



**VIT<sup>®</sup>**  
**AP**

## **CSE2007 : Database Management Systems**

**Name : Gudi Varaprasad**

**Reg. No. : 19BCE7048**

**School of Computer Science and Engineering**

**Lab Slot : L45 + L46**

**Date : 15 – 04 – 2021**

**Submitted to : Ms. Dhanavanthini madam**

### **SQL Joins**

1. Consider the following relations :

```
Run SQL Command Line

SQL> SELECT * FROM Trip_guides;

      ENO  ENAME                HRATE  MONTHSEMP
-----
      978  Siyabonge Nomvete      30      6
      942  Jean-Marc Ithier        35     12
      923  Tyrone Arendse          32      6
      982  Mathew Booth             25      0

SQL> SELECT * FROM Trip_locations;

      LCODE  LNAME
-----
      1024  Table Mountain
      1025  Robben Island
      1026  Kruger National Park
      1027  St Lucia

SQL> SELECT * FROM Trip_Expenditures;

      LCODE      ENO      HWORKED      TGSIZE
-----
      1024      978      5      8
      1027      942      8      4
      1025      923      3     20
      1026      982      6      8
      1024      978      5      8
      1025      978      3     16

6 rows selected.

SQL>
```

Answer the following using SQL Joins :

i. Display the employee names with their respective tour locations.

```
SQL> SELECT DISTINCT Trip_guides.ename, Trip_locations.lname FROM Trip_guides
2 LEFT JOIN Trip_Expenditures ON Trip_guides.eno = Trip_Expenditures.eno
3 LEFT JOIN Trip_locations ON Trip_locations.lcode = Trip_Expenditures.lcode;

ENAME                LNAME
-----
Jean-Marc Ithier     St Lucia
Mathew Booth         Kruger National Park
Siyabonge Nomvete    Robben Island
Siyabonge Nomvete    Table Mountain
Tyrone Arendse       Robben Island
```

ii. Display the employee\_numbers who had a trip to Table Mountain

```
SQL> SELECT DISTINCT Trip_guides.eno FROM Trip_guides
  2  LEFT JOIN Trip_Expenditions ON Trip_guides.eno = Trip_Expenditions.eno
  3  LEFT JOIN Trip_locations ON Trip_locations.lcode = Trip_Expenditions.lcode
  4  WHERE Trip_locations.lname = 'Table Mountain';

      ENO
-----
      978
```

iii. Display all the employees name regardless of whether they have undertaken a trip or not? (use left outer join)

```
SQL> SELECT ALL Trip_guides.ename FROM Trip_guides
  2  LEFT OUTER JOIN Trip_Expenditions ON Trip_guides.eno = Trip_Expenditions.eno
  3  LEFT OUTER JOIN Trip_locations ON Trip_locations.lcode = Trip_Expenditions.lcode;

ENAME
-----
Siyabonge Nomvete
Jean-Marc Ithier
Tyrone Arendse
Mathew Booth
Siyabonge Nomvete
Siyabonge Nomvete

6 rows selected.
```

iv. How do we return the employee numbers, names and locations of all guides, including those who have not yet given a trip

```
SQL> SELECT ALL Trip_guides.eno, Trip_guides.ename, Trip_locations.lname FROM Trip_guides
  2  JOIN Trip_Expenditions ON Trip_guides.eno = Trip_Expenditions.eno
  3  JOIN Trip_locations ON Trip_locations.lcode = Trip_Expenditions.lcode;

      ENO ENAME                LNAME
-----
      978 Siyabonge Nomvete    Table Mountain
      978 Siyabonge Nomvete    Table Mountain
      978 Siyabonge Nomvete    Robben Island
      923 Tyrone Arendse       Robben Island
      982 Mathew Booth         Kruger National Park
      942 Jean-Marc Ithier     St Lucia

6 rows selected.
```

v. Display all employees who've worked with a trip\_group of more than 10 people

```
SQL> SELECT Trip_guides.ename FROM Trip_guides WHERE Trip_guides.ename IN
  2  (
  3  SELECT DISTINCT Trip_guides.ename FROM Trip_guides
  4  JOIN Trip_Expenditions ON Trip_guides.eno = Trip_Expenditions.eno
  5  JOIN Trip_locations ON Trip_locations.lcode = Trip_Expenditions.lcode WHERE Trip_Expenditions.tgsize>10
  6  );

ENAME
-----
Siyabonge Nomvete
Tyrone Arendse
```

vi. Find the names of all the employees who have the same hourly rate as Siyabonge Nomvete. (self-join)

```
SQL> SELECT e1.ename FROM Trip_guides e1,Trip_guides e2 WHERE e1.hrate = e2.hrate AND e2.ename = 'Siyabonge Nomvete';

ENAME
-----
Siyabonge Nomvete
```

vii. Display the employee name and employee number who have worked for more than 3 hours.

```
SQL> SELECT Trip_guides.eno, Trip_guides.ename FROM Trip_guides WHERE Trip_guides.ename IN
2  (
3  SELECT DISTINCT Trip_guides.ename FROM Trip_guides
4  JOIN Trip_Expenditions ON Trip_guides.eno = Trip_Expenditions.eno
5  JOIN Trip_locations ON Trip_locations.lcode = Trip_Expenditions.lcode WHERE Trip_Expenditions.hworked>3
6  );

ENO ENAME
-----
978 Siyabonge Nomvete
942 Jean-Marc Ithier
982 Mathew Booth
```

viii. Display the total information of the trip guides along with their hourly rate in ascending order.

```
SQL> SELECT * FROM Trip_guides ORDER BY (hrate) ASC;

ENO ENAME                HRATE  MONTHSEMP
-----
982 Mathew Booth          25         0
978 Siyabonge Nomvete     30         6
923 Tyrone Arendse        32         6
942 Jean-Marc Ithier       35        12
```

ix. Display the trip location in which trip\_group have visited more than once.

```
SQL> SELECT Trip_Expenditions.lcode, Trip_locations.lname FROM Trip_Expenditions
2  LEFT JOIN Trip_locations ON Trip_Expenditions.lcode = Trip_locations.lcode
3  GROUP BY Trip_Expenditions.lcode, Trip_locations.lname HAVING count(Trip_Expenditions.lcode)>1;

LCODE LNAME
-----
1025 Robben Island
1024 Table Mountain
```