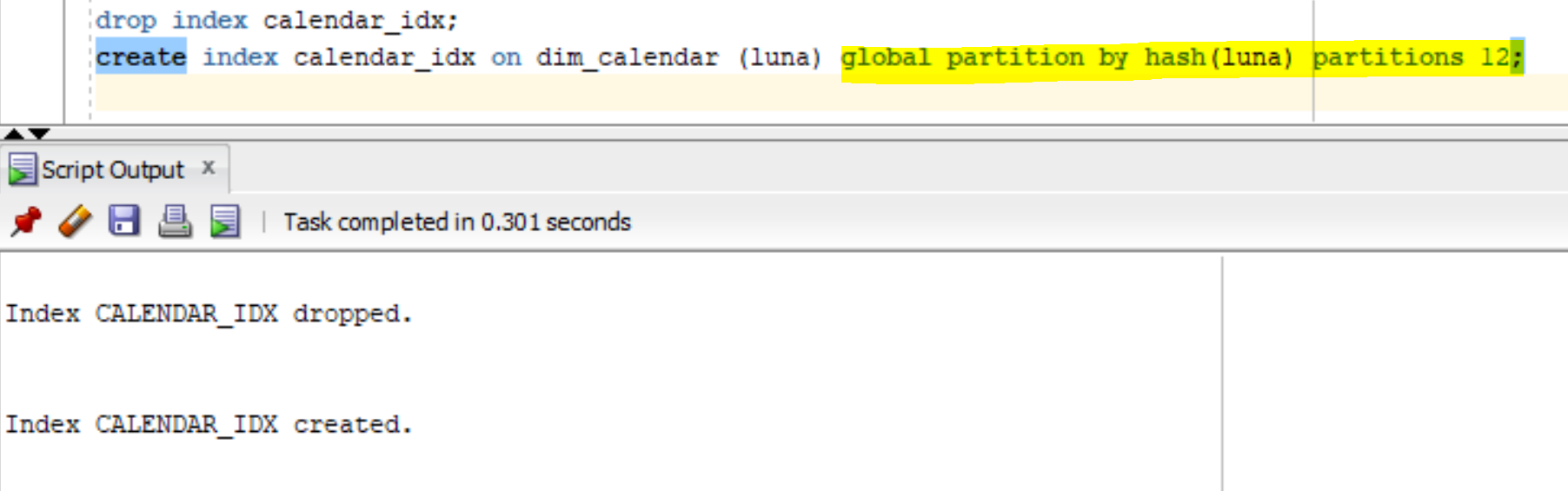
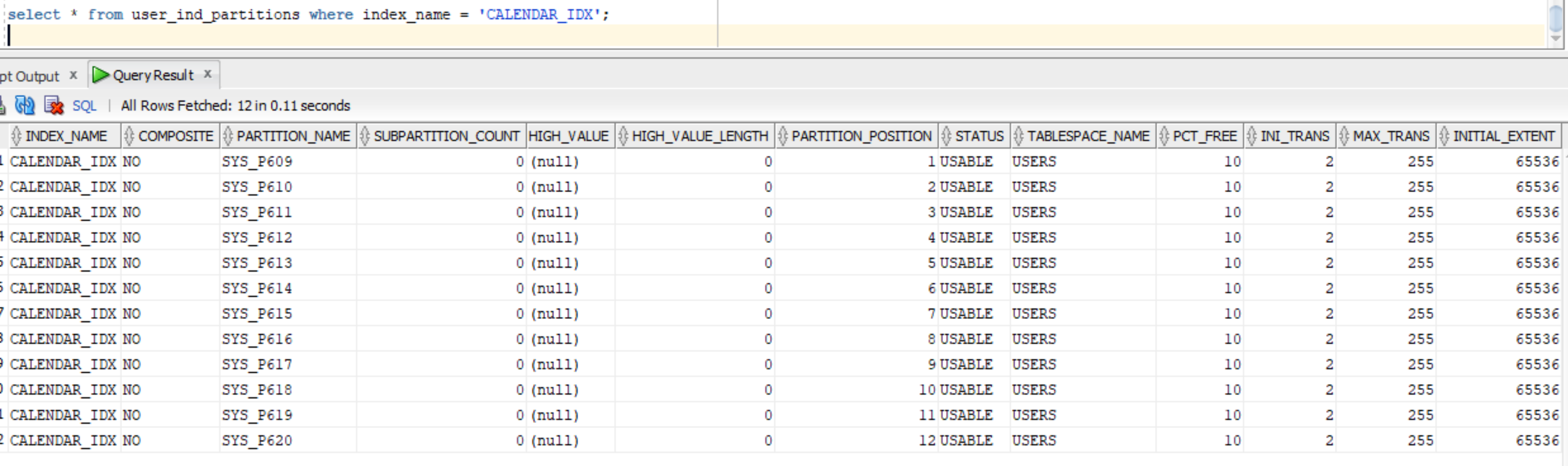
1. Creare index global



Consulatare dictionarul datelor pentru verificarea crearii partitiilor pe index

Folosire index

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_global' FOR

select /\*+ index(dim\_calendar calendar\_idx) \*/ \* from dim\_calendar where luna= 'June';

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_global','serial'));

Plan hash value: 2382551087

---------------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

---------------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 167 | 5344 | 8 (0)| 00:00:01 | | |

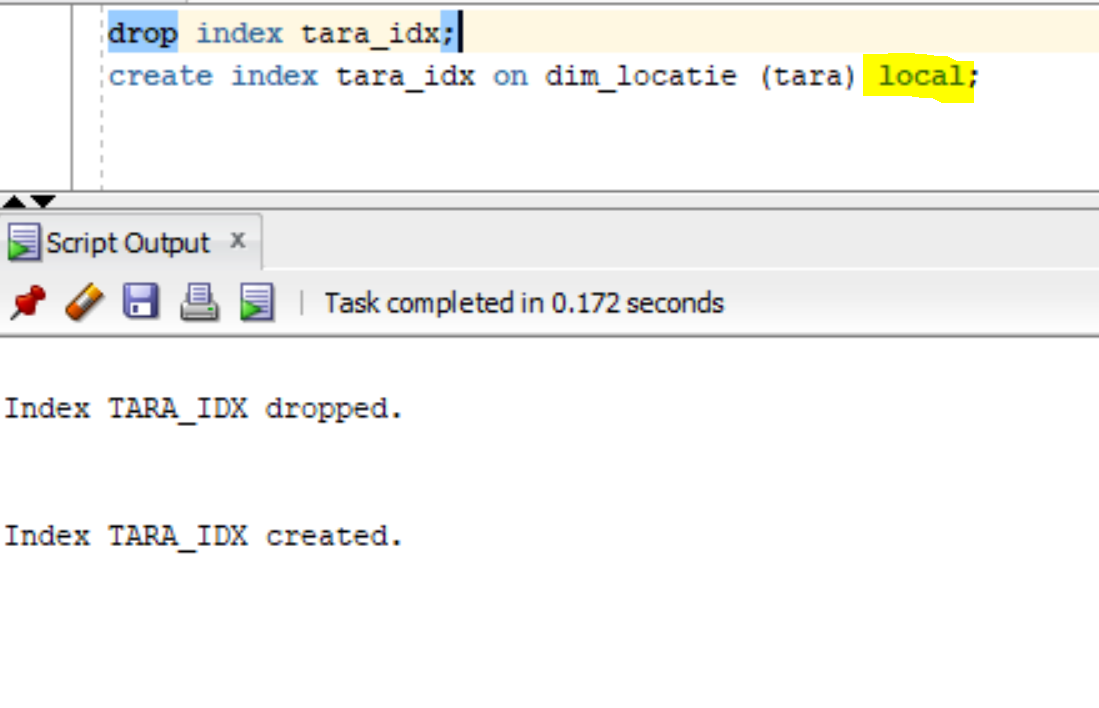
| 1 | PARTITION HASH SINGLE | | 167 | 5344 | 8 (0)| 00:00:01 | 7 | 7 |

| 2 | TABLE ACCESS BY INDEX ROWID BATCHED| DIM\_CALENDAR | 167 | 5344 | 8 (0)| 00:00:01 | | |

|\* 3 | INDEX RANGE SCAN | CALENDAR\_IDX | 167 | | 1 (0)| 00:00:01 | 7 | 7 |

---------------------------------------------------------------------------------------------------------------------

2. Creare index local



Folosire index

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_local' FOR

select \* from dim\_locatie where tara = 'Romania';

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_local','serial'));

Plan hash value: 1361974882

--------------------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

--------------------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 4 | 124 | 8 (0)| 00:00:01 | | |

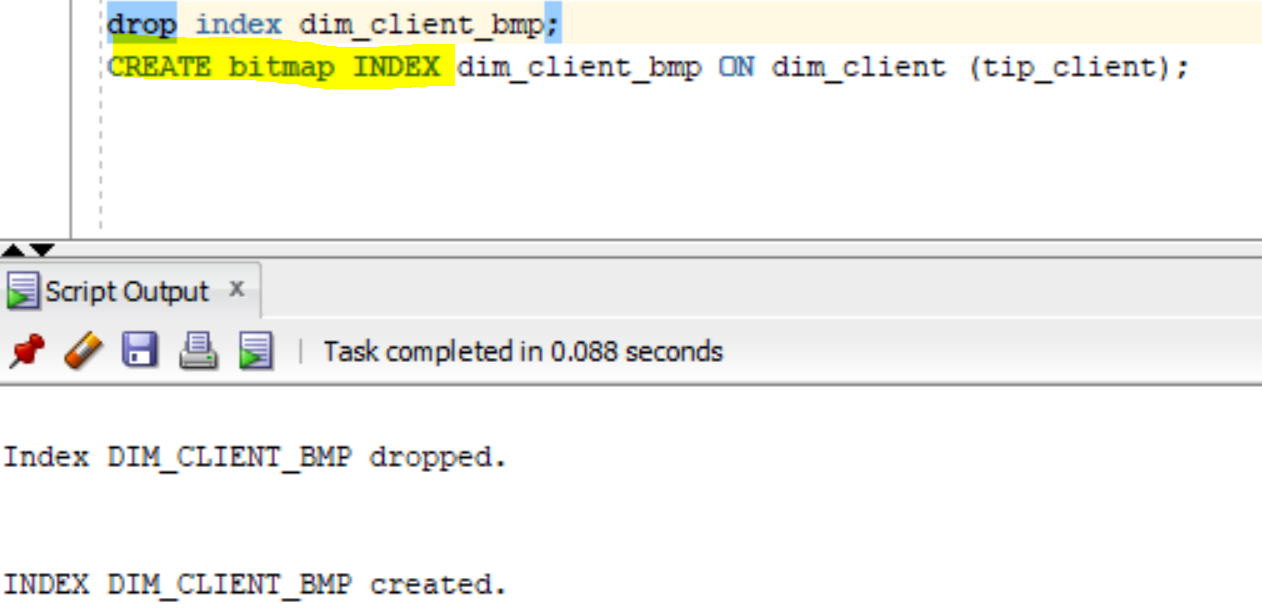
| 1 | PARTITION LIST ALL | | 4 | 124 | 8 (0)| 00:00:01 | 1 | 5 |

| 2 | TABLE ACCESS BY LOCAL INDEX ROWID BATCHED| DIM\_LOCATIE | 4 | 124 | 8 (0)| 00:00:01 | 1 | 5 |

|\* 3 | INDEX RANGE SCAN | TARA\_IDX | 4 | | 4 (0)| 00:00:01 | 1 | 5 |

--------------------------------------------------------------------------------------------------------------------------

3. Bitmap index



Folosire index

analyze index dim\_client\_bmp compute statistics;

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_bmp' FOR

select /\*+ index(dim\_client dim\_client\_bmp) \*/ \* from dim\_client where tip\_client = 'PF';

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_bmp','serial'));

Plan hash value: 2457481617

------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 6 | 156 | 7 (0)| 00:00:01 |

| 1 | TABLE ACCESS BY INDEX ROWID BATCHED| DIM\_CLIENT | 6 | 156 | 7 (0)| 00:00:01 |

| 2 | BITMAP CONVERSION TO ROWIDS | | | | | |

|\* 3 | BITMAP INDEX SINGLE VALUE | DIM\_CLIENT\_BMP | | | | |

4. Bitmap join index

create bitmap index bmp\_join\_idx on fact\_tranzactii (suma) from fact\_tranzactii f, dim\_comerciant d

where f.id\_comerciant = d.id\_comerciant local;

analyze index bmp\_join\_idx compute statistics;

alter session set star\_transformation\_enabled = true;

Folosire bitmap join index

EXPLAIN PLAN

SET STATEMENT\_ID = 's1\_index\_join\_bmp' FOR

select /\*+ STAR\_TRANSFORMATION \*/

/\*+ FACT(fact\_tranzactii) \*/

f.suma, d.nume

from fact\_tranzactii f, dim\_comerciant d

where f.id\_comerciant = d.id\_comerciant

and suma = 50;

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 's1\_index\_join\_bmp','serial'));

Plan hash value: 3558226070

--------------------------------------------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time | Pstart| Pstop |

--------------------------------------------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 1 | 37 | 1 (0)| 00:00:01 | | |

| 1 | NESTED LOOPS | | 1 | 37 | 1 (0)| 00:00:01 | | |

| 2 | NESTED LOOPS | | 1 | 37 | 1 (0)| 00:00:01 | | |

| 3 | PARTITION RANGE ALL | | 1 | 26 | 1 (0)| 00:00:01 | 1 |1048575|

|\* 4 | TABLE ACCESS BY LOCAL INDEX ROWID BATCHED| FACT\_TRANZACTII | 1 | 26 | 1 (0)| 00:00:01 | 1 |1048575|

| 5 | BITMAP CONVERSION TO ROWIDS | | | | | | | |

|\* 6 | BITMAP INDEX SINGLE VALUE | BMP\_JOIN\_IDX | | | | | 1 |1048575|

|\* 7 | INDEX UNIQUE SCAN | SYS\_C008471 | 1 | | 0 (0)| 00:00:01 | | |

| 8 | TABLE ACCESS BY INDEX ROWID | DIM\_COMERCIANT | 1 | 11 | 0 (0)| 00:00:01 | | |

--------------------------------------------------------------------------------------------------------------------------------