easiest way to convert PSQL to JSON is a temporary table.

```
create temporary table angprojj as
select
  ang.persnr,                joins are omitted,
  jsonb_build_object(        would be below...
    'persnr', ang.persnr,
    'name', min(ang.name),
    'projects', jsonb_agg(trim(proj.bezeichnung))
  ) as angwithproj
```

you can also directly **select json_build_object**
or **select json_build_object(jsonb_agg(tmp))**
also note the jsonb_agg(trim (proj.bezeichnung))
for simplicity, the joins on the picture are removed.

The jsonb_agg is necessary for aggregation
otherwise the join wouldn't work for JSON.

JSON-Daten abfragen

```
select persnr, angwithproj
from angprojj
where angwithproj->>'persnr' = 1001::text;

select persnr, angwithproj
from angprojj
where (angwithproj->>'name') like 'Marxer%'

select persnr, jsonb_pretty(angwithproj)
from angprojj
where angwithproj->'projects' @>
      to_jsonb('Uranus'::text)
```

Get as text:

```
select persnr, angwithproj->'projects' as projects
from angprojj;
```

=> 23 rows

CROSS JOIN zweier Tabellen. jsonb_array_elements_text() gibt „setof text" zurück:

```
select persnr, angwithproj->>'name' as persname, value as projname
from angprojj, jsonb_array_elements_text(angwithproj->'projects');
```

=> 29 rows

JOIN LATERAL = CROSS JOIN und Boolean: Output identisch mit oben :

```
select persnr, angwithproj->>'name' as persname, value as projname
from angprojj
cross join lateral jsonb_array_elements_text(angwithproj->'projects');
```

=> 29 rows

JSON | SQL