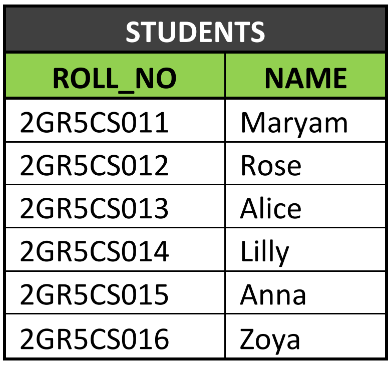
Day 30 – Student Report

Problem Statement:

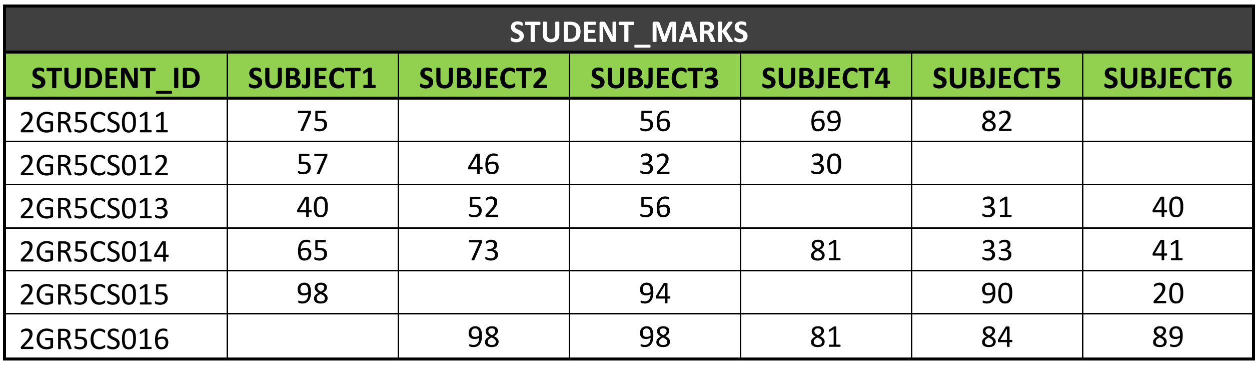
Given tables represent the marks scored by engineering students. **Create a report** to display the following results for each student.
 - Student_id, Student name
 - Total Percentage of all marks
 - Failed subjects (must be comma separated values in case of multiple failed subjects)
 - Result (if percentage >= 70% then 'First Class', if >= 50% & <=70% then 'Second class', if <=50% then 'Third class' else 'Fail'. The result should be Fail if a students fails in any subject irrespective of the percentage marks)

- The sequence of subjects in student_marks table match with the sequential id from subjects table.
- Students have the option to choose either 4 or 5 subjects only.

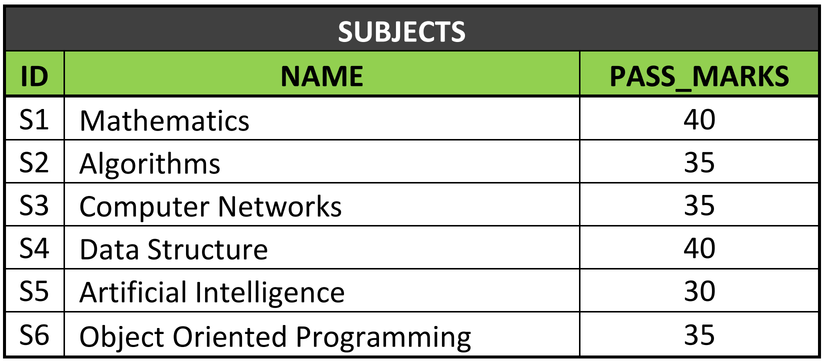
Input:



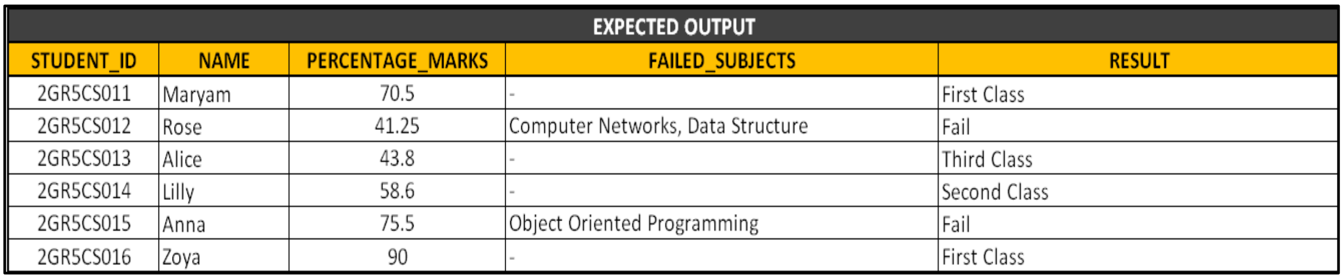
STUDENTS	
ROLL_NO	**NAME**
2GR5CS011	Maryam
2GR5CS012	Rose
2GR5CS013	Alice
2GR5CS014	Lilly
2GR5CS015	Anna
2GR5CS016	Zoya



STUDENT_MARKS						
STUDENT_ID	**SUBJECT1**	**SUBJECT2**	**SUBJECT3**	**SUBJECT4**	**SUBJECT5**	**SUBJECT6**
2GR5CS011	75		56	69	82	
2GR5CS012	57	46	32	30		
2GR5CS013	40	52	56		31	40
2GR5CS014	65	73		81	33	41
2GR5CS015	98		94		90	20
2GR5CS016		98	98	81	84	89



SUBJECTS		
ID	NAME	PASS_MARKS
S1	Mathematics	40
S2	Algorithms	35
S3	Computer Networks	35
S4	Data Structure	40
S5	Artificial Intelligence	30
S6	Object Oriented Programming	35



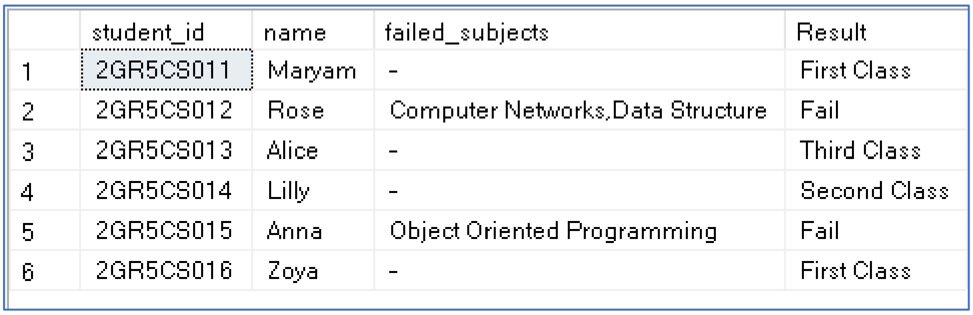
EXPECTED OUTPUT				
STUDENT_ID	NAME	PERCENTAGE_MARKS	FAILED_SUBJECTS	RESULT
2GR5CS011	Maryam	70.5	-	First Class
2GR5CS012	Rose	41.25	Computer Networks, Data Structure	Fail
2GR5CS013	Alice	43.8	-	Third Class
2GR5CS014	Lilly	58.6	-	Second Class
2GR5CS015	Anna	75.5	Object Oriented Programming	Fail
2GR5CS016	Zoya	90	-	First Class

SOLUTION

```
With mainCte as(
select a.id,sj.name,pass_marks
from (
           select name as id,ROW_NUMBER() over(order by name) as rn
           from sys.all_columns
           where object_id = OBJECT_ID('student_marks')
           and name like 'subject%'
      ) a
join (
           select *, ROW_NUMBER() over(order by id) as rn1
           from subjects
           where name is not null) sj
on sj.rn1=a.rn
),
finalCte as( --unpivoting

           select sm.student_id, m.marks,m.subjects
           from student_marks as sm
           CROSS APPLY
                      (
                    VALUES('subject1',subject1),
                          ('subject2',subject2),
                          ('subject3',subject3),
                             ('subject4',subject4),
                             ('subject5',subject5),
                             ('subject6',subject6)
```

```
        ) AS m(subjects,marks)


  ), LastCte as(

        select student_id,st.name,
        AVG(marks) as percentage_marks,
        STRING_AGG(case when marks<pass_marks then mct.name
                   else null end,',') as failed_subjects
        from finalCte as fct
        join mainCte as mct on mct.id=fct.subjects
        join students as st on fct.student_id=st.roll_no
        group by student_id,st.name
  )
  select student_id,name,
  ISNULL(failed_subjects,'-') as failed_subjects,
  case when failed_subjects IS not null then 'Fail'
        when percentage_marks>=70 then 'First Class'
        when percentage_marks>=50 and percentage_marks<=70 then
  'Second Class'
        when percentage_marks<=50 then 'Third Class'
        end as Result
from LastCte
```

FINAL OUTPUT

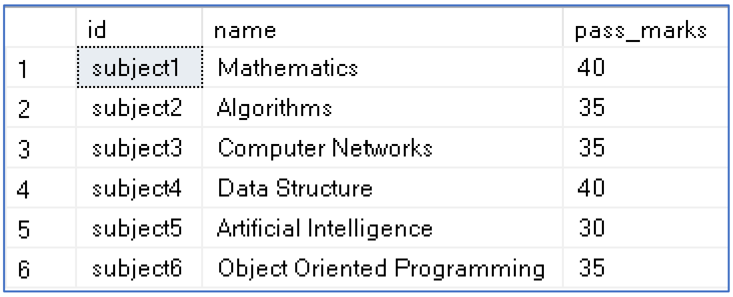


	student_id	name	failed_subjects	Result
1	2GR5CS011	Maryam	-	First Class
2	2GR5CS012	Rose	Computer Networks,Data Structure	Fail
3	2GR5CS013	Alice	-	Third Class
4	2GR5CS014	Lilly	-	Second Class
5	2GR5CS015	Anna	Object Oriented Programming	Fail
6	2GR5CS016	Zoya	-	First Class

SOLUTION WALKTHROUGH

1. In MainCte, we are matching student_marks columns with subjects table rows
   a. Extract all the column names from “student_marks” table along with row_number associated with each subject
   b. We need to join above result with subjects table to match the rows from subjects table with columns of student_marks table. Since, there is no common column in both tables, we will use row_number in both queries as joining clause.

```
With mainCte
as(
select a.id,sj.name,pass_marks
from (
           -- Query 1
           select name as id,ROW_NUMBER() over(order by name) as rn
           from sys.all_columns
           where object_id = OBJECT_ID('student_marks')
           and name like 'subject%'
     ) a
join (
           -- Query 2
           select *, ROW_NUMBER() over(order by id) as rn1
           from subjects
           where name is not null
     ) sj
on sj.rn1=a.rn
```

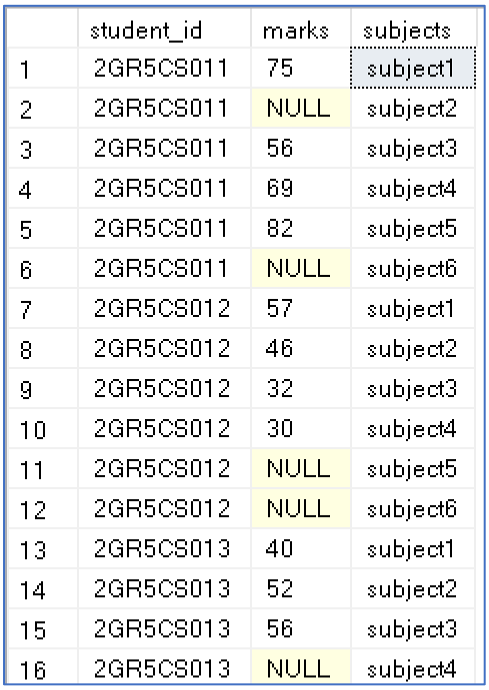


	id	name	pass_marks
1	subject1	Mathematics	40
2	subject2	Algorithms	35
3	subject3	Computer Networks	35
4	subject4	Data Structure	40
5	subject5	Artificial Intelligence	30
6	subject6	Object Oriented Programming	35

2. In FinalCte, we are unpivoting columns to rows so that we can map marks to each subject along with pass_marks later on.

Note: We cannot use **UNPIVOT** function here since it removes the NULL values when unpivoting the column, but we need those null values for comparison, so we will use **CROSS APPLY**

```
--unpivoting
finalCte as(

                 select sm.student_id, m.marks,m.subjects
                 from student_marks as sm
                 CROSS APPLY
                       (

                 VALUES('subject1',subject1),
                           ('subject2',subject2),
                           ('subject3',subject3),
                            ('subject4',subject4),
                            ('subject5',subject5),
                            ('subject6',subject6)
                       ) AS m(subjects,marks)
```

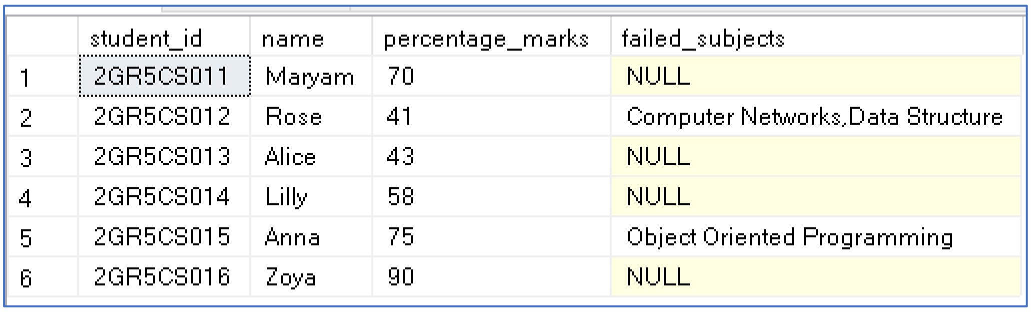


	student_id	marks	subjects
1	2GR5CS011	75	subject1
2	2GR5CS011	NULL	subject2
3	2GR5CS011	56	subject3
4	2GR5CS011	69	subject4
5	2GR5CS011	82	subject5
6	2GR5CS011	NULL	subject6
7	2GR5CS012	57	subject1
8	2GR5CS012	46	subject2
9	2GR5CS012	32	subject3
10	2GR5CS012	30	subject4
11	2GR5CS012	NULL	subject5
12	2GR5CS012	NULL	subject6
13	2GR5CS013	40	subject1
14	2GR5CS013	52	subject2
15	2GR5CS013	56	subject3
16	2GR5CS013	NULL	subject4

3. LastCte:
    a. Here, we are joining above 2 CTEs along with students table(to fetch name of student).
    b. Based on the condition, when marks are less than pass_marks, we consider it a failed subject and aggregate it with a comma.
    c. Find average of the marks scored by each student. Unike normal average calculation, we should not divide all the marks total by 6(total subject count), but we should take into consideration only those subjects for which no NULLs are present.

```
select student_id,st.name,
AVG(marks) as percentage_marks,
STRING_AGG(case when marks<pass_marks then mct.name else null
end,',') as failed_subjects
```

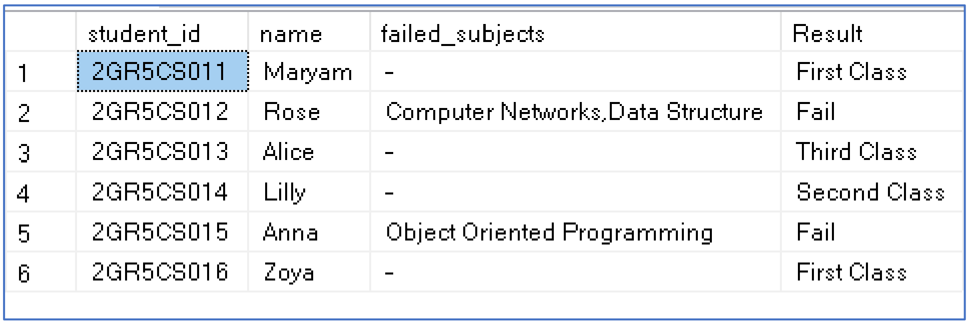
```
from finalCte as fct
join mainCte as mct on mct.id=fct.subjects
join students as st on fct.student_id=st.roll_no
 group by student_id,st.name
```



	student_id	name	percentage_marks	failed_subjects
1	2GR5CS011	Maryam	70	NULL
2	2GR5CS012	Rose	41	Computer Networks,Data Structure
3	2GR5CS013	Alice	43	NULL
4	2GR5CS014	Lilly	58	NULL
5	2GR5CS015	Anna	75	Object Oriented Programming
6	2GR5CS016	Zoya	90	NULL

4. Final Select Query:
   Find the Result based on conditions given in Problem statement.

```
select student_id,name,ISNULL(failed_subjects,'-') as
failed_subjects,
case when failed_subjects IS not null then 'Fail'
      when percentage_marks>=70 then 'First Class'
      when percentage_marks>=50 and percentage_marks<=70 then
'Second Class'
      when percentage_marks<=50 then 'Third Class'
      end as Result
 from LastCte
```



	student_id	name	failed_subjects	Result
1	2GR5CS011	Maryam	-	First Class
2	2GR5CS012	Rose	Computer Networks,Data Structure	Fail
3	2GR5CS013	Alice	-	Third Class
4	2GR5CS014	Lilly	-	Second Class
5	2GR5CS015	Anna	Object Oriented Programming	Fail
6	2GR5CS016	Zoya	-	First Class