Aufgabe 3: Performance-Verbesserung auf der Datenbank

select * from pg_indexes where schemaname = 'name ihres schemas';

```
1  select * from pg_indexes where schemaname = 'public';
```

Data Output | Explain | Messages | Notifications

	schemaname name	tablename name	indexname name	tablespace name	indexdef text
1	public	category	category_pkey	[null]	CREATE UNIQUE INDEX category_pkey ON public.category USING btree (id)
2	public	category	category_name_key	[null]	CREATE UNIQUE INDEX category_name_key ON public.category USING btre
3	public	question	question_pkey	[null]	CREATE UNIQUE INDEX question_pkey ON public.question USING btree (id)
4	public	player	player_pkey	[null]	CREATE UNIQUE INDEX player_pkey ON public.player USING btree (name)
5	public	game	game_pkey	[null]	CREATE UNIQUE INDEX game_pkey ON public.game USING btree (id)

\di

```
postgres=# \di
                    List of relations
 Schema |        Name         | Type  |  Owner   |  Table
--------+---------------------+-------+----------+----------
 public | category_name_key   | index | postgres | category
 public | category_pkey       | index | postgres | category
 public | game_pkey           | index | postgres | game
 public | player_pkey         | index | postgres | player
 public | question_pkey       | index | postgres | question
(5 rows)
```

select relname, n_live_tup from pg_stat_user_tables;

```
1  select relname, n_live_tup from pg_stat_user_tables;
2
```

Data Output | Explain | Messages | Notifications

	relname name	n_live_tup bigint
1	question	200
2	player	1000
3	question_answerlist	800
4	question	200
5	player	10000
6	game_myquestionmap	9351753
7	category	51
8	game	990901
9	category	51
10	game	100000

ANALYZE PLAYER;

ANALYZE GAME;

Es hat eigentlich nichts nach der beide ANALYZE geändert.

Query 1:

SELECT DISTINCT t0.NAME

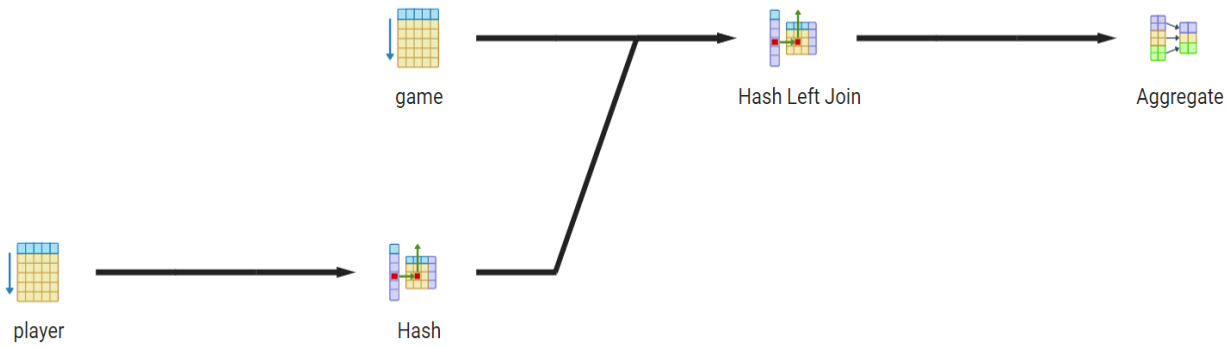
FROM public.Game t1 LEFT OUTER JOIN public.Player t0 ON (t0.NAME = t1.PLAYER_NAME) WHERE ((t1.STARTTIME >= '2021-07-08 11:07:55.516367+02') AND (t1.ENDTIME <= '2021-07-08 22:07:55.516367+02'));

Before creating index:

```
1  SELECT DISTINCT t0.NAME
2  FROM public.Game t1 LEFT OUTER JOIN public.Player t0 ON (t0.NAME = t1.PLAYER_NAME)
3  WHERE ((t1.STARTTIME >= '2021-07-08 11:07:55.516367+02')
4  AND (t1.ENDTIME <= '2021-07-08 22:07:55.516367+02'));
```

Messages

```
Successfully run. Total query runtime: 716 msec.
1000 rows affected.
```



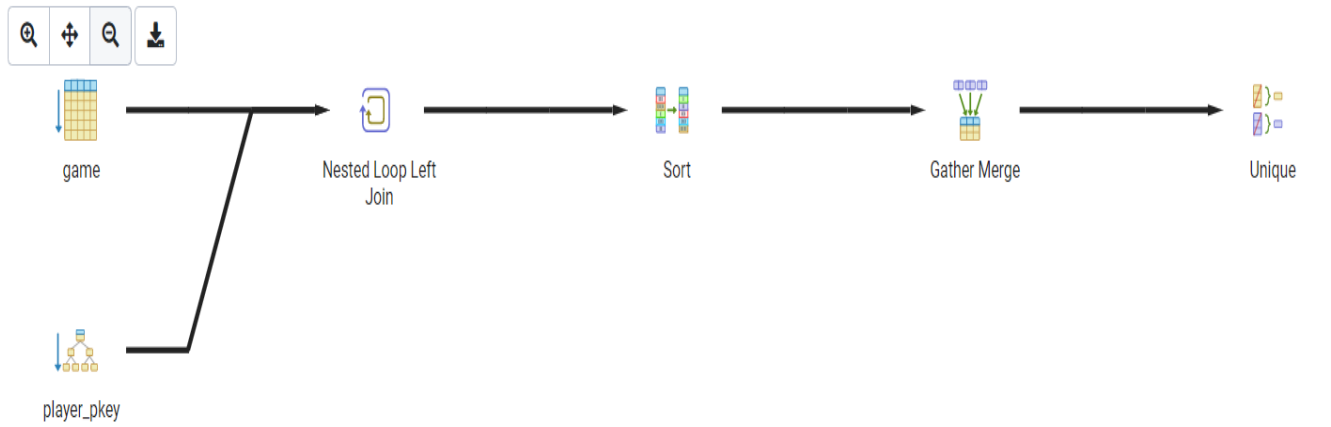
create index index_player_name_game on game (player_name);

After creating index:

```
1  SELECT DISTINCT t0.NAME
2  FROM public.Game t1 LEFT OUTER JOIN public.Player t0 ON
3  (t0.NAME = t1.PLAYER_NAME)
4  WHERE ((t1.STARTTIME >= '2021-07-08 21:02:58.838133+02') AND (t1.ENDTIME <= '2021-07-08 21:02:58.838133+02')
5
```

Messages

```
Successfully run. Total query runtime: 417 msec.
1 rows affected.
```



Query 2:

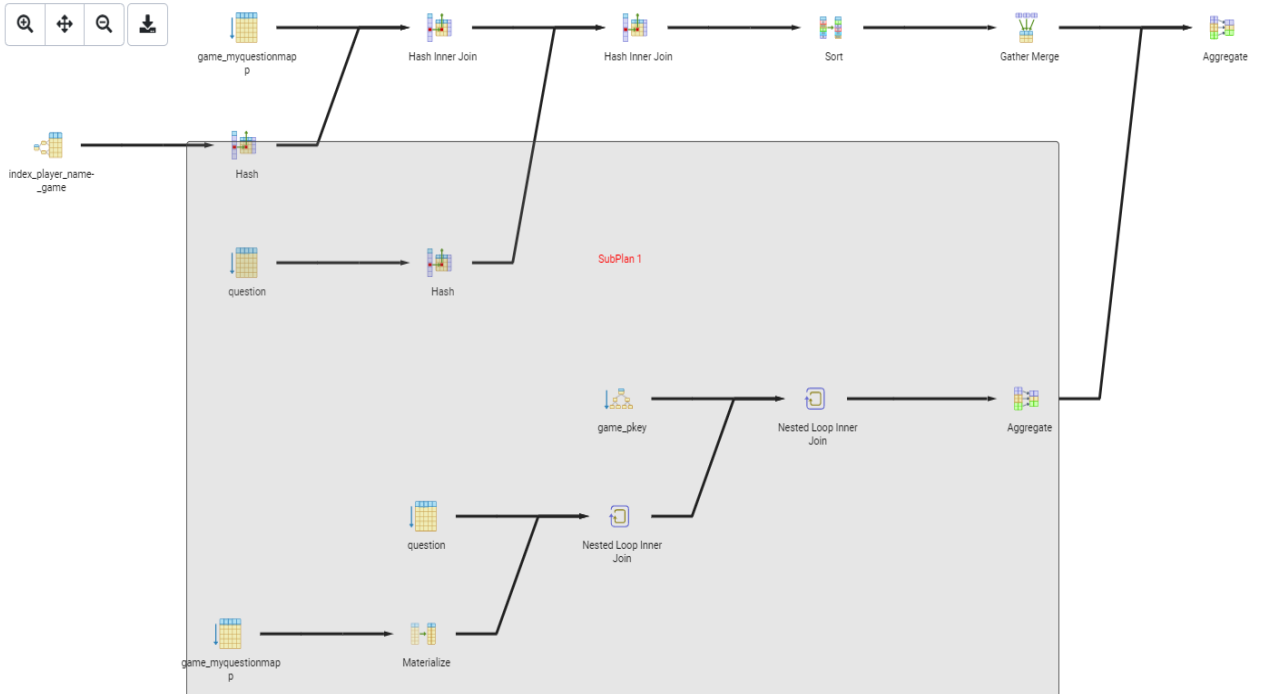
SELECT t0.ID, t0.STARTTIME, t0.ENDTIME, (SELECT COUNT(t5.antwort) FROM Game_MYQUESTIONMAP t5, public.Question t4, public.Game t6 WHERE (((t4.CORRECTANSWERINDEX = t5.antwort) AND (t0.ID = t6.ID)) AND ((t5.Game_ID = t6.ID) AND (t4.ID = t5.myquestionmap_KEY)))), COUNT(t2.antwort) FROM public.Game t0, Game_MYQUESTIONMAP t2, public.Question t1 WHERE ((t0.PLAYER_NAME = 'player 1') AND ((t2.Game_ID = t0.ID) AND (t1.ID = t2.myquestionmap_KEY))) GROUP BY t0.ID, t0.ENDTIME, t0.STARTTIME, t0.PLAYER_NAME;

Before creating index:

```
1  SELECT t0.ID, t0.STARTTIME, t0.ENDTIME, (SELECT COUNT(t5.antwort)
2  FROM Game_MYQUESTIONMAP t5, public.Question t4, public.Game t6
3  WHERE (((t4.CORRECTANSWERINDEX = t5.antwort) AND (t0.ID = t6.ID)) AND ((t5.Game_ID = t6.ID)
4  AND (t4.ID = t5.myquestionmap_KEY)))), COUNT(t2.antwort) FROM public.Game t0, Game_MYQUESTIONMAP t2, public.Question
5  GROUP BY t0.ID, t0.ENDTIME, t0.STARTTIME, t0.PLAYER_NAME;
```

Messages

```
Successfully run. Total query runtime: 1 min 22 secs.
100 rows affected.
```



After creating index:

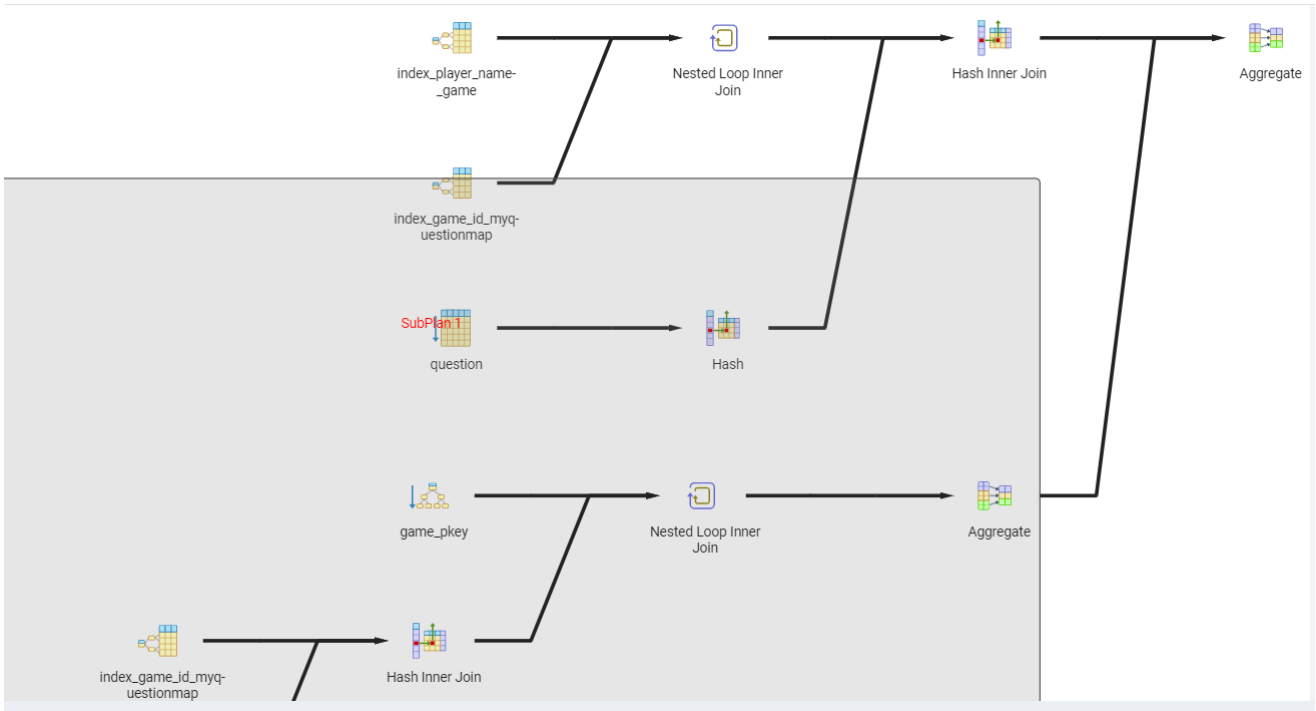
create index index_game_id_myquestionmap on game_myquestionmap(game_id);

create index index_myquestionmap_key on game_myquestionmap(myquestionmap_key);

```
SELECT t0.ID, t0.STARTTIME, t0.ENDTIME, (SELECT COUNT(t5.antwort)
FROM Game_MYQUESTIONMAP t5, public.Question t4, public.Game t6
WHERE (((t4.CORRECTANSWERINDEX = t5.antwort) AND (t0.ID = t6.ID)) AND ((t5.Game_ID = t6.ID)
AND (t4.ID = t5.myquestionmap_KEY)))), COUNT(t2.antwort) FROM public.Game t0, Game_MYQUESTIONMAP t2,
public.Question t1 WHERE ((t0.PLAYER_NAME = 'player 1') AND ((t2.Game_ID = t0.ID) AND (t1.ID = t2.myquestionmap_KEY))
GROUP BY t0.ID, t0.ENDTIME, t0.STARTTIME, t0.PLAYER_NAME;
```

Messages

```
Successfully run. Total query runtime: 68 msec.
100 rows affected.
```



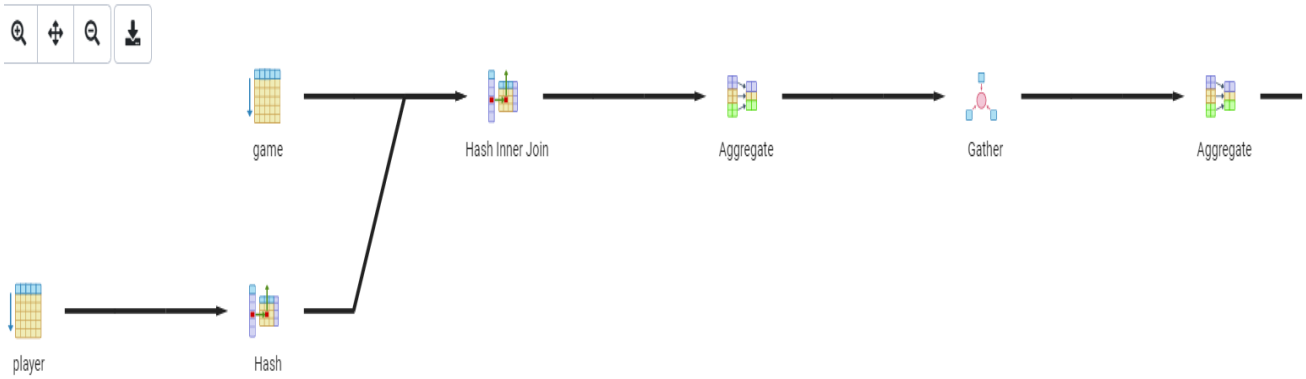
Query 3:

SELECT t0.NAME, COUNT(t1.ID) FROM public.Player t0, public.Game t1 WHERE (t0.NAME = t1.PLAYER_NAME) GROUP BY t0.NAME ORDER BY COUNT(t1.ID);

```
SELECT t0.NAME, COUNT(t1.ID) FROM public.Player t0, public.Game t1
WHERE (t0.NAME = t1.PLAYER_NAME) GROUP BY t0.NAME ORDER BY COUNT(t1.ID);
```

Messages

```
Successfully run. Total query runtime: 286 msec.
10000 rows affected.
```



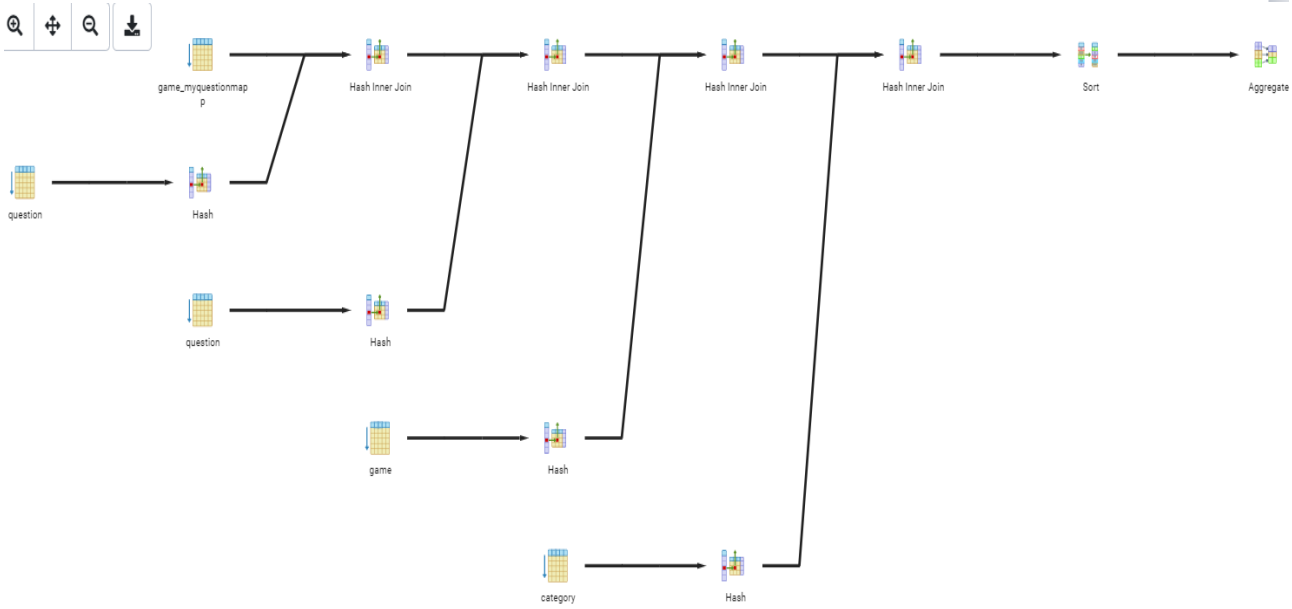
Query 4:

SELECT t0.ID, COUNT(DISTINCT(t1.ID)) FROM public.Question t4, Game_MYQUESTIONMAP t3, public.Question t2, public.Game t1, public.Category t0 WHERE ((t4.ID = t2.ID) AND (((t3.Game_ID = t1.ID) AND (t2.ID = t3.myquestionmap_KEY)) AND (t4.Category_id = t0.ID))) GROUP BY t0.ID ORDER BY COUNT(DISTINCT(t1.ID));

```
SELECT t0.ID, COUNT(DISTINCT(t1.ID))
FROM public.Question t4, Game_MYQUESTIONMAP t3, public.Question t2, public.Game t1, public.Category t0
WHERE ((t4.ID = t2.ID) AND (((t3.Game_ID = t1.ID) AND (t2.ID = t3.myquestionmap_KEY)) AND (t4.Category_id = t0.ID)))
GROUP BY t0.ID ORDER BY COUNT(DISTINCT(t1.ID));
```

Messages

```
Successfully run. Total query runtime: 12 secs 944 msec.
51 rows affected.
```



create index question_category_id on question(category_id);