име – символен низ. Цялата информация за вашата СУБД трябва да се съхранява на диска във формат, който трябва да проектирате и опишете като част от решението.

При работа с тази СУБД трябва да можете да използвате следните команди:

1. За работа с таблици:

- a. **CreateTable** - създава нова таблица по подадено име и списък от имената и типовете на съставлящите я колони; Трябва да има възможност за задаване на **стойности по подразбиране** или **автоматично-генерирани стойности**. Също за някоя колона може да укажете да бъде **създаден индекс** с цел по-бързо търсене в нея.
- b. **DropTable** - премахва таблица по нейното име;
- c. **ListTables** - извежда списък от имената на всички налични таблици;
- d. **TableInfo** - извежда информация (схема и брой записи, заемано пространство и др.) за таблица по подадено име.

2. За работа с данните:

. **Select** - Изпълнява заявка за извличане на данни от вашата система. Трябва да се поддържа **WHERE** клауза при която можете да филтрирате записи чрез изрази включващи оператори за сравнение (равенство и наредба) върху посочените колони. Трябва да можете да сглобявате по-сложни изрази чрез логически операции (**AND**, **NOT** и **OR**) и скоби. Също така трябва да можете да задавате подредба на резултатните редове чрез **ORDER BY** и да премахвате повторения, чрез **DISTINCT**. Не се изисква поддържане на **JOIN**, но реализацията му ще носи допълнителни точки;

a. **Remove** - Премахва определени редове от таблица. Те се посочват с **WHERE** клауза, според описанието за **Select**;

b. **Insert** - Добавя един или повече нови редове в таблица.

Трябва да имате възможност да работите със системата през прост конзолен интерфейс. Трябва да прихващате всякакви грешки по време на работа. Също при стартиране на програмата или чрез специална команда трябва да проверявате дали данните, които сте съхранили не са повредени по някакъв начин (например заради външна намеса или лош сектор на диска).

Вашата база данни може да стане по-голяма от количеството оперативна памет. Въпреки това, опитайте да използвате оперативната памет за да може операциите да бъдат максимално бързи. Така решения, които съхраняват изцяло съдържанието на базата в паметта, не са коректни, както и такива, които работят изцяло върху диска, без да

съхраняват никаква информация в паметта, чрез която да реализират операциите по-бързо.

Примерна употреба:

```
FMISql> CreateTable Sample (ID:Int, Name:String, Value:Int) Index ON ID
```

Table Sample created!

```
FMISql> ListTables
```

There is 1 table in the database:

Sample

```
FMISql> TableInfo Simple
```

There is no such table!

```
FMISql> TableInfo Sample
```

Table Sample : (ID:Int, Indexed; Name:String; Value:Int)

Total 0 rows (0 KB data) in the table

```
FMISql> Insert INTO Sample {(1, "Test", 1), (2, "something else", 100)}
```

Two rows inserted.

```
FMISql> Select Name FROM Sample WHERE ID != 5 AND Value < 50
```

```
| Name |
```

```
-----
```

```
| "Test" |
```

Total 1 row selected

```
FMISql> Select * FROM Sample WHERE ID != 5 AND Name > "Baba" ORDER BY Name
```

```
| ID |      Name      | Val |
```

```
-----
```

```
| 2 | "something else" | 100 |
```

```
| 1 |      "Test"     |   1 |
```

Total 2 rows selected

```
FMISql> Quit
```

Goodbye