

Лабораторная работа №3

Разработка NoSQL базы данных и спецификаций прикладной программы

1. План выполнения работы

Данная лабораторная работа включает:

- освоение прикладного интерфейса СУБД BerkeleyDB;
- разработку конвертора базы данных Postgres в набор баз данных BerkeleyDB;
- разработку требований к приложению.

2. Ход работы

Далее приведен пример основных моментов выполнения работы.

2.1. Технические требования

Для каждой таблицы базы данных PostgreSQL должна быть создана соответствующая база данных BerkeleyDB.

В качестве ключей в базах данных BerkeleyDB должны использоваться первичные ключи таблиц PostgreSQL, а в качестве значений необходимо использовать сериализованные значения столбцов.

Значения столбцов таблиц базы данных PostgreSQL должны быть сериализованы в формате JSON для использования в качестве значений в базе данных BerkeleyDB.

Конвертор должен представлять из себя консольное приложение. При запуске приложение должно выполнить конвертацию и завершиться.

2.2. Разработка конвертора

В таблице 2.1 приведено описание формата хранения данных, конвертированных из таблиц PostgreSQL в базы данных BerkeleyDB. В полях таблицы приведены названия столбцов таблиц PostgreSQL.

Таблица 2.1 – Формат хранения данных в BerkeleyDB

Таблица PostgreSQL	BerkeleyDB	
	Ключ	Значение (JSON)
students	id	{"employee_id", "group_id", "name", "passport"}
groups	id	{"number"}

Продолжение таблицы 2.1

marks	id	{"employee_id", "student_id", "subject_id", "mark", "date"}
lessons	id	{"group_id", "subject_id", "lesson_type_id", "week", "weekday", "room"}
employees	id	{"position_id", "name", "passport"}
subjects	id	{"name", "description"}
lesson_types	id	{"name"}
positions	id	{"name"}
employees_subjects	{employee_id}_{subject_id}	{"employee_id", "subject_id"}

Конвертер работает по следующему алгоритму:

1. Подключение к базе данных PostgreSQL.
2. Получение такой информации о таблицах как их название, название столбцов и т.д., а также получение содержимого таблиц.
3. Создание баз данных BerkeleyDB и их заполнение.
4. Закрытие открытых соединений с базами данных.

Для извлечения некоторой информации из PostgreSQL могут использоваться следующие запросы:

1. Запрос для получения имен всех таблиц схемы public:

```
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'public'
```

2. Запрос для получения названия столбцов таблицы table:

```
SELECT column_name
FROM information_schema.columns
WHERE table_name = 'table'
```

3. Запрос для получения данных таблицы table по столбцам, перечисленным в строке columns через запятую:

```
SELECT {columns} FROM table
```

После завершения выполнения приложения для каждой таблицы PostgreSQL будет создана база данных BerkeleyDB и с помощью утилиты db_dump можно просмотреть ее содержимое. В качестве примера, на рисунке 2.1 приведен результат выполнения данной утилиты для базы данных marks.db:

```
VERSION=3
format=print
type=hash
db_pagesize=4096
HEADER=END
11
{"employee_id": 14, "student_id": 11, "subject_id": 2, "mark": 6, "date": "2025-02-25"}
13
{"employee_id": 14, "student_id": 13, "subject_id": 5, "mark": 8, "date": "2025-03-05"}
15
{"employee_id": 22, "student_id": 15, "subject_id": 4, "mark": 9, "date": "2025-03-15"}
17
{"employee_id": 23, "student_id": 16, "subject_id": 1, "mark": 8, "date": "2025-03-25"}
19
{"employee_id": 30, "student_id": 18, "subject_id": 2, "mark": 6, "date": "2025-01-07"}
20
{"employee_id": 32, "student_id": 23, "subject_id": 4, "mark": 9, "date": "2025-01-12"}
22
{"employee_id": 34, "student_id": 23, "subject_id": 1, "mark": 10, "date": "2025-01-22"}
24
{"employee_id": 34, "student_id": 23, "subject_id": 6, "mark": 7, "date": "2025-02-02"}
26
{"employee_id": 41, "student_id": 23, "subject_id": 4, "mark": 8, "date": "2025-02-12"}
```

Рисунок 2.1 – Содержимое marks.db