**DWBI**

**QUESTION 1**

Q1. Create a copy of customers table and name it customers\_copy\_btree\_&lt;rollno&gt;.

Create individual b-tree indexes on the following columns of the table

customers\_copy\_btree\_&lt;rollno&gt; :

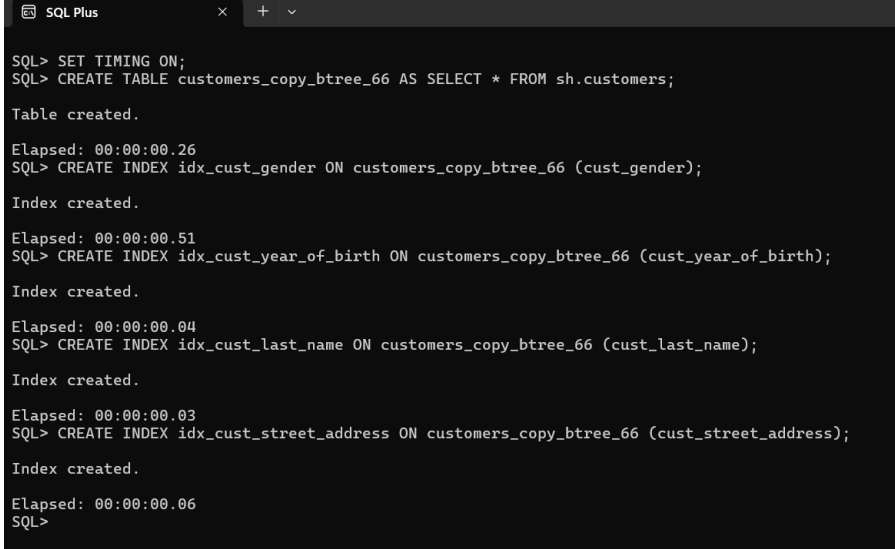
(a) cust\_gender

(b) cust\_year\_of\_birth

(c) cust\_last\_name

(d) cust\_street\_address

How long does it take to create the indexes?

****

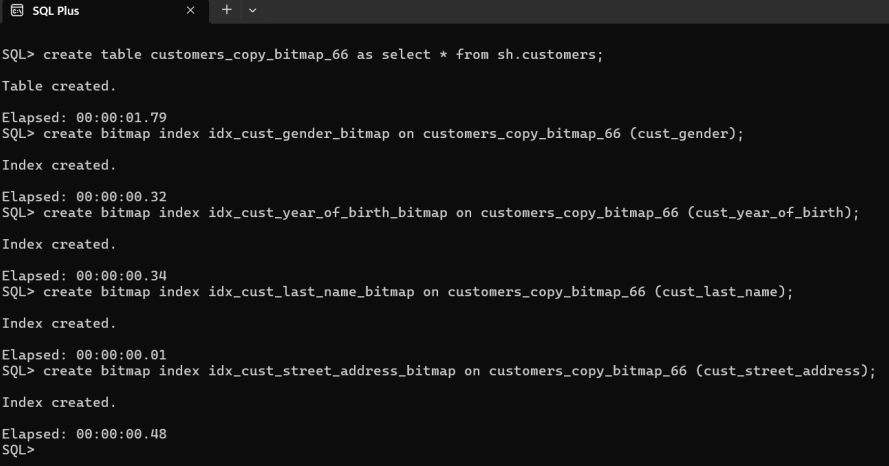
It takes about 0.5 ms each to create index on 4 difference columns

**QUESTION 2**

Q2. Create bitmap indexes on the above columns. How long does it take to create bitmap indexes? Compare it

with the results of btree index creation.

For Q1 and Q2, make a comparison table.

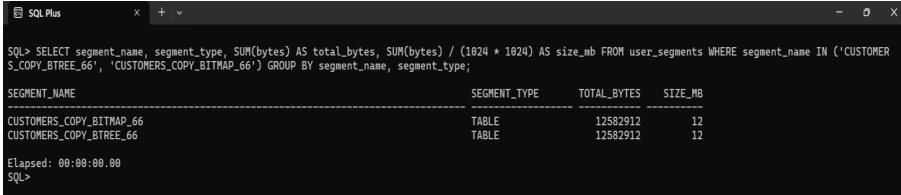
****

**Here also it takes near about 0.5 ms each for creating bitmap indexes.**

**QUESTION 3**

Q3. Find the size of each segment: customers\_copy\_bitmap and customers\_copy\_btree

(Hint : Use users\_segment table)

****

**We can see that both segments are having 12 mb size.**

**Question 4:**

Q4. Do as directed :

a. Create function based index on Employee table of HR schema. Function should be on salary attribute based on

commission percentage.

b. Find out list of employees having commission percentage less than 50000.

c. Create function based index on employee name for Upper and lower function.

d. Create user table with attributes (UserId, UserName, Gender)

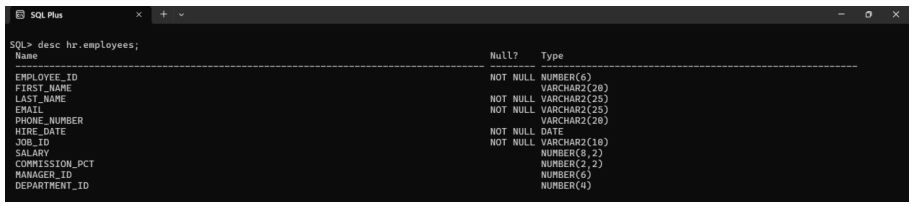
e. Insert 10000 records in user table

f. Build regular index on Username

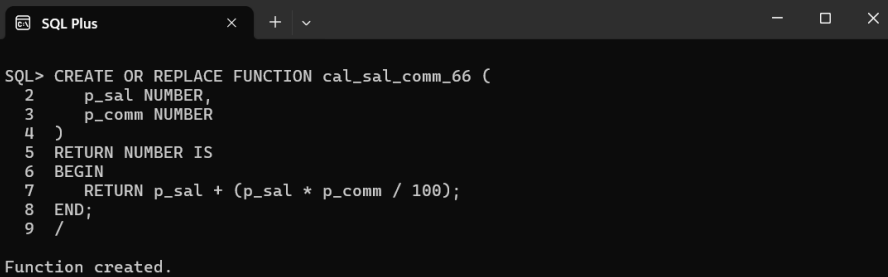
g. Build function based index on user name based on Upper function

h. Compare the response time and comment.

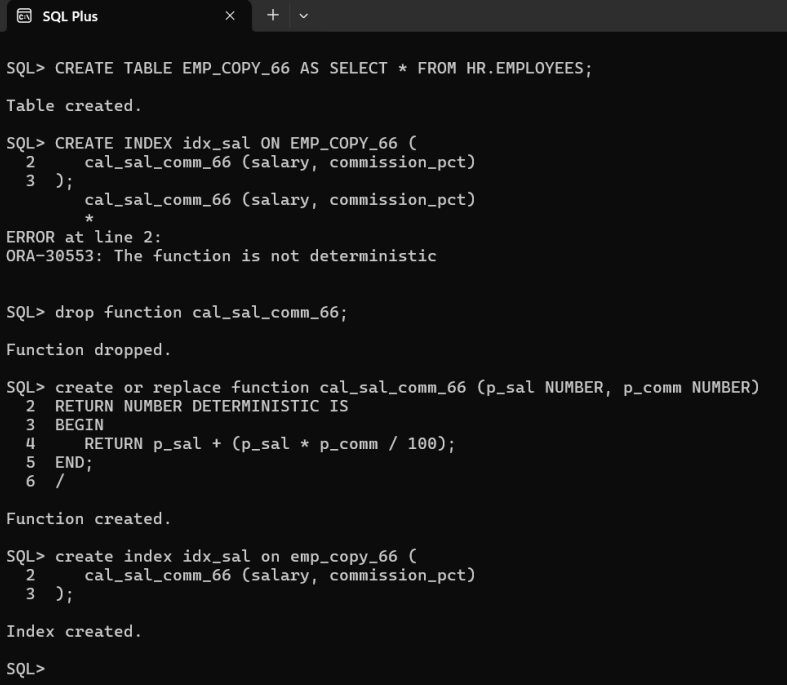
**Query Plan: First we see the employees tables from HR schema to understand all the attributes present in employee table.**

****

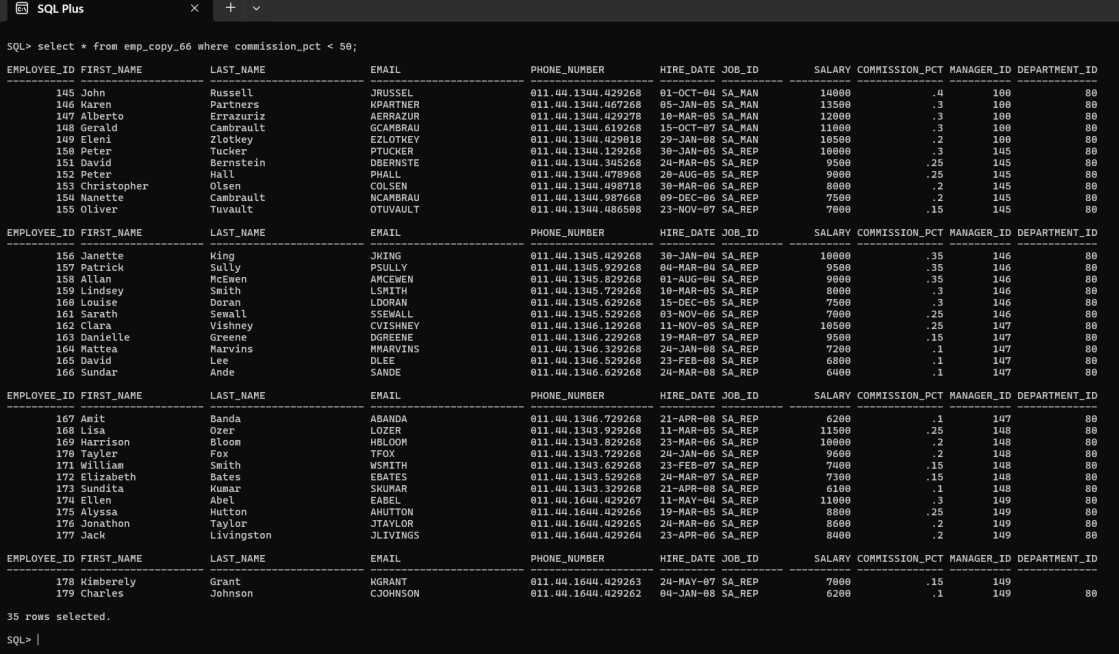
**4.1**

****

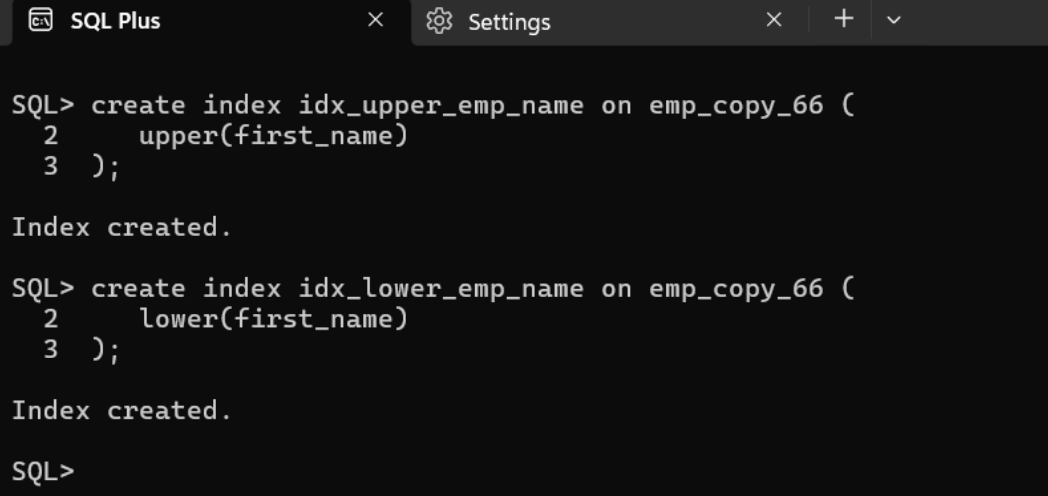
**We need out function to be deterministic in order to use it as index.**

****

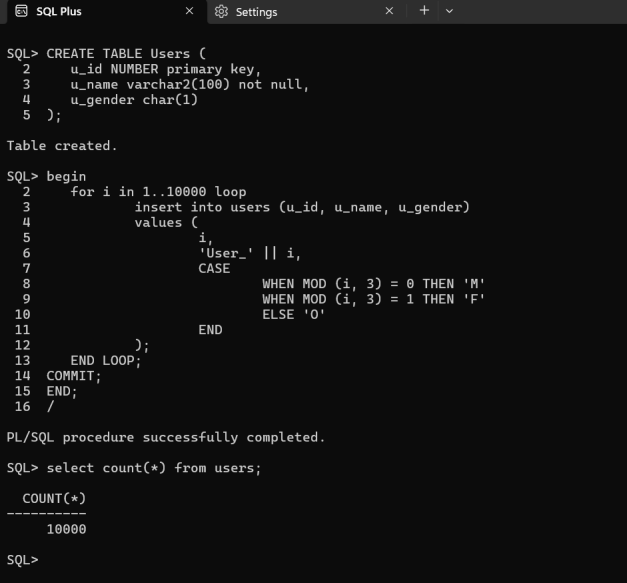
**4.2**

****

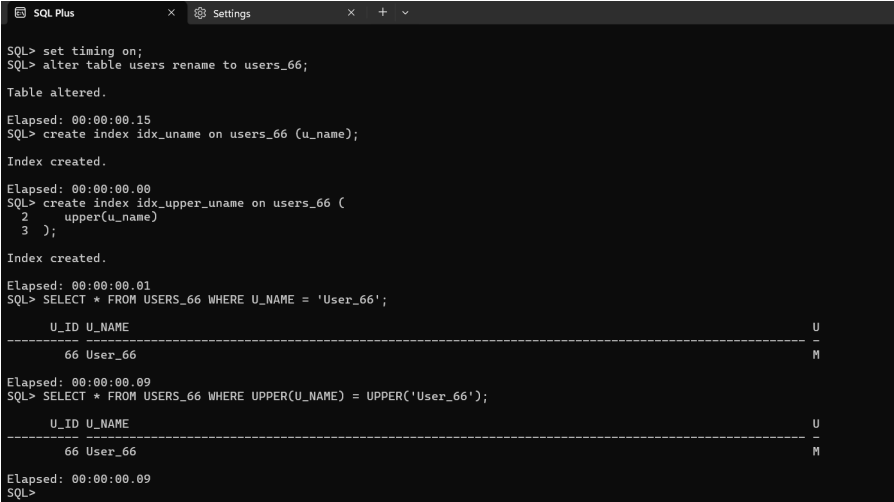
**4.3**

****

**4.4 and 4.5**

****

**4.6 and 4.7**

****

**Both queries take about the same time to execute i.e. 0.9ms on my device.**

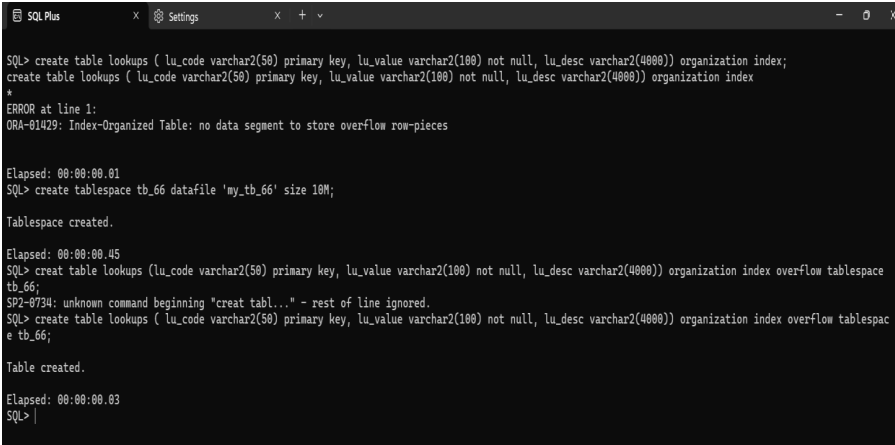
**Q5. Do as directed :**

a. Create an IOT look\_ups with the attributes (lookup\_code, lookup\_value, lookup\_description).

b. Constraint: lookup\_code should be primary key

c. lookup\_description should be in overflow area.

**First we need a tablespace to work on : I have created a tablespace named tb\_66**

****

**Question 6:**

a. Create a Index Organized Table(IOT) emp\_iot based on hr.employees

b. Create a Index Organized Table(IOT) emp101\_emp based on hr.employees. Place

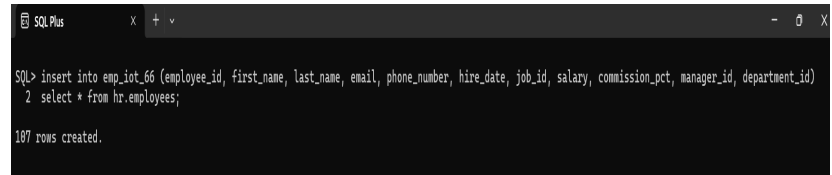
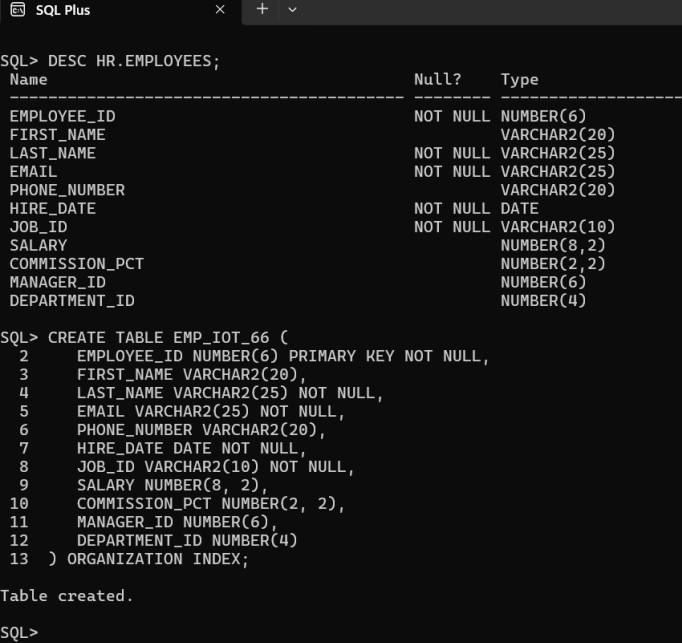
the column hiredate in overflow area.

c. Compare the timings of executing select all from employees,emp\_iot, and

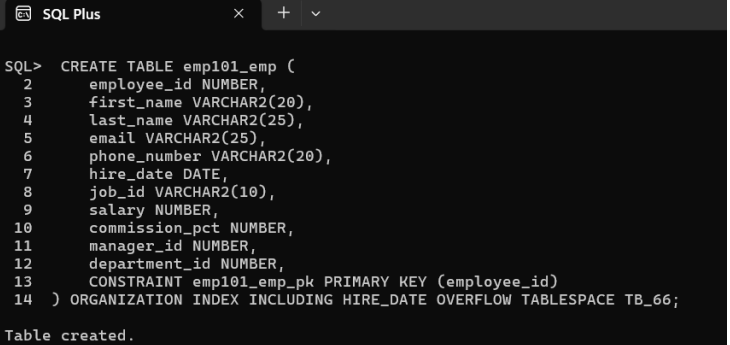
emp101\_iot. Comment on your observations.

**Query Plan : We need to check the attributes of Employees table to create a replica and perform required IOT operation hence we first check employees and then we can execute and create emp\_iot\_66 table.**

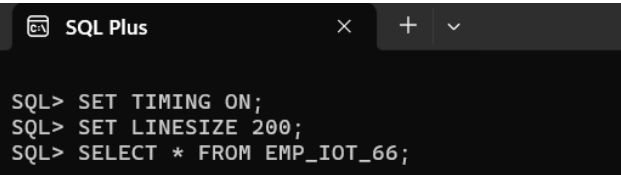
**6.1**

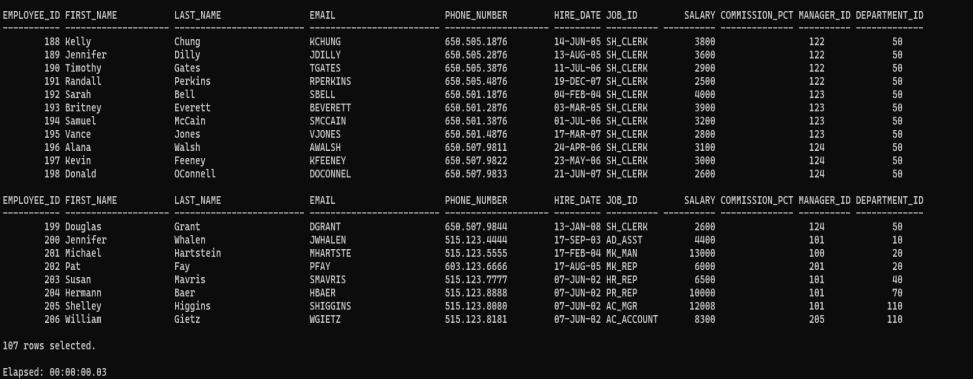
****

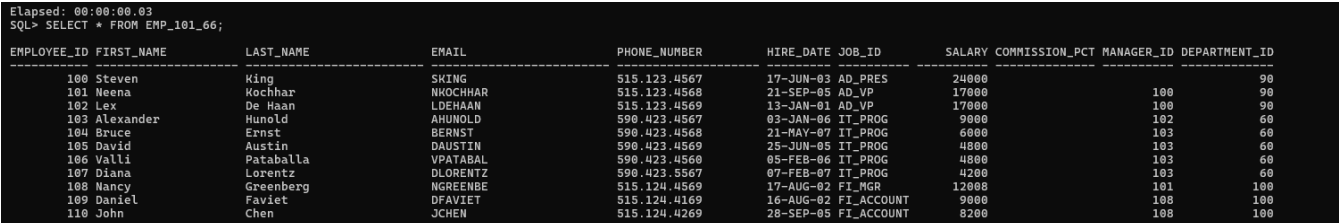
**6.2**

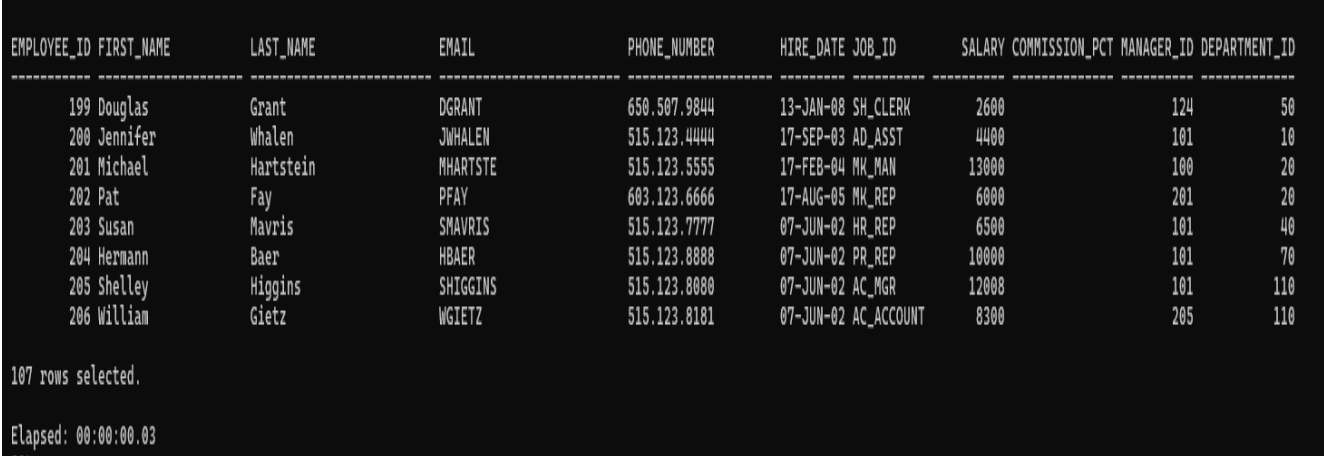
****

**6.3**

****

****

****

****

**CONCLUSION:**

In conclusion, we have successfully implemented indexes and IOT in Oracle, including creating a copy of the customers table, creating individual b-tree and bitmap indexes, comparing their creation times, and analyzing their sizes. We also created function-based indexes, an IOT with an overflow area, and compared the response times of regular and function-based indexes. Finally, we created two Index Organized Tables (IOTs) and compared their query execution times. These exercises demonstrate the power of indexing and IOTs in optimizing database performance.