

Лабораторная работа №4

Разработка прикладной программы с использованием NoSQL базы данных

1. План выполнения работы

В данной лабораторной работе выполняется программирование прикладной программы с использованием прикладного интерфейса СУБД BerkeleyDB.

Так же, как и в лабораторной работе №1 и №2 прикладная программа должна состоять из серверной части и пользовательского интерфейса. Функционал прикладной программы должен быть такой же, как и в работе №1 и №2.

2. Ход работы

Далее приведен пример основных моментов выполнения работы.

2.1. Технические требования

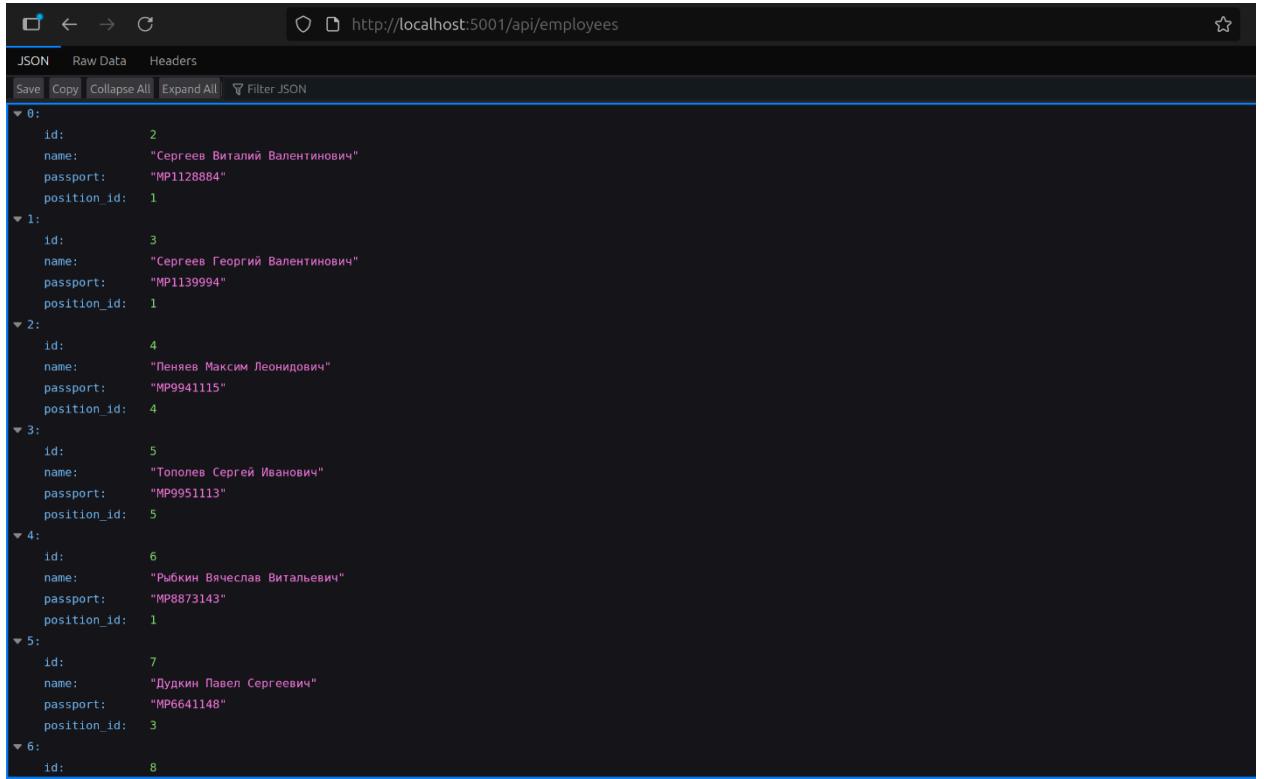
Серверная часть должна взаимодействовать с набором баз данных BerkeleyDB, где каждая база данных представляет отдельную таблицу реляционной схемы из предыдущих работ, а формат хранения данных должен быть таким же, как в работе №3.

В остальном, для данной работы можно выделить такие же технические требования к прикладной программе, как и для работы №1 и №2 исключая требования, предполагающие использование только реляционной базы данных, а требования, относящиеся к таблицам реляционной базы данных, рассматривать как требования к базам данных BerkeleyDB.

2.2. Разработка серверной части

Так же, как и в работе №1 серверная часть представляет из себя HTTP-сервер и имеет такой же функционал, но вместо взаимодействия с PostgreSQL взаимодействие осуществляется с BerkeleyDB. BerkeleyDB не поддерживает SQL, а также различные реляционные механизмы, позволяющие, например, организовывать связи между таблицами. Таким образом, все запросы к базам данных BerkeleyDB, выполняемые серверной частью, необходимо реализовывать на программном уровне. В качестве примера будет показан запрос на получение всех данных из базы данных employees.db. Для этого пользователю необходимо выполнить GET-запрос следующего формата: `http:// {адрес`

хоста} : {порт} /api/employees. Результат запроса в браузере изображен на рисунке 2.1:



The screenshot shows a browser window with the URL `http://localhost:5001/api/employees`. The page displays a JSON array of employee records. Each record contains four fields: id, name, passport, and position_id. The data is presented in a hierarchical tree view where each record is expandable.

id	name	passport	position_id
2	"Сергеев Виталий Валентинович"	"MP1128884"	1
3	"Сергеев Георгий Валентинович"	"MP1139994"	1
4	"Пеняев Максим Леонидович"	"MP9941115"	4
5	"Тополев Сергей Иванович"	"MP9951113"	5
6	"Рыбкин Вячеслав Витальевич"	"MP8873143"	1
7	"Дудкин Павел Сергеевич"	"MP6641148"	3
8			

Рисунок 2.1 – Результат запроса в браузере

2.3. Разработка пользовательского интерфейса

Пользовательский интерфейс остается таким же, как и в работе №2, за исключением того, что операции, выполняемые в отношении таблиц реляционной базы данных, теперь выполняются в отношении баз данных BerkeleyDB.

Таким образом, вместо меню «Tables» теперь используется меню «Databases». Для выбора активной базы данных используется меню «Databases», вызываемое сочетанием клавиш «Alt+d». Вид меню «Databases» изображен на рисунке 2.2.

University DB Admin

File Databases Operations

Filter: lessons
students
employees
subjects
marks
groups
positions
lesson_types
employees_subjects

	group_id	subject_id	lesson_type_id	week	weekday	room
26	6	1	4	4	241	
27	4	1	1	6	270	
28	6	1	2	3	209	
29	5	1	1	2	206	
30	1	4	3	1	337	
31	1	1	4	3	438	
32	6	3	3	1	267	
33	2	3	4	1	202	
34	15	6	3	2	376	
35	16	5	1	3	171	
36	13	1	4	2	453	
37	12	4	4	1	440	
38	9	5	4	3	340	
39	16	4	1	1	364	
40	1	2	4	2	153	
41	16	5	1	4	421	
42	7	6	4	2	347	
43	9	4	3	4	486	
44	1	4	3	2	194	
45	5	2	3	1	412	
46	13	4	3	4	244	
47	5	6	1	2	492	
48	4	1	4	2	294	
49	9	4	1	4	399	
50	8	2	1	3	401	
51	12	2	1	4	250	
52	7	6	4	1	229	
53	14	6	4	4	150	
54	2	2	4	2	438	
55	4	1	4	3	289	
56	13	6	1	1	489	
57	1	6	3	1	7	292
58	7	6	3	4	3	281
59	13	5	4	2	5	324
60	10	2	3	2	7	341
61	5	1	4	2	2	414
62	7	6	2	2	1	256

Рисунок 2.2 – Вид меню «Databases»