Laboratory 9 - NoSQL

Author: Piotr Copek, Zuzanna Micorek

Date: 25.06.2025

Task 1. Display the name of the student with identified by 5.

Number of returned items: 1

Query:

```
FOR sub IN subjects
    FILTER s._key == "5"
    RETURN s.STUDENT_NAME
```

Results:

```
[
  "MESUREA"
]
```

Task 2. Display the names of subjects starting with C.

Number of returned items: 2.

Query:

```
FOR sub IN subjects
    FILTER SUBSTRING(sub.SUBJECT_NAME, 0, 1) == "C"
    RETURN sub.SUBJECT_NAME
```

Result:

```
[
  "COMPUTER ARCHITECTURE",
  "COMPUTER NETWORKS"
]
```

Number of returned items: 4

Query:

```
    FILTER s.MAJOR_ID == "6"
    RETURN s.STUDENT_NAME
```

Result:

```
[
  "BANDSMAN",
  "PLAY",
  "GLAIR",
  "COLD"
]
```

Number of returned items: 4

Query:

```
LET studentsWithGrade2 = (
    FOR g IN grades
        FILTER g.GRADE == 2
        RETURN DISTINCT g.STUDENT_ID
)

    FILTER s.GENDER == "F" AND s._key IN studentsWithGrade2
    SORT s.STUDENT_NAME ASC
    RETURN s.STUDENT_NAME
```

Result:

```
[
  "BLACK",
  "DODGE",
  "GLAIR",
  "TURK"
]
```

Task 5. Prepare the list of lectures (subject with type='Lecture') given to COMPUTER CONSTRUCTION students.

Number of returned items: 10

Query:

```
LET compConstStudents = (
    FOR s IN students
        FILTER s.MAJOR_ID == "1"
        RETURN s._key
)

FOR g IN grades
    FILTER g.STUDENT_ID IN compConstStudents
    FOR sub IN subjects
        FILTER sub._key == g.SUBJECT_ID AND sub.SUB_TYPE_ID == "LCT"
        RETURN DISTINCT sub.SUBJECT_NAME
```

Result:

```
[
  "COMPUTER ARCHITECTURE",
  "DIGITAL MODELING",
  "THEORY OF COMPUTER SCIENCE",
  "NUMERICAL METHODS",
  "ORACLE",
  ...
]
```

Task 6. Display in reverse alphabetical order the names of male students who are in major PROCESS CONTROL or ELECTROTECHNOLOGY

Number of returned items: 3

Query:

```
FOR s IN students
    FILTER s.GENDER == "M" AND (s.MAJOR_ID == "7" OR s.MAJOR_ID == "5")
    SORT s.STUDENT_NAME
    RETURN s.STUDENT_NAME
```

Result:

```
[
  "KRACYPAN",
  "REX",
  "WARES"
]
```

Task 7. Display the names of subjects with at least one grade 2 given. The names mustn’t repeat.

Number of returned items: 5

Query:

```
LET subjectsWithGrade2 = (
    FOR g IN grades
        FILTER g.GRADE == 2
        RETURN DISTINCT g.SUBJECT_ID
)

FOR sub IN subjects
    FILTER sub._key IN subjectsWithGrade2
    RETURN sub.SUBJECT_NAME
```

Result:

```
[
  "DATABASES",
  "THEORY OF COMPUTER SCIENCE",
  "ORACLE",
  "TABAKS",
  "DISTRIBUTED DATABASES"
]
```

Task 8. Display the names of subjects with their parent subjects.

Number of returned items:

Query: 4

```
FOR sub IN subjects
    FILTER sub.SUPER_SUB_ID != null

    LET parentSubject = FIRST(
        FOR parent IN subjects
            FILTER parent._key == sub.SUPER_SUB_ID
            RETURN parent.SUBJECT_NAME
    )

    FILTER parentSubject != null

    RETURN {
        subject_name: sub.SUBJECT_NAME,
        super_sub: parentSubject
    }
```

Result:

```
[
  {
    "subject_name": "DATABASES",
    "super_sub": "DATABASES"
  }, ...
]
```

Task 9. Display the number of COMPUTER SCIENCE students.

Number of returned items: 1

Query:

```
RETURN COUNT(
    FOR s IN students
        FILTER s.MAJOR_ID == "3"
        RETURN 1
)
```

Result:

```
[
  15
]
```

Task 10. Display an average of grades of student COLLEGE.

Number of returned items: 1

Query:

```
LET collegeGrades = (
    FOR g IN grades
        FOR s IN students
            FILTER s.STUDENT_NAME == "COLLEGE" AND g.STUDENT_ID == s._key
            RETURN g.GRADE
)
RETURN { avg: AVG(collegeGrades) }
```

Result:


```
[
  {
    "avg": 3.3333333333333335
  }
]
```


Task 11. Display a number of students of MINING MACHINERY major.

Number of returned items: 1

Query:

```
RETURN {
    ent: COUNT(
        FOR s IN students
            FILTER s.MAJOR_ID == "10"
            RETURN 1
    )
}
```

Result:

```
[
  {
    "ent": 4
  }
]
```

Task 12. Display the lowest and the highest grade got by each student having name starting with ‘B’.

Number of returned items: 6

Query:

```
FOR s IN students
    FILTER LEFT(s.STUDENT_NAME, 1) == "B"
    LET studentGrades = (
        FOR g IN grades
            FILTER g.STUDENT_ID == s._key
            RETURN g.GRADE
    )
    FILTER LENGTH(studentGrades) > 0
    RETURN {
        student_id: s._key,
        student_name: s.STUDENT_NAME,
        min: MIN(studentGrades),
        max: MAX(studentGrades)
    }
```

Result:

```
[
  {
    "student_id": "12",
    "student_name": "BLACK",
    "min": 2,
    "max": 5
  }, ...
]
```


Task 13. Display number of different names of students of MINING MACHINERY major.

Number of returned items: 1

Query:

```
LET distinctNames = (
    FOR s IN students
        FILTER s.MAJOR_ID == "10"
        COLLECT name = s.STUDENT_NAME
        RETURN name
)
RETURN { ent: COUNT(distinctNames) }
```

Result:


```
[
  {
    "ent": 2
  }
]
```


Task 14. For each major display the dates of birth of the oldest students.

Number of returned items: 10

Query:

```
FOR s IN students
    COLLECT major_id = s.MAJOR_ID INTO groups
    LET oldest = MIN(
        FOR member IN groups
            RETURN member.s.DATE_OF_BIRTH
    )
    // Look up the major name
    LET major_name = FIRST(
        FOR m IN majors
            FILTER m._key == major_id
            RETURN m.MAJOR_NAME
    )
    RETURN {
        major_id: major_id,
        major_name: major_name,
        min_date_of_birth: oldest
    }
```

Result:

```
[
  {
    "major_id": "1",
    "major_name": "SOFTWARE",
    "min_date_of_birth": "1960-02-25 00:00:00"
  }, ...
]
```

Task 15. For each student with name starting with B display the number of grades that he or she got in each subject.

Number of returned items: 11

Query:

```
FOR s IN students
    FILTER LEFT(s.STUDENT_NAME, 1) == "B"
    FOR g IN grades
        FILTER g.STUDENT_ID == s._key
        FOR sub IN subjects
            FILTER sub._key == g.SUBJECT_ID
            COLLECT studentName = s.STUDENT_NAME,
                subjectName = sub.SUBJECT_NAME INTO groups
            RETURN {
                student_name: studentName,
                subject_name: subjectName,
                count: LENGTH(groups)
            }
```

Result:


```
[
  {
    "student_name": "BANDSMAN",
    "subject_name": "DATABASES",
    "count": 1
  },
  {
    "student_name": "BIGG",
    "subject_name": "ORACLE",
    "count": 2
  }, ...
]
```


Task 16. Display the names of subjects which have more than 8 grades given.

Number of returned items: 6

Query:

```
FOR g IN grades
    COLLECT subjectId = g.SUBJECT_ID INTO groups
    FILTER LENGTH(groups) > 8
    FOR sub IN subjects
        FILTER sub._key == subjectId
        RETURN {
            subject_name: sub.SUBJECT_NAME
        }
```

Result:


```
[
  {
    "subject_name": "DATABASES"
  },
  {
    "subject_name": "ALGEBRA"
  }, ...
]
```


Task 17. For each student from SOFTWARE major display the number of subjects in which he or she has grades (number of subjects not the number of grades!).

Number of returned items: 16

Query:

```
FOR s IN students
    FILTER s.MAJOR_ID == "1"
    LET gradedSubjects = (
        FOR g IN grades
            FILTER g.STUDENT_ID == s._key
            COLLECT subjectId = g.SUBJECT_ID
            RETURN subjectId
    )
    RETURN {
        student_id: s._key,
        student_name: s.STUDENT_NAME,
        count: LENGTH(gradedSubjects)
    }
```

Result:


```
[
  {
    "student_id": "3",
    "student_name": "CAR",
    "count": 1
  },
  {
    "student_id": "4",
    "student_name": "KRAUS",
    "count": 1
  }, ...
]
```


Task 18. Display the names of students who are younger than student CASAN.

Number of returned items: 6

Query:

```
LET casan_dob = FIRST(
    FOR s IN students
        FILTER s.STUDENT_NAME == "CASAN"
        RETURN s.DATE_OF_BIRTH
)

FOR s IN students
    FILTER s.DATE_OF_BIRTH > casan_dob
    RETURN s.STUDENT_NAME
```

Result:

```
[
  "SCRIBBLER",
  "RAGE",
  "MEEK",
  "MILLIONAIRE",
  "MILLIONAIRE",
  "MILLIONAIRE"
]
```

Task 19. Display the names of students whose average of grades is higher than student HAT.

Number of returned items: 6

Query:

```
LET hat_avg = FIRST(
    LET hat_grades = (
        FOR g IN grades
            FOR s IN students
                FILTER s.STUDENT_NAME == "HAT" AND g.STUDENT_ID == s._key
                RETURN g.GRADE
    )
    RETURN AVG(hat_grades)
)

FOR s IN students
    LET student_avg = AVG(
        FOR g IN grades
            FILTER g.STUDENT_ID == s._key
            RETURN g.GRADE
    )
    FILTER student_avg > hat_avg
    RETURN s.STUDENT_NAME
```

Result:

```
[
  "MARSHAL",
  "KRAUS",
  "MESUREA",
  "CASAN",
  "GREY",
  "GREGG"
]
```

Task 20. Display the names of majors having more students than COMPUTER SCIENCE.

Number of returned items: 2

Query:

```
LET cs_count = LENGTH(
    FOR s IN students
        FILTER s.MAJOR_ID == "3"
        RETURN s
)

LET major_counts = (
    FOR s IN students
        COLLECT major_id = s.MAJOR_ID INTO groups
        RETURN {
            major_id: major_id,
            count: LENGTH(groups)
        }
)

FOR m IN majors
    FOR mc IN major_counts
        FILTER m._key == mc.major_id AND mc.count > cs_count
        RETURN m.MAJOR_NAME
```

Result:

```
[
  "SOFTWARE",
  "COMPUTER CONSTRUCTION"
]
```