04b_grouping

Grouping and Aggregates

University DB

```
set search_path to university;
```

1. Find the number of offerings for each year

```
select year, count(oid)
from offering
group by year;
```

2. Same results because `oid` cannot be bull

```
select year, count(*)
from offering
group by year;
```

3. Similar, but different results because of null values

```
select year, count(iid)
from offering
group by year;
```

4. Find the number of course offerings for each instructor

```
select iid, count(*)
from offering
group by iid;
```

5. Get the number of students enrolled in each course offering

```
select oid, count(sid)
from enrollment
group by oid;
```

6. Get the number of students enrolled in each course offering
   - with course codes, names and sections

```
select e.oid, code, name, section, count(sid) as n_students
from enrollment e
     inner join offering o on e.oid = o.oid
```

```
        inner join course c on o.cid = c.cid
    group by e.oid, code, name, section;
```

7. What about the offerings without any enrollments?

```
select o.oid, code, name, section, count(sid) as n_students
from enrollment e
        right join offering o on e.oid = o.oid
        inner join course c on o.cid = c.cid
group by o.oid, code, name, section
order by n_students desc;
```

8. Get the number of students enrolled in each course offering
   - with course codes, names and sections, but only for course offerings with less than 3 enrollments

```
select o.oid, code, name, section, count(sid) as n_students
from enrollment e
     right join offering o on e.oid = o.oid
     inner join course c on o.cid = c.cid
group by o.oid, code, name, section
having count(sid) < 3;
```

9. Same, but with less than 1 enrollment

```
select o.oid, code, name, section, count(sid) as n_students
from enrollment e
        right join offering o on e.oid = o.oid
        inner join course c on o.cid = c.cid
group by o.oid, code, name, section
having count(sid) < 1;
```

10. better

```
select o.oid, code, name, section
from enrollment e
        right join offering o on e.oid = o.oid
        inner join course c on o.cid = c.cid
where e.oid is null;
```

Exercises

1. Get the number of offerings for each instructor
2. Get the number of offerings for instructor with id 1, for each year
3. Get the number of offerings for each instructor for each year
   - what do we do with instructors without any offerings?
4. Get the number of offerings for each instructor for each semester and each year
   - what do we do with semesters without any offerings?
   - what do we do with semesters without offerings for some instructors?
     - for example, many instructors teach only in the fall and winter

semesters, but not in the summer semesters; do we list these off-semesters for each instructor with a count of 0?