*-- 1 --*

drop table if exists students;

drop table if exists courses;

drop table if exists enrollment;

create table if not exists students

(

student\_id serial primary key,

first\_name varchar(100),

last\_name varchar(100),

birthdate date,

enrollment\_year integer

);

create table if not exists courses

(

course\_id serial primary key,

course\_name varchar(100),

credit\_hours integer

);

create table if not exists enrollment

(

enrollment\_id serial primary key,

student\_id integer *references* students (student\_id) *on delete cascade*,

course\_id integer *references* courses (course\_id) *on delete cascade*,

grade integer

);

*-- 2 --*

INSERT INTO students (first\_name, last\_name, birthdate, enrollment\_year)

VALUES ('Alice', 'Johnson', '2003-05-14', 2021),

('Bob', 'Smith', '2002-09-23', 2020),

('Charlie', 'Brown', '2001-12-02', 2019),

('Diana', 'Prince', '2003-03-08', 2021),

('Ethan', 'Hunt', '2004-07-16', 2022),

('Fiona', 'Gallagher', '2002-11-25', 2020),

('George', 'Miller', '2001-02-17', 2019),

('Hannah', 'Wells', '2003-08-05', 2021),

('Ian', 'Wright', '2004-04-12', 2022),

('Jenny', 'Lewis', '2002-10-30', 2020);

INSERT INTO courses (course\_name, credit\_hours)

VALUES ('Mathematics', 3),

('History', 3),

('Physics', 4),

('Biology', 4),

('Chemistry', 4),

('Computer Science', 3),

('English Literature', 3),

('Economics', 3),

('Philosophy', 2),

('Art History', 3);

INSERT INTO enrollment (student\_id, course\_id, grade)

VALUES (1, 1, 5),

(1, 2, 4),

(1, 3, 5),

(2, 4, 3),

(2, 5, 4),

(2, 6, 5),

(3, 7, 4),

(3, 8, 5),

(3, 9, 3),

(4, 10, 5),

(4, 1, 3),

(4, 2, 4),

(5, 3, 4),

(5, 4, 3),

(5, 5, 4),

(6, 6, 3),

(6, 7, 4),

(6, 8, 5),

(7, 9, 5),

(7, 10, 3),

(7, 1, 4),

(8, 2, 5),

(8, 3, 3),

(8, 4, 3),

(9, 5, 4),

(9, 6, 4),

(9, 7, 5),

(10, 8, 4),

(10, 9, 3),

(10, 10, 4);

*-- 3 --*

*-- 3.1 --*

select students.first\_name,

students.last\_name,

students.birthdate

from students;

select s.first\_name,

s.last\_name,

s.birthdate,

c.course\_name

from students s

join

enrollment e on s.student\_id = e.student\_id

join

courses c ON e.course\_id = c.course\_id

where c.course\_name = 'Mathematics';

select s.first\_name,

s.last\_name,

avg(e.grade) as gpa

from students s

join

enrollment e on s.student\_id = e.student\_id

group by s.student\_id, s.first\_name, s.last\_name

having avg(e.grade) < 4;

*-- 3.2 --*

select s.first\_name,

s.last\_name,

c.course\_name

from students s

join

enrollment e on s.student\_id = e.student\_id

join

courses c on e.course\_id = c.course\_id;

select s.first\_name,

s.last\_name

from students s

left join

enrollment e on s.student\_id = e.student\_id

where e.student\_id is null;

*-- 3.3 --*

select c.course\_name,

count(e.student\_id) as student\_count

from courses c

left join

enrollment e ON c.course\_id = e.course\_id

group by c.course\_name

order by student\_count desc;

select c.course\_name,

count(e.student\_id) as student\_count

from courses c

left join

enrollment e on c.course\_id = e.course\_id

group by c.course\_name

order by student\_count desc

limit 1;

*-- 3.4 --*

select first\_name, last\_name

from students

order by last\_name;

select s.first\_name, s.last\_name

from students s

join enrollment e on s.student\_id = e.student\_id

join courses c on e.course\_id = c.course\_id

where s.enrollment\_year > 2015

and c.course\_name = 'History';

*-- 3.5 --*

with course\_counts as (select student\_id,

COUNT(\*) AS course\_count

from enrollment

group by student\_id),

average\_course\_count as (select avg(course\_count) as avg\_courses\_per\_student

from course\_counts)

select s.first\_name,

s.last\_name,

cc.course\_count

from students s

join course\_counts cc on s.student\_id = cc.student\_id,

average\_course\_count avg\_cc

where cc.course\_count > avg\_cc.avg\_courses\_per\_student;

with average\_grades as (select student\_id,

avg(grade) as avg\_grade

from enrollment

group by student\_id),

lowest\_avg\_grade as (select min(avg\_grade) as min\_avg\_grade

from average\_grades),

students\_with\_lowest\_avg as (select student\_id

from average\_grades

where avg\_grade = (select min\_avg\_grade from lowest\_avg\_grade))

select s.first\_name,

s.last\_name,

c.course\_name

from students s

join enrollment e on s.student\_id = e.student\_id

join courses c on e.course\_id = c.course\_id

where s.student\_id in (select student\_id from students\_with\_lowest\_avg);

*-- 3.6 --*

update enrollment

set grade = 3

where grade = 4;

delete

from students

where student\_id not in (select distinct student\_id

from enrollment);

*-- 4 --*

select c.course\_name as "Course Name",

round(avg(e.grade), 2) as "Average Score"

from courses c

left join enrollment e on c.course\_id = e.course\_id

group by c.course\_name

order by c.course\_name;

select c.course\_name as "Course Name",

round(AVG(e.grade), 2) as "Average Score"

from courses c

join enrollment e on c.course\_id = e.course\_id

where e.grade is not null

group by c.course\_name

order by c.course\_name;