

Задана база данных, содержащая следующие таблицы:

Таблица **Employees**

Поле	Тип	Ограничения
code	integer	Primary key
name	varchar(24)	Not null
last_name	varchar(32)	Not null
login	varchar(16)	Not null
psw	varchar(16)	

Таблица **EmployeeHistory**

Поле	Тип	Ограничения
id	integer	Primary key
position	varchar(24)	Not null
manager	integer	> 0
hire	date	Not null
dismiss	date	>= hire
code	integer	Foreign key (code)

Пусть определён следующий интерфейс

```
public interface History
{
    String[] getHistory (String name);
    String[] getHistory (int code);
    boolean login (String user, String password);
    boolean logout ();
}
```

В данном интерфейсе методы имеют следующее назначение:

- метод login() обеспечивает регистрацию клиента в системе. Метод проверяет, что в таблице Employees есть запись, поле login которой совпадает со значением параметра метода user, а поле psw с точностью до регистра совпадает со значением параметра метода password. В этом, и только в этом, случае пользователь регистрируется в системе. Метод возвращает значение true при условии, если клиент успешно подключён к системе. В противном случае метод возвращает логическое значение false;
- метод getHistory (int code) позволяет получить историю занятости сотрудника по его коду. Метод получает выборку из таблицы EmployeeHistory, состоящей из всех записей, в которых поле code содержит значение, совпадающее со значением параметра метода code. Метод возвращает массив строк, каждая из которых представляют одну запись таблицы EmployeeHistory;

- метод `getHistory (String name)` позволяет получить историю занятости сотрудника по его фамилии. Метод получает выборку из таблицы `EmployeeHistory`, состоящей из всех записей, в которых поле `last_name` содержит значение, совпадающее со значением параметра метода `name`. Метод возвращает массив строк, каждая из которых представляют одну запись таблицы `EmployeeHistory`;
- метод `logout()` производит отключение пользователя от системы. После вызова этого метода пользователь теряет возможность получать информацию с помощью метода `getHistory()`. Метод возвращает значение `true`, если пользователь успешно отключён от системы. Если метод был вызван для пользователя, который не был подключён к системе, то метод возвращает значение `false`.

Напишите распределенное приложение, которое обеспечивает доступ произвольному количеству клиентов к заданной базе посредством промежуточного сервера. Приложение должно состоять из трех компонентов:

- ✓ Клиентской части приложения, которая обеспечивает пользовательский интерфейс. Клиентская часть приложения должна позволять пользователю приложения опосредованно вызывать все методы, определённые в интерфейсе `History`, и представлять результаты этих вызовов.
- ✓ Серверной части приложения, которая обеспечивает многопользовательский режим работы. Сервер должен обеспечивать подключение клиентов приложения, получение запросов от клиентов, трансляцию клиентских запросов серверу базы данных и передачу результатов запросов от сервера базы данных клиентам.
- ✓ Сервера базы данных, который обеспечивает доступ к базе данных по технологии `JDBC`. Данный компонент должен соответствовать шаблону проектирования `Singleton`.

Вид взаимодействия между отдельными компонентами приложения определяется индивидуально преподавателем.