Exam prep questions:

1. Load the values.
2. Count the number of students – select count(1) from students;
3. Distribution of students across different programs – select student\_population\_year\_ref, count(1) from students group by student\_population\_year\_ref; or with student\_population\_code\_ref instead of year.
4. Joints – (left joint), to calculate the average age, average duration of the sessions.

Sol - SELECT avg (DATE\_PART('year', now()) - DATE\_PART('year', contact\_birthdate)) as Average from contacts;

1. Corelate the queries to find overlapping between session, provide list of session which are overlapping, based on time rooms like startdate etc.

\*\*\*\*5 minutes per questions. Total 90 minutes\*\*\*\*

1. Find the students(Fn, Ln) which is the most absent or find the course which has more absents. (need to use orderby name by default if its person or course if its course name)
2. find the course which has the most absent students

sol - select attendance\_course\_ref, sum(attendance\_presence)

from attendance

where attendance\_presence <> 0

group by attendance\_course\_ref

order by sum(attendance\_presence)

limit 1

1. find the teachers who are not giving any courses

sol –

select distinct teacher\_epita\_email from teachers left join sessions on teacher\_epita\_email = session\_prof\_ref  
where session\_prof\_ref is null ;

(or) SELECT a.teacher\_epita\_email, c.contact\_first\_name, c.contact\_last\_name  
FROM teachers a  
LEFT JOIN sessions b  
ON a.teacher\_epita\_email=b.session\_prof\_ref  
LEFT JOIN contacts c  
ON a.teacher\_contact\_ref = c.contact\_email  
WHERE b.session\_prof\_ref IS NULL;

1. find the students who are not graded

sol - select \* from grades where grade\_score is null;

Sol for 6 only with email - select attendance\_student\_ref, sum (attendance\_presence)from attendance group by attendance\_student\_ref order by sum asc limit 1;

Sol with name - select contacts.contact\_last\_name, contacts.contact\_first\_name, sum (attendance\_presence)from attendance

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

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

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

order by sum asc

limit 1; (or no need for limits) we can add rank to differentiate the equivalent

create table attendance

(

attendance\_student\_ref text,

attendance\_population\_year\_ref integer,

attendance\_course\_ref text,

attendance\_course\_rev integer,

attendance\_session\_date\_ref timestamp,

attendance\_session\_start\_time text,

attendance\_session\_end\_time text,

attendance\_presence integer

);

Put timestamp for all the dates for good results – by Sivasankari! The great

Extras;

SELECT AVG(grade\_score) FROM grades

SELECT AVG((CURRENT\_DATE - contact\_birthdate)/365) AS AGE FROM contacts;

SELECT TO\_TIMESTAMP(session\_start\_time, 'HH24') AS v1,  
TO\_TIMESTAMP(session\_end\_time, 'HH24') AS v2 FROM sessions;

**Q1: join:**

**select st.student\_epita\_email, ct.contact\_email from students st join contacts ct**

**on st.student\_contact\_ref = ct.contact\_email where**

**st.student\_enrollment\_status = 'completed' order by st.student\_epita\_email;**

**Q2: Produce a count of the number of city each student:**

**select contact\_city, count(\*)**

**from contacts group by contact\_city**

**order by contact\_city;**

**Q3: Total attendance Presence ref by course and student:**

**select sum(attendance\_presence) as "Total Present" from attendance where attendance\_student\_ref ='jamal.vanausdal@epita.fr' and attendance\_course\_ref='PM\_AGILE' group by attendance\_student\_ref;**

**Q4: Total attendance Presence ref by course and student with timings:**

**select attendance\_student\_ref, sum(attendance\_presence) as "Total Present"**

**from attendance where attendance\_session\_start\_time >= '11:00:00'**

**and attendance\_session\_start\_time < '13:00:00' group by attendance\_student\_ref order by sum(attendance\_presence) ;**

**Q5: using method extract:**

**select attendance\_student\_ref, extract (month from attendance\_session\_date\_ref) as month, sum(attendance\_presence) as "Total Present" from attendance where extract (year from attendance\_session\_date\_ref ) = 2021 group by attendance\_student\_ref order by sum(attendance\_student\_ref);**

**select extract (year from timestamp '05-06-2021');**

**Q6: count:**

**select count(distinct attendance\_student\_ref) as Refered\_stu from attendance;**

**Q7:SUM:**

**select sum(attendance\_presence) as "Total Present” from attendance where attendance\_student\_ref ='jamal.vanausdal@epita.fr' group by attendance\_presence having sum(attendance\_presence) >= 5**

**order by attendance\_presence;**

**Q8: selecting particular emaila= and course:**

**select sum(attendance.attendance\_presence) as "Total Present" , attendance.attendance\_course\_ref**

**from attendance where attendance.attendance\_course\_ref='SE\_ADV\_JAVA' and attendance.attendance\_student\_ref='marjory.mastella@epita.fr'**

**group by attendance.attendance\_course\_ref;**

**Q10:calculating avg age of student:**

**SELECT avg (DATE\_PART('year', now()) - DATE\_PART('year', contact\_birthdate)) from contacts;**

**SELECT (DATE\_PART('year', now()) - DATE\_PART('year', contact\_birthdate)) as age from contacts;**

**Q11:sum of atnce:**

**select attendance\_student\_ref, sum (attendance\_presence) from attendance group by attendance\_student\_ref order by sum asc;**

**Q12:select particular**

**select contacts.contact\_last\_name, contacts.contact\_first\_name, sum (attendance\_presence)from attendance  
left join students on students.student\_epita\_email = attendance.attendance\_student\_ref  
left join contacts on contacts.contact\_email = students.student\_contact\_ref  
group by contacts.contact\_last\_name, contacts.contact\_first\_name  
order by sum asc  
limit 1;**

**Q13: find most of absent in particular subject**

**select attendance\_course\_ref, sum(attendance\_presence)  
from attendance  
where attendance\_presence <> 0  
group by attendance\_course\_ref  
order by count(attendance\_presence)  
limit 1;**

**Q14: prof not taking sessions:**

**select distinct teacher\_epita\_email from teachers left join sessions on teacher\_epita\_email = session\_prof\_ref**

**where session\_prof\_ref is null ;**

**SELECT a.teacher\_epita\_email, c.contact\_first\_name, c.contact\_last\_name**

**FROM teachers a**

**LEFT JOIN sessions b**

**ON a.teacher\_epita\_email=b.session\_prof\_ref**

**LEFT JOIN contacts c**

**ON a.teacher\_contact\_ref = c.contact\_email**

**WHERE b.session\_prof\_ref IS NULL;**

**Q15: student is not graded:**

**select \* from grades where grade\_score is null;**

**Q16: count**

**Select student\_population\_code\_ref , count(1) from students**

**Group by student\_population\_code\_ref;**