**Add column to a table:**

ALTER TABLE table\_name

ADD column\_name column\_type NULL/NOT NULL;

**Update columns’ value:**

UPDATE *table\_name*  
SET *column1* = *value1*, *column2* = *value2*, ...  
WHERE *condition*;

Ex: update contacts

set contact\_age = date\_part('year', age(contact\_birthdate))

-- number of students

select count(\*) from students

--get students population in each year

select student\_population\_year\_ref, count(student\_epita\_email) from students s group by student\_population\_year\_ref

--get students population in each program

select student\_population\_code\_ref, count(student\_epita\_email) from students s group by student\_population\_code\_ref

--calculate age from DOB

select contact\_first\_name, contact\_last\_name ,contact\_birthdate, date\_part('year', age(contact\_birthdate)) from contacts c

--add age column to contacts

ALTER TABLE contacts

add contact\_age integer null

--add age to age list from dob

update contacts

set contact\_age = date\_part('year', age(contact\_birthdate))

--or

update contacts as c1 set contact\_age =

(SELECT date\_part('year',age(contact\_birthdate)) as c\_age

FROM contacts as c2 where c1.contact\_email=c2.contact\_email);

--avg students age

select avg(contact\_age) as avg\_student\_age

from contacts c inner join students s

on c.contact\_email = s.student\_contact\_ref

--avg session duration for a course

select avg(EXTRACT(EPOCH FROM TO\_TIMESTAMP(session\_end\_time, 'HH24:MI:SS')::TIME - TO\_TIMESTAMP(session\_start\_time, 'HH24:MI:SS')::TIME)/3600) as duration

from sessions as s left join courses as c

on c.course\_code=s.session\_course\_ref

where c.course\_code='SE\_ADV\_DB'

--student with most absences

select count(a.attendance\_student\_ref) as absences,

c.contact\_first\_name, c.contact\_last\_name

from contacts as c

left join students as s on s.student\_contact\_ref=c.contact\_email

left join attendance as a on s.student\_epita\_email=a.attendance\_student\_ref

where a.attendance\_presence=0

group by c.contact\_first\_name, c.contact\_last\_name

order by absences desc

limit 1

--find course with most absences

select c.course\_name, count(a.attendance\_presence)

from attendance a inner join courses c

on attendance\_course\_ref = c.course\_code

where attendance\_presence = 0

group by c.course\_name

order by count desc

limit 1

-- find students who are not graded

select s.student\_epita\_email, g.grade\_score

from students s right join grades g

on s.student\_epita\_email = g.grade\_student\_epita\_email\_ref

where g.grade\_score is null

--teachers that are not present in any session

select t.teacher\_epita\_email from teachers t

left outer join sessions s

on teacher\_epita\_email = s.session\_prof\_ref

where s.session\_prof\_ref is null

--list of teachers who attended the total sessions

select con.contact\_first\_name, con.contact\_last\_name, tea.teacher\_contact\_ref, count(session\_prof\_ref)

from teachers tea

inner join contacts con

on con.contact\_email = tea.teacher\_contact\_ref

inner join sessions sess

on tea.teacher\_epita\_email = sess.session\_prof\_ref

group by con.contact\_first\_name, con.contact\_last\_name, tea.teacher\_contact\_ref

order by count

--find the DSA students details with grades

select c.contact\_first\_name , c.contact\_last\_name , s.student\_population\_code\_ref , g.grade\_course\_code\_ref as course\_name, g.grade\_score

from students s inner join grades g

on s.student\_epita\_email = g.grade\_student\_epita\_email\_ref

inner join contacts c on s.student\_contact\_ref = c.contact\_email

where s.student\_population\_code\_ref = 'DSA'

--attendance percentage for a given student for all courses enrolled in

select (sum\_attendance/total\_attendance::float)\*100 as attendance\_percentage, attendance\_student\_ref, attendance\_course\_ref from

(

select count(\*) as total\_attendance, sum(attendance\_presence) as sum\_attendance, attendance\_student\_ref,attendance\_course\_ref from attendance

where attendance\_student\_ref ='jamal.vanausdal@epita.fr'

group by attendance\_student\_ref, attendance\_course\_ref

) as res

order by attendance\_percentage

--avg grade for DSA students

select s.student\_population\_code\_ref, avg(g.grade\_score) as avg\_grade from grades g

left join students s on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

where s.student\_population\_code\_ref = 'DSA'

group by s.student\_population\_code\_ref

--all students avg grade

select c.contact\_first\_name, c.contact\_last\_name, s.student\_epita\_email , avg(grade\_score)

from grades g inner join students as s

on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts as c

on s.student\_contact\_ref = c.contact\_email

group by c.contact\_first\_name,c.contact\_last\_name, s.student\_epita\_email

order by avg desc

--list courses taught by teachers

select c.course\_code, c2.contact\_first\_name, c2.contact\_last\_name

from sessions s inner join courses c

on s.session\_course\_ref = c.course\_code

inner join teachers t

on s.session\_prof\_ref = t.teacher\_epita\_email

inner join contacts c2 on t.teacher\_contact\_ref = c2.contact\_email

select distinct con.contact\_first\_name, con.contact\_last\_name, sess.session\_course\_ref

from teachers tea

inner join contacts con

on con.contact\_email = tea.teacher\_contact\_ref

inner join sessions sess

on tea.teacher\_epita\_email = sess.session\_prof\_ref

--teachers not giving any course

select \* from teachers t

left join sessions s

on s.session\_prof\_ref = t.teacher\_epita\_email

left join courses c on s.session\_course\_ref = c.course\_code

where s.session\_course\_ref is null;

-- number of students

select count(\*) from students

--get students population in each year

select student\_population\_year\_ref, count(student\_epita\_email) from students s group by student\_population\_year\_ref

--get students population in each program

select student\_population\_code\_ref, count(student\_epita\_email) from students s group by student\_population\_code\_ref

--calculate age from DOB

select contact\_first\_name, contact\_last\_name ,contact\_birthdate, date\_part('year', age(contact\_birthdate)) from contacts c

--add age column to contacts

ALTER TABLE contacts

add contact\_age integer null

--add age to age list from dob

update contacts

set contact\_age = date\_part('year', age(contact\_birthdate))

--or

update contacts as c1 set contact\_age =

(SELECT date\_part('year',age(contact\_birthdate)) as c\_age

FROM contacts as c2 where c1.contact\_email=c2.contact\_email);

--avg students age

select avg(contact\_age) as avg\_student\_age

from contacts c inner join students s

on c.contact\_email = s.student\_contact\_ref

--avg session duration for a course

select avg(EXTRACT(EPOCH FROM TO\_TIMESTAMP(session\_end\_time, 'HH24:MI:SS')::TIME - TO\_TIMESTAMP(session\_start\_time, 'HH24:MI:SS')::TIME)/3600) as duration

from sessions as s left join courses as c

on c.course\_code=s.session\_course\_ref

where c.course\_code='SE\_ADV\_DB'

--student with most absences

select count(a.attendance\_student\_ref) as absences,

c.contact\_first\_name, c.contact\_last\_name

from contacts as c

left join students as s on s.student\_contact\_ref=c.contact\_email

left join attendance as a on s.student\_epita\_email=a.attendance\_student\_ref

where a.attendance\_presence=0

group by c.contact\_first\_name, c.contact\_last\_name

order by absences desc

limit 1

--find course with most absences

select c.course\_name, count(a.attendance\_presence)

from attendance a inner join courses c

on attendance\_course\_ref = c.course\_code

where attendance\_presence = 0

group by c.course\_name

order by count desc

limit 1

-- find students who are not graded

select s.student\_epita\_email, g.grade\_score

from students s right join grades g

on s.student\_epita\_email = g.grade\_student\_epita\_email\_ref

where g.grade\_score is null

--teachers that are not present in any session

select t.teacher\_epita\_email from teachers t

left outer join sessions s

on teacher\_epita\_email = s.session\_prof\_ref

where s.session\_prof\_ref is null

--list of teachers who attended the total sessions

select con.contact\_first\_name, con.contact\_last\_name, tea.teacher\_contact\_ref, count(session\_prof\_ref)

from teachers tea

inner join contacts con

on con.contact\_email = tea.teacher\_contact\_ref

inner join sessions sess

on tea.teacher\_epita\_email = sess.session\_prof\_ref

group by con.contact\_first\_name, con.contact\_last\_name, tea.teacher\_contact\_ref

order by count

--find the DSA students details with grades

select c.contact\_first\_name , c.contact\_last\_name , s.student\_population\_code\_ref , g.grade\_course\_code\_ref as course\_name, g.grade\_score

from students s inner join grades g

on s.student\_epita\_email = g.grade\_student\_epita\_email\_ref

inner join contacts c on s.student\_contact\_ref = c.contact\_email

where s.student\_population\_code\_ref = 'DSA'

--attendance percentage for a given student for all courses enrolled in

select (sum\_attendance/total\_attendance::float)\*100 as attendance\_percentage, attendance\_student\_ref, attendance\_course\_ref from

(

select count(\*) as total\_attendance, sum(attendance\_presence) as sum\_attendance, attendance\_student\_ref,attendance\_course\_ref from attendance

where attendance\_student\_ref ='jamal.vanausdal@epita.fr'

group by attendance\_student\_ref, attendance\_course\_ref

) as res

order by attendance\_percentage

--avg grade for DSA students

select s.student\_population\_code\_ref, avg(g.grade\_score) as avg\_grade from grades g

left join students s on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

where s.student\_population\_code\_ref = 'DSA'

group by s.student\_population\_code\_ref

--all students avg grade

select c.contact\_first\_name, c.contact\_last\_name, s.student\_epita\_email , avg(grade\_score)

from grades g inner join students as s

on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts as c

on s.student\_contact\_ref = c.contact\_email

group by c.contact\_first\_name,c.contact\_last\_name, s.student\_epita\_email

order by avg desc

--list courses taught by teachers

select c.course\_code, c2.contact\_first\_name, c2.contact\_last\_name

from sessions s inner join courses c

on s.session\_course\_ref = c.course\_code

inner join teachers t

on s.session\_prof\_ref = t.teacher\_epita\_email

inner join contacts c2 on t.teacher\_contact\_ref = c2.contact\_email

select distinct con.contact\_first\_name, con.contact\_last\_name, sess.session\_course\_ref

from teachers tea

inner join contacts con

on con.contact\_email = tea.teacher\_contact\_ref

inner join sessions sess

on tea.teacher\_epita\_email = sess.session\_prof\_ref

--teachers not giving any course

select \* from teachers t

left join sessions s

on s.session\_prof\_ref = t.teacher\_epita\_email

left join courses c on s.session\_course\_ref = c.course\_code

where s.session\_course\_ref is null;

--1 list of students in a given year period and program

select \* from students

where student\_population\_period\_ref = 'SPRING'

and student\_population\_year\_ref = '2021'

and student\_population\_code\_ref = 'SE'

--2 get nb of enrolled of enrolled students in a given year period and program

select count(1) from students

where student\_population\_period\_ref = 'SPRING'

and student\_population\_year\_ref = '2021'

and student\_population\_code\_ref = 'SE'

--3 get all defined exams for a course from grades table

select g.grade\_course\_code\_ref, g.grade\_exam\_type\_ref from grades g

where g.grade\_course\_code\_ref ='SE\_ADV\_JAVA'

group by g.grade\_course\_code\_ref, g.grade\_exam\_type\_ref

--4 get all grades for a given student

select c.contact\_first\_name,c.contact\_last\_name, g.grade\_course\_code\_ref, g.grade\_score from grades g

inner join students s on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts c on s.student\_contact\_ref = c.contact\_email

where g.grade\_student\_epita\_email\_ref='jamal.vanausdal@epita.fr'

--5 get all grades for a specific exam

select g.grade\_course\_code\_ref,g.grade\_exam\_type\_ref,g.grade\_score from grades g

where g.grade\_course\_code\_ref = 'SE\_ADV\_JS'

--6 get students ranks in an exam for a course

select c.contact\_first\_name, c.contact\_last\_name, g.grade\_course\_code\_ref, g.grade\_exam\_type\_ref, g.grade\_score,

rank() over(order by g.grade\_score desc) as rnk

from grades g inner join students s

on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts c

on s.student\_contact\_ref = c.contact\_email

where g.grade\_course\_code\_ref = 'SE\_ADV\_JS'

--7 get students ranks in all exams for a course

select c.contact\_first\_name, c.contact\_last\_name, g.grade\_course\_code\_ref, g.grade\_exam\_type\_ref, g.grade\_score,

rank() over(partition by g.grade\_exam\_type\_ref order by g.grade\_score desc) as rnk

from grades g inner join students s

on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts c

on s.student\_contact\_ref = c.contact\_email

where g.grade\_course\_code\_ref = 'SE\_ADV\_JAVA'

--8 get students ranks in all exams and all courses

select c.contact\_first\_name, c.contact\_last\_name, g.grade\_course\_code\_ref, g.grade\_exam\_type\_ref, g.grade\_score,

rank() over(partition by g.grade\_exam\_type\_ref,g.grade\_course\_code\_ref order by g.grade\_score desc) as rnk

from grades g inner join students s

on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts c

on s.student\_contact\_ref = c.contact\_email

--9 Get all courses for one program

select p.program\_assignment, c.course\_name, p.program\_course\_code\_ref

from programs p inner join courses c on p.program\_course\_code\_ref = c.course\_code

where p.program\_assignment = 'SE'

--10 Get courses in common between 2 programs

select c.course\_name

from courses c inner join programs p

on c.course\_code = p.program\_course\_code\_ref

where p.program\_assignment ='DSA'

intersect

select c.course\_name

from courses c inner join programs p

on c.course\_code = p.program\_course\_code\_ref

where p.program\_assignment ='CS'

--11 Get all programs following a certain course

select p.program\_assignment

from courses c inner join programs p

on c.course\_code = p.program\_course\_code\_ref

where c.course\_code = 'AI\_DATA\_SCIENCE\_IN\_PROD'

--12 get course with the biggest duration

with course\_duration\_rank as (

select duration, c.course\_name,

rank() over(order by duration desc) as rnk

from courses c

)

select duration ,course\_name, rnk

from course\_duration\_rank

where rnk = 1;

--13 Get courses with the same duration

select course\_name, duration from courses where duration in (

select duration from courses

group by duration having count(\*) > 1

)

order by duration desc

--14 Get all sessions for a specific course

select s.session\_course\_ref,s.session\_type, s.session\_date from sessions s

where s.session\_course\_ref = 'AI\_DATA\_PREP'

--15 Get all sessions for a certain period

select s.session\_course\_ref,s.session\_type, s.session\_date from sessions s

where s.session\_date between '2020-11-01' and '2020-11-30'

--16 Get one student attendance sheet

select c.contact\_first\_name, s.student\_epita\_email , a.attendance\_session\_date\_ref, a.attendance\_course\_ref, a.attendance\_presence from attendance a

left join students s on s.student\_epita\_email = a.attendance\_student\_ref

left join contacts c on c.contact\_email = s.student\_contact\_ref

where s.student\_epita\_email = 'jamal.vanausdal@epita.fr'

--17 Get one student summary of attendance / SAME

--18 Get student with most absences

select count(a.attendance\_student\_ref) as absences,

c.contact\_first\_name, c.contact\_last\_name

from contacts as c

left join students as s on s.student\_contact\_ref=c.contact\_email

left join attendance as a on s.student\_epita\_email=a.attendance\_student\_ref

where a.attendance\_presence=0

group by c.contact\_first\_name, c.contact\_last\_name

order by absences desc

limit 1

--1 hard questions - get all exams for a specific course

select e.exam\_course\_code, e.exam\_weight, e.exam\_type from exams e

where e.exam\_course\_code ='SE\_ADV\_JAVA'

--2 get all grades for a specific student

select c.contact\_first\_name,c.contact\_last\_name, g.grade\_course\_code\_ref, g.grade\_score from grades g

inner join students s on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts c on s.student\_contact\_ref = c.contact\_email

where g.grade\_student\_epita\_email\_ref='jamal.vanausdal@epita.fr'

--3 Get the final grades for a student on a specifique course or all courses

select s.student\_epita\_email ,g.grade\_course\_code\_ref, sum(e.exam\_weight \* g.grade\_score)/sum(e.exam\_weight) from grades g

inner join exams e on g.grade\_course\_code\_ref = e.exam\_course\_code

inner join students s on g.grade\_student\_epita\_email\_ref =s.student\_epita\_email

where s.student\_epita\_email ='jamal.vanausdal@epita.fr'

group by g.grade\_course\_code\_ref, s.student\_epita\_email

--4 Get the students with the top 5 scores for specific course

with total\_grade\_course as (

select c.contact\_first\_name, c.contact\_last\_name ,g.grade\_course\_code\_ref, sum(e.exam\_weight \* g.grade\_score)/sum(e.exam\_weight) as total\_grade,

rank() over (partition by g.grade\_course\_code\_ref order by sum(e.exam\_weight \* g.grade\_score)/sum(e.exam\_weight) desc) as rnk

from grades g

inner join exams e on g.grade\_course\_code\_ref = e.exam\_course\_code

inner join students s on g.grade\_student\_epita\_email\_ref = s.student\_epita\_email

inner join contacts c on s.student\_contact\_ref = c.contact\_email

group by g.grade\_course\_code\_ref, c.contact\_first\_name, c.contact\_last\_name

)

select contact\_first\_name, contact\_last\_name, grade\_course\_code\_ref, total\_grade, rnk

from total\_grade\_course

where rnk <=5 and grade\_course\_code\_ref ='DT\_RDBMS'

--5 Get the students with the top 5 scores for specific course

--6 Get the class average for a course

select g.grade\_course\_code\_ref, (sum(e.exam\_weight \* g.grade\_score)/sum(e.exam\_weight)::float) as class\_average

from grades g inner join exams e on g.grade\_course\_code\_ref = e.exam\_course\_code

inner join students s on g.grade\_student\_epita\_email\_ref =s.student\_epita\_email

group by g.grade\_course\_code\_ref