Friends Query

1. Names of all students who are friends with someone named Gabriel.

Solution.

//

```
select name
from Highschooler h,
        (select id2
        from Highschooler h, Friend f
        where  h.name = 'Gabriel'
        and h.id = f.id1)
where h.id = id2
order by name
```

//

Alexis
Andrew
Cassandra
Jessica
Jordan

2. For every student who likes someone 2 or more grades younger than themselves, return that student's name and grade, and the name and grade of the student they like.

John	12	Haley	10

Solution.

//

```
select h1.name, h1.grade, h2.name, h2.grade
from Highschooler h1, Highschooler h2, Likes l
where h1.id = l.id1 and h2.id = l.id2
```

```
and h1.grade - h2.grade >= 2
//
```

3. For every pair of students who both like each other, return the name and grade of both students. Include each pair only once, with the two names in alphabetical order.

Cassandra	9	Gabriel	9
Jessica	11	Kyle	12

Solution.

```
//
select h1.name, h1.grade, h2.name, h2.grade
from Highschooler h1, Highschooler h2, Likes l1, Likes l2
where h1.id = l1.id1 and h2.id = l1.id2
and h1.id = l2.id2 and h2.id = l2.id1
and h1.name < h2.name
//
```

4. Find all students who do not appear in the Likes table (as a student who likes or is liked) and return their names and grades. Sort by grade, then by name within each grade.

Jordan	9
Tiffany	9
Logan	12

Solution.

```
//
select name, grade
from Highschooler h
where h.id not in
        (select id
        from Highschooler h, Likes l
        where h.id = l.id1
        union
        select id
```

```
          from Highschooler h, Likes l
          where h.id = l.id2)
//
```

5. For every situation where student A likes student B, but we have no information about whom B likes (that is, B does not appear as an ID1 in the Likes table), return A and B's names and grades.

Alexis	11	Kris	10
Austin	11	Jordan	12
Brittany	10	Kris	10
John	12	Haley	10

Solution.

```
//
select h1.name, h1.grade, h2.name, h2.grade
from Highschooler h1, Highschooler h2, Likes l
where h1.id = l.id1 and h2.id = l.id2
and l.id2 not in (select l.id1 from Likes l)
//
```

6. Find names and grades of students who only have friends in the same grade. Return the result sorted by grade, then by name within each grade.

Jordan	9
Brittany	10
Haley	10
Kris	10

Gabriel	11
John	12
Logan	12

Solution.

```
//
select name, grade
from Highschooler h
where h.id not in(select f.id1
                  from Highschooler h1, Highschooler h2, Friend f
                  where h1.id = f.id1 and h2.id = f.id2
                  and h1.grade <> h2.grade)
order by grade, name
//
```

7. For each student A who likes a student B where the two are not friends, find if they have a friend C in common (who can introduce them!). For all such trios, return the name and grade of A, B, and C.

Andrew	10	Cassandra	9	Gabriel	9
Austin	11	Jordan	12	Andrew	10
Austin	11	Jordan	12	Kyle	12

Solution.

```
//
select h1.name, h1.grade, h2.name, h2.grade, h3.name, h3.grade
from Highschooler h1, Highschooler h2, Highschooler h3, Likes l, Friend f1, Friend f2
where h2.id <> (select id1
                 from Friend f
                where h1.id = f.id2)
and h1.id = l.id1 and h2.id = l.id2
and h1.id = f1.id1 and h3.id = f1.id2
and h2.id = f2.id1 and h3.id = f2.id2
```

//

8. Find the difference between the number of students in the school and the number of different first names.

2

Solution.

//

```
select numstu-numname
from(select count(*) as numstu
         from Highschooler h),
         (select count(distinct name) as numname
         from Highschooler h)
```

//

9. Find the name and grade of all students who are liked by more than one other student.

Cassandra	9
Kris	10

Solution.

//

```
select distinct name, grade
from Highschooler h, Likes l
where h.id = l.id2
and id2 in (select id2
              from Likes l
              where id1 not in (select id1
                                  from Likes l
                                   group by id2)
               )
```

//