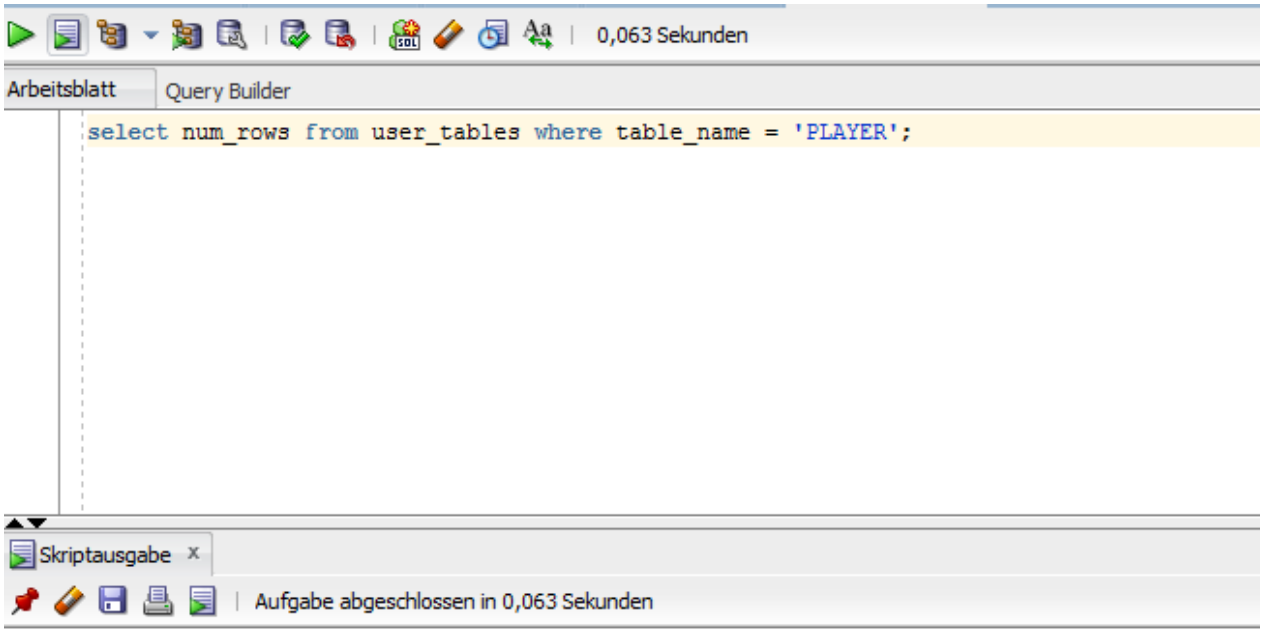
Ahmad Mustain Billah **Sascha Bauer**

Praktrikum 5 Datenbanken2 – Performance Verbesserungen

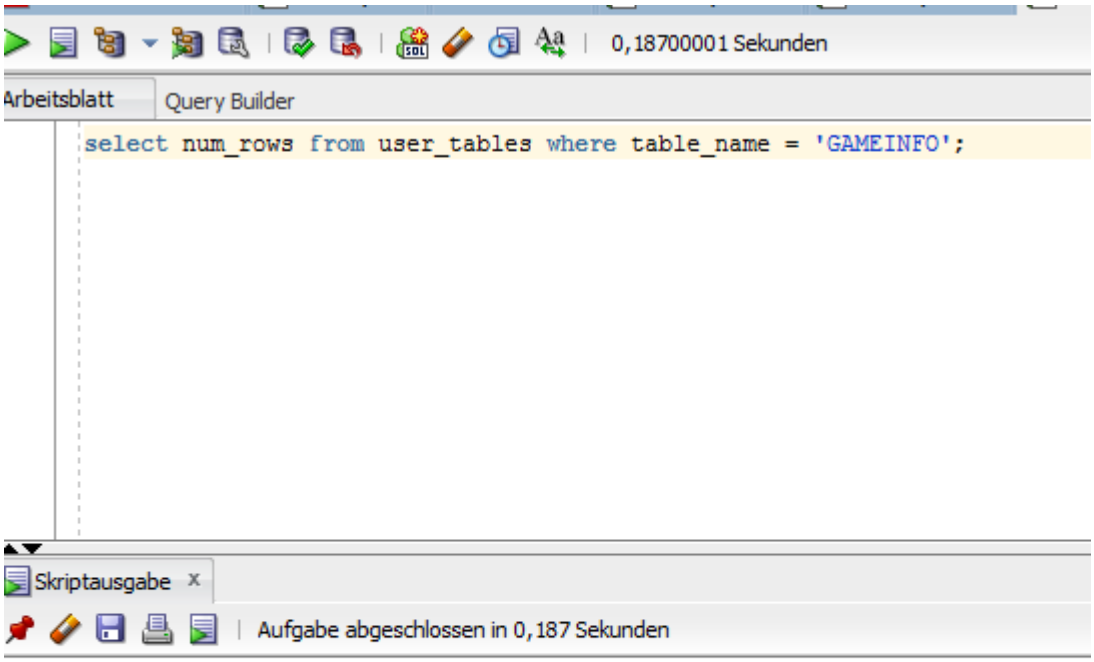
1. Prüfen Sie, für welche Tabellen und welche Spalte(n) es auf der Datenbank bereits einen Index gibt. Hinweis: Die Systemtabelle user indexes enthalt die Metadaten zu allen Indexen Ihres Datenbankschemas:
   select * from user indexes

   Alle existierenden Indizes sind in der Datei Indizes.txt oder wahlweise Indizes.xml zu finden. Die Ausgliederung in eine andere Datei dient der Übersicht.

2. Prüfen Sie, welchen Wert die Systemtabelle user tables für die Anzahl der Datensätze (num rows) in den Tabellen PLAYER und GAME und die assoziierten Tabellen enthält:
   select num rows from user tables where table name = '<TABLENAME>';

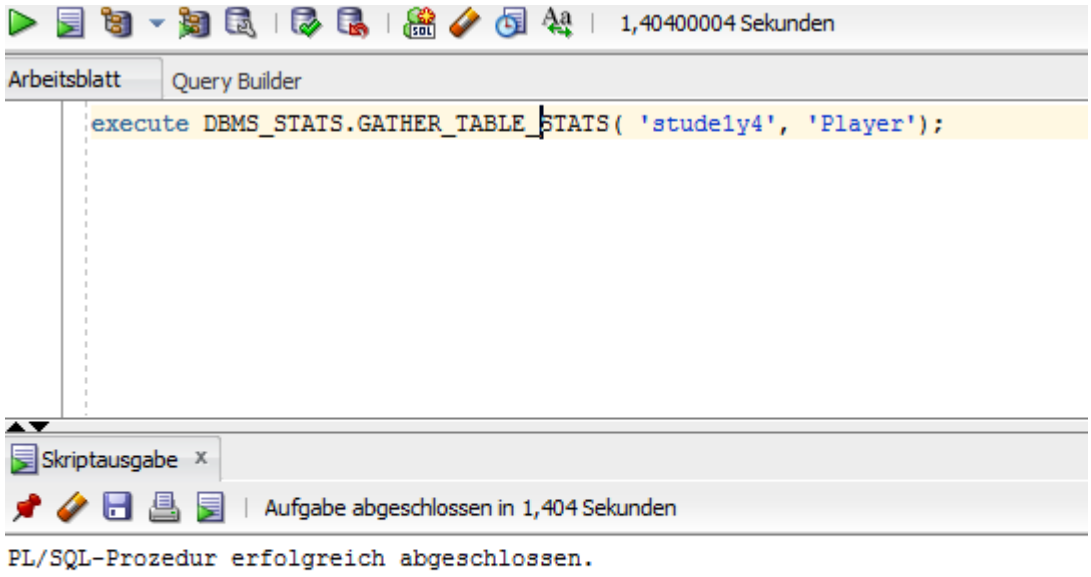


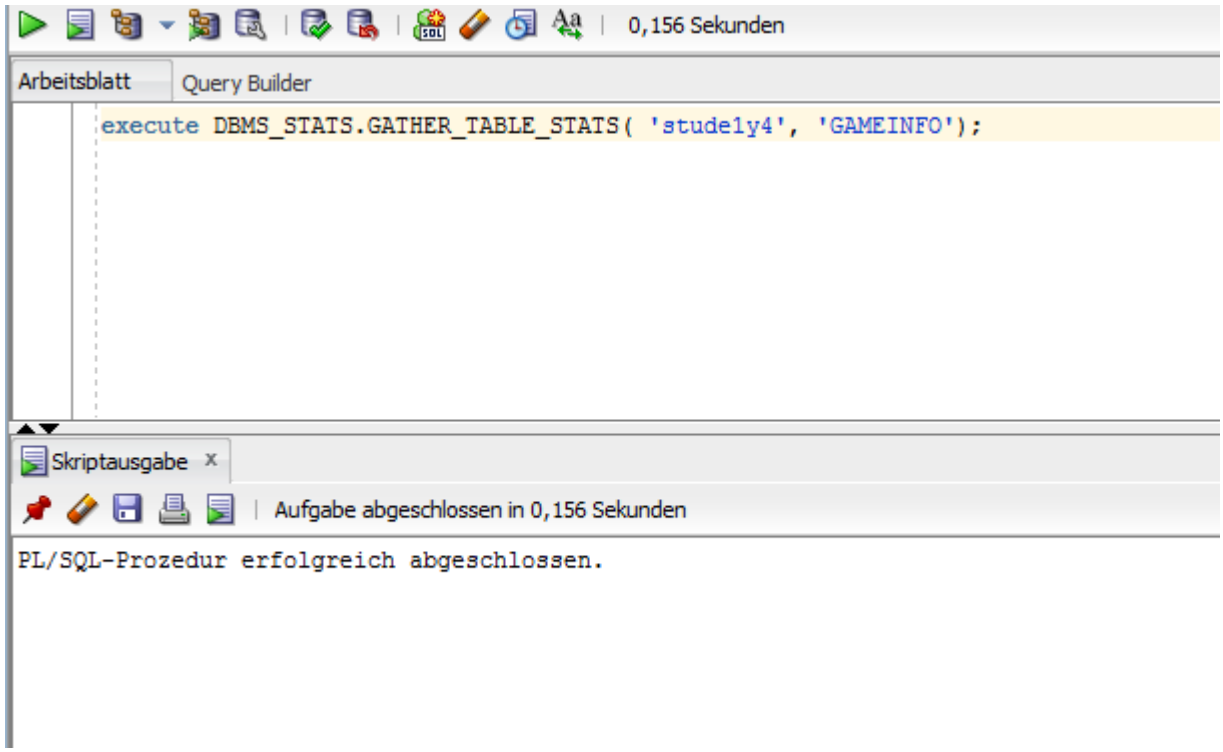
Ahmad Mustain Billah **Sascha Bauer**



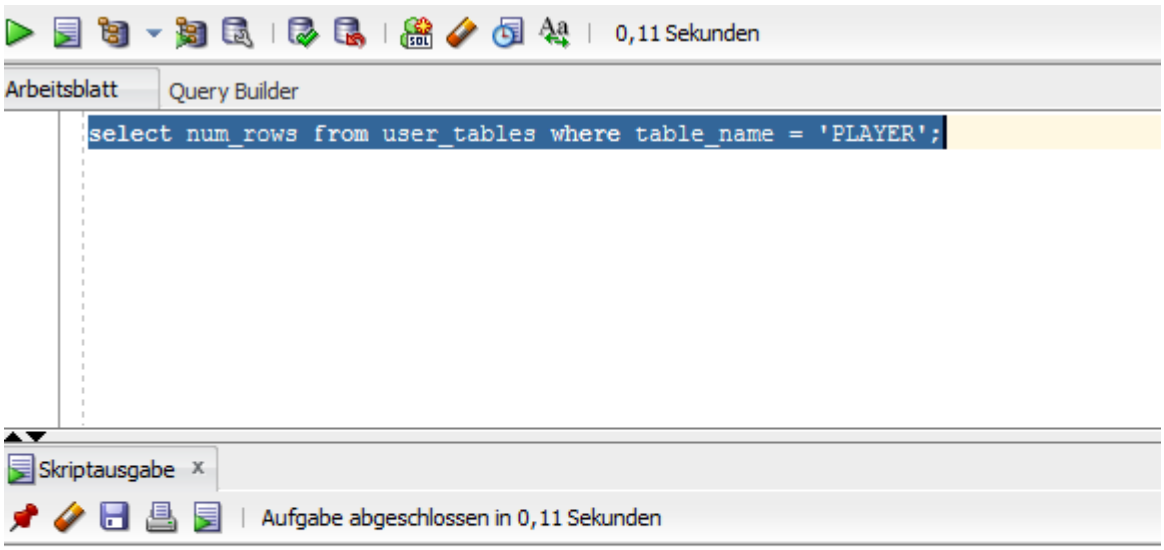
3. Die Tabellenstatistik kann explizit mit dem folgenden Prozeduraufruf aktualisiert werden:
execute DBMS STATS.GATHER TABLE STATS('<schemaname>', '<TABLENAME>');
Prüfen Sie nach Ausführung der Prozedur erneut die Anzahl der Zeilen num rows in den o.g. Tabellen.

Ahmad Mustain Billah **Sascha Bauer**

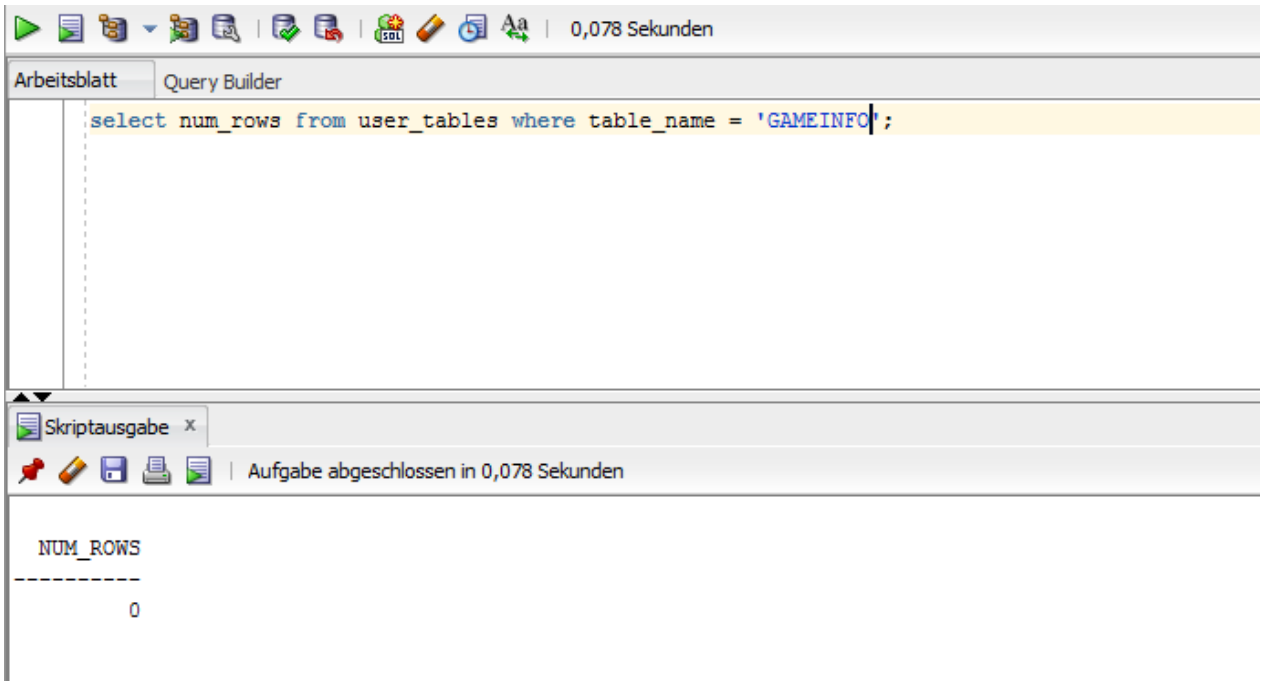




Ahmad Mustain Billah **Sascha Bauer**




```
  NUM_ROWS
----------
         0
```

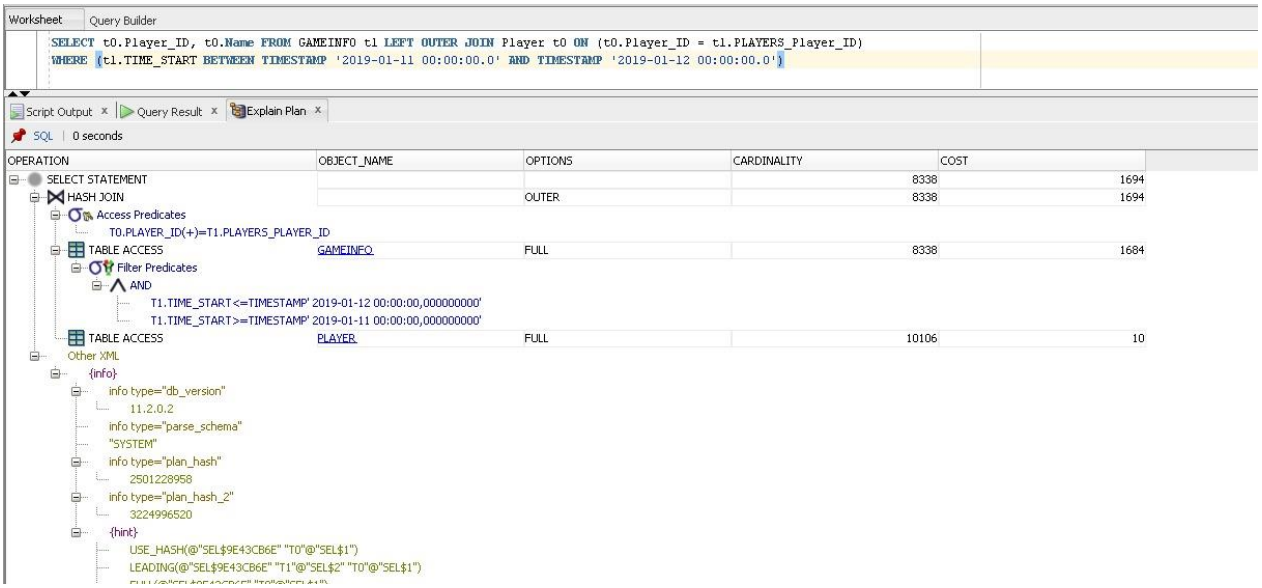



```
  NUM_ROWS
----------
         0
```

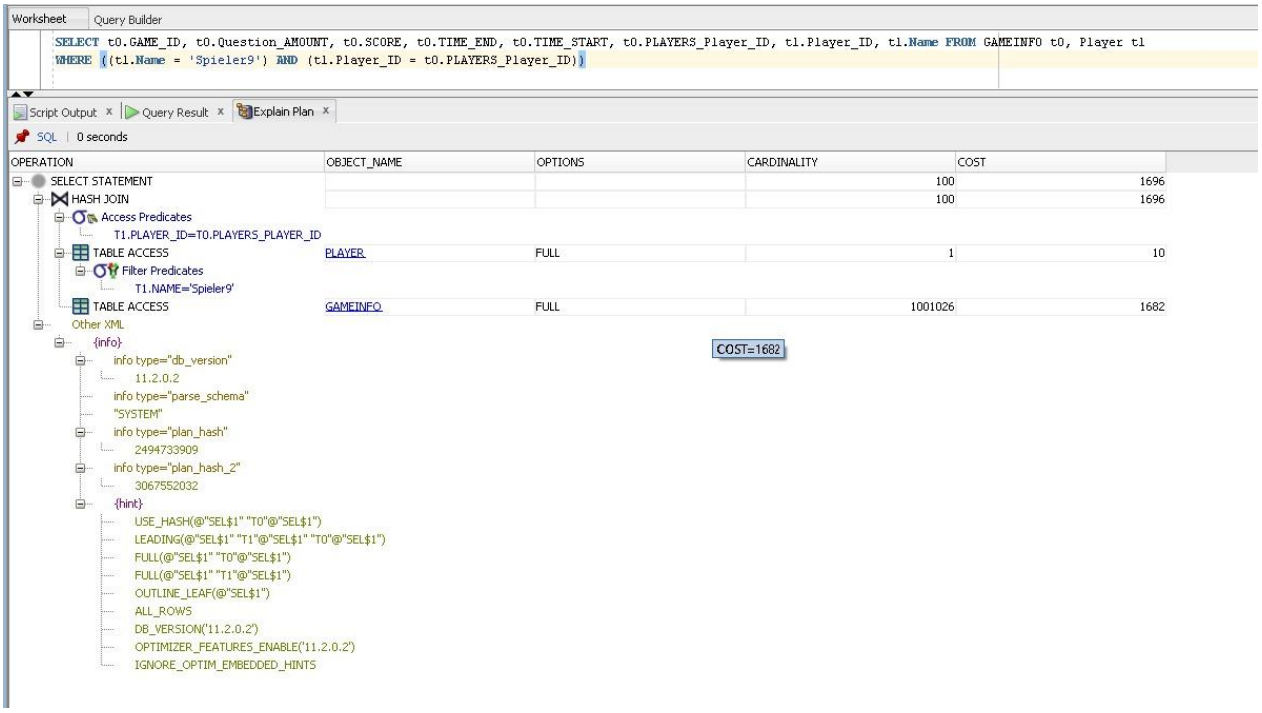
Ahmad Mustain Billah **Sascha Bauer**

4. Generieren Sie nochmals die Explains für die Queries, die Sie in der Vorbereitung zu diesem Praktikum erstellt haben. Was hat sich geändert? Welche Kosten werden nun für die Ausführung einer Query angegeben?

Analyse1:

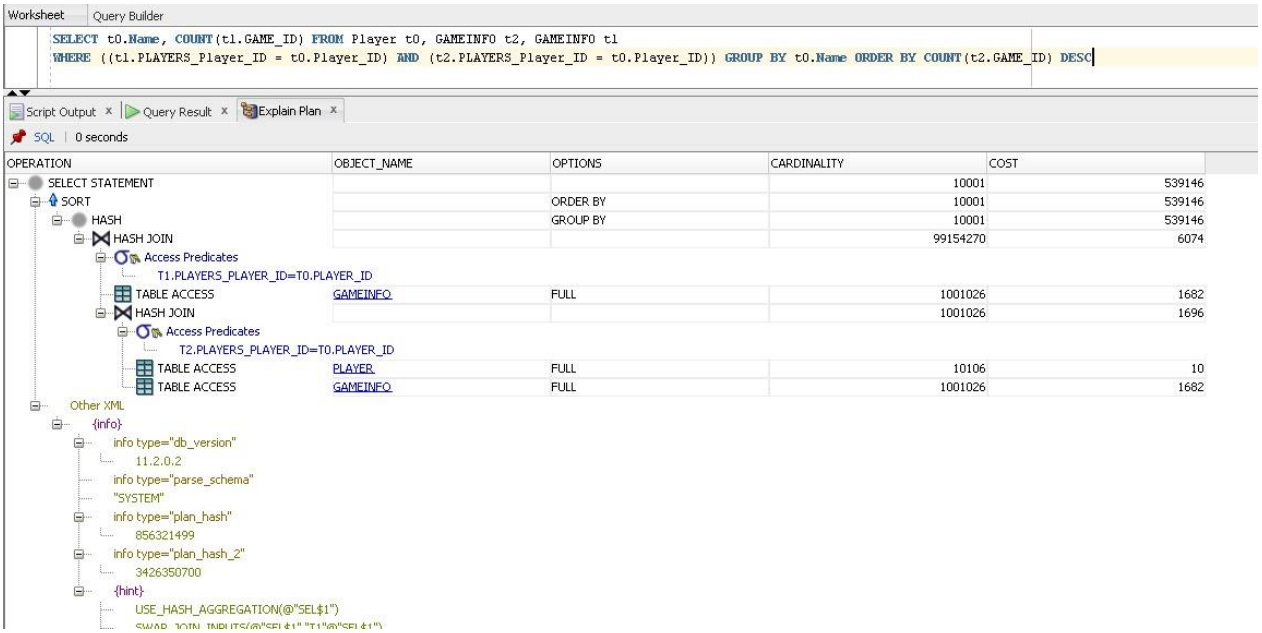


Analyse2:



Ahmad Mustain Billah **Sascha Bauer**

Analyse3:



Worksheet | Query Builder

```
SELECT t0.Name, COUNT(t1.GAME_ID) FROM Player t0, GAMEINFO t2, GAMEINFO t1
WHERE ((t1.PLAYERS_Player_ID = t0.Player_ID) AND (t2.PLAYERS_Player_ID = t0.Player_ID)) GROUP BY t0.Name ORDER BY COUNT(t2.GAME_ID) DESC
```

Script Output | Query Result | Explain Plan

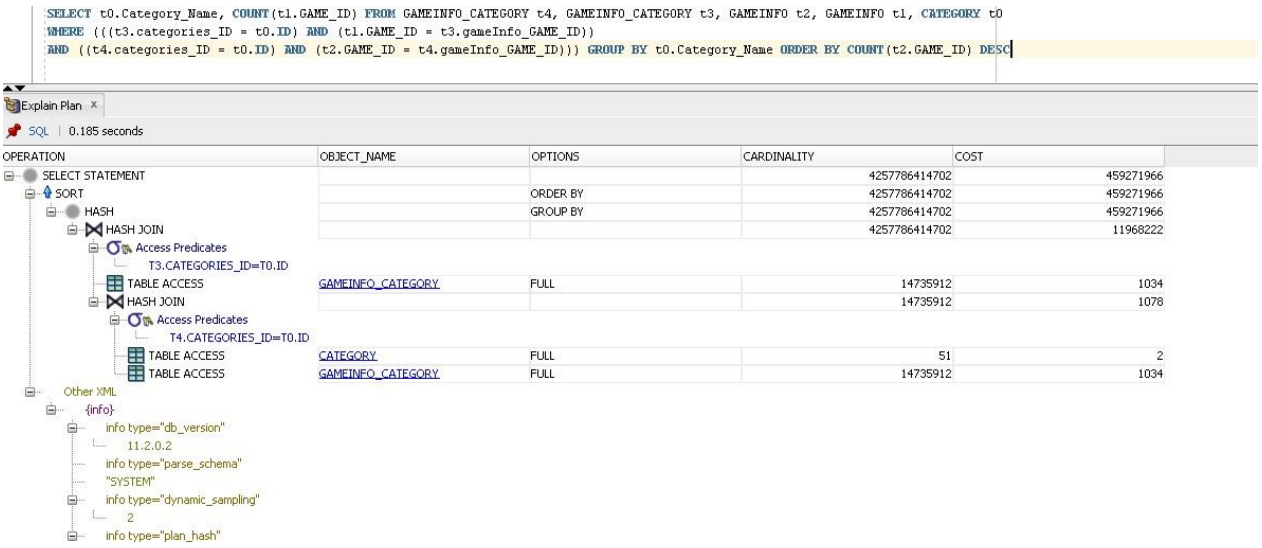
SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10001	539146
SORT		ORDER BY	10001	539146
HASH		GROUP BY	10001	539146
HASH JOIN			99154270	6074
Access Predicates				
T1.PLAYERS_PLAYER_ID=T0.PLAYER_ID				
TABLE ACCESS	GAMEINFO	FULL	1001026	1682
HASH JOIN			1001026	1696
Access Predicates				
T2.PLAYERS_PLAYER_ID=T0.PLAYER_ID				
TABLE ACCESS	PLAYER	FULL	10106	10
TABLE ACCESS	GAMEINFO	FULL	1001026	1682

Other XML

```
{info}
  info type="db_version"
    11.2.0.2
  info type="parse_schema"
  "SYSTEM"
  info type="plan_hash"
    856321499
  info type="plan_hash_2"
    3426350700
  {hint}
    USE_HASH_AGGREGATION(@"SEL$1")
```

Analyse4:



```
SELECT t0.Category_Name, COUNT(t1.GAME_ID) FROM GAMEINFO_CATEGORY t4, GAMEINFO_CATEGORY t3, GAMEINFO t2, GAMEINFO t1, CATEGORY t0
WHERE (((t3.categories_ID = t0.ID) AND (t1.GAME_ID = t3.gameInfo_GAME_ID))
AND ((t4.categories_ID = t0.ID) AND (t2.GAME_ID = t4.gameInfo_GAME_ID))) GROUP BY t0.Category_Name ORDER BY COUNT(t2.GAME_ID) DESC
```

Explain Plan

SQL | 0.185 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			4257786414702	459271966
SORT		ORDER BY	4257786414702	459271966
HASH		GROUP BY	4257786414702	459271966
HASH JOIN			4257786414702	11968222
Access Predicates				
T3.CATEGORIES_ID=T0.ID				
TABLE ACCESS	GAMEINFO_CATEGORY	FULL	14735912	1034
HASH JOIN			14735912	1078
Access Predicates				
T4.CATEGORIES_ID=T0.ID				
TABLE ACCESS	CATEGORY	FULL	51	2
TABLE ACCESS	GAMEINFO_CATEGORY	FULL	14735912	1034

Other XML

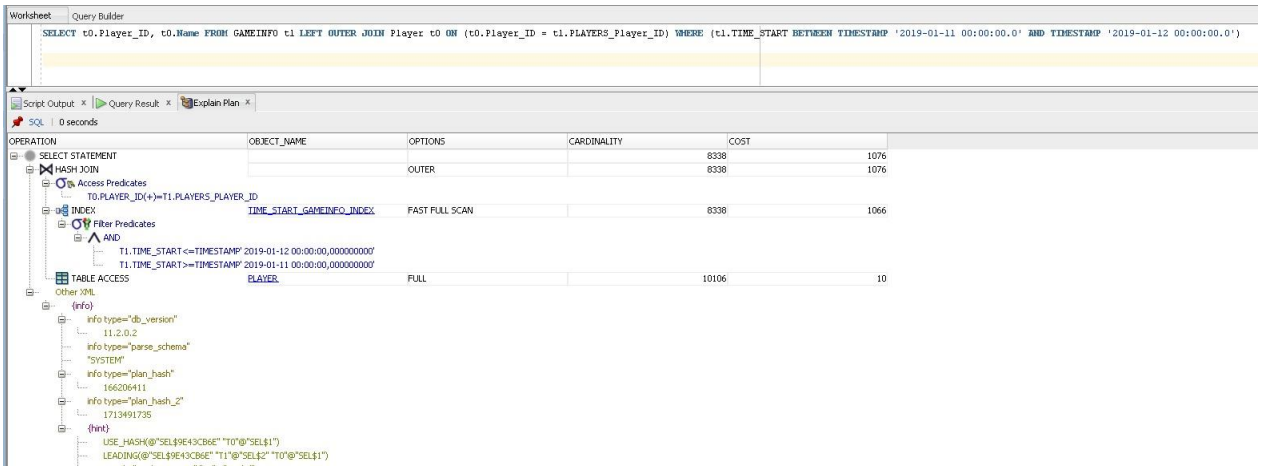
```
{info}
  info type="db_version"
    11.2.0.2
  info type="parse_schema"
  "SYSTEM"
  info type="dynamic_sampling"
    2
  info type="plan_hash"
```

5. Welche Spalten sind sinnvoll mit einem zusätzlichen Index zu belegen, um einen günstigeren Anfrageplan für die Verbundabfragen zu erhalten? Implementieren Sie die entsprechenden Indexe: create index <indexname> on <tablename> (col1[, col2, ...])
   und lassen Sie anschließend die Explains nochmals generieren.
   Welche Änderungen im Abfrageplan, in den Kosten und in der Ausführungszeit stellen Sie fest? Optimieren Sie, bis Sie f□ur mindestens drei der Abfragen eine wesentliche Verbesserung erreichen.

Für den 1.Query erstellen wir ein Index für die Spalten PLAYERS_PLAYER_ID (fk von Tabelle PLAYER) und TIME_START in der Tabelle GAMEINFO.

create index time_start_gameinfo_index on GAMEINFO (PLAYERS_PLAYER_ID, TIME_START);

Analyse1 anschließend:

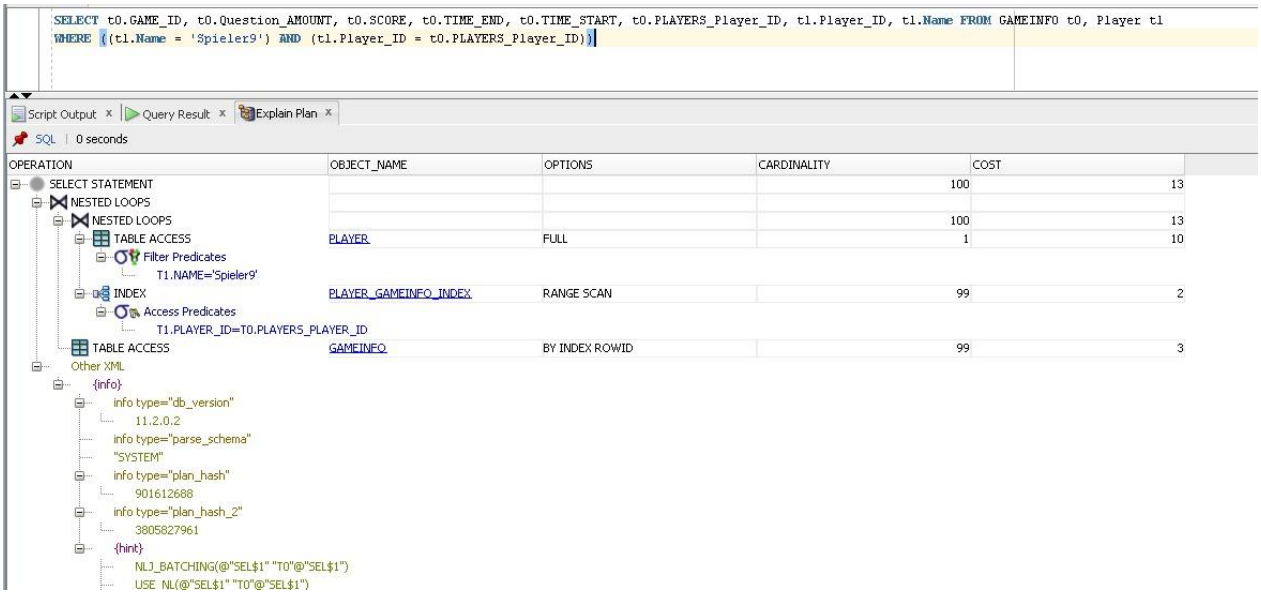


Ahmad Mustain Billah **Sascha Bauer**

Für den 2.- und 3.Querie erstellen wir einen Index für die Spalte PLAYERS_PLAYER_ID (fk von Tabelle PLAYER) in der Tabelle GAMEINFO.

create index player_gameinfo_index on GAMEINFO (PLAYERS_PLAYER_ID);

Analyse2 anschließend:



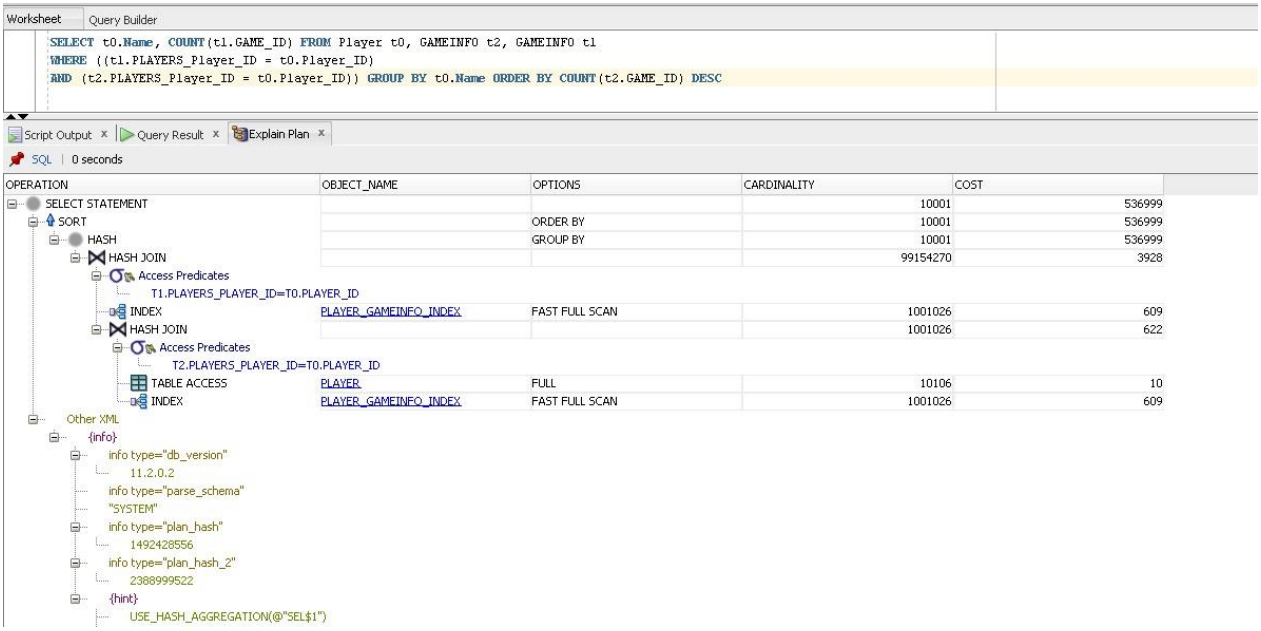
```
SELECT t0.GAME_ID, t0.Question_AMOUNT, t0.SCORE, t0.TIME_END, t0.TIME_START, t0.PLAYERS_Player_ID, t1.Player_ID, t1.Name FROM GAMEINFO t0, Player t1
WHERE ((t1.Name = 'Spieler9') AND (t1.Player_ID = t0.PLAYERS_Player_ID))
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			100	13
NESTED LOOPS				
NESTED LOOPS			100	13
TABLE ACCESS	PLAYER	FULL	1	10
Filter Predicates				
T1.NAME='Spieler9'				
INDEX	PLAYER_GAMEINFO_INDEX	RANGE SCAN	99	2
Access Predicates				
T1.PLAYER_ID=T0.PLAYERS_PLAYER_ID				
TABLE ACCESS	GAMEINFO	BY INDEX ROWID	99	3
Other XML				
{info}				
info type="db_version"				
11.2.0.2				
info type="parse_schema"				
"SYSTEM"				
info type="plan_hash"				
901612688				
info type="plan_hash_2"				
3805827961				
{hint}				
NLJ_BATCHING(@"SEL$1" "T0"@"SEL$1")				
USE_NL(@"SEL$1" "T0"@"SEL$1")				

Analyse3 anschließend:



```
SELECT t0.Name, COUNT(t1.GAME_ID) FROM Player t0, GAMEINFO t2, GAMEINFO t1
WHERE ((t1.PLAYERS_Player_ID = t0.Player_ID)
AND (t2.PLAYERS_Player_ID = t0.Player_ID)) GROUP BY t0.Name ORDER BY COUNT(t2.GAME_ID) DESC
```

Worksheet | Query Builder

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10001	536999
SORT		ORDER BY	10001	536999
HASH		GROUP BY	10001	536999
HASH JOIN			99154270	3928
Access Predicates				
T1.PLAYERS_PLAYER_ID=T0.PLAYER_ID				
INDEX	PLAYER_GAMEINFO_INDEX	FAST FULL SCAN	1001026	609
HASH JOIN			1001026	622
Access Predicates				
T2.PLAYERS_PLAYER_ID=T0.PLAYER_ID				
TABLE ACCESS	PLAYER	FULL	10106	10
INDEX	PLAYER_GAMEINFO_INDEX	FAST FULL SCAN	1001026	609
Other XML				
{info}				
info type="db_version"				
11.2.0.2				
info type="parse_schema"				
"SYSTEM"				
info type="plan_hash"				
1492428556				
info type="plan_hash_2"				
2388999522				
{hint}				
USE_HASH_AGGREGATION(@"SEL$1")				