Name: Sarthak Jain

Programme: B.Tech CSE (Spec. Al&ML) (Batch 5)
Course: Advanced Database Management Systems

LAB 7

Title: To understand the concepts of Index.

Objective: Students will be able to implement the concept of index.

O. Create a table of table name: EMPLOYEES and add 6 rows

| Column Name | Data Type | Width | Attributes |
|---------------|-----------|-------|------------|
| Employee_id | character | 10 | PK |
| First_Name | character | 30 | NN |
| Last_Name | character | 30 | NN |
| DOB | date | - | - |
| Salary | number | 25 | NN |
| Department_id | character | 10 | - |

Ans.0

INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('0x23487', 'Ravi', 'Kumar', to_date('05-12-1998', 'dd-mm-yyyy'), 50000, '0x34543');

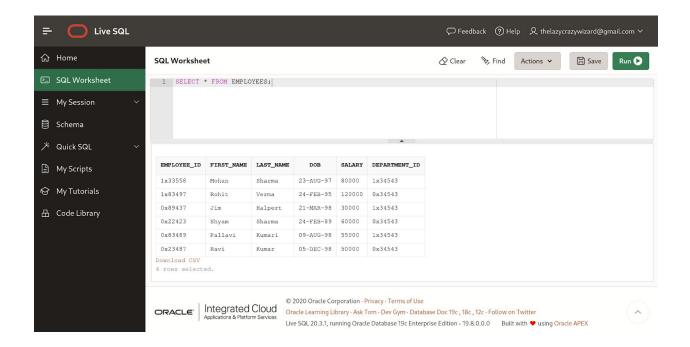
INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('1x33558', 'Mohan', 'Sharma', to_date('23-08-1997', 'dd-mm-yyyy'), 80000, '1x34543');

INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('1x83497', 'Rohit', 'Verma', to_date('24-02-1995', 'dd-mm-yyyy'), 120000, '0x34543');

INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('0x89437', 'Jim', 'Halpert', to_date('21-03-1998', 'dd-mm-yyyy'), 30000, '1x34543');

INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('0x22423', 'Shyam', 'Sharma', to_date('24-02-1989', 'dd-mm-yyyy'), 60000, '0x34543');

INSERT INTO EMPLOYEES (Employee_id, First_Name, Last_Name, DOB, Salary, Department_id) VALUES ('0x83489', 'Pallavi', 'Kumari', to_date('09-08-1998', 'dd-mm-yyyy'), 55000, '1x34543');

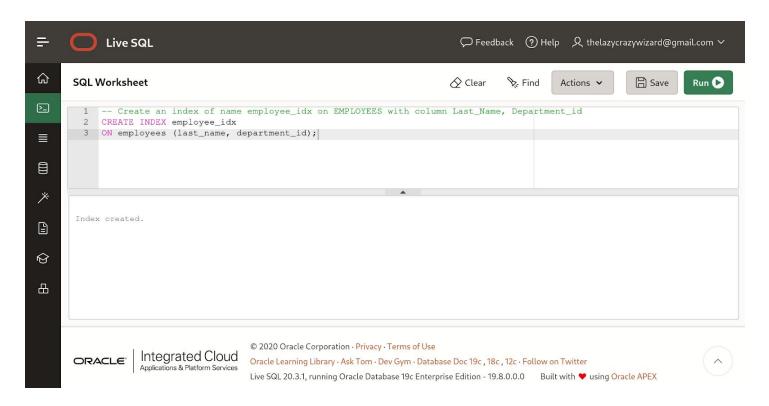


1. Execute the following index related queries:

1.1 Create an index of name employee_idx on EMPLOYEES with column Last_Name, Department_id

Ans.1.1.

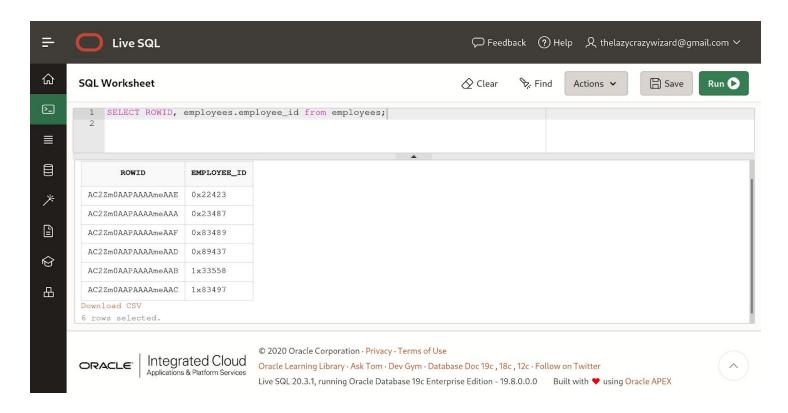
CREATE INDEX employee_idx
ON employees (last_name, department_id);



1.2 Create a unique index on employee_id column of the EMPLOYEES and Find the ROWID for the table.

Ans.1.2.

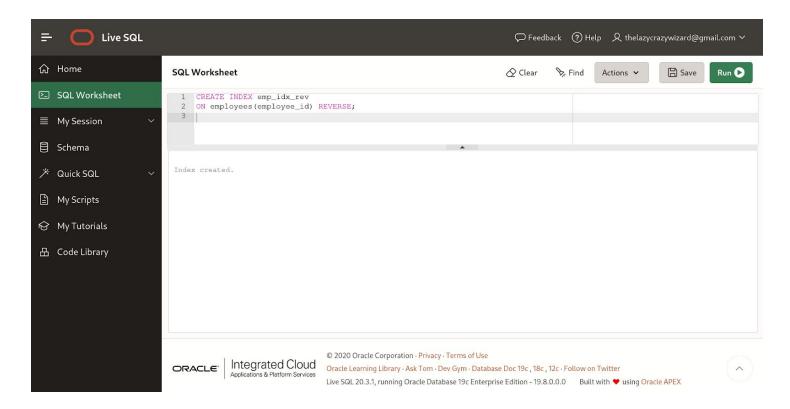
SELECT ROWID, employees.employee_id from employees;



1.3 Create a reverse index on employee_id column of the EMPLOYEES.

Ans.1.3.

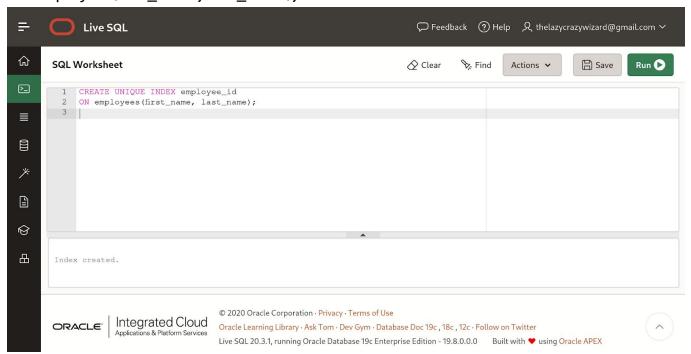
CREATE INDEX emp_idx_rev
ON employees(employee_id) REVERSE;



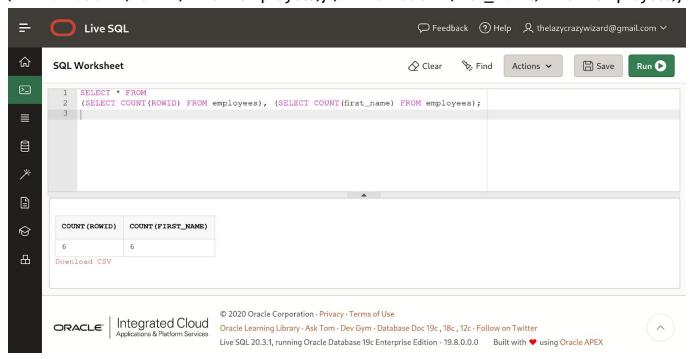
1.4 Create a unique and composite index on employee_id and check whether there is duplicity of tuples or not.

Ans.1.4.

CREATE UNIQUE INDEX employee_id ON employees(first_name, last_name);



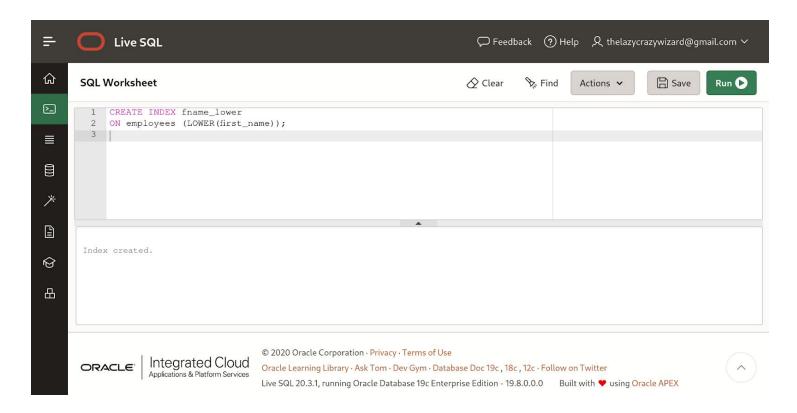
SELECT * FROM (SELECT COUNT(ROWID) FROM employees), (SELECT COUNT(first_name) FROM employees);



1.5 Create Function-based indexes defined on the SQL functions UPPER(column_name) or LOWER(column_name) to facilitate case-insensitive searches(on column Last_Name).

Ans.1.5.

CREATE INDEX fname_lower
ON employees (LOWER(first_name));



1.6 Drop the function based index on column Last_Name.

Ans.1.6.

DROP INDEX fname_lower;

