Wilhelm

- Are views and materialized views used in all queries that benefit from using them? Can any query be made easier to understand by storing part of it in a view? Can performance be improved by using a materialized view?
  We think you have used views in the queries that benefit them correctly but you have not used materialized views.

- Did you change the database design to simplify these queries? If so, was the database design worsened in any way just to make it easier to write these particular queries?

- Is there any correlated subquery, that is a subquery using values from the outer query? Remember that correlated subqueries are slow since they are evaluated once for each row processed in the outer query.
  We do not think you have used correlated subquery.

- Are there unnecessarily long and complicated queries? Are you for example using a UNION clause where it's not required?
  There is no UNION clause

- Analyze the query plan for at least one of your queries using the command EXPLAIN (or EXPLAIN ANALYZE), which is available in both Postgres and MySQL. Where in the query does the DBMS spend most time? Is that reasonable? If you have time, also consider if the query can be rewritten to execute faster, but you're not required to do that.
  The explain analyze part is well written

general problems:
Instead of presenting the amount of seats left it should be grouped by the three categories.

Magnus

- Can any query be made easier to understand by storing part of it in a view? Can performance be improved by using a materialized view?
  We cannot find any views in the code but it could have benefited by implementing views.

- Did you change the database design to simplify these queries? If so, was the database design worsened in any way just to make it easier to write these particular queries?

- Is there any correlated subquery, that is a subquery using values from the outer query? Remember that correlated subqueries are slow since they are evaluated once for each row processed in the outer query.

We did not find any correlated subqueries.

- Are there unnecessarily long and complicated queries? Are you for example using a UNION clause where it’s not required?
  The UNION clause is not implemented correctly

- Analyze the query plan for at least one of your queries using the command EXPLAIN (or EXPLAIN ANALYZE), which is available in both Postgres and MySQL. Where in the query does the DBMS spend most time? Is that reasonable? If you have time, also consider if the query can be rewritten to execute faster, but you’re not required to do that.
  We did not find the explain analyze section

general problems:
The queries should of been in your github

person 3

- Can any query be made easier to understand by storing part of it in a view? Can performance be improved by using a materialized view?
  views and materialized views are present

- Did you change the database design to simplify these queries? If so, was the database design worsened in any way just to make it easier to write these particular queries?

- Is there any correlated subquery, that is a subquery using values from the outer query? Remember that correlated subqueries are slow since they are evaluated once for each row processed in the outer query.
  no correlated subqueries

- Are there unnecessarily long and complicated queries? Are you for example using a UNION clause where it’s not required?
  no union clause used

- Analyze the query plan for at least one of your queries using the command EXPLAIN (or EXPLAIN ANALYZE), which is available in both Postgres and MySQL. Where in the query does the DBMS spend most time? Is that reasonable? If you have time, also consider if the query can be rewritten to execute faster, but you’re not required to do that.
  The explain analyze part is well written

general problems:
the code could have been written in a little better way, consider indentation.