

--4.1

```
SELECT SUM(participants) AS "Number of participants", course_code FROM CourseInstance
GROUP BY course_code
ORDER BY [Number of participants] DESC ,course_code ASC
```

Number of participants	course_code
63	a450
53	a730
41	a480
14	a290

--4.2

```
SELECT COUNT(*) AS "Number of grades" , student_number FROM CourseGrade WHERE grade <> 0
GROUP BY student_number
ORDER BY [Number of grades] DESC, student_number ASC
```

Number of grades	student_number
2	o210
2	o298
2	o410
2	o473
2	o649
1	o148
1	o348
1	o349
1	o354
1	o548
1	o654

(11 rows affected)

--4.3a

```
SELECT student_number, AVG(grade) AS "Average grade" FROM CourseGrade
GROUP BY student_number
ORDER BY student_number ASC
```

student_number	Average grade
o148	4
o210	2
o298	3
o348	5
o349	4
o354	2
o410	2
o473	1
o548	2
o649	4
o654	5

--4.3b

```
SELECT student_number, CAST(AVG(grade * 1.0) AS decimal(4,2)) AS "Average grade" FROM
CourseGrade
GROUP BY student_number
ORDER BY student_number ASC
```

student_number	Average grade
o148	4.00
o210	2.00
o298	3.00
o348	5.00
o349	4.00
o354	2.00
o410	2.50
o473	1.67
o548	2.00
o649	4.00
o654	5.00

(11 rows affected)

--4.4

```
SELECT teacher_number, COUNT(*) as "Number of course instances" FROM CourseInstance WHERE
teacher_number IS NOT NULL
GROUP BY teacher_number
ORDER BY teacher_number
```

teacher_number	Number of course instances
h180	1
h290	1
h303	5
h430	1
h560	1
h784	1

(6 rows affected)

--4.5

```
SELECT teacher_number, COUNT( DISTINCT course_code) as "Number of course instances" FROM
CourseInstance WHERE teacher_number IS NOT NULL
GROUP BY teacher_number
ORDER BY [Number of course instances] DESC, teacher_number ASC
```

teacher_number	Number of course instances
h303	2
h180	1
h290	1
h430	1
h560	1
h784	1

(6 rows affected)

--4.6

```
SELECT teacher_number, COUNT(*) AS "Number of course instances" FROM CourseInstance
WHERE teacher_number IS NOT NULL
GROUP BY teacher_number HAVING COUNT(*) > 1
ORDER BY teacher_number
```

teacher_number	Number of course instances
h303	5

--4.7

```
SELECT student_number, CAST(AVG(grade*1.0) as DECIMAL(4,2)) AS "Average grade",
MAX(grade) AS "Highest grade", MIN(grade) AS "Lowest grade" FROM CourseGrade
GROUP BY student_number
ORDER BY [Average grade], student_number
```

student_number	Average grade	Highest grade	Lowest grade
o473	1.67	3	0
o210	2.00	3	1
o354	2.00	2	2
o548	2.00	2	2
o410	2.50	3	2
o298	3.00	3	3
o148	4.00	4	4
o349	4.00	4	4
o649	4.00	4	4
o348	5.00	5	5
o654	5.00	5	5

(11 rows affected)

--4.8

```
SELECT student_number, CAST(AVG(grade*1.0) as DECIMAL(4,2)) AS "Average grade",
MAX(grade) AS "Highest grade", MIN(grade) AS "Lowest grade" FROM CourseGrade
GROUP BY student_number HAVING CAST(AVG(grade*1.0) as DECIMAL(4,2)) > 3
ORDER BY [Average grade], student_number
```

student_number	Average grade	Highest grade	Lowest grade
o148	4.00	4	4
o349	4.00	4	4
o649	4.00	4	4
o348	5.00	5	5
o654	5.00	5	5

(5 rows affected)

--4.9

```
SELECT participants, course_code, instance_number FROM CourseInstance
WHERE participants > (SELECT AVG(participants) FROM CourseInstance)
ORDER BY participants DESC, course_code ASC, instance_number ASC
```

participants	course_code	instance_number
25	a480	2

22	a730	3
18	a450	4

(3 rows affected)

--4.10

```
SELECT surname, first_name, student_number FROM Student
WHERE NOT EXISTS (SELECT * FROM CourseGrade WHERE CourseGrade.student_number =
Student.student_number AND grade <> 0)
ORDER BY surname ASC, first_name ASC, student_number ASC
```

surname	first_name	student_number
Hullerus	Greta	o558
Kaarlo	Kuikka	o889

(2 rows affected)

--4.11

```
SELECT surname, first_name, teacher_number, salary FROM Teacher
WHERE salary > (SELECT MAX(salary) FROM Teacher WHERE campus_code = 'c222')
ORDER BY surname ASC, first_name ASC, teacher_number ASC
```

surname	first_name	teacher_number	salary
Kiisseli	Marja	h999	25000.00
Kokki	Seppo	h180	15780.00
Tahko	Olka	h560	15899.00
Vainio	Veera	h784	16049.00
Virta	Emma	h430	18100.00

(5 rows affected)

--4.12

```
SELECT course_code, course_name FROM Course
WHERE EXISTS (SELECT * FROM CourseGrade WHERE CourseGrade.course_code =
Course.course_code AND grade <> 0)
ORDER BY course_code ASC
```

course_code	course_name
a290	Java Basics
a450	Cloud Computing
a480	Data Management
a730	E-business

(4 rows affected)

--4.13

```
SELECT course_code, course_name FROM Course
WHERE NOT EXISTS (SELECT * FROM CourseGrade WHERE CourseGrade.course_code =
Course.course_code AND grade <> 0)
ORDER BY course_code ASC
```

course_code	course_name
a500	Database Developer
a510	Database Design
a800	Finnish 1

(3 rows affected)

--4.14

```
SELECT teacher_number, surname, first_name FROM Teacher
WHERE NOT EXISTS (SELECT * FROM CourseInstance WHERE CourseInstance.teacher_number =
Teacher.teacher_number AND teacher_number IS NOT NULL)
ORDER BY surname ASC, first_name ASC, teacher_number ASC
```

teacher_number	surname	first_name
h999	Kiisseli	Marja
h777	Matikka	Mauri
h714	Tahko	Olka

(3 rows affected)

--4.15

```
SELECT course_name FROM Course WHERE Course.course_code IN
(SELECT CourseInstance.course_code FROM CourseInstance
LEFT JOIN Teacher ON CourseInstance.teacher_number = Teacher.teacher_number
GROUP BY course_code HAVING COUNT(DISTINCT Teacher.campus_code) > 1)
```

course_name
Data Management
E-business

(2 rows affected)

--4.16

```
SELECT gender, CAST(COUNT(*) * 100 * 1.0 / (SELECT COUNT(*) FROM Student) AS
DECIMAL(4,1)) AS "%" FROM Student
GROUP BY gender
ORDER BY [%] DESC
```

gender	%
M	53.8
F	46.2

(2 rows affected)

--4.18

```
WITH TempTable(student_number, avgCOUNT)
AS ( SELECT student_number, CAST(AVG(grade * 1.0) AS decimal(4,2)) AS "Average grade"
FROM CourseGrade
GROUP BY student_number)
SELECT Student_number, avgCOUNT as "Average grade" From TempTable
WHERE avgCOUNT = (SELECT MAX(avgCOUNT) FROM TempTable)
```

Student_number	Average grade
o348	5.00
o654	5.00

(2 rows affected)

--4.19

```
WITH TempTab (decade, teacher_born)
AS (SELECT CONCAT(floor(RIGHT(DATENAME(YEAR,birth_date),2)/10)*10, 's') AS 'decades',
COUNT(*) AS 'teachers born' FROM Teacher GROUP BY
CONCAT(floor(RIGHT(DATENAME(YEAR,birth_date),2)/10)*10, 's'))
SELECT decade, teacher_born AS 'teachers born' FROM TempTab
ORDER BY decade ASC
```

decade	teachers born
40s	1
50s	1
60s	2
70s	3
80s	2

(5 rows affected)

--4.20

```
WITH TempTable(gender,surname, first_name, birth_date, ageCOUNT)
AS (SELECT gender,surname, first_name, birth_date, DATEDIFF(YEAR,birth_date, GETDATE())
AS 'Age' FROM Student)
SELECT gender,surname, first_name, birth_date, ageCOUNT as 'age' FROM TempTable
WHERE ageCOUNT = (SELECT MIN(ageCOUNT) FROM TempTable)
```

gender	surname	first_name	birth_date	age
M	Keto	Matti	1986-05-17	35
M	Leino	Fredrik	1986-11-22	35