Assignment_Indexing_v4.pdf

Task 1

The database models the employees of a big comapny with a lot of data. It contains the following tables:

- employees
- departments
- cars
- phonecalls
- postalareas

The database contains 5 tables. All but one tables have a relation with the employees table. The only table without a relation is the phonecall table.

Task 2

```
Run | New Tab | JSON
SELECT FirstName, Salary FROM employee WHERE LastName = 'Virtanen'; 11ms
```

```
Run | New Tab | JSON
SELECT Sum(Price) FROM phonecall WHERE `PhoneNumber` = '041-951114' 1.7s
```

```
Run | New Tab | JSON
SELECT phonecall.`CallID`, phonecall.`Price`
FROM employee
JOIN phonecall ON phonecall.`PhoneNumber` = employee.`PhoneNumber`
WHERE employee.`LastName` = 'Virtanen' 1s
```

The queries are slow due to the lack of indexes. The queries are slow because the database has to scan the whole table to find the requested data.

Task 3

```
Run | New Tab
ALTER TABLE employee ADD key (`LastName`); 303ms
```

```
Run | New Tab | JSON
SELECT FirstName, Salary FROM employee WHERE LastName = 'Virtanen'; 2ms
```

We've speed up the querie by about 8ms by adding an index to the LastName field.

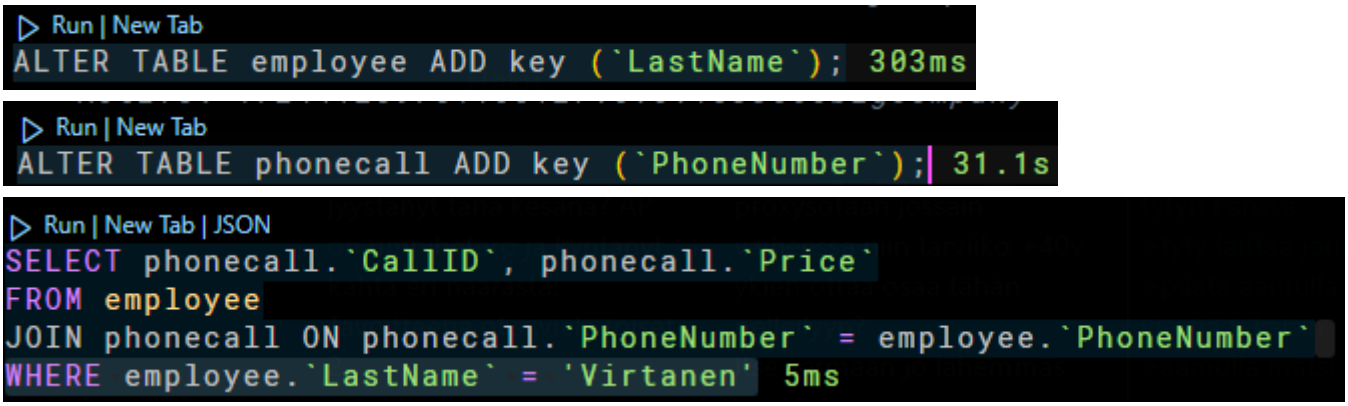
Task 4

```
Run | New Tab
ALTER TABLE phonecall ADD key (`PhoneNumber`); 31.1s
```

```
Run | New Tab | JSON
SELECT Sum(Price) FROM phonecall WHERE `PhoneNumber` = '041-951114' 3ms
```

We've speed up the querie from 1.7s to just 3ms.

Task 5



This query uses the lastname and phonenumbers fields. With these imporovements we've sped ut the query from 1s to 5ms.