**Experiment-6:** To Implement the concept of Joins.

**Student Name: -Neha Sharma UID: -20BCS4576**

**Branch: - (CSE\_IOT)                                     Section/Group: - 20BIT (A)**

**Semester: 3rd Date of Performance: 10/7/2021**

**Subject Name: - DATABASE MANAGEMENT SYSTEM LAB**

**Subject Code: 21O-20CSP-233\_20BIT-1\_B**

**1. Aim/Overview of the practical:**

To implement the concept of Joins.

**2. Task to be done:**

Implementation of Join commands of SQL with proper Input queries syntax and the output.

**3. Theme/Interests definition (For creative domains):**

**1. Viewing Data In The Tables-:**

**Command: -** Selecting all rows and all columns.

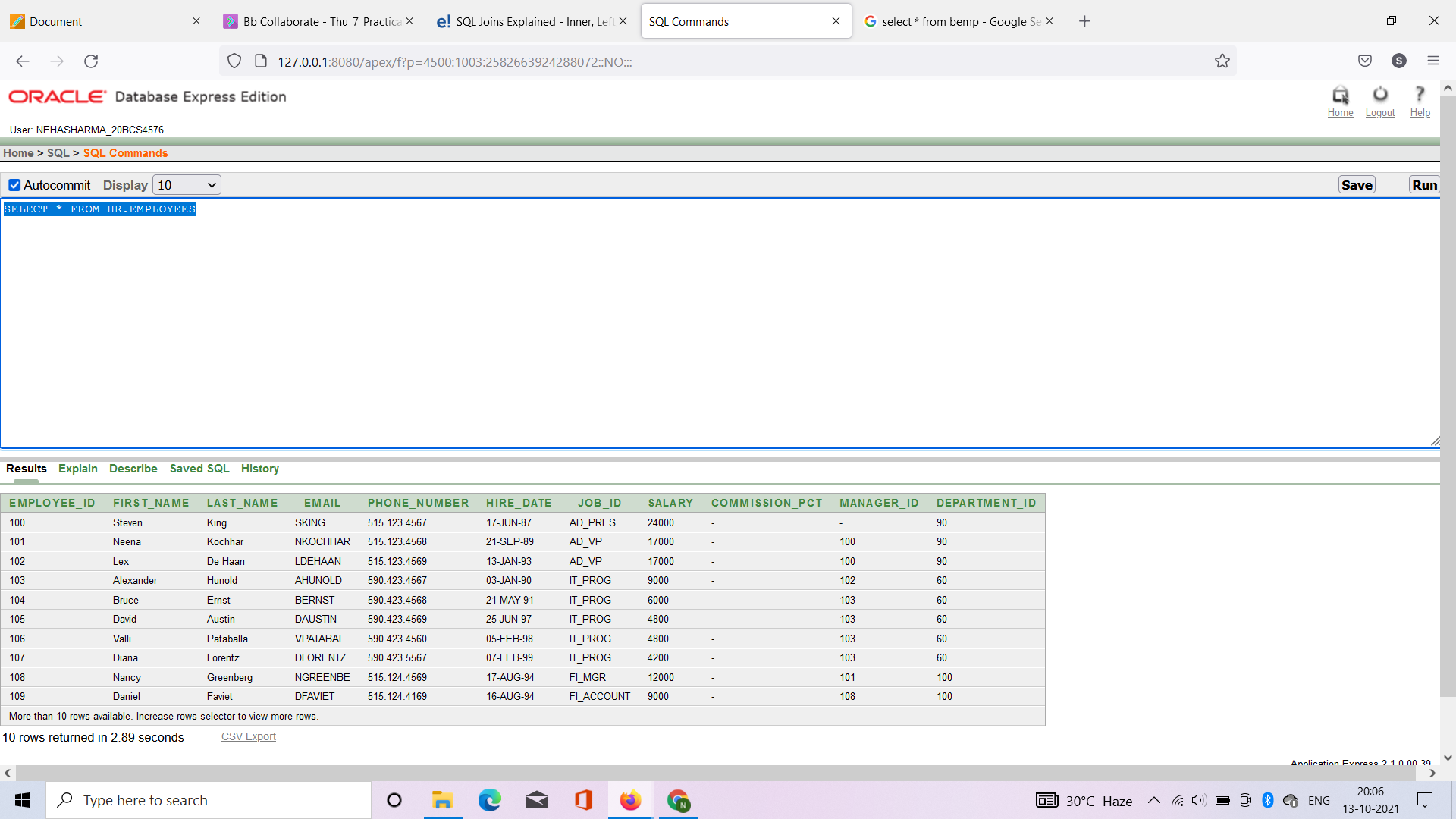
**Purpose: -** All rows and all columns form already exiting table.

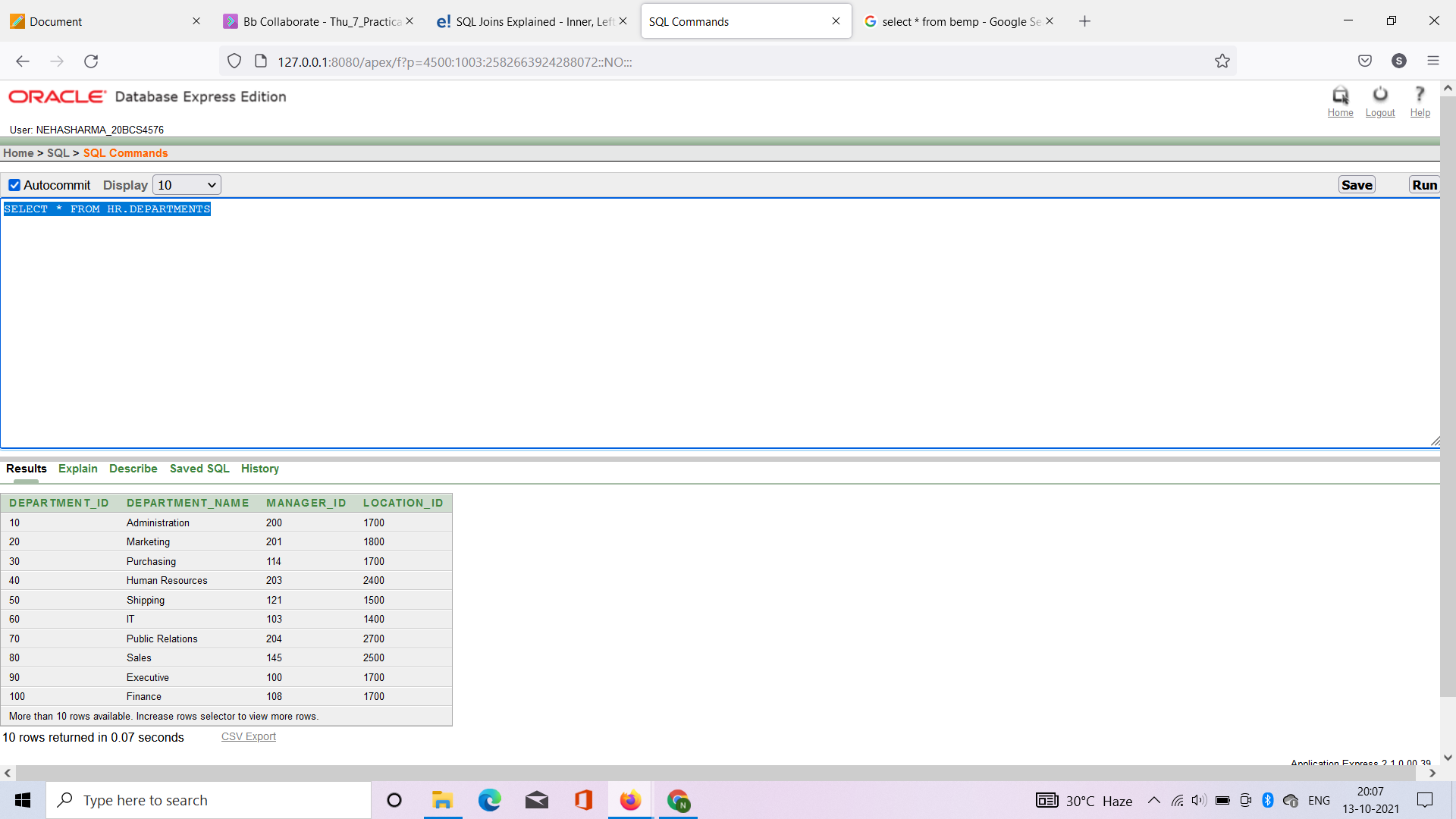
**Syntax: -** *Select \* from <table name>*

**Example: -** SELECT \* FROM HR.EMPLOYEES;

SELECT \* FROM HR.DEPARTMENTS

**Output: -**

****

****

**2. Creating copy existing tables of employees and departments as bemp and bdept respectively**.

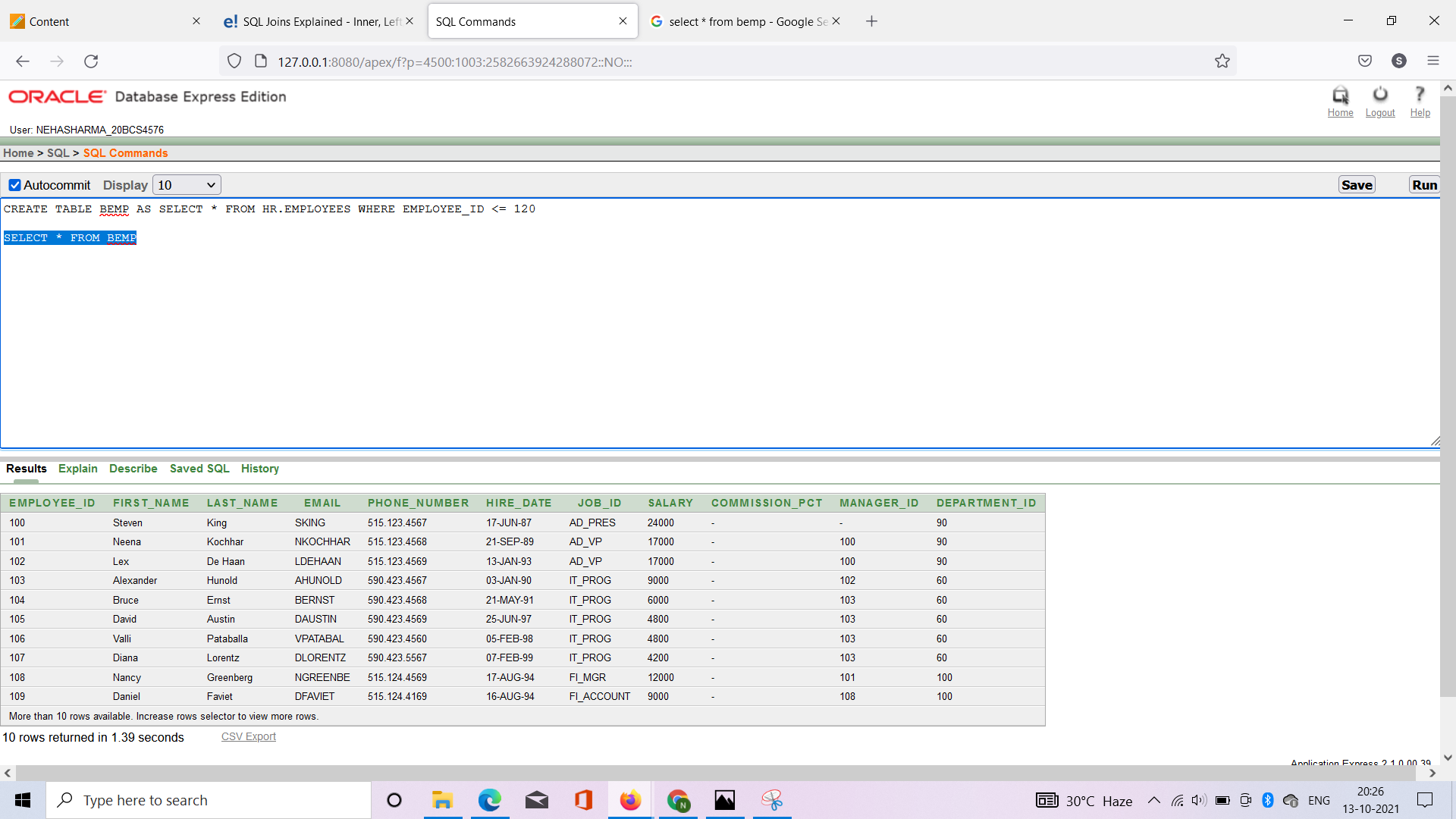
**Command: -** copy  **Purpose: -** Creating copy existing tables

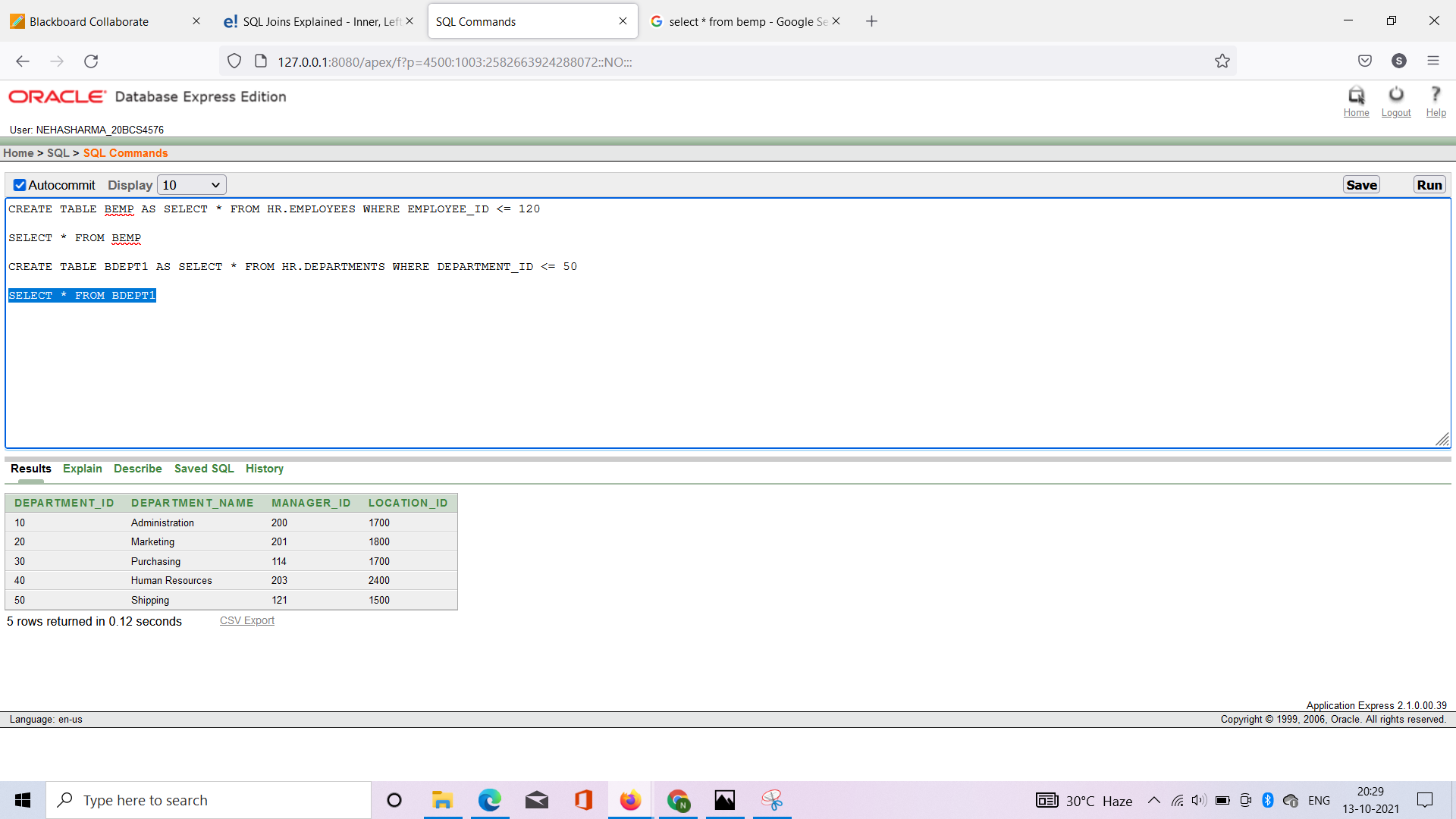
**Syntax: *-*** *create table <new table name> as select from < old table name> where <col 1 .col 2>*

**Example: -** 1)CREATE TABLE BEMP AS SELECT \* FROM HR.EMPLOYEES WHERE EMPLOYEE\_ID <= 120 SELECT \* FROM BEMP

2) CREATE TABLE DDEPT1 AS SELECT \* FROM HR.DEPARTMENTS WHERE DEPARTMENT\_ID <= 150 SELECT \* FROM BDEPT1

**Output: -1)**

****

****

**3. Cross join**

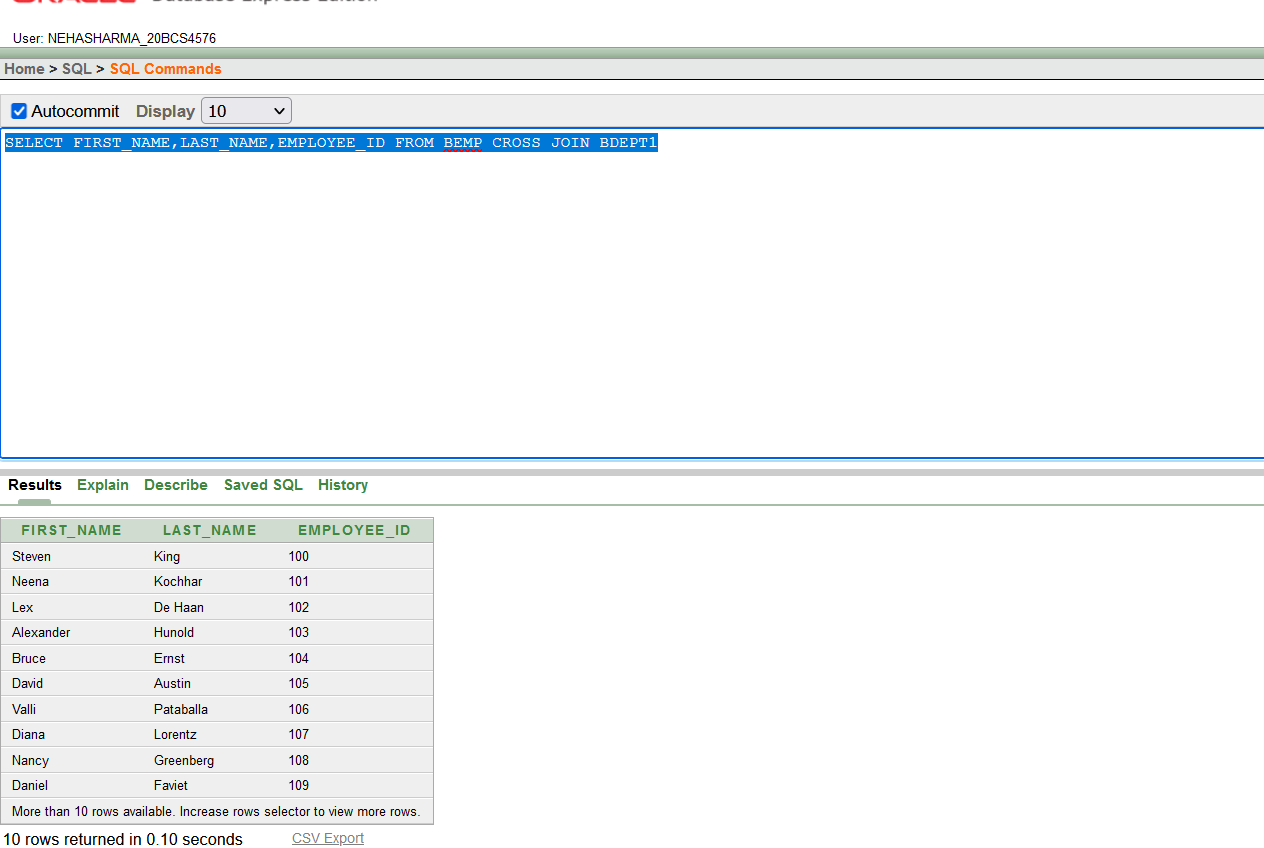
**Command: -** cross join **Purpose: -** The cross join is used to generate a paired combination of each row of the first table with each row of the second table.

**Syntax: -** *SELECT ColumnName\_1, ColumnName\_2, ColumnName\_n FROM <Table\_1>*

*CROSS JOIN <Table\_2>*

**Example: -** SELECT FIRST\_NAME,LAST\_NAME,EMPLOYEE\_ID,MANAGER\_ID FROM BEMP CROSS JOIN BDEPT1

**Output: -**

****

1. **Inner join-**

The INNER JOIN keyword selects records that have matching values in both tables.

1. **Inner Join: With ON keyword**

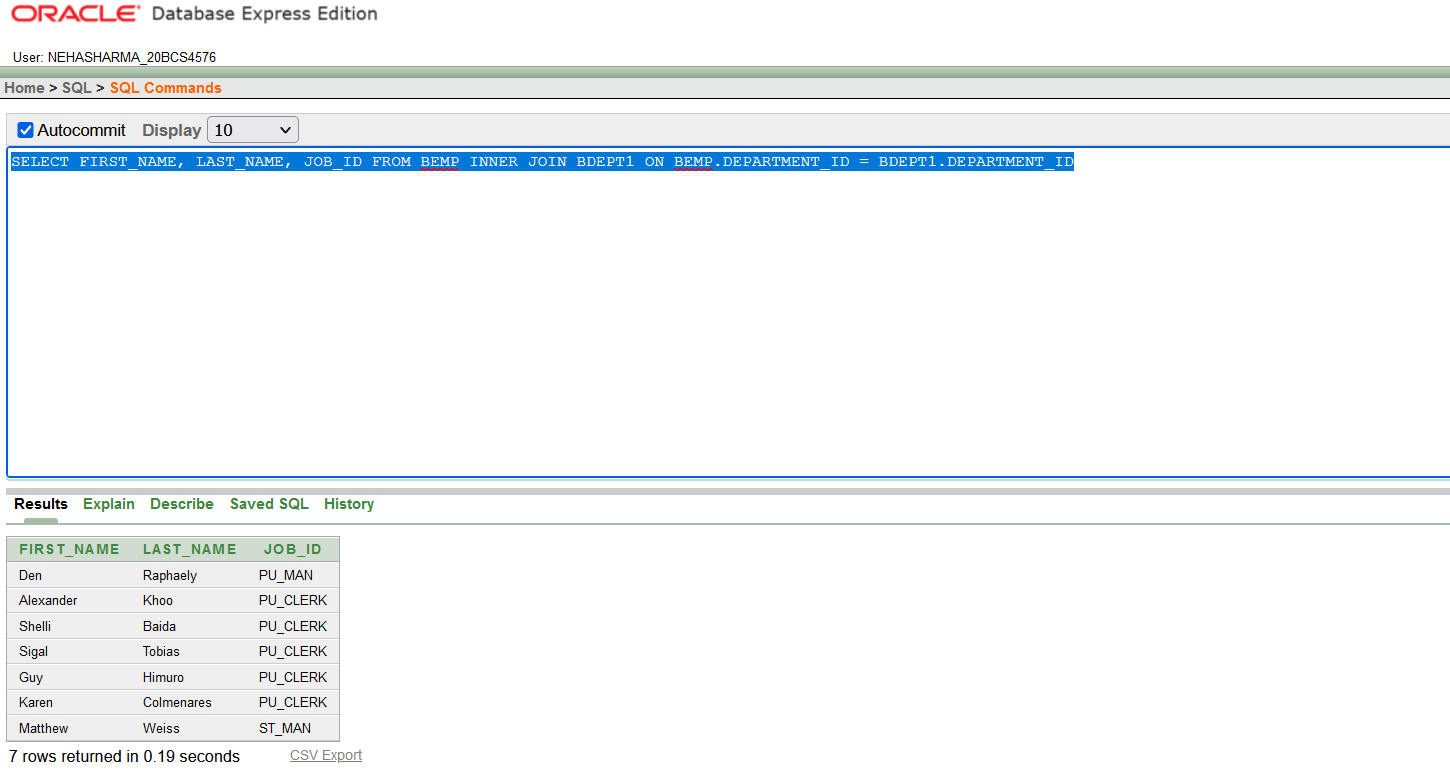
**Command: -** INNER JOIN \_ON

**Purpose: -** The INNER JOIN keyword selects records that have matching values in both table

**Syntax: -** *SELECT COLUMN\_NAME(S) FROM TABLE1*  *INNER JOIN TABLE2* *ON TABLE1.COLUMN\_NAME = TABLE2.COLUMN\_NAME;*

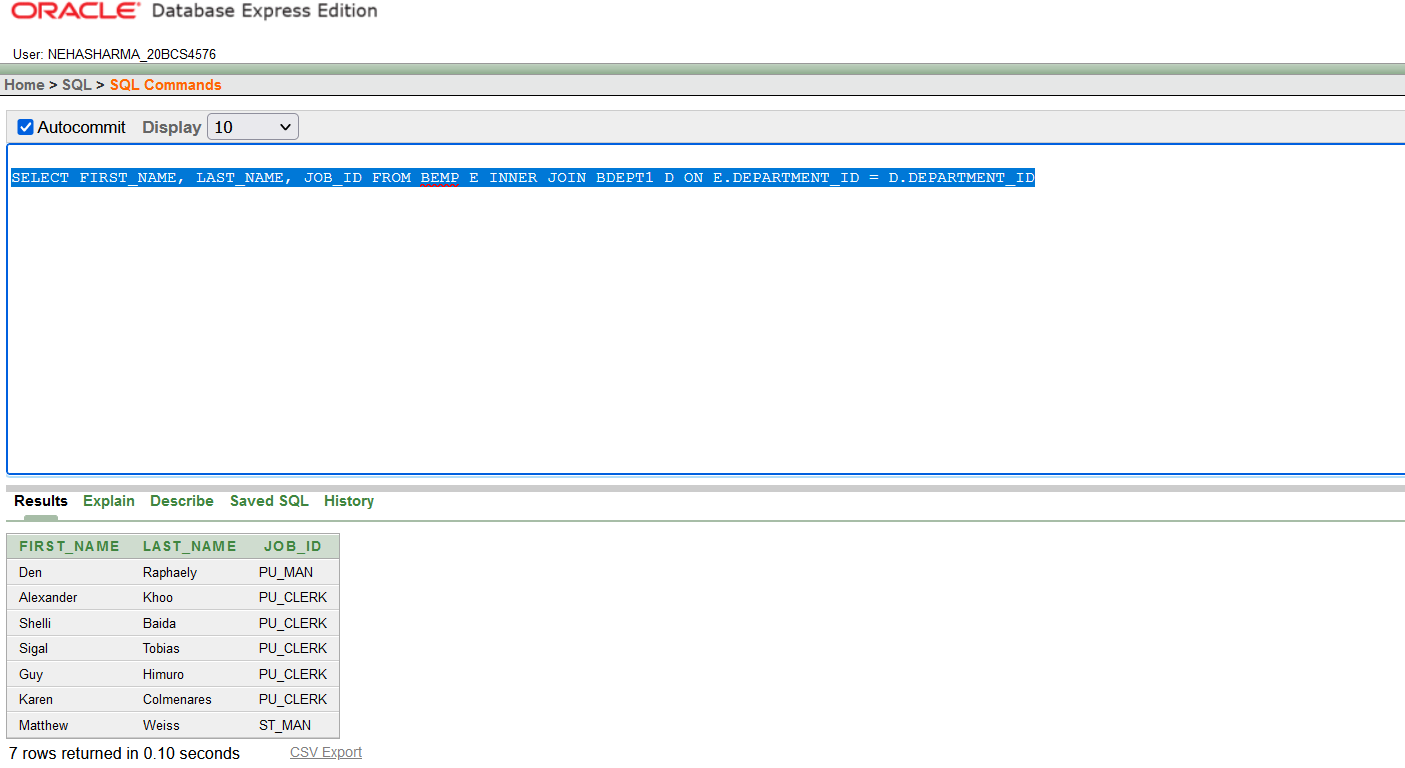
**Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP INNER JOIN BDEPT1 ON BEMP.DEPARTMENT\_ID = BDEPT1.DEPARTMENT\_ID

**OUTPUT**



* **Using Table Aliases :**

*SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP E INNER JOIN BDEPT1 D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID*



1. **Inner Join: With USING keyword**

**Command: -** INNER JOIN \_USING

**Purpose: -** The INNER JOIN keyword selects records that have matching values in both table

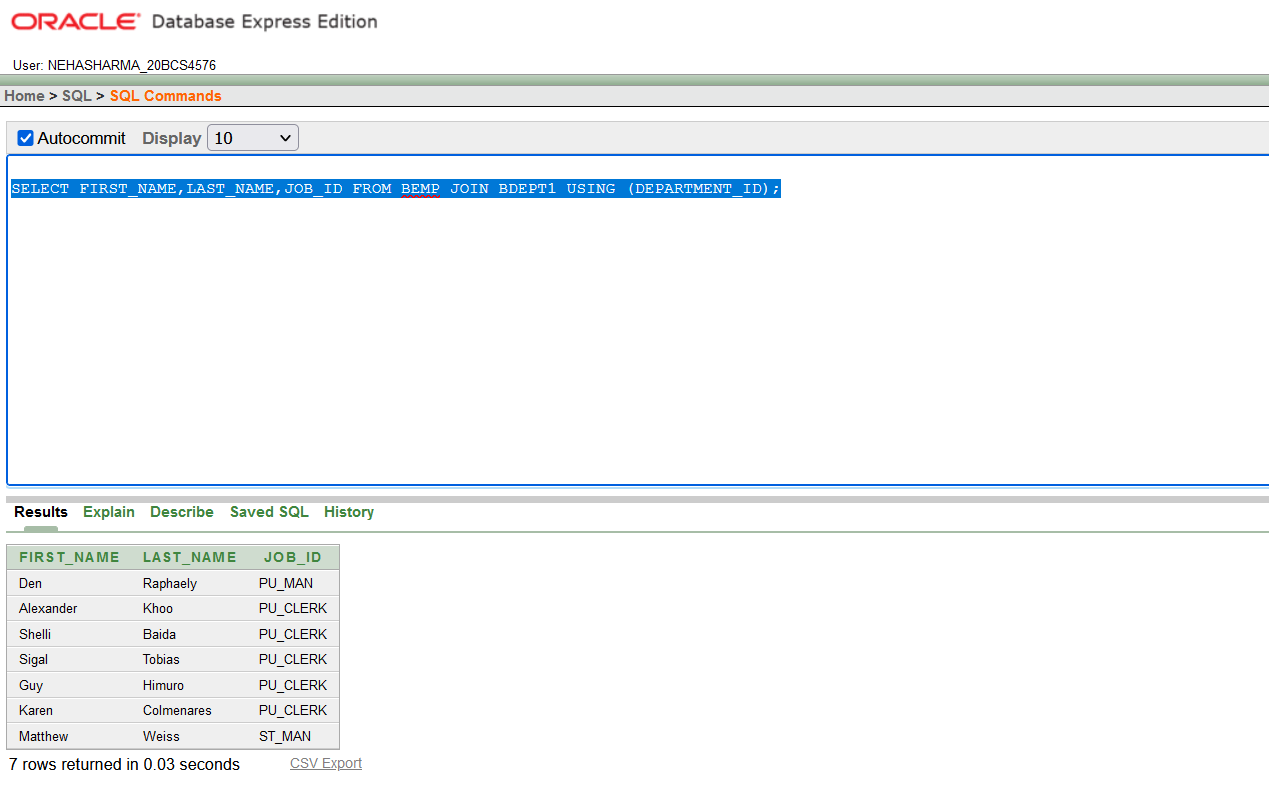
**Syntax: -**

*SELECT COLUMN\_NAME(S) FROM TABLE1*  *INNER JOIN TABLE2* *USING TABLE1.JOIN TABLE2(COLUMN \_NAME);*

**Example: -**

SELECT FIRST\_NAME,LAST\_NAME,JOB\_ID FROM BEMP JOIN BDEPT1 USING (DEPARTMENT\_ID);

**OUTPUT**



1. **Inner Join: With**  **where clause**

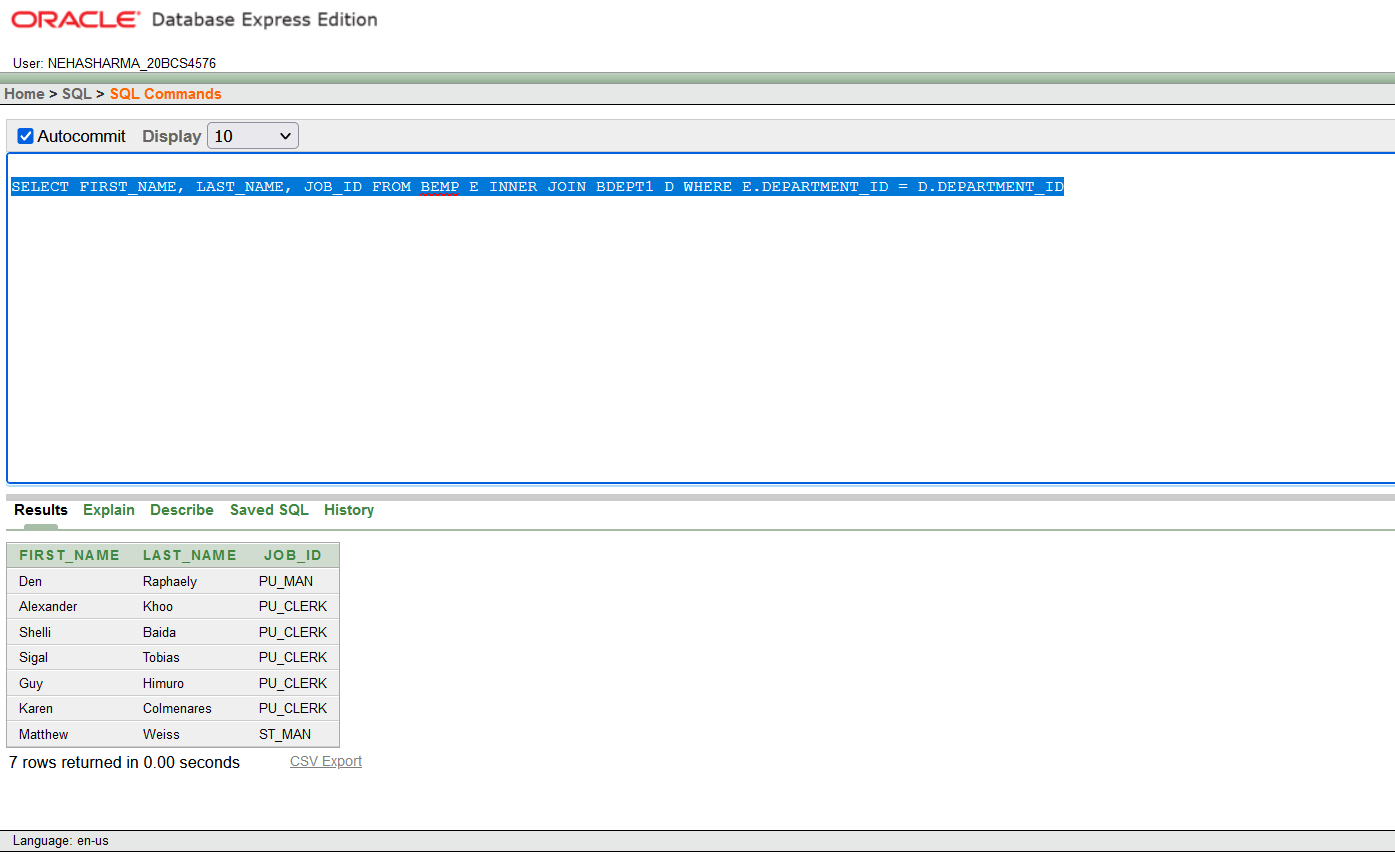
**Command: -** INNER JOIN \_where clause (condition)

**Purpose: -** The INNER JOIN keyword selects records that have matching values in both table

**Syntax: -** *SELECT COLUMN\_NAME(S) FROM TABLE1*  *INNER JOIN TABLE2* *WHERE TABLE1.COLUMN\_NAME = TABLE2.COLUMN\_NAME*

**Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP E INNER JOIN BDEPT1 D WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID

**OUTPUT**



1. **Inner join :Non equi join**

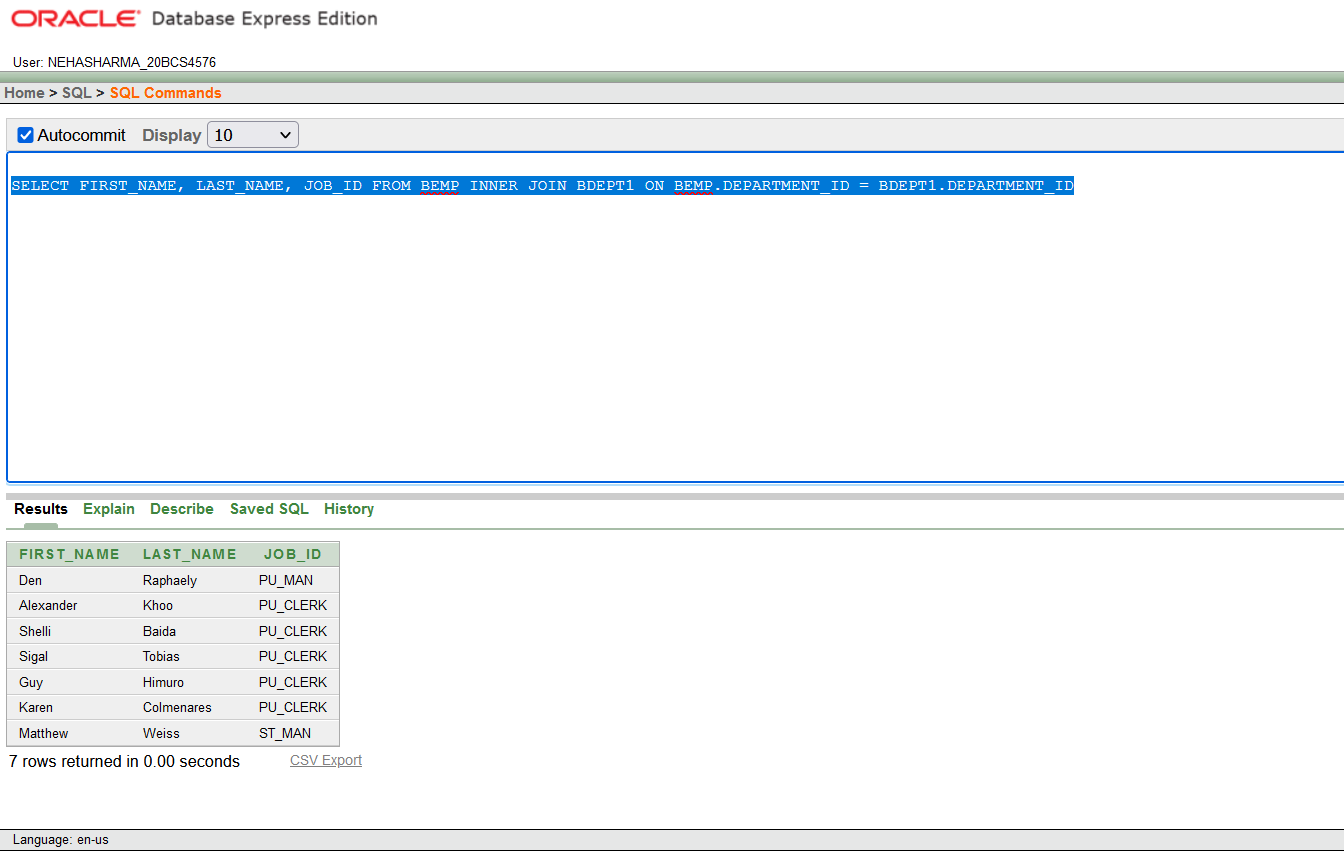
**Command: -** INNER JOIN \_On

**Purpose: -** The INNER JOIN keyword selects records that have matching values in both table

**Syntax: -** *SELECT COLUMN\_NAME(S) FROM TABLE1*  *INNER JOIN TABLE2* *ON TABLE1.COLUMN\_NAME >/< TABLE2.COLUMN\_NAME*

**Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP INNER JOIN DDEPT ON BEMP.DEPARTMENT\_ID = BDEPT1.DEPARTMENT\_ID

**Output :**-



**2. Natural Join**

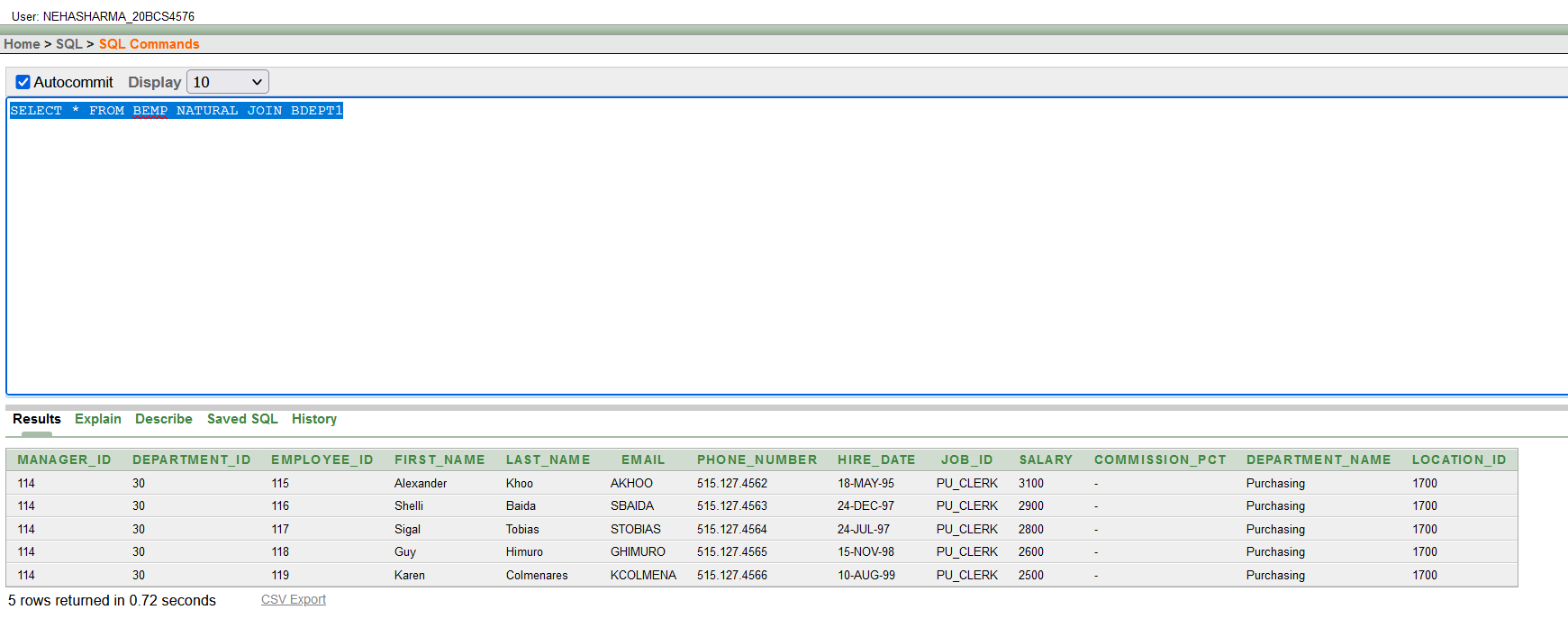
**Command: -** natural join

**Purpose: -**  It selects rows from the two tables that have equal values in all matched columns. If the columns having the same names have different data types, an error is returned.

**Syntax: -** *SELECT \* from bemp natural join bdept1*

**Example: -** SELECT \* FROM BEMP NATURAL JOIN BDEPT1

**OUTPUT –**



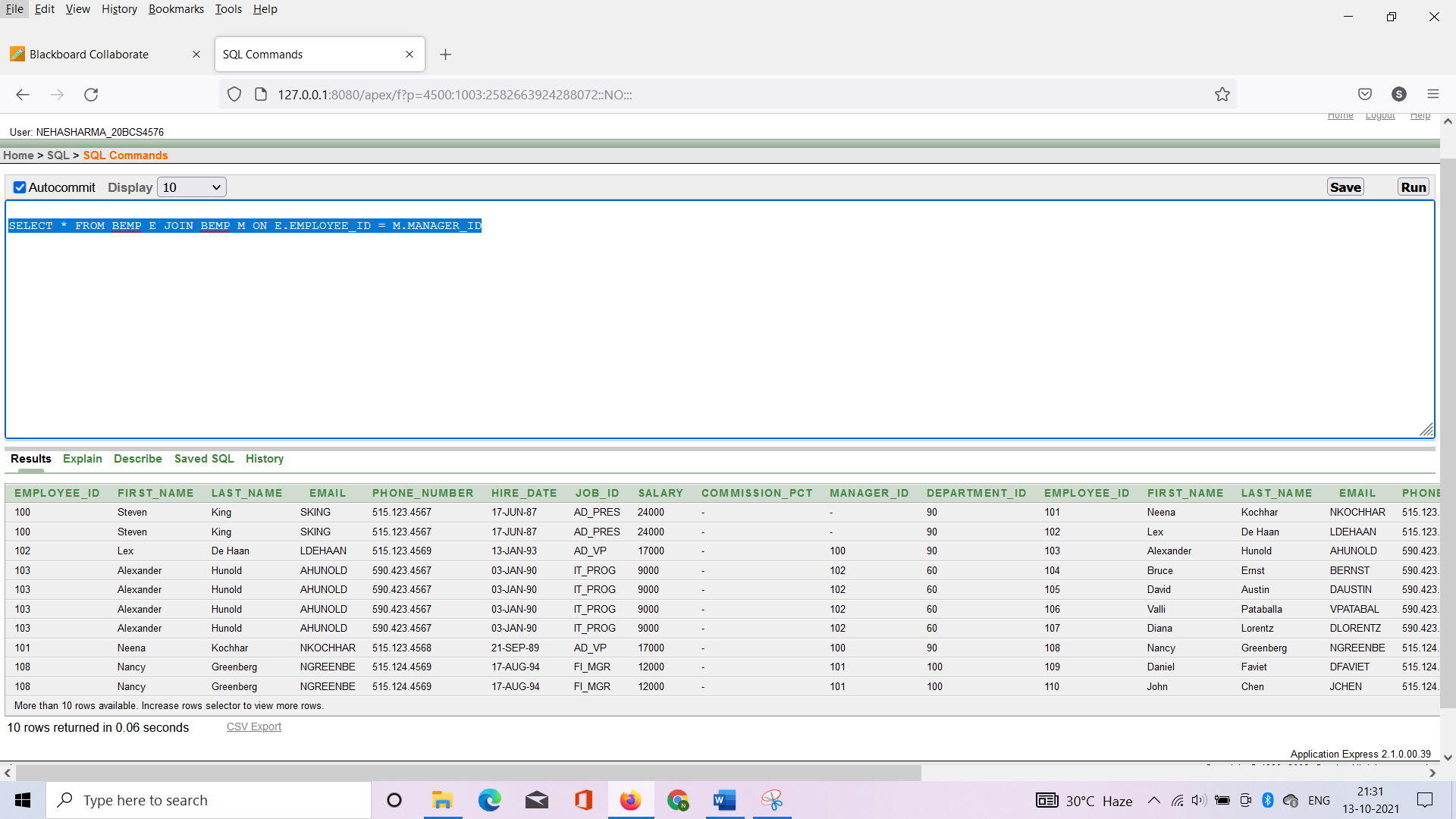
**3. Self-Join-**

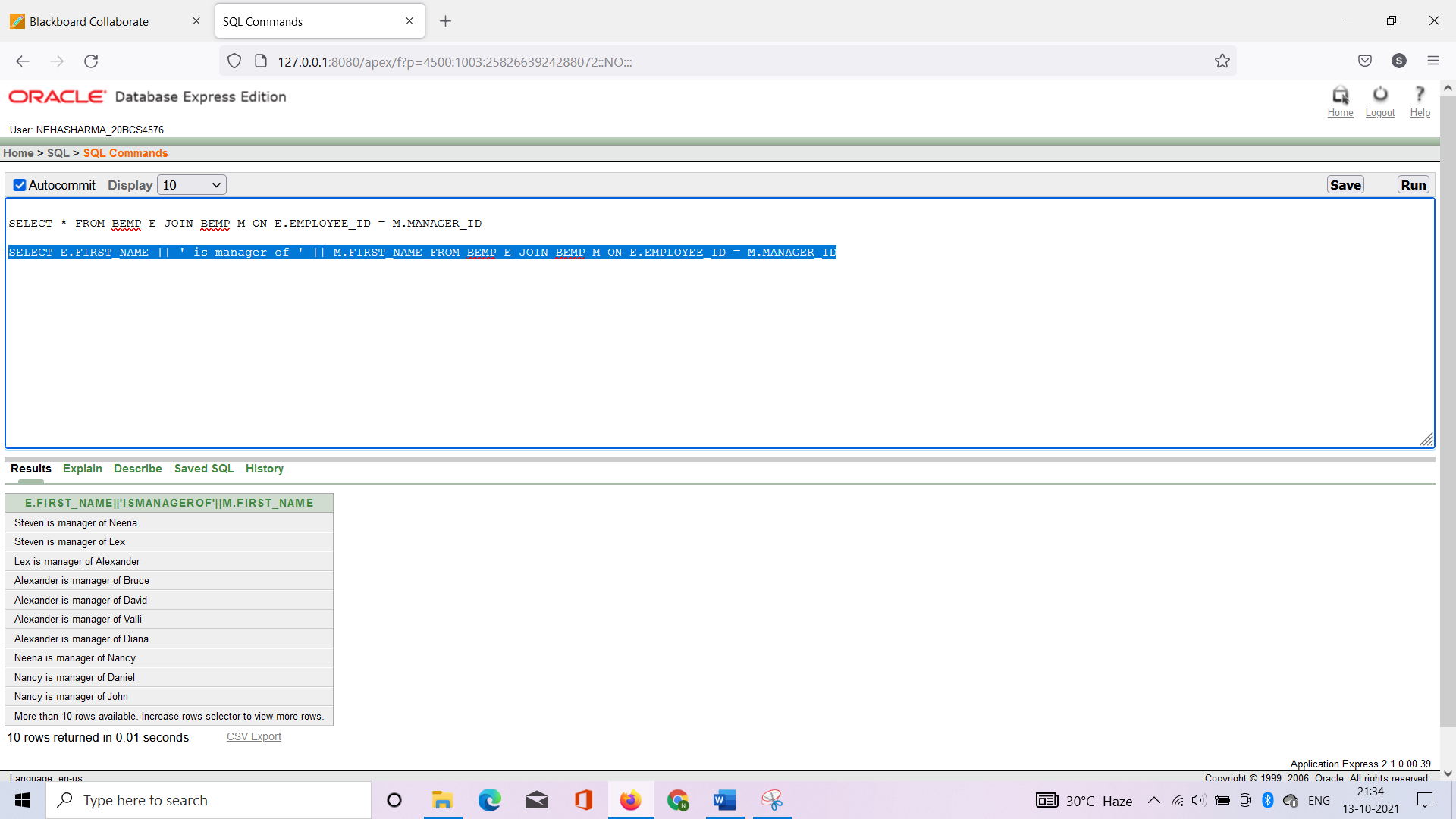
**Purpose:** - A self-join is a regular join, but the table is joined with itself – this is extremely useful for comparisons within a table. Joining a table with itself means that each row of the table is combined with itself and with every other row of the table.

**Syntax: -** *SELECT column name from table1 join table 1 on (column\_name)*

**Example: -**

**Output –**

****



1. **Outer join-**

* The Outer join returns all rows from both the participating tables which satisfy the join condition along with rows which do not satisfy the join condition.

1. **Outer join : left join**

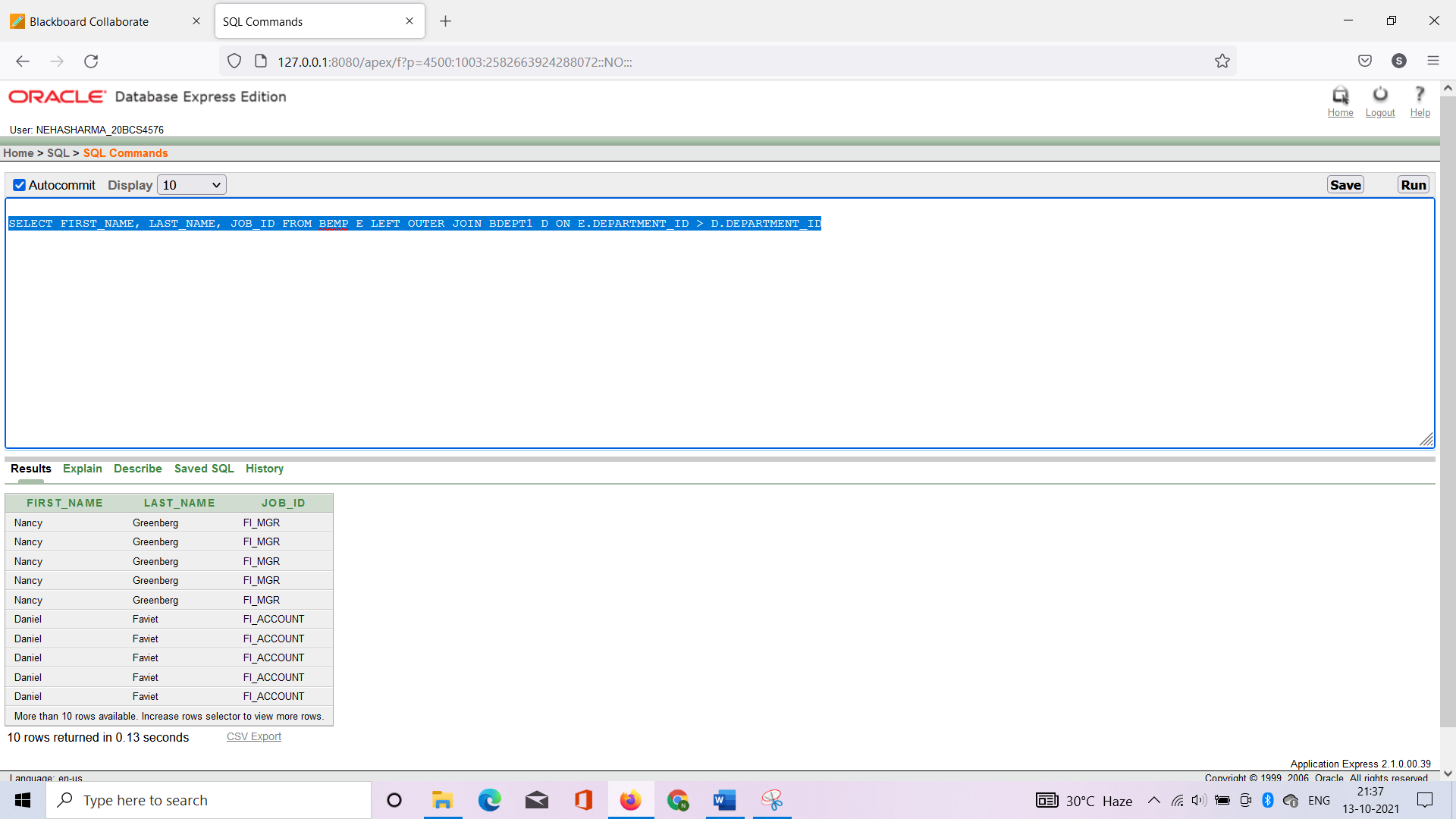
**Command: -** left outer join

**Purpose: -** The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

**Syntax: -** *select column \_name from table1 left outer join table 2 on  (column 1 > column );*

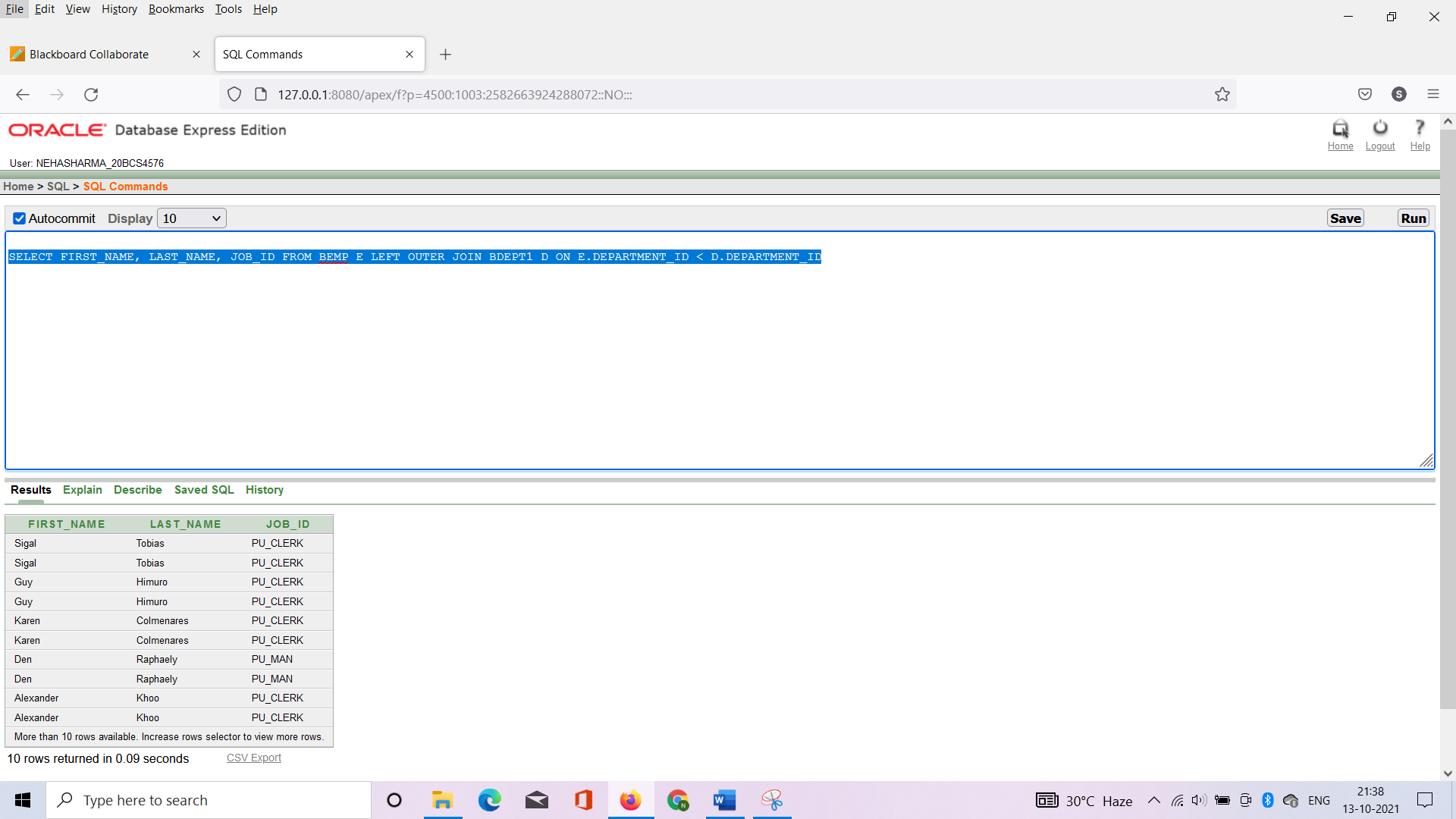
**Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP LEFT OUTER JOIN BDEPT1 ON E.DEPARTMENT\_ID > D.DEPARTMENT\_ID

**OUTPUT**



**2.Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP LEFT OUTER JOIN BDEPT1 ON BEMP.DEPARTMENT\_ID < DDEPT.DEPARTMENT\_ID

**OUTPUT**

****

1. **Outer join : right join**

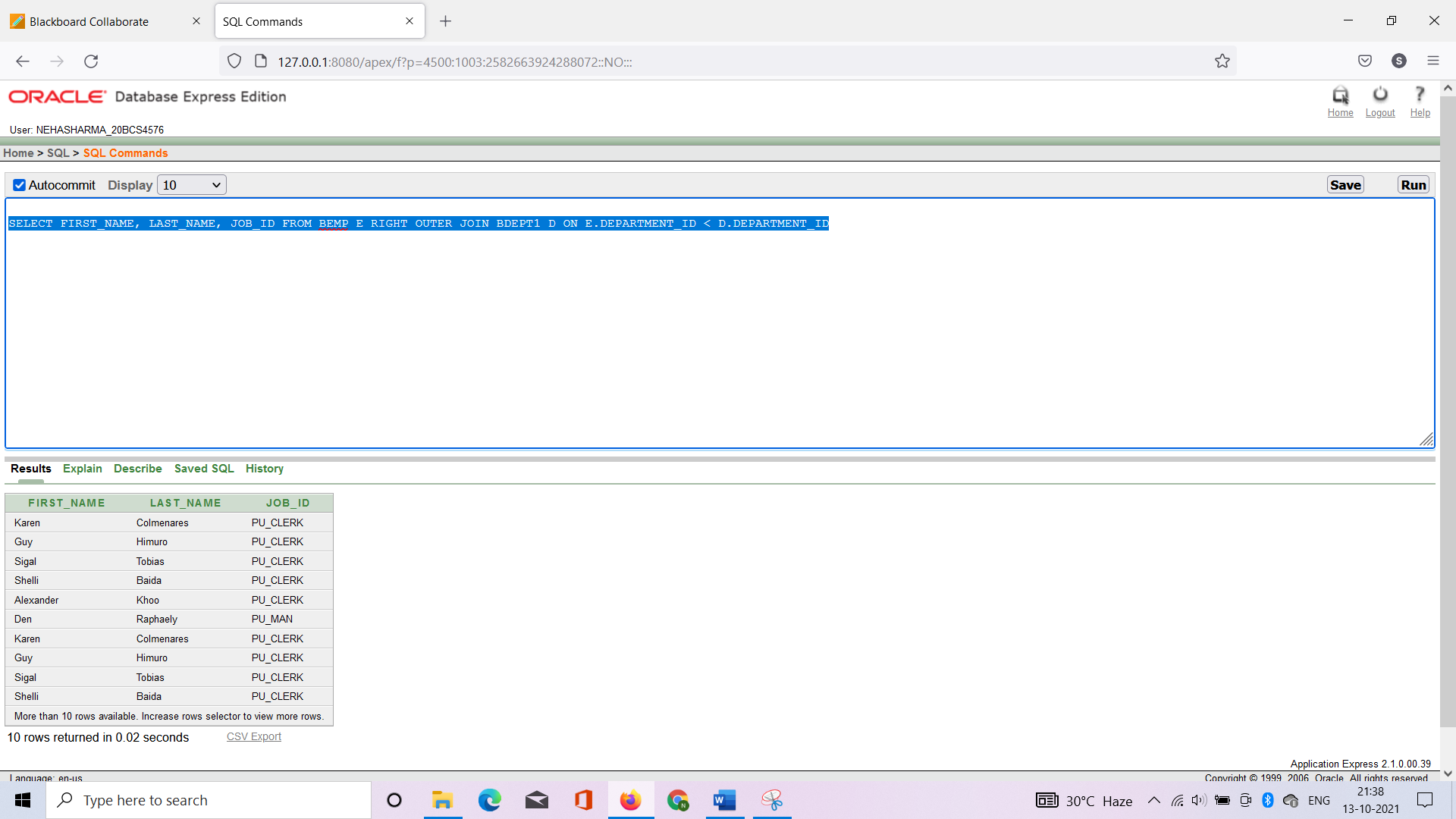
**Command: -** right outer join

**Purpose: -** the right join keyword returns all records from the right table (table2), and the matched records from the left table (table1). the result is null from the left side, when there is no match.

**Syntax: -** *select column \_name from table1 right outer join table 2 on  (column 1 < column );*

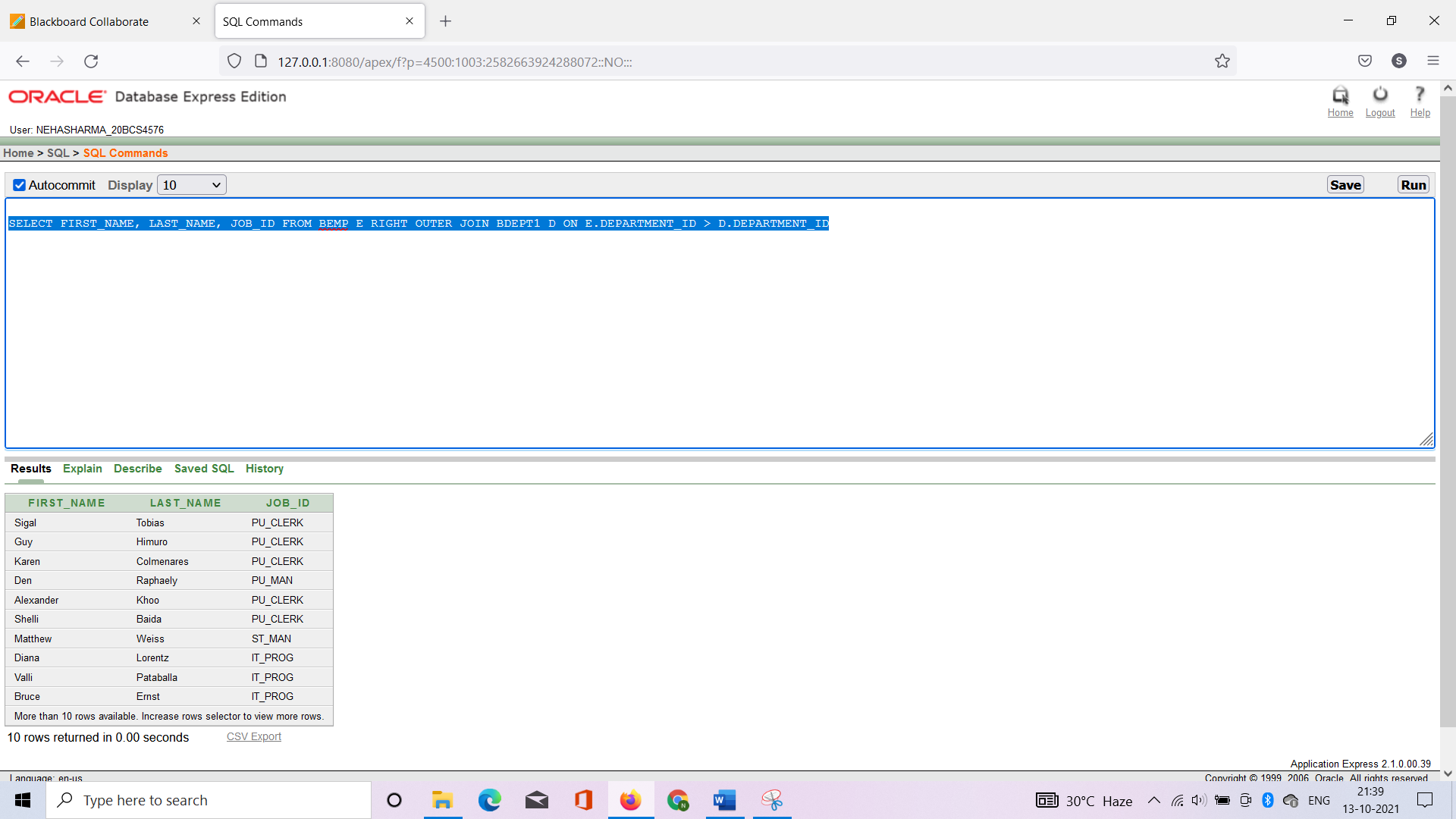
**Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP RIGHT OUTER JOIN BDEPT1 ON E.DEPARTMENT\_ID< D.DEPARTMENT\_ID

**OUTPUT:**

****

**2.Example: -** SELECT FIRST\_NAME, LAST\_NAME, JOB\_ID FROM BEMP RIGHT OUTER JOIN BDEPT1 ON E.DEPARTMENT\_ID>D.DEPARTMENT\_ID

**OUTPUT:**

****

1. **Outer join : right join**

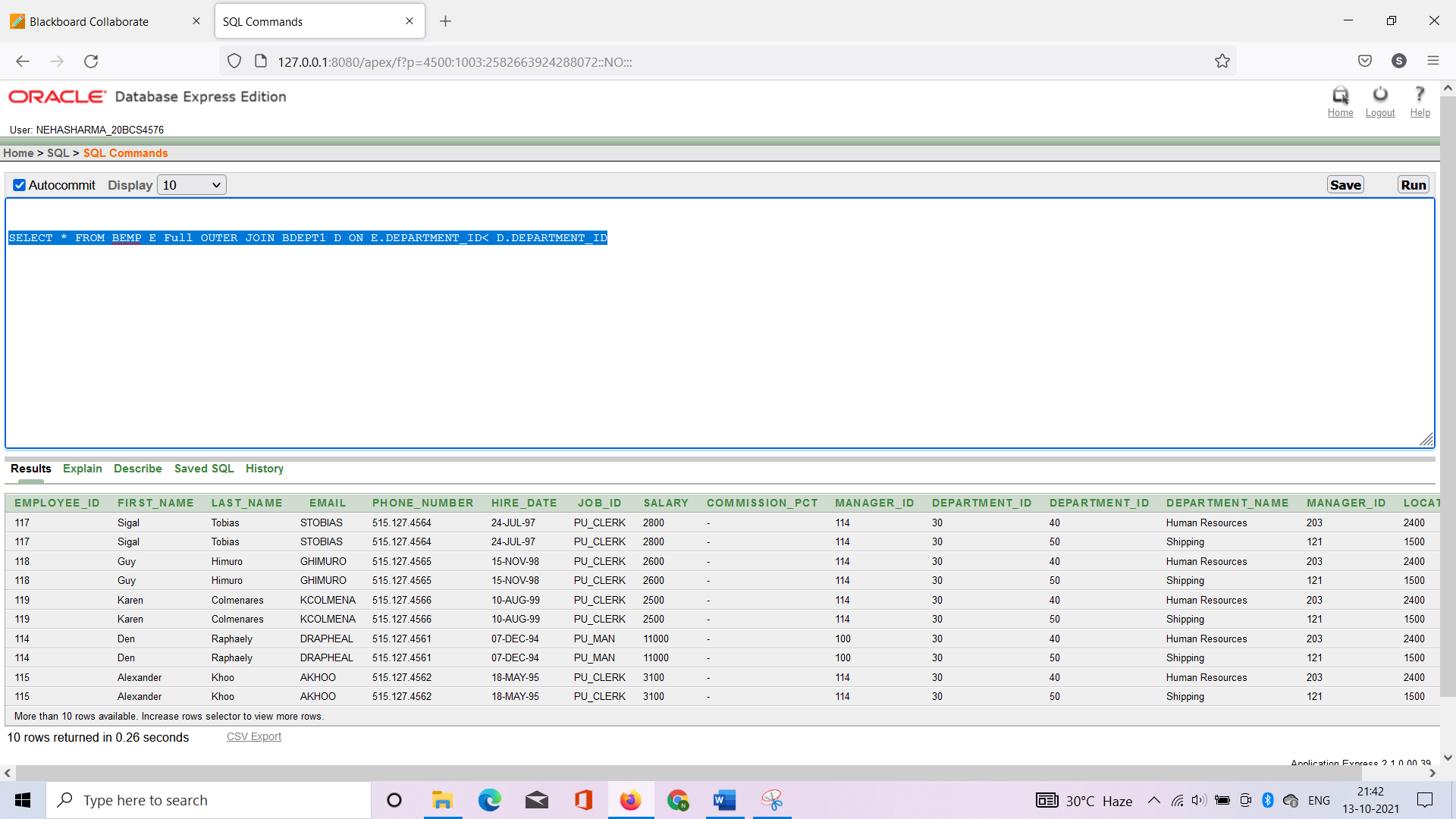
**Command: -** full outer join

**Purpose: -** the full outer join keyword returns all records when there is a match in left (table1) or right (table2) table records.

**Syntax: -** *select column \_name from table1 full outer join table 2 on  (column 1 < column );*

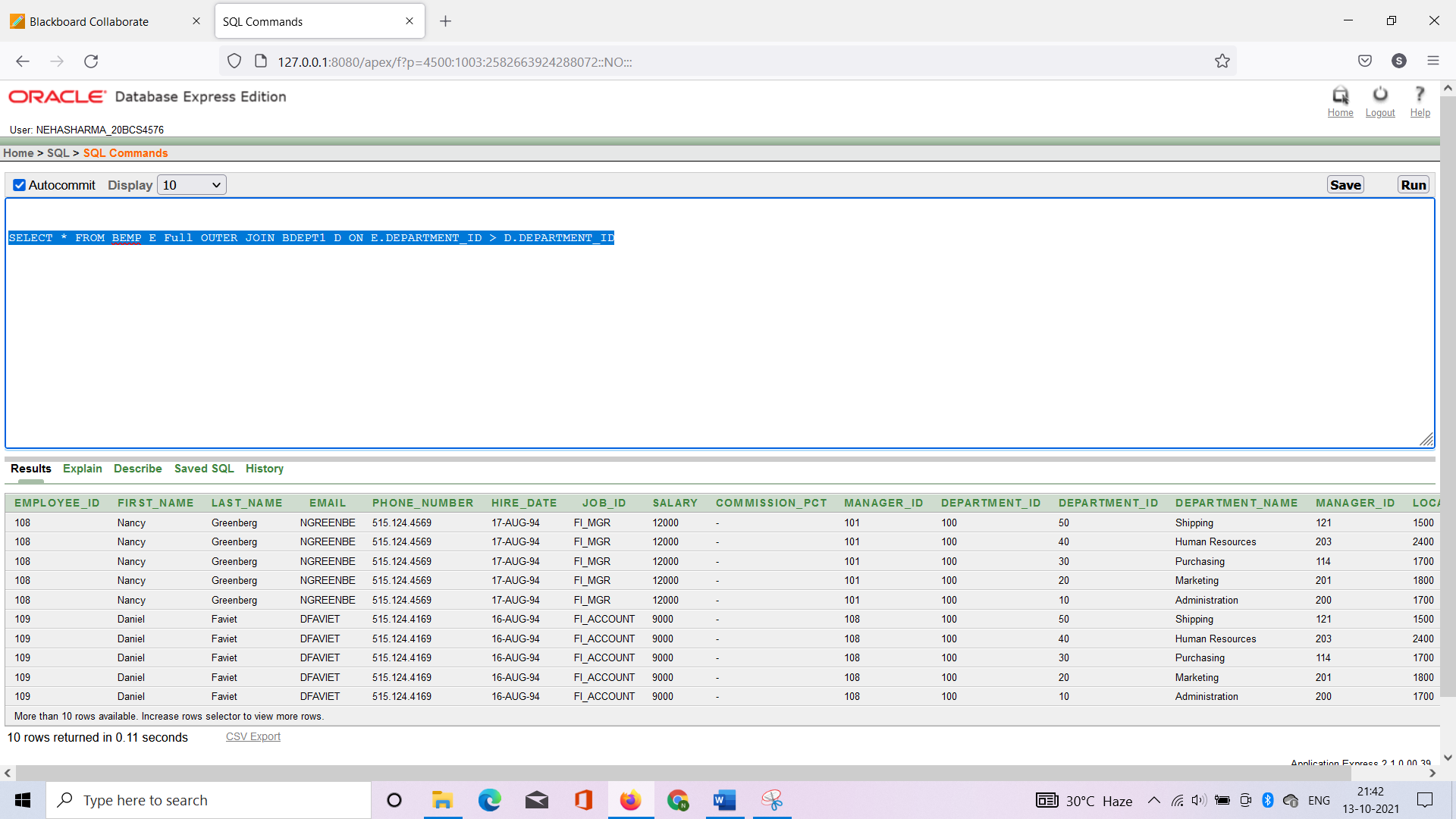
**Example: -** SELECT \* FROM BEMP Full OUTER JOIN BDEPT1 ON E.DEPARTMENT\_ID< D.DEPARTMENT\_ID

**Output -**



**2. Example: -** SELECT \* FROM BEMP full OUTER JOIN BDEPT1 ON E.DEPARTMENT\_ID> D.DEPARTMENT\_ID

**Output -**

****

**5. Result/Output/Writing Summary:**

We observed the different syntax of implementing the concept of Joins.

**Learning outcomes (What I have learnt):**

1. After completing this experiment we get to know SQL commands and its implementation on objects like JOINS and their different types like inner joins and outer join with sub parts of them, and string manipulation using Learned about alter table commands.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |