

Use **mvn spring-boot:run -Dspring-boot.run.arguments="Who is head of department Math»** to search the head of department

```
2021-12-22 13:22:15.088 INFO 63192 --- [main] c.e.TestProject.TestProjectApplication : Starting request processing!
2021-12-22 13:22:15.089 INFO 63192 --- [main] c.e.TestProject.controller.Controller : Initializing...
2021-12-22 13:22:15.094 INFO 63192 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Searching head of Math department
2021-12-22 13:22:15.105 INFO 63192 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Search successfully completed

SEARCH RESULT :
Alex Alexandrov works in Math department with salary 2000.0 on a Head position.

2021-12-22 13:22:15.238 INFO 63192 --- [main] c.e.TestProject.TestProjectApplication : Program was finished successfully
```

Use **mvn spring-boot:run -Dspring-boot.run.arguments="Show Math statistics»** to Show {department_name} statistics.

```
SEARCH RESULT :
2021-12-22 13:23:38.385 INFO 63278 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Collecting Math department statistics
2021-12-22 13:23:38.517 INFO 63278 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Search successfully completed
Math department has 2 assistant
Math department has 2 associate_professor
Math department has 1 professor
```

Use **mvn spring-boot:run -Dspring-boot.run.arguments="Show the average salary for the department Math»** to Show the average salary for the department {department_name}.

```
2021-12-22 13:24:37.721 INFO 63326 --- [main] c.e.TestProject.TestProjectApplication : Starting request processing!
2021-12-22 13:24:37.721 INFO 63326 --- [main] c.e.TestProject.controller.Controller : Initializing...
2021-12-22 13:24:37.866 INFO 63326 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Counting avarage department Math salary
Alex Alexandrov works in Math department with salary 2000.0 on a Head position.
Max Mustermann works in Math department with salary 500.0 on a Employer position.
Igor Igorov works in Math department with salary 400.0 on a Employer position.
Berthold Hochl works in Math department with salary 600.0 on a Employer position.
Michail Mihailov works in Math department with salary 600.0 on a Employer position.
Total department salary is : 4100.0
2021-12-22 13:24:37.872 INFO 63326 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Search successfully completed

SEARCH RESULT :
Average Math department salary is : 820.0

2021-12-22 13:24:37.876 INFO 63326 --- [main] c.e.TestProject.TestProjectApplication : Program was finished successfully
```

Use **mvn spring-boot:run -Dspring-boot.run.arguments="Show count of employee for Math»** to Show count of employee for {department_name}.

```
2021-12-22 13:25:19.005 INFO 63363 --- [main] c.e.TestProject.TestProjectApplication : Starting request processing!
2021-12-22 13:25:19.005 INFO 63363 --- [main] c.e.TestProject.controller.Controller : Initializing...
2021-12-22 13:25:19.010 INFO 63363 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Counting department Math employers
2021-12-22 13:25:19.151 INFO 63363 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Search successfully completed

SEARCH RESULT :
Math department has : 5 Employers

2021-12-22 13:25:19.155 INFO 63363 --- [main] c.e.TestProject.TestProjectApplication : Program was finished successfully
```

Use **mvn spring-boot:run -Dspring-boot.run.arguments="Global search by go»** for Global Search.

```
SEARCH RESULT :
2021-12-22 13:26:08.515 INFO 63409 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Searching by go keyword
2021-12-22 13:26:08.629 INFO 63409 --- [main] c.e.TestProject.dao.WorkRepositoryImpl : Search completed
Goga | Gogovich | professor
Igor | Igorov | assistant

2021-12-22 13:26:08.631 INFO 63409 --- [main] c.e.TestProject.TestProjectApplication : Program was finished successfully
```

Was chosen MySQL database. I used MAMP as a local server.

The screenshot shows the phpMyAdmin interface in a web browser. The address bar indicates the URL: `localhost:8888/phpMyAdmin5/index.php?route=/database/structure&server=1&db=BotsCrew`. The left sidebar shows the database structure tree with 'BotsCrew' selected. The main panel displays the 'Struktur' (Structure) tab for the 'BotsCrew' database. It lists five tables: 'department', 'employer_department', 'hibernate_sequence', 'lector', and 'work'. Each table row includes icons for actions like 'Anzeigen' (Show), 'Struktur' (Structure), 'Suche' (Search), 'Einfügen' (Insert), 'Leeren' (Empty), and 'Löschen' (Delete). Below the table list, there is a 'Filter' section and a 'Drucken' (Print) button. At the bottom, there is a 'Erzeuge Tabelle' (Create Table) section with fields for 'Name' and 'Anzahl der Spalten' (Number of columns), set to 4.

Tabelle	Aktion	Datensätze	Typ	Kollation	Größe	Überhang
department	[Anzeigen] [Struktur] [Suche] [Einfügen] [Leeren] [Löschen]	4	MyISAM	utf8_general_ci	2,1 KiB	-
employer_department	[Anzeigen] [Struktur] [Suche] [Einfügen] [Leeren] [Löschen]	30	MyISAM	utf8_general_ci	3,5 KiB	-
hibernate_sequence	[Anzeigen] [Struktur] [Suche] [Einfügen] [Leeren] [Löschen]	1	MyISAM	utf8_general_ci	1,0 KiB	-
lector	[Anzeigen] [Struktur] [Suche] [Einfügen] [Leeren] [Löschen]	10	MyISAM	utf8_general_ci	2,4 KiB	-
work	[Anzeigen] [Struktur] [Suche] [Einfügen] [Leeren] [Löschen]	17	MyISAM	utf8_general_ci	4,8 KiB	-
5 Tabellen	Gesamt	62	InnoDB	utf8_general_ci	13,8 KiB	0 B

The screenshot shows the phpMyAdmin interface displaying the data of the 'work' table. The address bar indicates the URL: `localhost:8888/phpMyAdmin5/index.php?route=/sql&db=BotsCrew&table=work&pos=0`. The left sidebar shows the database structure tree with 'BotsCrew' selected. The main panel displays the 'Anzeigen' (View) tab for the 'work' table. It shows a green status bar indicating 'Zeige Datensätze 0 - 16 (17 insgesamt, Die Abfrage dauerte 0.0008 Sekunden.)'. Below this, there is a SQL query editor showing the query: `SELECT * FROM `work``. The table data is displayed in a grid with columns: 'id', 'position', 'salary', 'department_id', and 'lector_id'. The data rows show various employees and their details.

	id	position	salary	department_id	lector_id
<input type="checkbox"/>	15	Employer	1000	1	5
<input type="checkbox"/>	16	Employer	100	2	5
<input type="checkbox"/>	17	Head	2000	3	6
<input type="checkbox"/>	18	Employer	500	3	7
<input type="checkbox"/>	19	Employer	500	4	7
<input type="checkbox"/>	20	Head	3000	1	8
<input type="checkbox"/>	21	Employer	100	4	8
<input type="checkbox"/>	22	Employer	400	3	9
<input type="checkbox"/>	23	Head	2000	4	10
<input type="checkbox"/>	24	Head	1500	2	11
<input type="checkbox"/>	25	Employer	600	4	11
<input type="checkbox"/>	26	Employer	500	4	12
<input type="checkbox"/>	27	Employer	600	3	13
<input type="checkbox"/>	28	Employer	600	1	13
<input type="checkbox"/>	29	Employer	600	3	14
<input type="checkbox"/>	30	Employer	700	1	14
<input type="checkbox"/>	31	Employer	800	1	14

What I wanted to do, but I had not enough time is :

Improve project Structure.

Use Interfaces instead implementations. (Lectors and Departments for example) with adding design patterns.

Use @Query instead of Custom Repositories.

Improve results format and Output process.

Add Error Handling

Cohesion and coupling improvements are also needed.