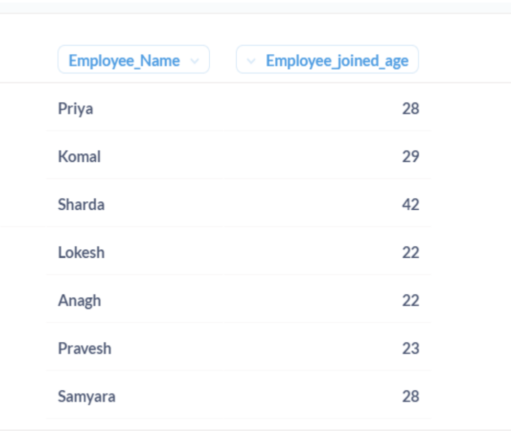
School Management Project SQL (Education Analytics)

Datasets Used:

```
select * from employee_d
select * from course_d
select * from class_d
select * from student_d
select * from ratings_d
```

#Identify the age of the employee when they joined the school within the Employee table. [Hint: Use Employee_Birthdate and Employee_since columns].

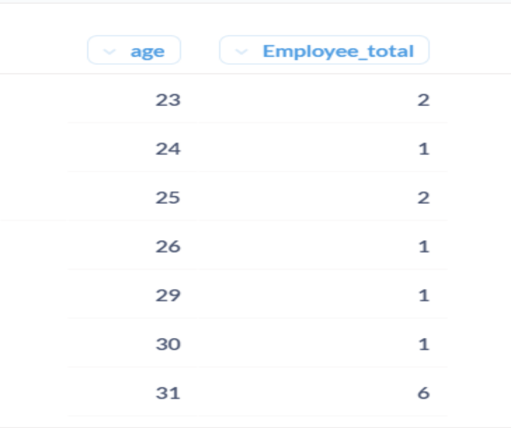
```
select *, (Employee_since - Year(Employee_Birthdate)) as Employee_joined_age
from employee_d
```



Employee_Name	Employee_joined_age
Priya	28
Komal	29
Sharda	42
Lokesh	22
Anagh	22
Pravesh	23
Samyara	28

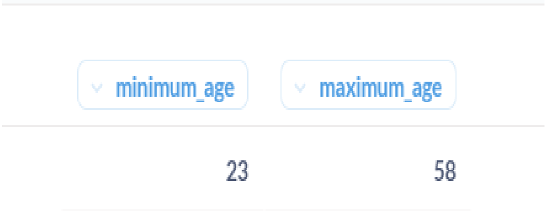
#Calculate the total employees belonging to each age/age-group.

```
select age, count(*) as Employee_total
from employee_d
group by age
order by age asc
```



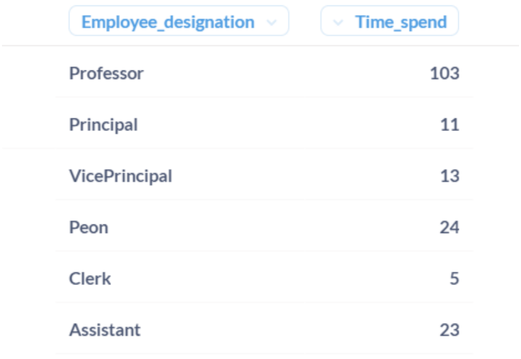
age	Employee_total
23	2
24	1
25	2
26	1
29	1
30	1
31	6

```
#Calculate the min and max of age of employees.

select min(age) as minimum_age,
       max(age) as maximum_age
from employee_d
```



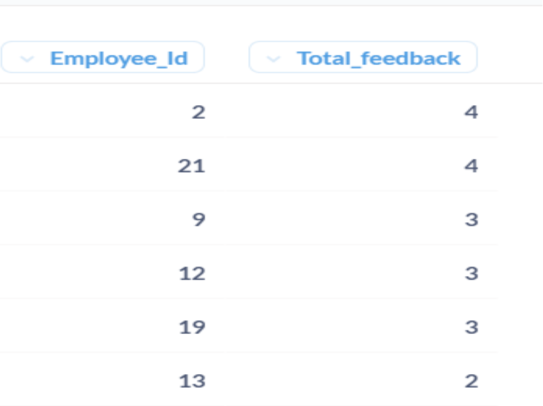
minimum_age	maximum_age
23	58

```
#Identify the time spent by employees in school grouped by
#Their designation

select Employee_designation, sum(age-Employee_joined_age) as Time_spend
from
(select *, (Employee_since - Year(Employee_Birthdate)) as Employee_joined_age
 from employee_d) t1
 group by Employee_designation
```



Employee_designation	Time_spend
Professor	103
Principal	11
VicePrincipal	13
Peon	24
Clerk	5
Assistant	23

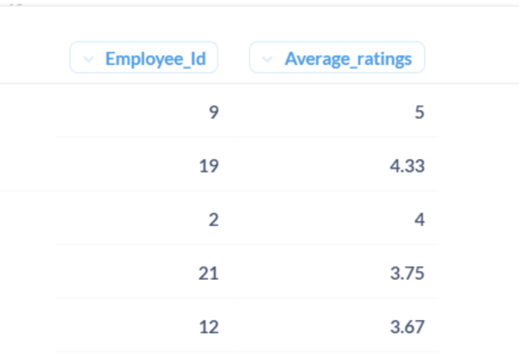
```
#Calculate following feedback statistics:
#Total number of feedbacks for an employee on employee id

select a.Employee_Id, count(b.Rating) as Total_feedback
from employee_d_a a
join ratings_d b
on a.Employee_Id = b.Employee_Id
group by a.Employee_Id
order by Total_feedback desc
```



Employee_Id	Total_feedback
2	4
21	4
9	3
12	3
19	3
13	2

#Average rating of an employee having at least 3 feedbacks.

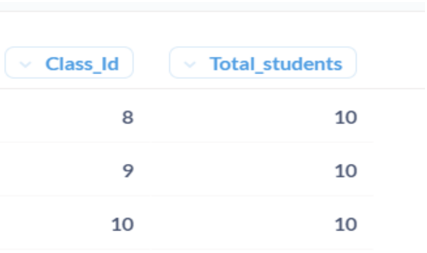
```
select a.Employee_Id, avg(b.rating) as Average_ratings
from employee_d a
join ratings_d b
on a.Employee_Id = b.Employee_Id
   join (select count(Rating) as Total_feedback , Employee_Id
         from ratings_d
         group by Employee_Id) c
   on b.Employee_Id = c.Employee_Id
where c.Total_feedback >= 3
group by a.Employee_Id
order by Average_ratings desc
```



Employee_Id	Average_ratings
9	5
19	4.33
2	4
21	3.75
12	3.67

#Identify the total number of students by
#Class_Id

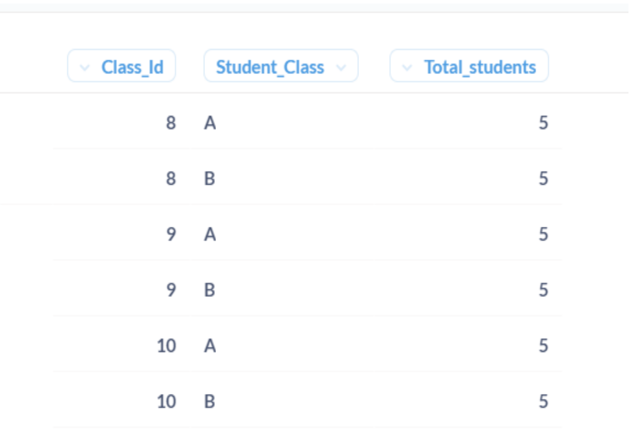
```
select Class_Id, count(Student_Id) as Total_students
from student_d
group by Class_Id
```



Class_Id	Total_students
8	10
9	10
10	10

#Class_Id and Student_Class

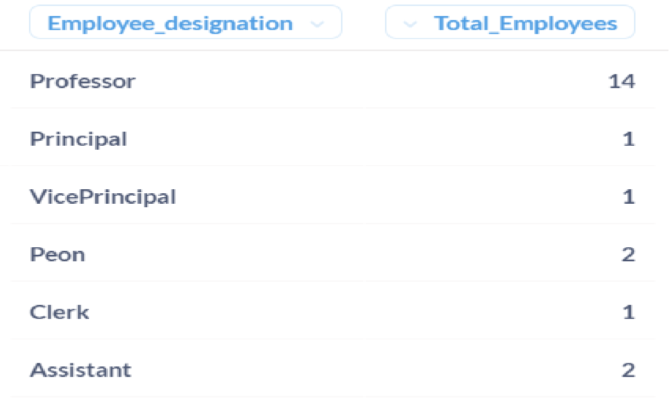
```
select Class_Id, Student_Class, count(Student_Id) as Total_students
from student_d
group by Class_Id, Student_Class
```



Class_Id	Student_Class	Total_students
8	A	5
8	B	5
9	A	5
9	B	5
10	A	5
10	B	5

#Total number of employees by Employee designation

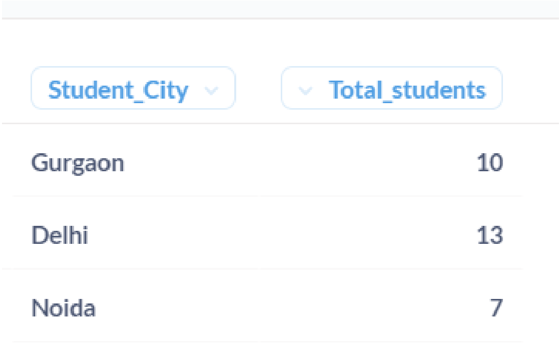
```
select Employee_designation, count(*) as Total_Employees
from employee_d
group by Employee_designation
```



Employee_designation	Total_Employees
Professor	14
Principal	1
VicePrincipal	1
Peon	2
Clerk	1
Assistant	2

#Total number of students by each city

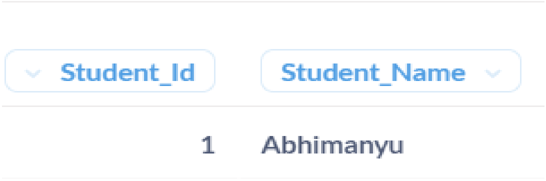
```
select Student_City, count(*) as Total_students
from student_d
group by Student_City
```



Student_City	Total_students
Gurgaon	10
Delhi	13
Noida	7

#Class 8A student from Gurgaon has been a stellar performer whole year. Get the name of the student.

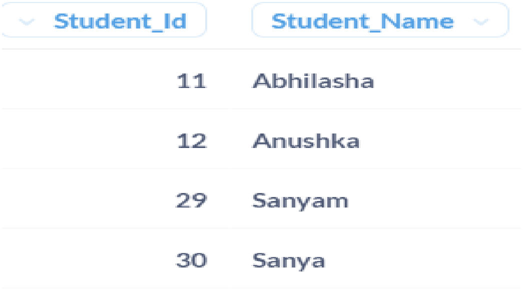
```
select Student_Id, Student_Name
from student_d
where Student_City = 'Gurgaon' and (Class_Id = 8 and Student_Class = 'A')
```



Student_Id	Student_Name
1	Abhimanyu

#Class 9A and 10B students from Delhi are fantastic musicians and just gave an outstanding performance in a national level event. Get the name of the students.
#(Try to solve this query using only AND, OR operation as well. Try to explain where it could fail by applying IN operation)

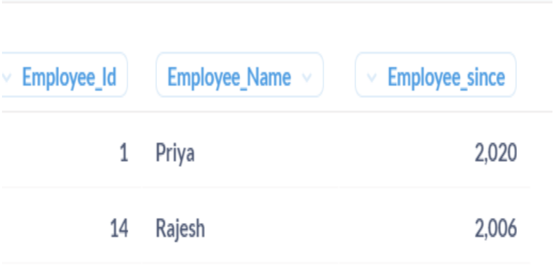
```
select Student_Id, Student_Name
from student_d
where Student_City = 'Delhi' and ((Class_Id = 9 and Student_Class = 'A') or (Class_Id = 10
and Student_Class = 'B'))
```



Student_Id	Student_Name
11	Abhilasha
12	Anushka
29	Sanyam
30	Sanya

#Professor from Gurgaon who are with us since 2006 and 2020 has been a fantastic duo to carry out the science projects
#on state level with school students and got prize from state CM. Get the name of professors.

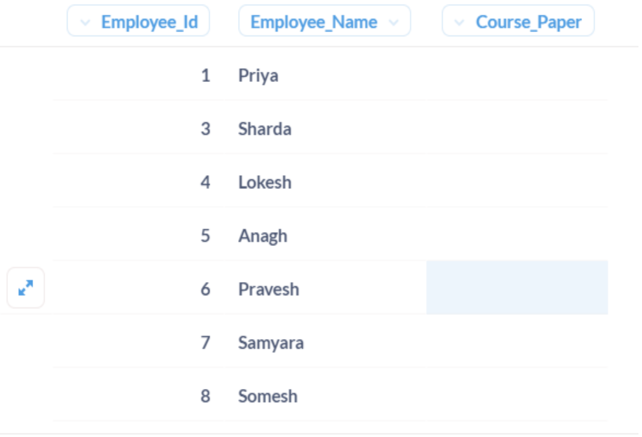
```
select Employee_Id, Employee_Name, Employee_since
from employee_d
where Employee_City = 'Gurgaon' and (Employee_since = 2006 or Employee_since = 2020)
```



Employee_Id	Employee_Name	Employee_since
1	Priya	2,020
14	Rajesh	2,006

#School management wants to identify all the professor names and their Ids that are not mapped with any courses as of now.
#Also, identify professor info who are currently mapped to a course. [Hint: Use DISTINCT with JOINs and IS NULL in WHERE for matching up null values]

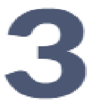
```
select DISTINCT a.Employee_Id, a.Employee_Name, b.Course_Paper
from employee_d a
left join course_d b
on a.Employee_Id = b.Professor_Id
where b.Course_Paper is NULL
```



Employee_Id	Employee_Name	Course_Paper
1	Priya	
3	Sharda	
4	Lokesh	
5	Anagh	
6	Pravesh	
7	Samyara	
8	Somesh	

#Get the total professors that are currently a class teacher.

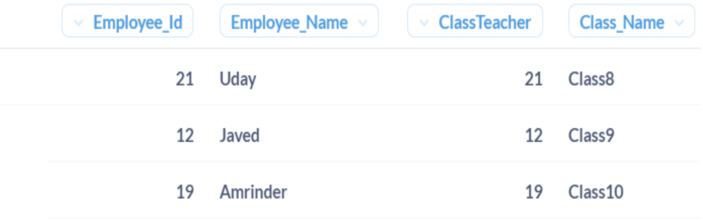
```
select count(*)
from
(select a.Employee_Id, b.ClassTeacher, b.Class_Name
 from employee_d a
 left join class_d b
 on a.Employee_Id = b.ClassTeacher
 where a.Employee_Id = b.ClassTeacher) t1
```



3

#Get the id and name of professors who are currently a class teacher.

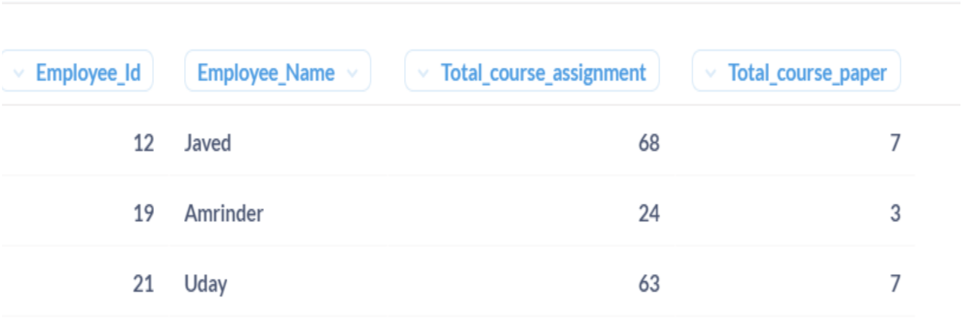
```
select a.Employee_Id, a.Employee_Name, b.ClassTeacher, b.Class_Name
from employee_d a
left join class_d b
on a.Employee_Id = b.ClassTeacher
where a.Employee_Id = b.ClassTeacher
```



Employee_Id	Employee_Name	ClassTeacher	Class_Name
21	Uday	21	Class8
12	Javed	12	Class9
19	Amrinder	19	Class10

#Find the total Assignments and paper by each class teacher in a class.

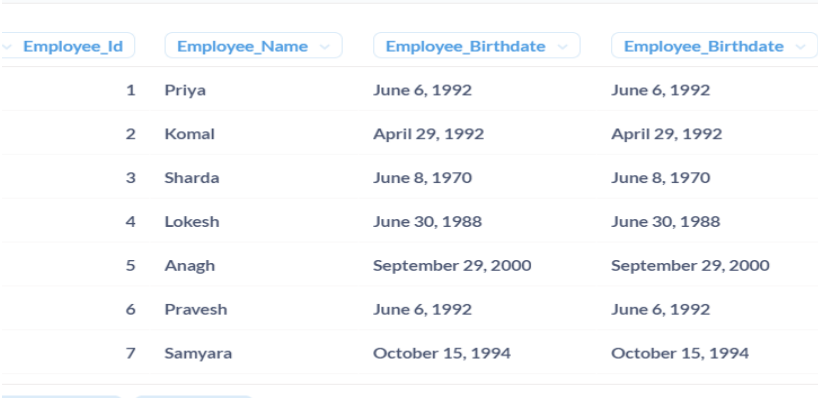
```
select a.Employee_Id, a.Employee_Name, sum(c.Course_Assignments) as
Total_course_assignment, sum(Course_Paper) as Total_course_paper
from employee_d a
join class_d b
on a.Employee_Id = b.ClassTeacher
join course_d c
    on c.Professor_Id = b.ClassTeacher
group by a.Employee_Id, a.Employee_Name
```



Employee_Id	Employee_Name	Total_course_assignment	Total_course_paper
12	Javed	68	7
19	Amrinder	24	3
21	Uday	63	7

#School management wants to know the employees having the birth date on the same day to plan for the leaves they provide to employees.
#Please check which of the 2 employees in employee table have birth date on the same day.
[Hint: Use Self JOIN]

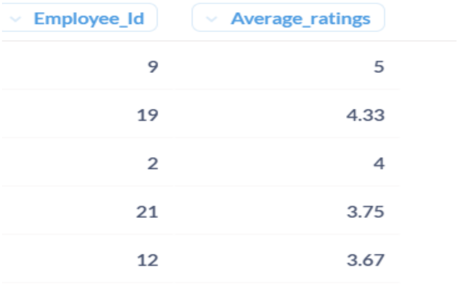
```
select a.Employee_Id, b.Employee_Name, a.Employee_Birthdate, b.Employee_Birthdate
from employee_d a
join employee_d b
on a.Employee_Id = b.Employee_Id
where a.Employee_Birthdate = b.Employee_Birthdate
```



Employee_Id	Employee_Name	Employee_Birthdate	Employee_Birthdate
1	Priya	June 6, 1992	June 6, 1992
2	Komal	April 29, 1992	April 29, 1992
3	Sharda	June 8, 1970	June 8, 1970
4	Lokesh	June 30, 1988	June 30, 1988
5	Anagh	September 29, 2000	September 29, 2000
6	Pravesh	June 6, 1992	June 6, 1992
7	Samyara	October 15, 1994	October 15, 1994

#Get the TOP 2 employees name and their ratings who got the best overall ratings from students.

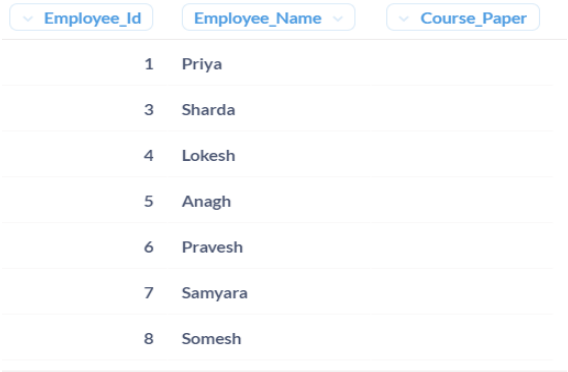
```
select a.Employee_Id, avg(b.rating) as Average_ratings
from employee_d a
join ratings_d b
on a.Employee_Id = b.Employee_Id
   join (select count(Rating) as Total_feedback , Employee_Id
          from ratings_d
          group by Employee_Id) c
   on b.Employee_Id = c.Employee_Id
where c.Total_feedback >= 3
group by a.Employee_Id
order by Average_ratings desc
limit 2
```



Employee_Id	Average_ratings
9	5
19	4.33
2	4
21	3.75
12	3.67

#Get the professors that aren't involved in any courses as of now.

```
select DISTINCT a.Employee_Id, a.Employee_Name, b.Course_Paper
from employee_d a
left join course_d b
on a.Employee_Id = b.Professor_Id
where b.Course_Paper is NULL
```

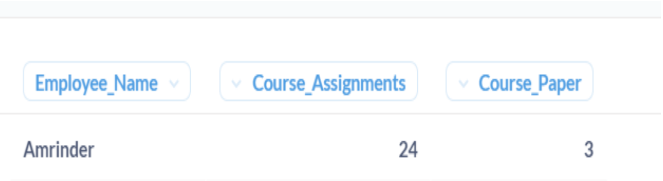


Employee_Id	Employee_Name	Course_Paper
1	Priya	
3	Sharda	
4	Lokesh	
5	Anagh	
6	Pravesh	
7	Samyara	
8	Somesh	

#Professors are busy with assignments and papers they have given to students and may not have the time to attend counselling.

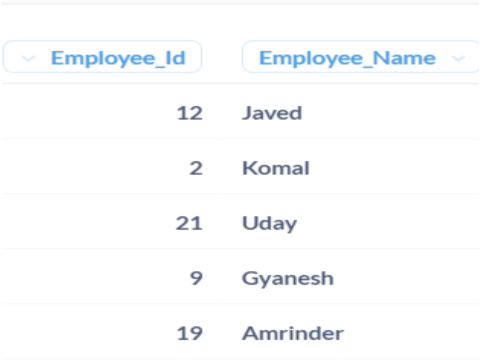
#Get the employee name where the average paper >= 3 and assignments >20.

```
select a.Employee_Name, b.Course_Assignments, b.Course_Paper
from employee_d a
join course_d b
on a.Employee_Id = b.Professor_Id
join (select avg(Course_Paper) as Average_course_paper, Professor_Id
      from course_d
      group by Professor_Id) c
on b.Professor_Id = c.Professor_Id
where c.Average_course_paper >= 3 and b.Course_Assignments > 20
```



Employee_Name	Course_Assignments	Course_Paper
Amrinder	24	3

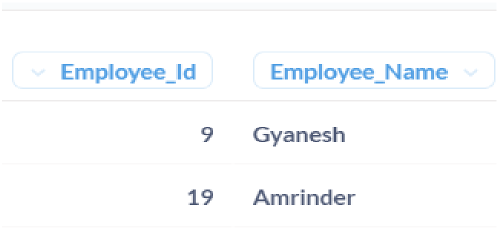
```
#Get the employees that are rated by students. [Solve below 2 with IN and EXIST both
operators]
#Get the employees with more than 2 students ratings. [Note: Identify number of students]

select Employee_Id, Employee_Name
from
(select count(b.Student_Id) as voted_students, a.Employee_Id, a.Employee_Name
from employee_d a
join ratings_d b
on a.Employee_Id = b.Employee_Id
group by a.Employee_Id) t1
where voted_students > 2

#Alternate coding_invaders

select Employee_Id, Employee_Name
from employee_d
where Employee_Id IN     (select Employee_Id
                          from
                         (select count(Student_Id) as voted_students, Employee_Id
                          from ratings_d
                          group by Employee_Id)t1
                           where voted_students > 2 )
```



Employee_Id	Employee_Name
12	Javed
2	Komal
21	Uday
9	Gyanesh
19	Amrinder

#Get the employees with an average rating of more than 4 and are rated by more than 3 students.

```
select Employee_Id, Employee_Name
from employee_d
where Employee_Id IN
                              (select Employee_Id
                              from
                              (select avg(Rating) as Average_ratings, count(Student_Id) as
voted_students, Employee_Id
                              from ratings_d
                              group by Employee_Id)t1
                              where Average_ratings > 4 and voted_students >= 3)
```



Employee_Id	Employee_Name
9	Gyanesh
19	Amrinder