1. Important Topics learnt:

1.1. Java:

1- Working with spring boot repositories, services, controllers etc.
2- Injection dependencies (@bean, @autowired)
3- Threads (thread pooling mainly) and Sockets (HttpClient and similar) using Java.
4- Database connection manager (JdbcTemplate).
5- ActiveMQ message queueing system and implemented it.
6- Logging using log4j in advanced techniques.
7- The mvc model of spring and daos and similar stuff.
8- Loading the changeable and constants data from the database.
9- Transactional annotations in Spring and how do they work.
10- Using maven and fixing the issues in pom.xml and similar.

1.2.Database:

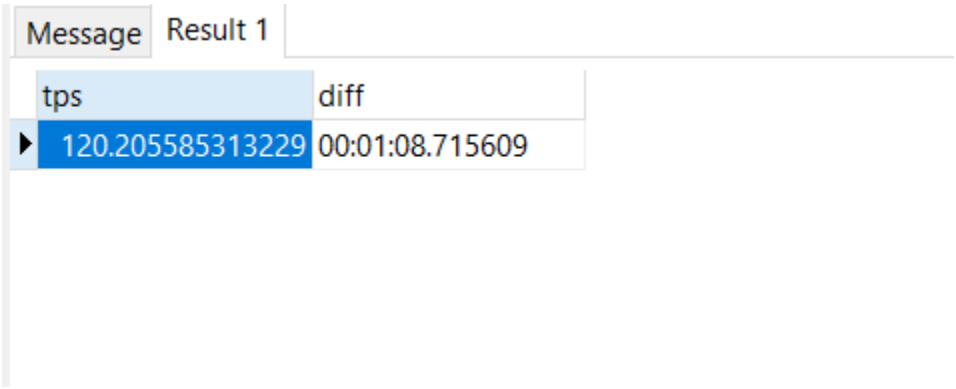
1- Using POSTGRES in general, setting it up and connecting to it.
2- Learned more about joining tables.
3- UPDATE with returning and some new queries.
4- Locking and concurrency in database.
5- Learned more about transactions and their types.
6- Partitioning and sub partitioning.

2. Performance wise

2.1. Before completing the whole structure of the project

Performance TPS used to range up to 350-550 TPS.

2.2. After completing



2.2.1. TPS after the whole project is finished

```
SELECT
  ( COUNT ( * ) / EXTRACT ( EPOCH FROM ( MAX ( subquery.finish_stamp ) - MIN ( subquery.requested_timestamp ) ) ) ) AS tps,
  ( MAX ( subquery.finish_stamp ) - MIN ( subquery.requested_timestamp ) ) AS diff
FROM
  ( SELECT * FROM requests INNER JOIN requests_history ON requests.request_id = requests_history.id ) AS subquery;
--UPDATE requests set processed = FALSE;
```

2.2.2. Query used to measure the TPS

3. Example requests

Link: http://localhost:85/deduct/?channelId=1

Type: POST

Body (JSON):

```
{
    "senderId":"1",
    "receiverId": "2",
    "amount": 50.00
}
```

Channel Id is passed as request parameter