# Normalized Schema Diagram

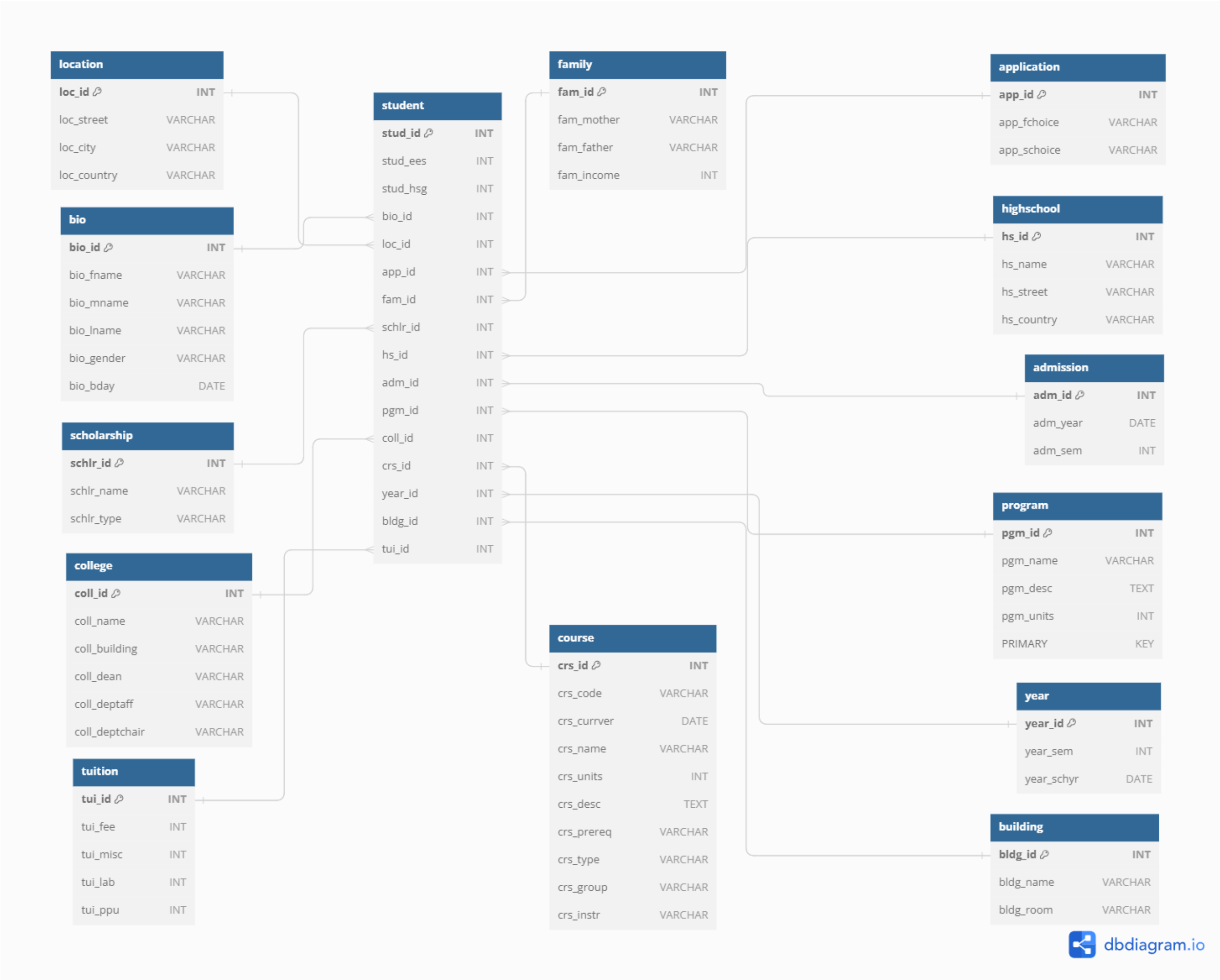


Figure 1.1: This schema was built using the engine of DB Diagram, the schema is available through this link

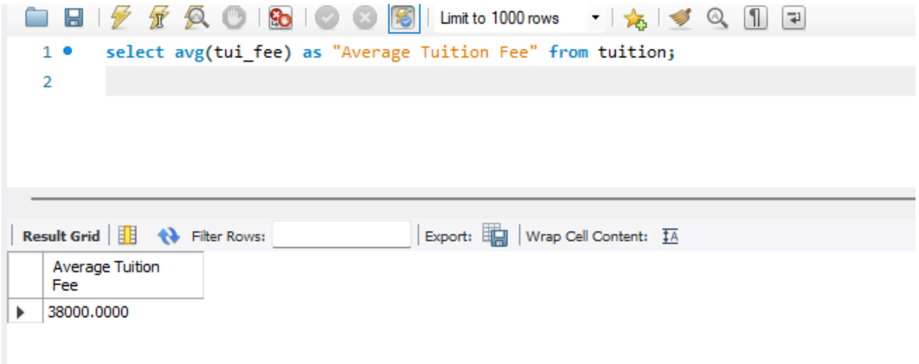
# Questions

## Question 1

What is the average tuition fee?

***Solution:***

```
select avg(tui_fee) as "Average Tuition Fee" from tuition;
```



```
select avg(tui_fee) as "Average Tuition Fee" from tuition;
```

| Average Tuition Fee |
|---|
| 38000.0000 |

Figure 2.1: Question 1 Query and Output

## Question 2

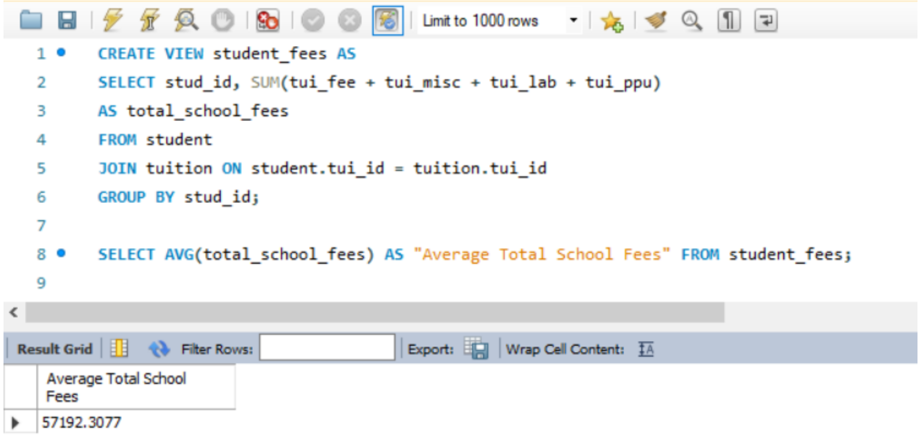
What is the average total school fees?

Using the SUM will add the various components of school fees and grouping it by Student ID will make it one row per student, and using the AVG will get the average total school fees in general

***Solution:***

```
CREATE VIEW student_fees AS
SELECT stud_id, SUM(tui_fee + tui_misc + tui_lab + tui_ppu)
AS total_school_fees
FROM student
JOIN tuition ON student.tui_id = tuition.tui_id
GROUP BY stud_id;

SELECT AVG(total_school_fees) AS "Average Total School Fees" FROM student_fees;
```



```
CREATE VIEW student_fees AS
SELECT stud_id, SUM(tui_fee + tui_misc + tui_lab + tui_ppu)
AS total_school_fees
FROM student
JOIN tuition ON student.tui_id = tuition.tui_id
GROUP BY stud_id;

SELECT AVG(total_school_fees) AS "Average Total School Fees" FROM student_fees;
```

| Average Total School Fees |
|---|
| 57192.3077 |

Figure 2.2: Question 2 Query and Output

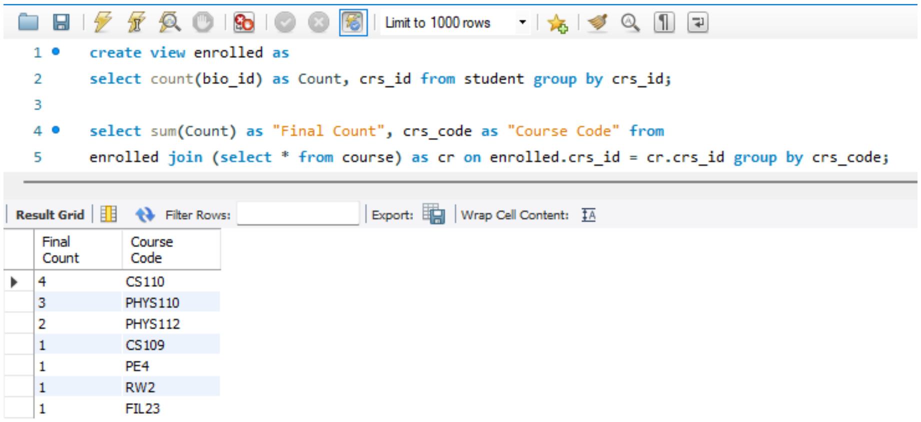
## Question 3

How many students are enrolled per subject?

***Solution:***

```
create view enrolled as
select count(bio_id) as Count, crs_id from student group by crs_id;

select sum(Count) as "Final Count", crs_code as "Course Code" from enrolled
join (select * from course) as cr on enrolled.crs_id = cr.crs_id group by crs_code;
```



```
create view enrolled as
select count(bio_id) as Count, crs_id from student group by crs_id;

select sum(Count) as "Final Count", crs_code as "Course Code" from
enrolled join (select * from course) as cr on enrolled.crs_id = cr.crs_id group by crs_code;
```

| Final Count | Course Code |
|---|---|
| 4 | CS110 |
| 3 | PHYS110 |
| 2 | PHYS112 |
| 1 | CS109 |
| 1 | PE4 |
| 1 | RW2 |
| 1 | FIL23 |

Figure 2.3: Question 3 Query and Output

## Question 4

How many subjects does Taylor Sheesh have?

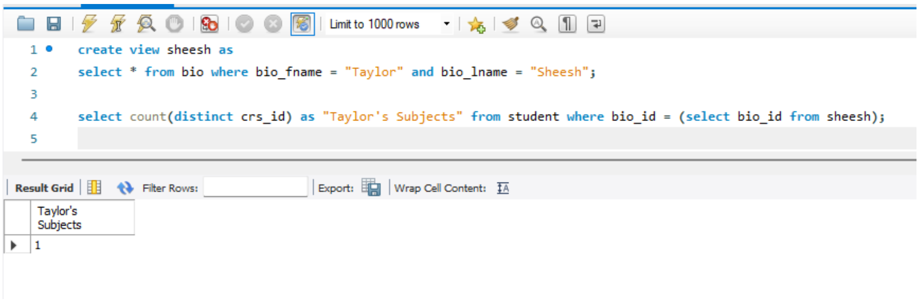
By using the `distinct` keyword to only get the courses that are connected to Taylor Sheesh's `bio_id`, we can acquire the count

***Solution:***

```
create view sheesh as
```

```
select * from bio where bio_fname = "Taylor" and bio_lname = "Sheesh";

select count(distinct crs_id) as "Taylor’s Subjects" from student where
bio_id = (select bio_id from sheesh);
```



```
create view sheesh as
select * from bio where bio_fname = "Taylor" and bio_lname = "Sheesh";

select count(distinct crs_id) as "Taylor's Subjects" from student where bio_id = (select bio_id from sheesh);
```

```
Taylor's
Subjects
1
```

Figure 2.4: Question 4 Query and Output

### Question 5

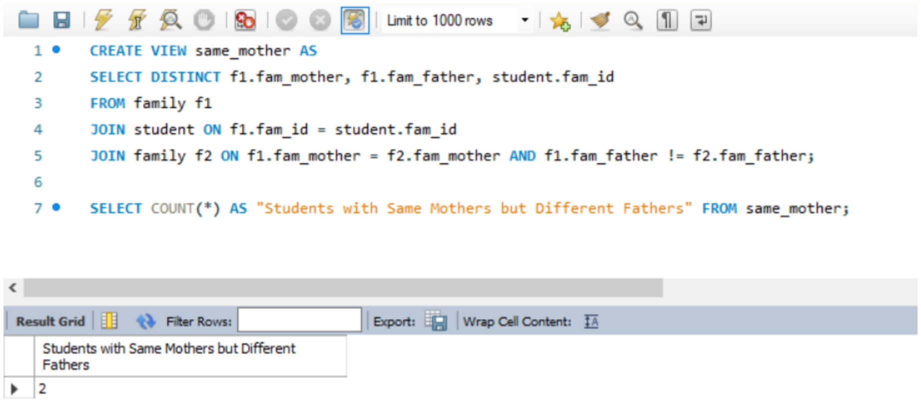
How many students have the same mother but different fathers?

Doing a self-join on family table created two instances to check if they have the same mother but different fathers with the help of the DISTINCT keyword to help eliminate duplicates

***Solution:***

```
CREATE VIEW same_mother AS
SELECT DISTINCT f1.fam_mother, f1.fam_father, student.fam_id
FROM family f1
JOIN student ON f1.fam_id = student.fam_id
JOIN family f2 ON f1.fam_mother = f2.fam_mother AND f1.fam_father != f2.fam_father;

SELECT COUNT(*) AS "Students with Same Mothers but Different Fathers" FROM same_mother;
```



```
CREATE VIEW same_mother AS
SELECT DISTINCT f1.fam_mother, f1.fam_father, student.fam_id
FROM family f1
JOIN student ON f1.fam_id = student.fam_id
JOIN family f2 ON f1.fam_mother = f2.fam_mother AND f1.fam_father != f2.fam_father;

SELECT COUNT(*) AS "Students with Same Mothers but Different Fathers" FROM same_mother;
```

```
Students with Same Mothers but Different
Fathers
2
```

Figure 2.5: Question 5 Query and Output

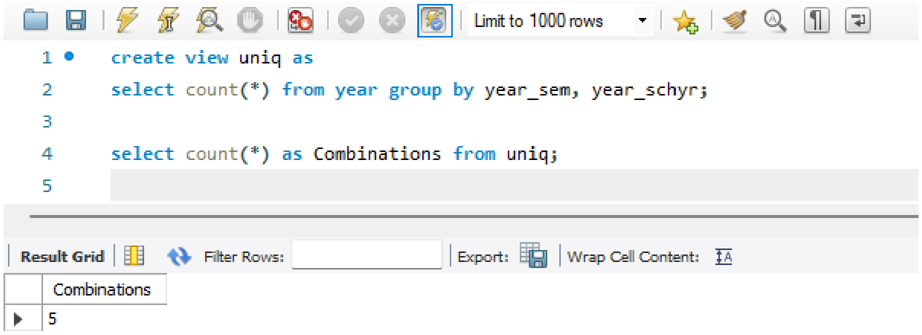
### Question 6

What are the combinations of semester and school year?

***Solution:***

```
create view uniq as
select count(*) from year group by year_sem, year_schyr;

select count(*) as Combinations from uniq;
```



```
create view uniq as
select count(*) from year group by year_sem, year_schyr;

select count(*) as Combinations from uniq;
```

```
Combinations
5
```

Figure 2.6: Question 6 Query and Output

### Question 7

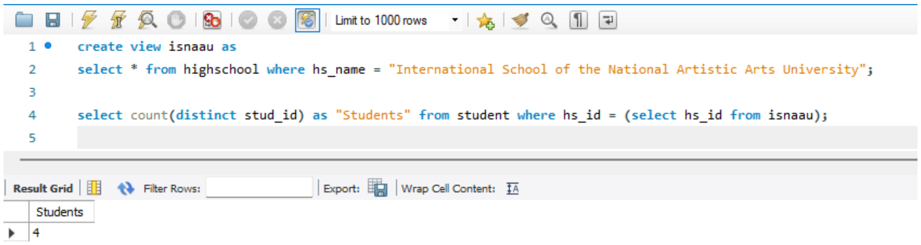
How many students does International School of the National Artistic Arts University have?

The I.S.N.A.A.U. has a total of 4 unique students in the database

***Solution:***

```
create view isnaau as
select * from highschool where hs_name
= "International School of the National Artistic Arts University";

select count(distinct stud_id) as "Students" from student
where hs_id = (select hs_id from isnaau);
```



```
create view isnaau as
select * from highschool where hs_name = "International School of the National Artistic Arts University";

select count(distinct stud_id) as "Students" from student where hs_id = (select hs_id from isnaau);
```

```
Students
4
```

Figure 2.7: Question 7 Query and Output

### Question 8

How many courses are held in all the buildings?

Two views were created: (1) For joining the course and building tables and (2) For eliminating duplicates, then it counted the courses per building it is being held

***Solution:***

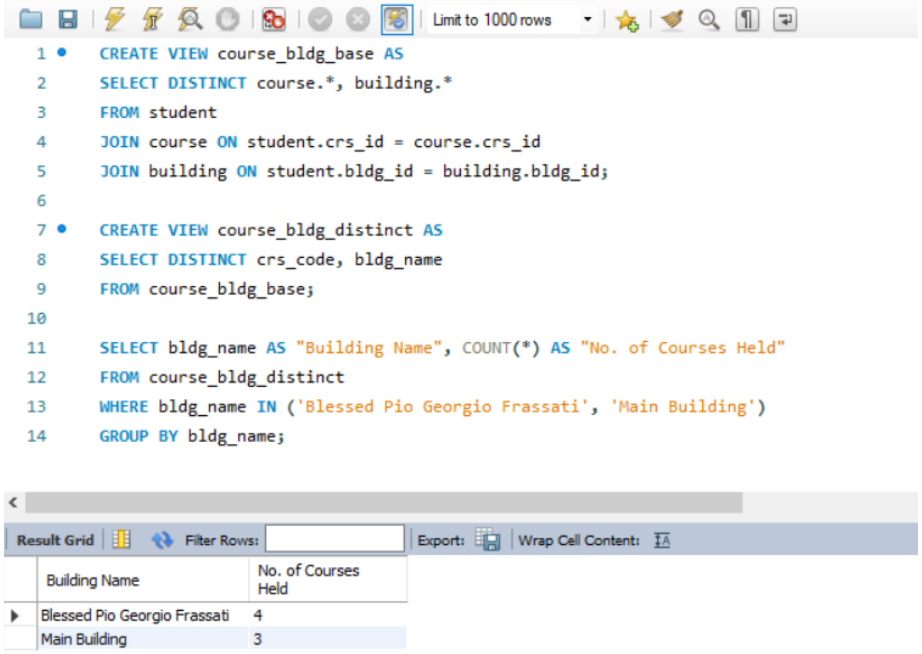
```
CREATE VIEW course_bldg_base AS
```

```
SELECT DISTINCT course.*, building.*
FROM student
JOIN course ON student.crs_id = course.crs_id
JOIN building ON student.bldg_id = building.bldg_id;

CREATE VIEW course_bldg_distinct AS
SELECT DISTINCT crs_code, bldg_name
FROM course_bldg_base;

SELECT bldg_name AS "Building Name", COUNT(*) AS "No. of Courses Held"
FROM course_bldg_distinct
WHERE bldg_name IN (’Blessed Pio Georgio Frassati’, ’Main Building’)
GROUP BY bldg_name;
```

---



```
CREATE VIEW course_bldg_base AS
SELECT DISTINCT course.*, building.*
FROM student
JOIN course ON student.crs_id = course.crs_id
JOIN building ON student.bldg_id = building.bldg_id;

CREATE VIEW course_bldg_distinct AS
SELECT DISTINCT crs_code, bldg_name
FROM course_bldg_base;

SELECT bldg_name AS "Building Name", COUNT(*) AS "No. of Courses Held"
FROM course_bldg_distinct
WHERE bldg_name IN ('Blessed Pio Georgio Frassati', 'Main Building')
GROUP BY bldg_name;
```

| Building Name | No. of Courses Held |
|---|---|
| Blessed Pio Georgio Frassati | 4 |
| Main Building | 3 |

Figure 2.8: Question 8 Query and Output

## Question 9

What is the average family income per scholarship?

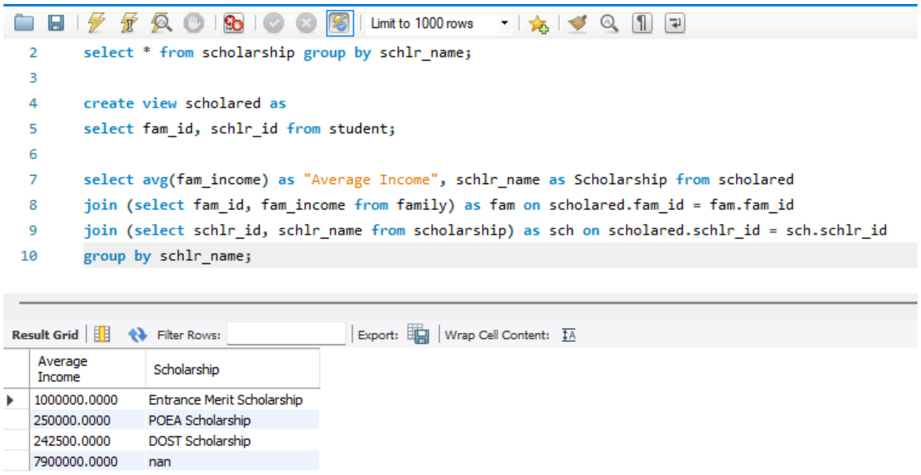
***Solution:***

```
create view scships as
select * from scholarship group by schlr_name;

create view scholared as
select fam_id, schlr_id from student;

select avg(fam_income) as "Average Income", schlr_name as Scholarship from scholared
join (select fam_id, fam_income from family) as fam
    on scholared.fam_id = fam.fam_id
join (select schlr_id, schlr_name from scholarship) as sch
    on scholared.schlr_id = sch.schlr_id
```

```
group by schlr_name;
```



```
select * from scholarship group by schlr_name;

create view scholared as
select fam_id, schlr_id from student;

select avg(fam_income) as "Average Income", schlr_name as Scholarship from scholared
join (select fam_id, fam_income from family) as fam on scholared.fam_id = fam.fam_id
join (select schlr_id, schlr_name from scholarship) as sch on scholared.schlr_id = sch.schlr_id
group by schlr_name;
```

| Average Income | Scholarship |
|---|---|
| 1000000.0000 | Entrance Merit Scholarship |
| 250000.0000 | POEA Scholarship |
| 242500.0000 | DOST Scholarship |
| 7900000.0000 | nan |

Figure 2.9: Question 9 Query and Output

### Question 10

Whose family has the least amount of money left after paying total school fees?

***Solution:***

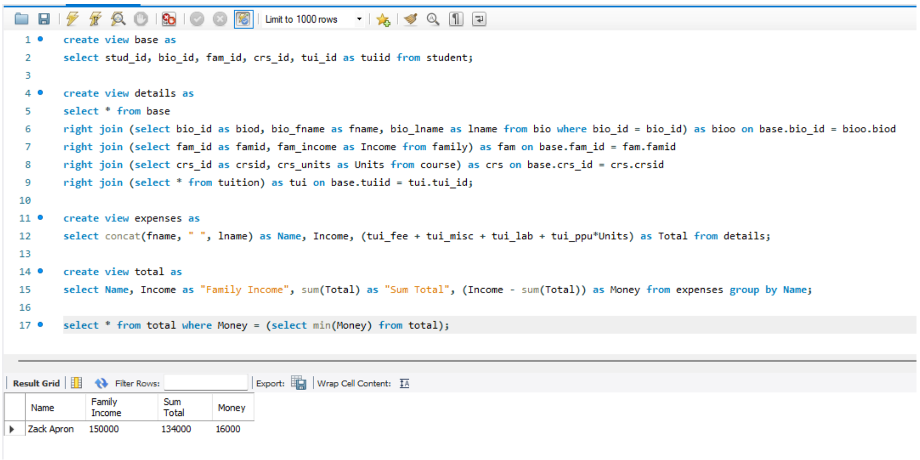
```
create view base as
select stud_id, bio_id, fam_id, crs_id, tui_id as tuiid from student;

create view details as
select * from base
right join
(select bio_id as biod, bio_fname as fname, bio_lname as lname from bio where bio_id = bio_id)
as bioo on base.bio_id = bioo.biod
right join
(select fam_id as famid, fam_income as Income from family) as fam on base.fam_id = fam.famid
right join
(select crs_id as crsid, crs_units as Units from course) as crs on base.crs_id = crs.crsid
right join
(select * from tuition) as tui on base.tuiid = tui.tui_id;

create view expenses as
select concat(fname, " ", lname) as Name, Income,
(tui_fee + tui_misc + tui_lab + tui_ppu*Units) as Total from details;

create view total as
select Name, Income as "Family Income", sum(Total) as "Sum Total", (Income - sum(Total))
as Money from expenses group by Name;

select * from total where Money = (select min(Money) from total);
```



```
create view base as
select stud_id, bio_id, fam_id, crs_id, tui_id as tuiid from student;

create view details as
select * from base
right join (select bio_id as biod, bio_fname as fname, bio_lname as lname from bio where bio_id = bio_id) as bioo on base.bio_id = bioo.biod
right join (select fam_id as famid, fam_income as Income from family) as fam on base.fam_id = fam.famid
right join (select crs_id as crsid, crs_units as Units from course) as crs on base.crs_id = crs.crsid
right join (select * from tuition) as tui on base.tuiid = tui.tui_id;

create view expenses as
select concat(fname, " ", lname) as Name, Income, (tui_fee + tui_misc + tui_lab + tui_ppu*Units) as Total from details;

create view total as
select Name, Income as "Family Income", sum(Total) as "Sum Total", (Income - sum(Total)) as Money from expenses group by Name;

select * from total where Money = (select min(Money) from total);
```

| Name | Family Income | Sum Total | Money |
|---|---|---|---|
| Zack Apron | 150000 | 134000 | 16000 |

Figure 2.10: Question 10 Query and Output

## Question 11

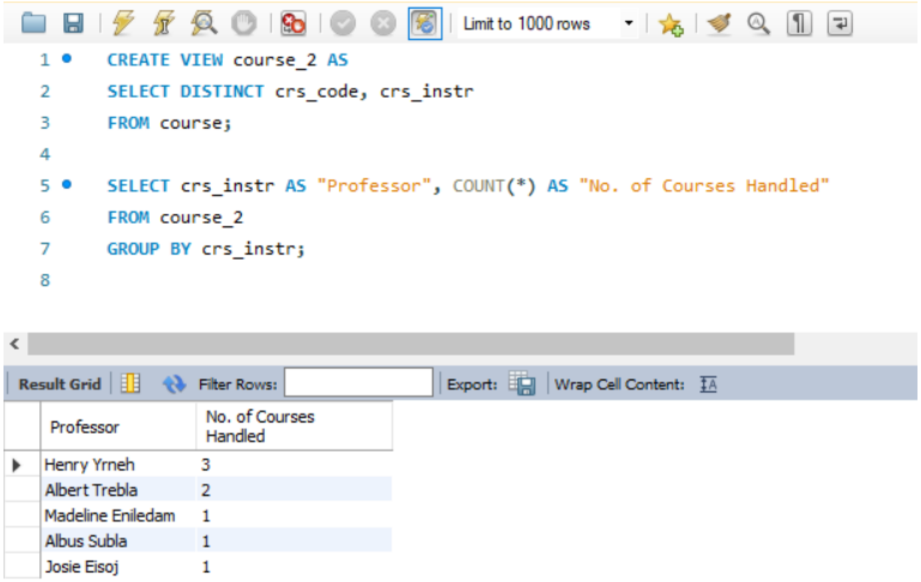
How many courses do each professor handle?

A view was created using the DISTINCT keyword to eliminate duplicate values and be used as the basis for counting the courses each professor handles

***Solution:***

```
CREATE VIEW course_2 AS
SELECT DISTINCT crs_code, crs_instr
FROM course;

SELECT bldg_name AS "Professor", COUNT(*) AS "No. of Courses Handled"
FROM course_2
GROUP BY crs_instr;
```



```
CREATE VIEW course_2 AS
SELECT DISTINCT crs_code, crs_instr
FROM course;

SELECT crs_instr AS "Professor", COUNT(*) AS "No. of Courses Handled"
FROM course_2
GROUP BY crs_instr;
```

| Professor | No. of Courses Handled |
|---|---|
| Henry Yrneh | 3 |
| Albert Trebla | 2 |
| Madeline Eniledam | 1 |
| Albus Subla | 1 |
| Josie Eisoj | 1 |

Figure 2.11: Question 11 Query and Output

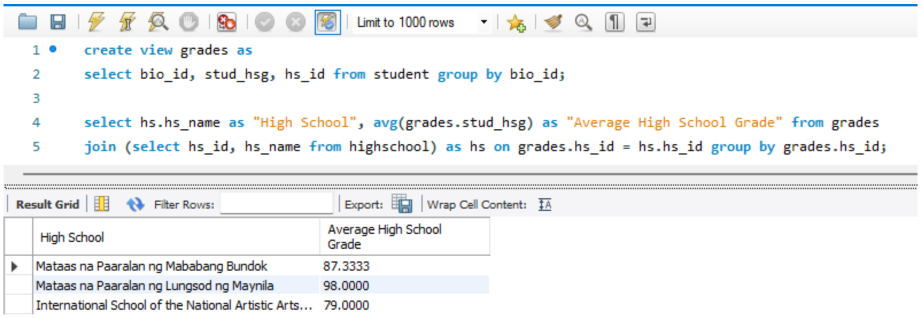
## Question 12

What is the average high school grade for each school?

***Solution:***

```
create view grades as
select bio_id, stud_hsg, hs_id from student group by bio_id;

select hs.hs_name as "High School", avg(grades.stud_hsg) as
"Average High School Grade" from grades
join (select hs_id, hs_name from highschool) as hs
    on grades.hs_id = hs.hs_id group by grades.hs_id;
```



```
create view grades as
select bio_id, stud_hsg, hs_id from student group by bio_id;

select hs.hs_name as "High School", avg(grades.stud_hsg) as "Average High School Grade" from grades
join (select hs_id, hs_name from highschool) as hs on grades.hs_id = hs.hs_id group by grades.hs_id;
```

| High School | Average High School Grade |
|---|---|
| Mataas na Paaralan ng Mababang Bundok | 87.3333 |
| Mataas na Paaralan ng Lungsod ng Maynila | 98.0000 |
| International School of the National Artistic Arts... | 79.0000 |

Figure 2.12: Question 12 Query and Output

# Thoughts and Experience

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.