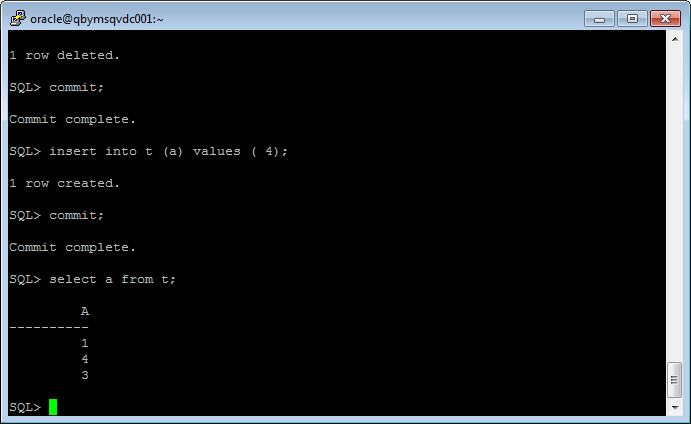
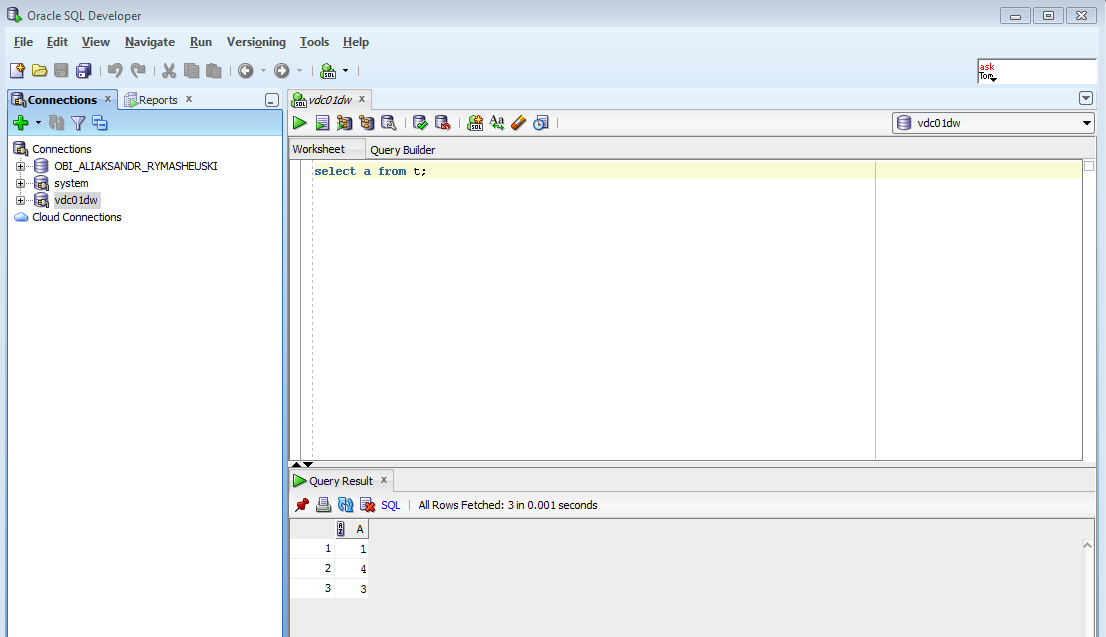
# 2. Heap Organized Tables

## 2.1 Task 1 – Heap Understanding

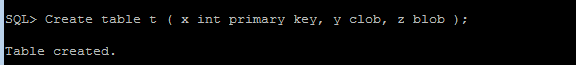




## 2.2 Task 2 – Understanding Low level of data abstraction: Heap Table Segments

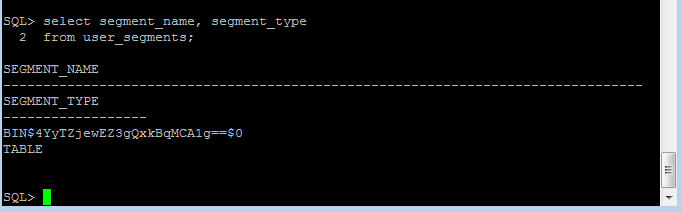
Step 1:

# Create table t ( x int primary key, y clob, z blob );



Step 2:

# select segment\_name, segment\_type 2 from user\_segments;



Step 3:

# Create table t

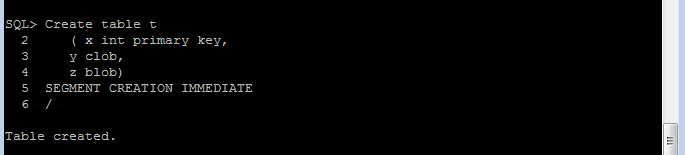
( x int primary key,

y clob,

z blob )

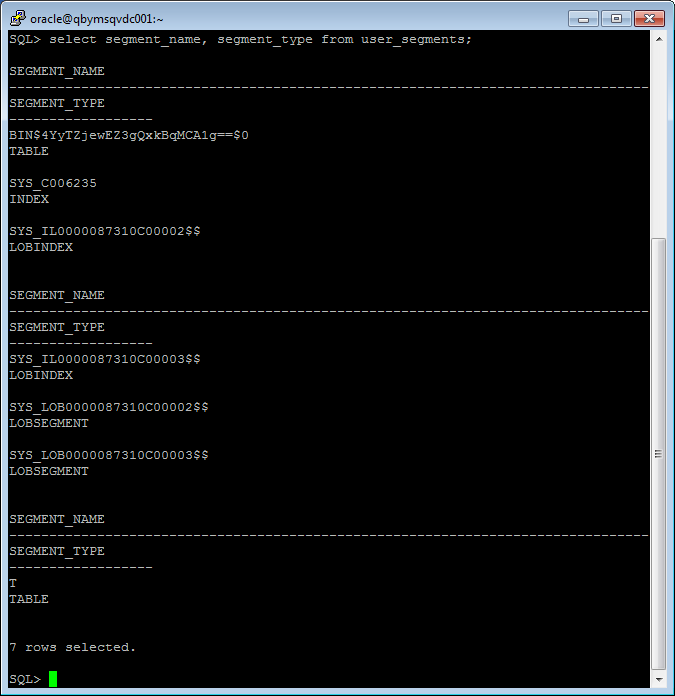
SEGMENT CREATION IMMEDIATE

/



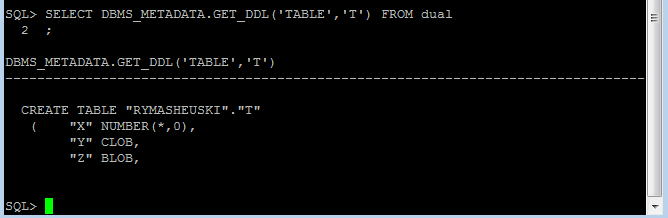
**Step 4:**

# select segment\_name, segment\_type 2 from user\_segments;



Step 5:

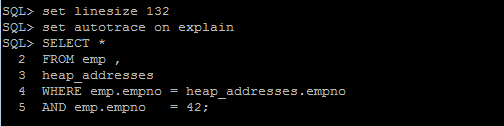
# SELECT DBMS\_METADATA.GET\_DDL('TABLE','T') FROM dual

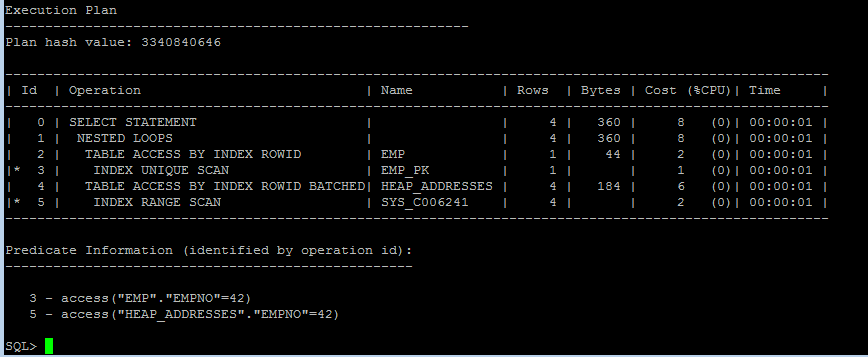


# 3. Index Organized Tables

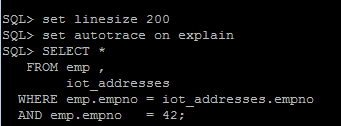
## 3.1 Task 3: Compare performance of using IOT tables

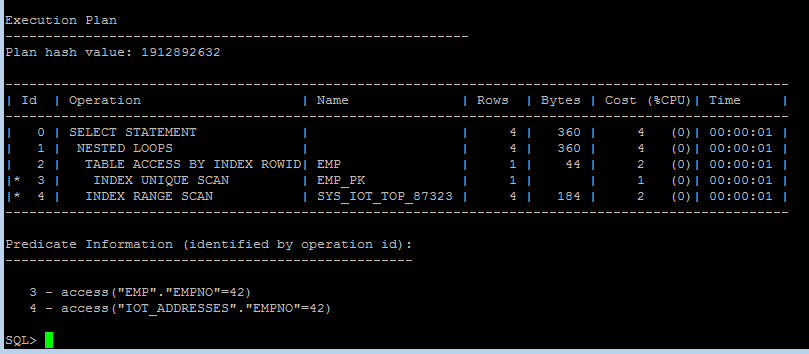
Heap-organized table





IOT

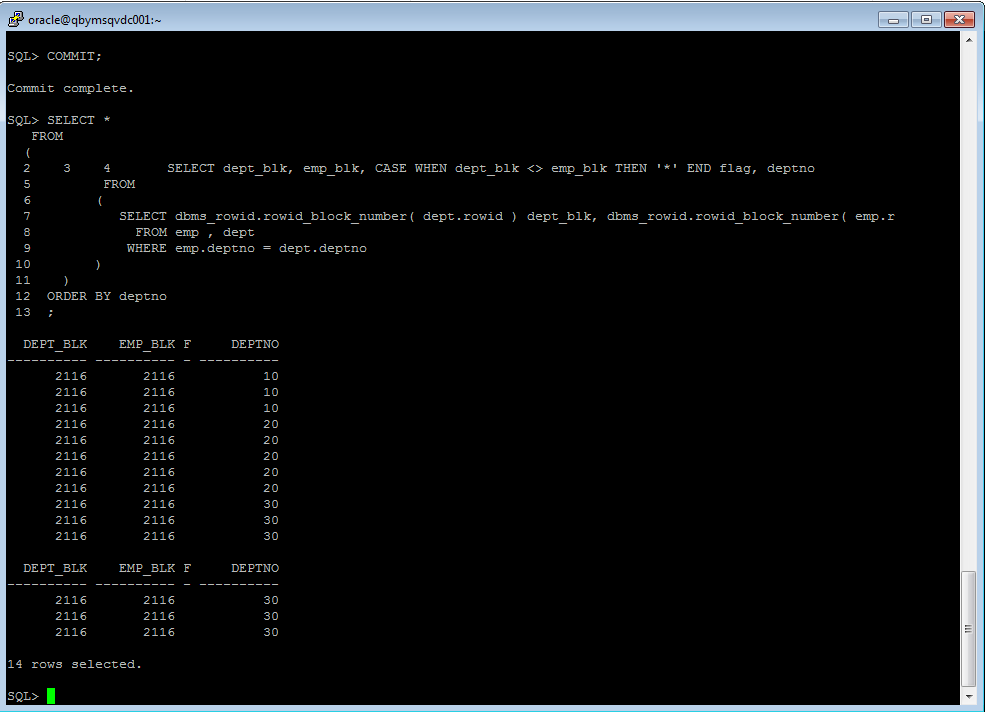




When Index-organized table is queried all the necessary data are contained in the index. Therefore, there is no need to perform access to the table by rowid. This explains why execution plan to the heap-organized table contains more rows than execution plan to the IOT (“Table access by index rowid batched”). Therefore, cost of the query to the heap-organized table bigger than cost of the query to the IOT.

# 4. Index Clustered Tables

## 4.1 Task 4: Analyses Cluster Storage by Blocks



When we created an index-clustered table we specify the size of 1024. This means that for each value of the cluster key Oracle will take 1024 bytes per block. Thus, one block (8Kb) can store a maximum of 7 values ​​cluster keys (8Kb/1024)=7). In our table we have only 3 distinct values of the deptno (clustering column) and all the values store in one block.

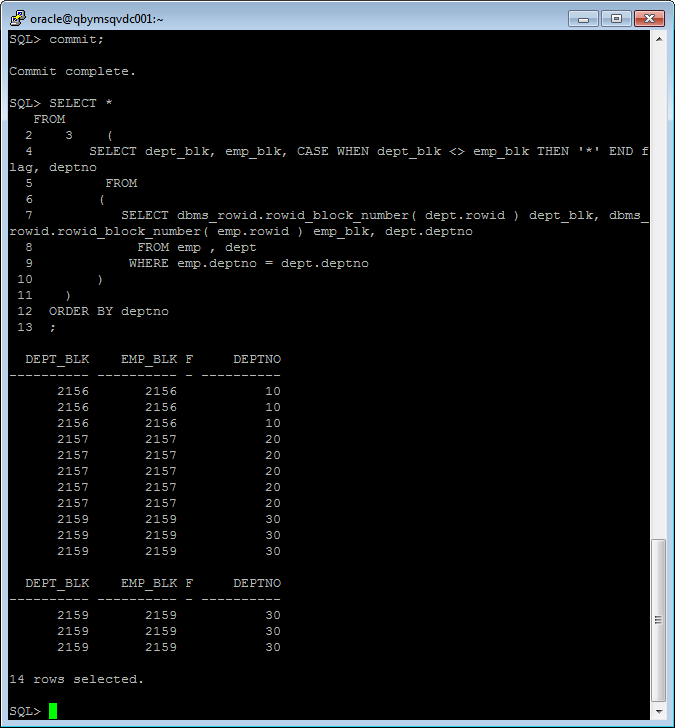
Advantages:

-fast join operations with tables that are stored in the cluster (tables are stored in pre-joining condition)

-rows with the same cluster key are stored in one block (or more blocks, but very densely)

# 5. Hash Clustered Tables

# 5.1 Task 5: Analyses Cluster Storage by Blocks



In the hash-clustered tables, you determine the number of hash keys your table will have. So, hash cluster pre-allocates enough space to hold (HASHKEYS/trunc(blocksize/SIZE)) bytes of data (this type of cluster don’t dynamically allocates space as needs it such as index clustered tables do). For our hash-clustered tables for each haskey is allocated one block.

Advantages:

* all rows corresponding to hash key is stored in one place on the HDD (files are not fragmented, if you set the correct options when you create cluster, such as number of haskeys and size)
* fast join operations with tables that are stored in the cluster (tables are stored in pre-joining condition)