-- Melihat semua data di table teachers

select \*

from teachers;

-- Melihat dosen yang salary-nya lebih besar dari 50000

select\*

from teachers

where salary>50000;

-- Menghitung dosen yang salarynya lebih dari 50000

select count(\*)

from teachers

where salary>50000;

-- Melihat dosen yang first\_name adalah "Samuel"

select\*

from teachers

where first\_name='Samuel';

-- Melihat dosen yang first\_name berawalan "Sam-"

Select \*

from teachers

where first\_name like 'Sam%';

-- Melihat dosen yang first\_name berakhiran "-sam"

Select \*

from teachers

where first\_name like '%sam'

-- Melihat dosen yang namanya mengandung tulisan "-sam-"

Select \*

from teachers

where first\_name like '%sam%'

-- Melihat gaji dosen tertinggi

select max(salary)

from teachers;

-- Melihat gaji dosen terendah

select Min(salary)

from teachers;

-- Melihat gaji dosen rata-rata

select avg(salary), floor(AVG(salary)), ceil(avg(salary))

from teachers;

-- Melihat dosen yang memiliki gaji tertinggi (ada namanya, bukan cuma angka aja)

-- INI CARA YANG SALAH KARENA HASILNYA GA VALID/AKURAT WALAU SYNTAXNYA JALAN

select first\_name, last\_name,max(salary)

from teachers;

-- CARA SALAH 2. Program tidak akan running

select first\_name, last\_name

from teachers,

where salary=max(salary);

-- Cara 3 INI BENAR, pake suvb query, mendekati seperti di python, if dalam if (?)

select first\_name, last\_name

from teachers

where salary=(

select max(salary)

from teachers);

-- CARA MELAKUKAN GROUPBY

-- Melihat gaji tertinggi di setiap universitas

select school, Max(salary)

from teachers

group by school;

-- (jadi yang school akan dikelompokan dan dicari nilai salary tertinggi dari masing2 kelompok school)

-- Melihat dosen yang memiliki gaji tertinggi di setiap universitas

select first\_name, last\_name, school, salary

from teachers

where (school, salary) IN ( #ini school dan salary dimasukin dalam kurung karena ngambil dua kolom

select school, max(salary)

from teachers

group by school

);

-- SQL JOINS

-- Tidak secara 'fisik' digabungkan, tapi bisa dipanggil

-- INNER, LEFT, RIGHT and OUTER JOIN

-- LATIHAN JOIN

-- Menggabungkan table teachers dengan table courses

select \*

FROM teachers

join courses on teachers.id=courses.teachers\_id;

-- Melihat dosen yang mengajar "Calculus"

select \*

FROM teachers

join courses on teachers.id=courses.teachers\_id

where courses.name='calculus';

-- Melihat jumlah mata kuliah di setiap universitas

-- universitas ada di tabel teachers

-- Matkul ada di tabel courses

select teachers.school, Count(courses.name) AS "Totalll Matkyuul ahseek"

from teachers

join courses on teachers.id=courses.teachers\_id

group by teachers.school

-- Melihat universitas apa saja yang total mata kuliahnya lebih dari "3"

select teachers.school as univ, count(courses.name) as total\_matkul

from teachers

join courses on teachers.id=courses.teachers\_id

group by teachers.school

having total\_matkul>3; #biasanya kalo ada HAVING pasti sepaket sama GROUP BY

-- Melihat first\_name beserta jumlah mata kuliah yang diajarkan dan diurutkan secara ascending berdasarkan jumlahnya

SELECT teachers.first\_name, Count(courses.name) AS total\_matkul

from teachers

join courses on teachers.id=courses.teachers\_id

group by teachers.first\_name

order by total\_matkul ASC;