Q1: **What are the communication methods between microservices in Lagom Framework? What are the uses cases of these methods? Why?**

In the Lagom framework, there are several communication methods between microservices. The appropriate communication method is chosen based on the requirements of the application.

Two types of communication :

* Synchronous Communication : The client sends a request and waits for a response from the service.
* Asynchronous Communication : The client sends a request but it doesn’t wait for a response from the service.

The first one; Service Calls, by using service calls, a microservice can request data or initiate actions from another API. These calls enable the retrieval of info from another service. As service calls are sync.

Second one; Message Broker, Enabling microservices to publish messages and other microservices to consume them by subscribing to the appropriate topic. Message broker are synchronous communication.

Third one; Entity Actions : A microservice can make requests to a specific entity within another microservice.

Last one ; Events : A microservice can disseminate messages to other microservices interested in a specific type of event. It works like message brokers but because of their structure they are simpler to use.

Q2 : **What are the methods to filter a non-primary key columns in CQL. Are they useful for production?**

1. Using the WHERE clause with a comparison operator to filter rows based on the values of non-primary key columns.
2. Using the ALLOW FILTERING option to allow Cassandra to filter rows based on the values of non-primary key columns.
3. Using the IN clause to filter rows based on whether a non-primary key column has a value that appears in a list of values.
4. Using the LIKE clause to filter rows based on whether a non-primary key column matches a specified pattern.
5. Using secondary indexes to index non-primary key columns and filter rows based on the values of those columns.

Q3: **Is denormalization useful for Cassandra architecture? Why?**

Denormalization in Cassandra can improve query performance by adding redundant copies of data to tables or rows. Denormalization allows data to be accessed from a single row or table rather than multiple tables or columns.

Q4: **What are the advantages of event sourcing?**

Event sourcing stores system state changes as a sequence of events, offering auditability, flexibility, simplicity, scalability, and resilience but it may require more complex data storage and processing.

Q5: **In a CQRS(Command Query Responsibility Segregation) paradigm, can we read a record from microservice state? Is this approach the best practice? Why?**

In CQRS, reading and writing are separated into command and query microservices. It is best practice to use the query microservice for reading to avoid tight coupling and optimize performance.

Q6: **Suppose an application that has many Lagom Microservices uses Cassandra Database(no-sql) for readside. How can you gather(make relation) of two different microservice’s data tables?**

To relate data from two different microservices in a Lagom application using Cassandra, consider using a materialized view, the WITH clause, a UDF( User-Defined Function ) or a UDA(User-Defined Aggregate). The best approach depends on the specific requirements and data involved.