Guillaume Minette de Saint Martin

Advanced data Technologies

Lab 4

1. **(First&Mandatory for everyone)** Create a view containing all fields of table STUDENT together with the average grade (field AVG_GRADE) of each student. Name the view STUD_GRADES.

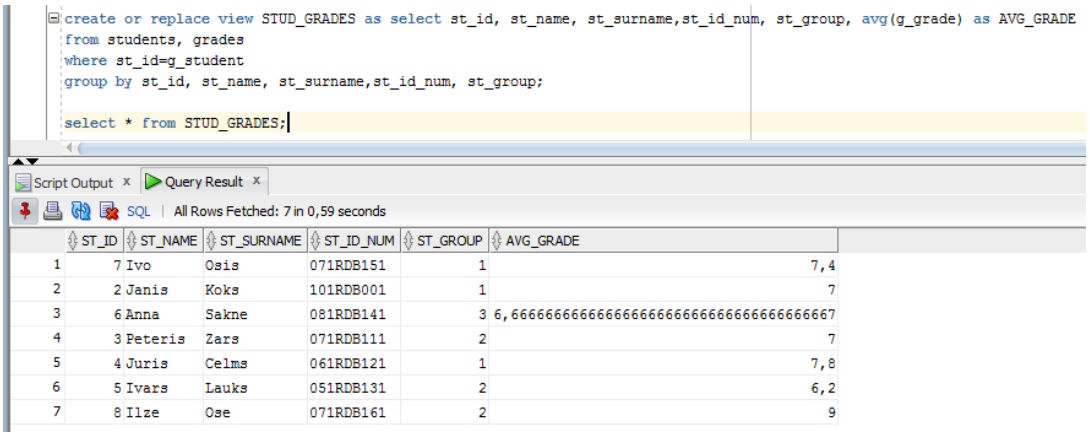
```
create or replace view STUD_GRADES as select st_id, st_name,
st_surname,st_id_num, st_group, avg(g_grade) as AVG_GRADE

from students, grades

where st_id=g_student

group by st_id, st_name, st_surname,st_id_num, st_group;


select * from STUD_GRADES;
```



```
create or replace view STUD_GRADES as select st_id, st_name, st_surname,st_id_num, st_group, avg(g_grade) as AVG_GRADE
from students, grades
where st_id=g_student
group by st_id, st_name, st_surname,st_id_num, st_group;

select * from STUD_GRADES;
```

Script Output | Query Result

SQL | All Rows Fetched: 7 in 0,59 seconds

	ST_ID	ST_NAME	ST_SURNAME	ST_ID_NUM	ST_GROUP	AVG_GRADE
1	7	Ivo	Osis	071RDB151	1	7,4
2	2	Janis	Koks	101RDB001	1	7
3	6	Anna	Sakne	081RDB141	3	6,6666666666666666666666666666666666667
4	3	Peteris	Zars	071RDB111	2	7
5	4	Juris	Celms	061RDB121	1	7,8
6	5	Ivars	Lauks	051RDB131	2	6,2
7	8	Ilze	Ose	071RDB161	2	9

Join tables then group by the displayed columns and add the average.

2. (Easy) Find how many group mates each student has.

```
select st_id, st_name, st_surname, st_group, (select count(
                                                b.st_id)
                                               from students b
                                               where a.st_id<>b.st_id and
                                               b.st_group=a.st_group
                                               ) as Mates
from students a;
```

```
select st_id, st_name, st_surname, st_group, (select count(
                                                b.st_id)
                                               from students b
                                               where a.st_id<>b.st_id and
                                               b.st_group=a.st_group
                                               ) as Mates
from students a;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 7 in 0,04 seconds

	ST_ID	ST_NAME	ST_SURNAME	ST_GROUP	MATES
1	2	Janis	Koks	1	2
2	3	Peteris	Zars	2	2
3	4	Juris	Celms	1	2
4	5	Ivars	Lauks	2	2
5	6	Anna	Sakne	3	0
6	7	Ivo	Osis	1	2
7	8	Ilze	Ose	2	2

Select the count of rows that correspond to the same group and a different id.

3. (Easy) Assign ranks to students of each group by average grade. Student having the highest average grade in each group receives rank 1, the second one receives rank 2, and so on.

```
select st_id, st_name, st_surname,st_id_num,
st_group,avg(g_grade),rank() over (partition by st_group order by
avg(g_grade) DESC ) as Rank

from students, grades

where st_id=g_student

group by st_id, st_name, st_surname,st_id_num, st_group;
```

```
select st_id, st_name, st_surname,st_id_num, st_group,avg(g_grade),rank() over (partition by st_group order by avg(g_grade) DESC ) as Rank
from students, grades
where st_id=g_student
group by st_id, st_name, st_surname,st_id_num, st_group;
```

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 7 in 0,014 seconds

	ST_ID	ST_NAME	ST_SURNAME	ST_ID_NUM	ST_GROUP	AVG(G_GRADE)	RANK
1	4	Juris	Celms	061RDB121	1	7,8	1
2	7	Ivo	Osis	071RDB151	1	7,4	2
3	2	Janis	Koks	101RDB001	1	7	3
4	8	Ilze	Ose	071RDB161	2	9	1
5	3	Peteris	Zars	071RDB111	2	7	2
6	5	Ivars	Lauks	051RDB131	2	6,2	3
7	6	Anna	Sakne	081RDB141	3	6,6666666666666666666666666666666666667	1

Join tables then group by needed displayed columns and use function rank() over a partition of rows using st_group column and order in descendant order.

4. (Easy) Find the difference between each student's average grade and the best average grade in his group.

```
select st_id, st_name, st_surname,st_id_num, st_group,avg(g_grade),
first_value(avg(g_grade)) over (

                    partition by st_group order by avg(g_grade) DESC )-
avg(g_grade) as Difference,

                    rank() over (partition by st_group order by
avg(g_grade) DESC ) as Rank

from students, grades

where st_id=g_student

group by st_id, st_name, st_surname,st_id_num, st_group;
```

```
select st_id, st_name, st_surname,st_id_num, st_group,avg(g_grade), first_value(avg(g_grade)) over (
                    partition by st_group order by avg(g_grade) DESC )-avg(g_grade) as Difference,
                    rank() over (partition by st_group order by avg(g_grade) DESC ) as Rank
from students, grades
where st_id=g_student
group by st_id, st_name, st_surname,st_id_num, st_group;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 7 in 0,013 seconds

	ST_ID	ST_NAME	ST_SURNAME	ST_ID_NUM	ST_GROUP	AVG(G_GRADE)	DIFFERENCE	RANK
1	4	Juris	Celms	061RDB121	1	7,8	0	1
2	7	Ivo	Osis	071RDB151	1	7,4	0,4	2
3	2	Janis	Koks	101RDB001	1	7	0,8	3
4	8	Ilze	Ose	071RDB161	2	9	0	1
5	3	Peteris	Zars	071RDB111	2	7	2	2
6	5	Ivars	Lauks	051RDB131	2	6,2	2,8	3
7	6	Anna	Sakne	081RDB141	3	6,6666666666666666666666666666666666667	0	1

First join tables then use function first_value() with partition over st_group to find the best average grade in this group and compute the difference with the average grade.

5. (Medium) For each student find
    a. Number of students that have at least the same average grade as he/she has.

```
select st_id, st_name, st_surname,st_id_num, st_group,avg(g_grade),
count(*) over (
                    order by avg(g_grade) rows unbounded preceding )-1 as
other
from students a, grades b
where st_id=g_student
group by st_id, st_name, st_surname,st_id_num, st_group;
```

```
select st_id, st_name, st_surname,st_id_num, st_group,avg(g_grade), count(*) over (
                    order by avg(g_grade) rows unbounded preceding )-1 as other
from students a, grades b
where st_id=g_student
group by st_id, st_name, st_surname,st_id_num, st_group;
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 7 in 0,022 seconds

	ST_ID	ST_NAME	ST_SURNAME	ST_ID_NUM	ST_GROUP	AVG(G_GRADE)	OTHER
1	5	Ivars	Lauks	051RDB131	2	6,2	0
2	6	Anna	Sakne	081RDB141	3	6,6666666666666666666666666666666666667	1
3	2	Janis	Koks	101RDB001	1	7	2
4	3	Peteris	Zars	071RDB111	2	7	3
5	7	Ivo	Osis	071RDB151	1	7,4	4
6	4	Juris	Celms	061RDB121	1	7,8	5
7	8	Ilze	Ose	071RDB161	2	9	6

Join tables then group by needed displayed columns and count rows that are contained between the current average grade and the lower ones minus 1 not to count the current rows.

b. Number of the students that have average grade higher than their average grade by at least 1 mark.

```
select st_id, st_name, st_surname,st_id_num, st_group,avg_grade,
count(*) over (

                    order by avg_grade range between 1 following and
unbounded following)  other

from STUD_GRADES a;
```

```
select st_id, st_name, st_surname,st_id_num, st_group,avg_grade, count(*) over (
                    order by avg_grade range between 1 following and unbounded following)  other
from STUD_GRADES a;
```

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 7 in 0,013 seconds

	ST_ID	ST_NAME	ST_SURNAME	ST_ID_NUM	ST_GROUP	AVG_GRADE	OTHER
1	5	Ivars	Lauks	051RDB131	2	6,2	3
2	6	Anna	Sakne	081RDB141	3	6,6666666666666666666666666666666666667	2
3	2	Janis	Koks	101RDB001	1	7	1
4	3	Peteris	Zars	071RDB111	2	7	1
5	7	Ivo	Osis	071RDB151	1	7,4	1
6	4	Juris	Celms	061RDB121	1	7,8	1
7	8	Ilze	Ose	071RDB161	2	9	0

U

By using STUD_GRADES view, count the number of rows that are between the current row - 1 and the following ordering by grade average.

6. (Medium) Find which grades differ the most from the corresponding teacher's average grade.

```
create or replace view TEA_GRADES as select T_ID, T_NAME, T_TITLE,
avg(g_grade) as AVG_GRADE

from teachers, grades, courses

where c_id=g_course and

c_teacher=t_id

group by T_ID, T_NAME, T_FIRSTNAME, T_TITLE;



select unique(teachers.T_ID), teachers.T_NAME,
teachers.T_TITLE,first_value(g_grade) over( partition by teachers.t_id
order by (abs(g_grade)-AVG_GRADE)) as  other

from teachers , grades, courses, TEA_GRADES

where c_id=g_course and

    c_teacher=teachers.t_id and

    teachers.t_id=TEA_GRADES.t_id;
```

First create a view displaying mark average by teacher then join this view with needed tables. Use the rank() function to select the first row of the partition by teachers ordering by the difference of their grades and their average grade.

7. (Hard) How many days after the first exam of each student was passed were the other exams passed by him?

8. (Hard) Find which grades differ more than 1 mark from the highest grade of the corresponding student.

9. (Hard) Divide students into four groups by their average nark. The first 25% of students have to be assigned to the first group, the second ones to the second and so on.