**RDBMS PROJECT**

**TOPIC: PIZZA DILIVERY MANAGEMENT SYSTEM**

**BY:**

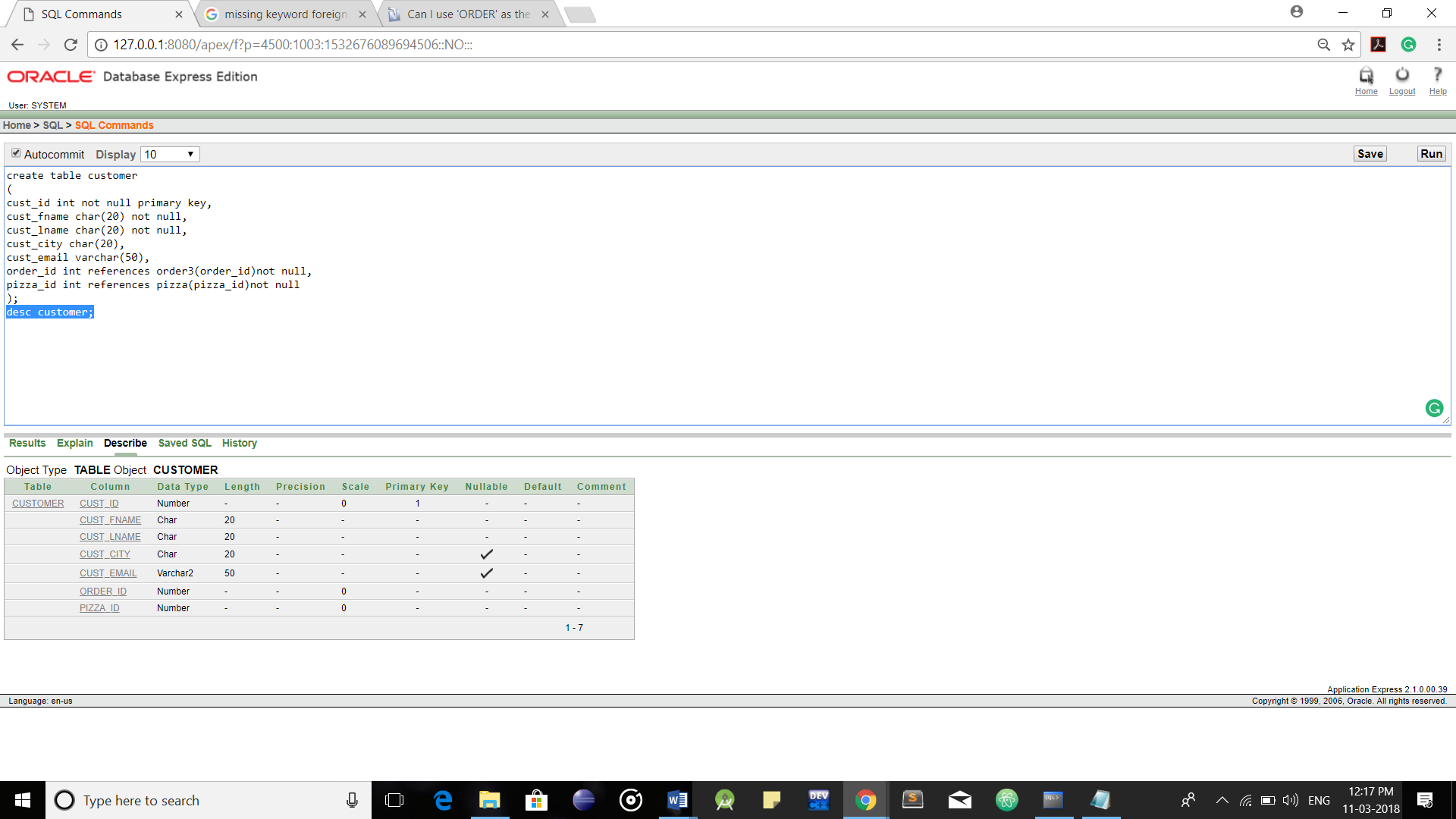
**1.JAYESH SHELAR (1606109)**

**2.RAKESH SHINDE (1606112)**

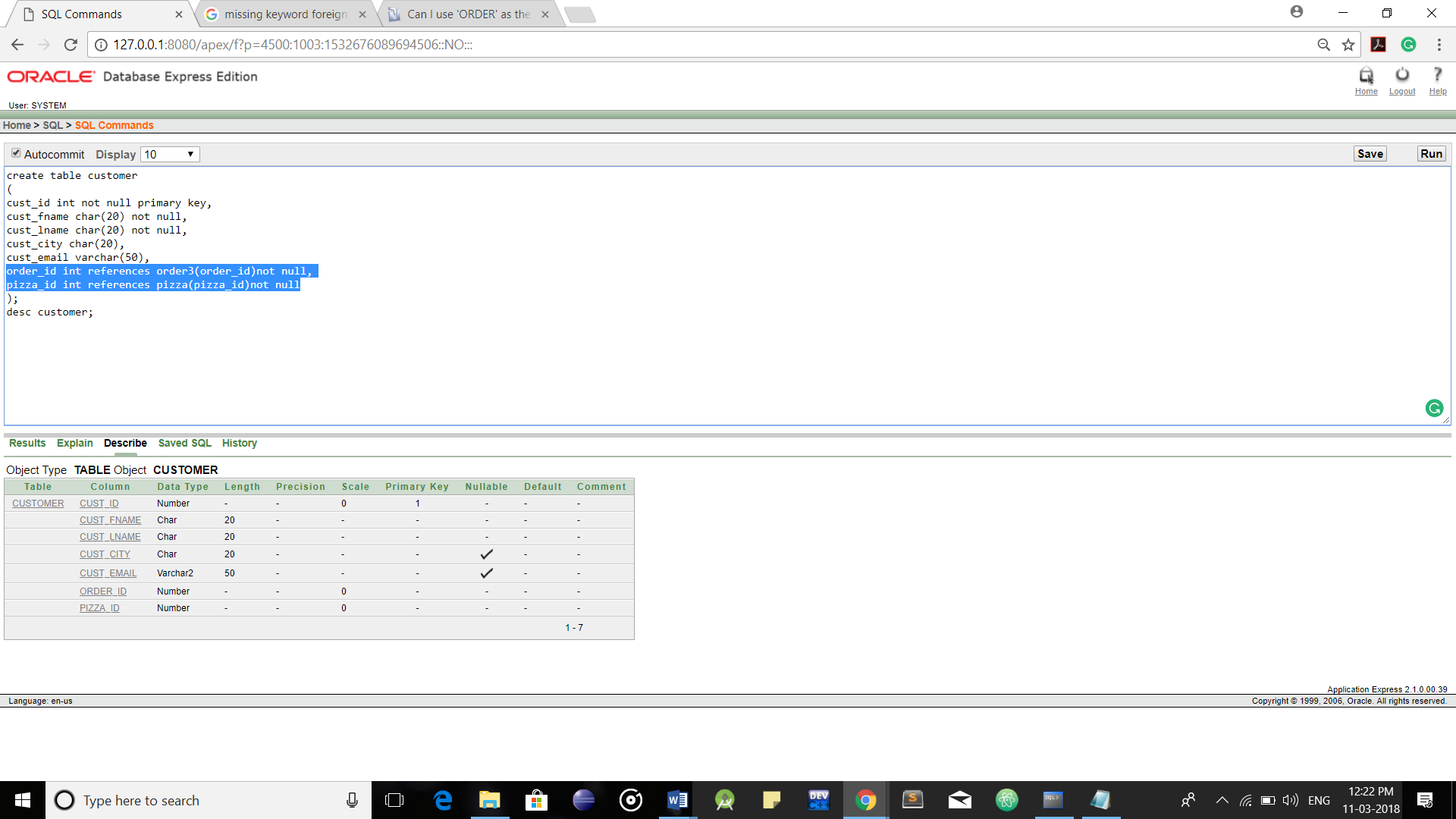
**3.SHREYAS MANTE (1606116)**

**4.DHIRAJ SURVE (1606118)**

**Creating a Table:**



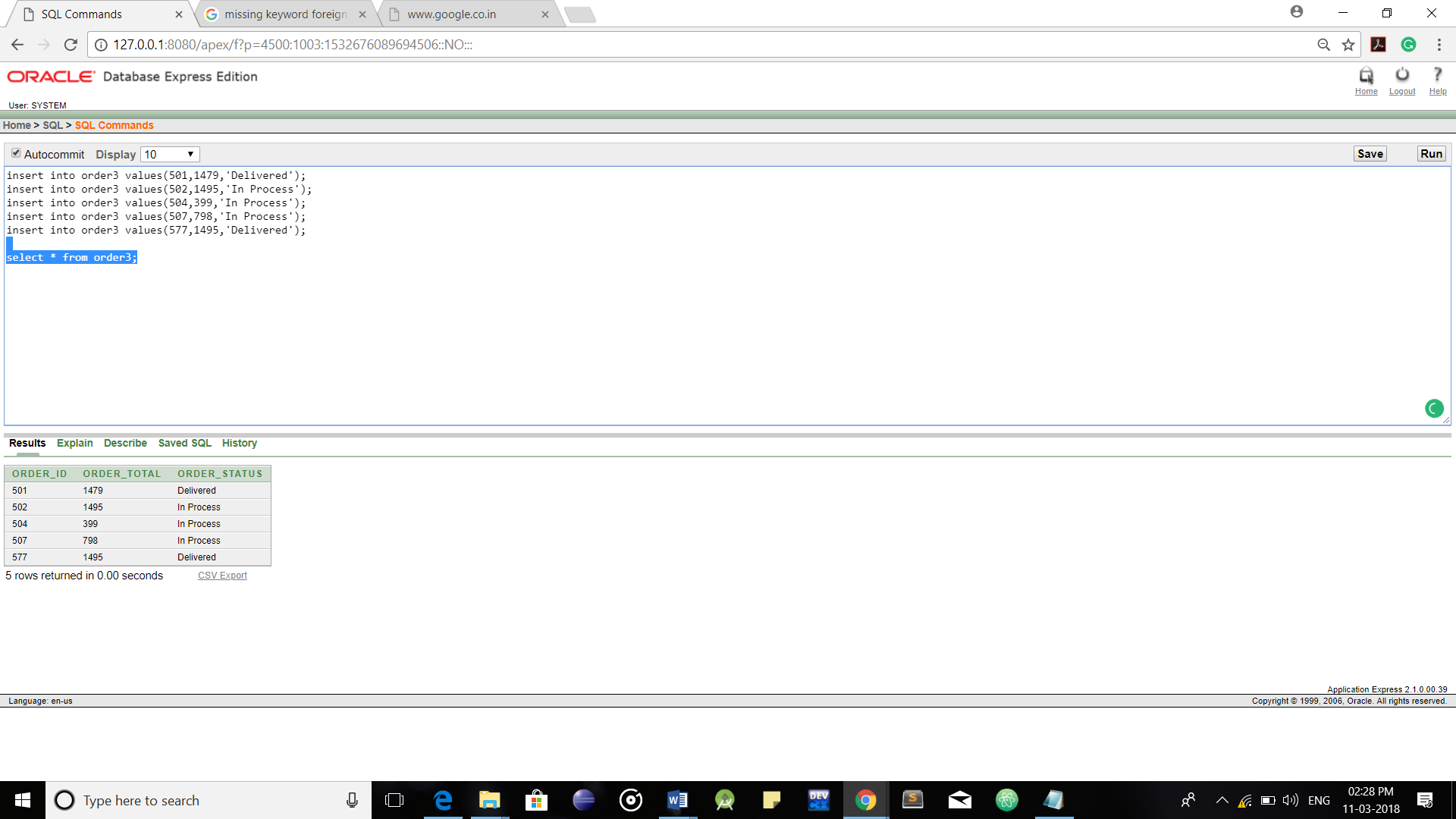
**Primary Key and Foreign Keys:**

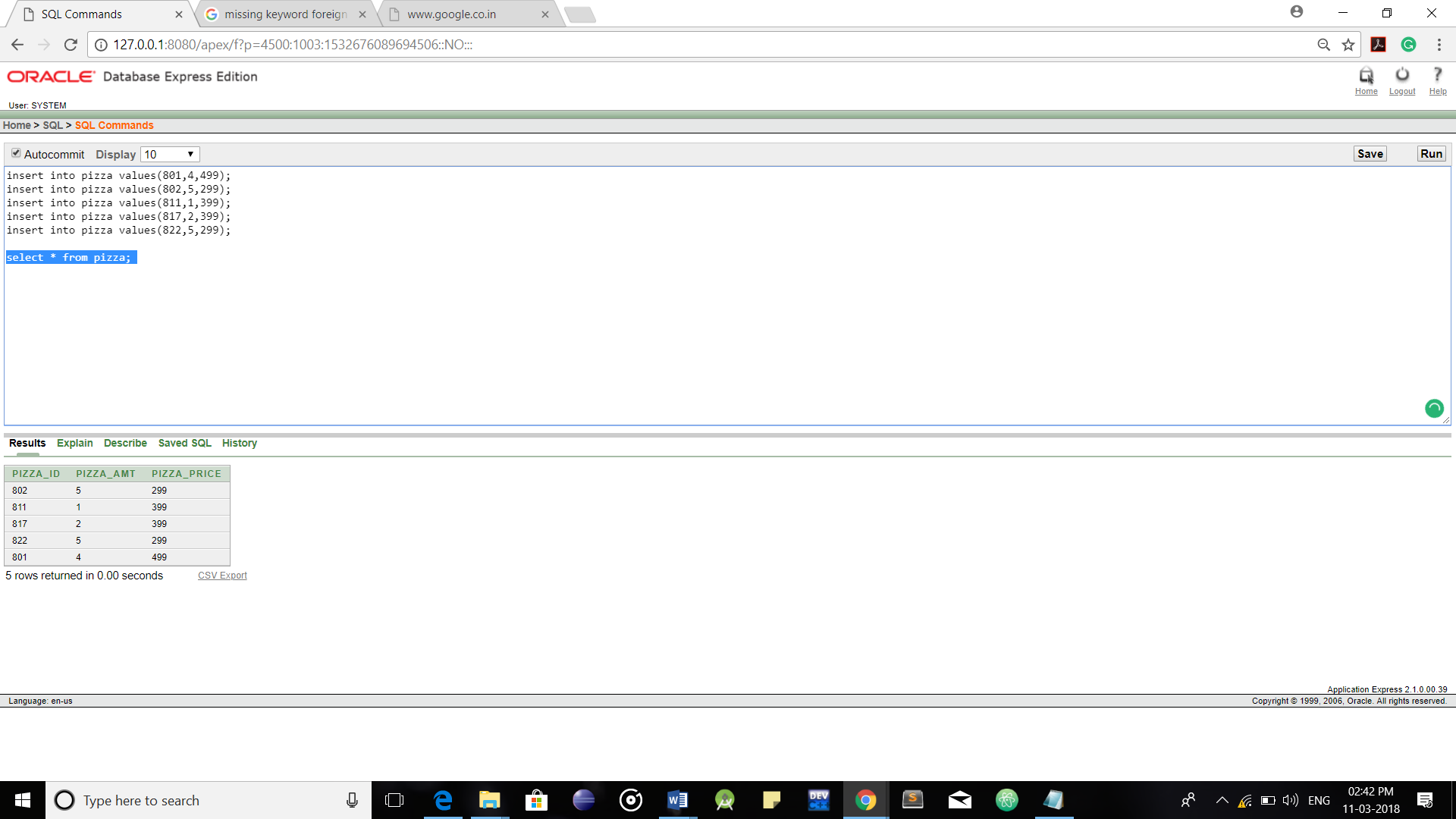


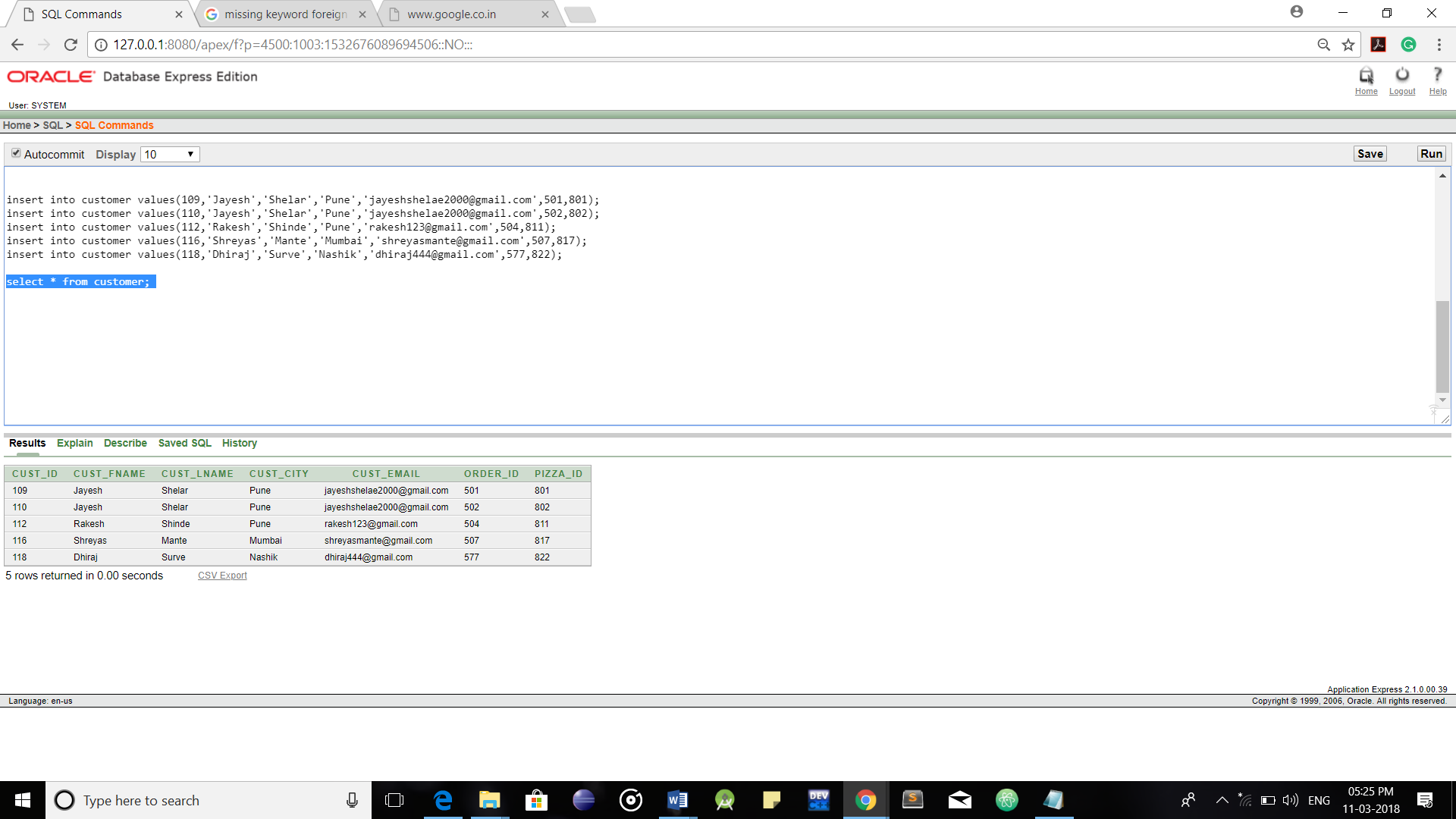
Primary Key

Foreign Keys

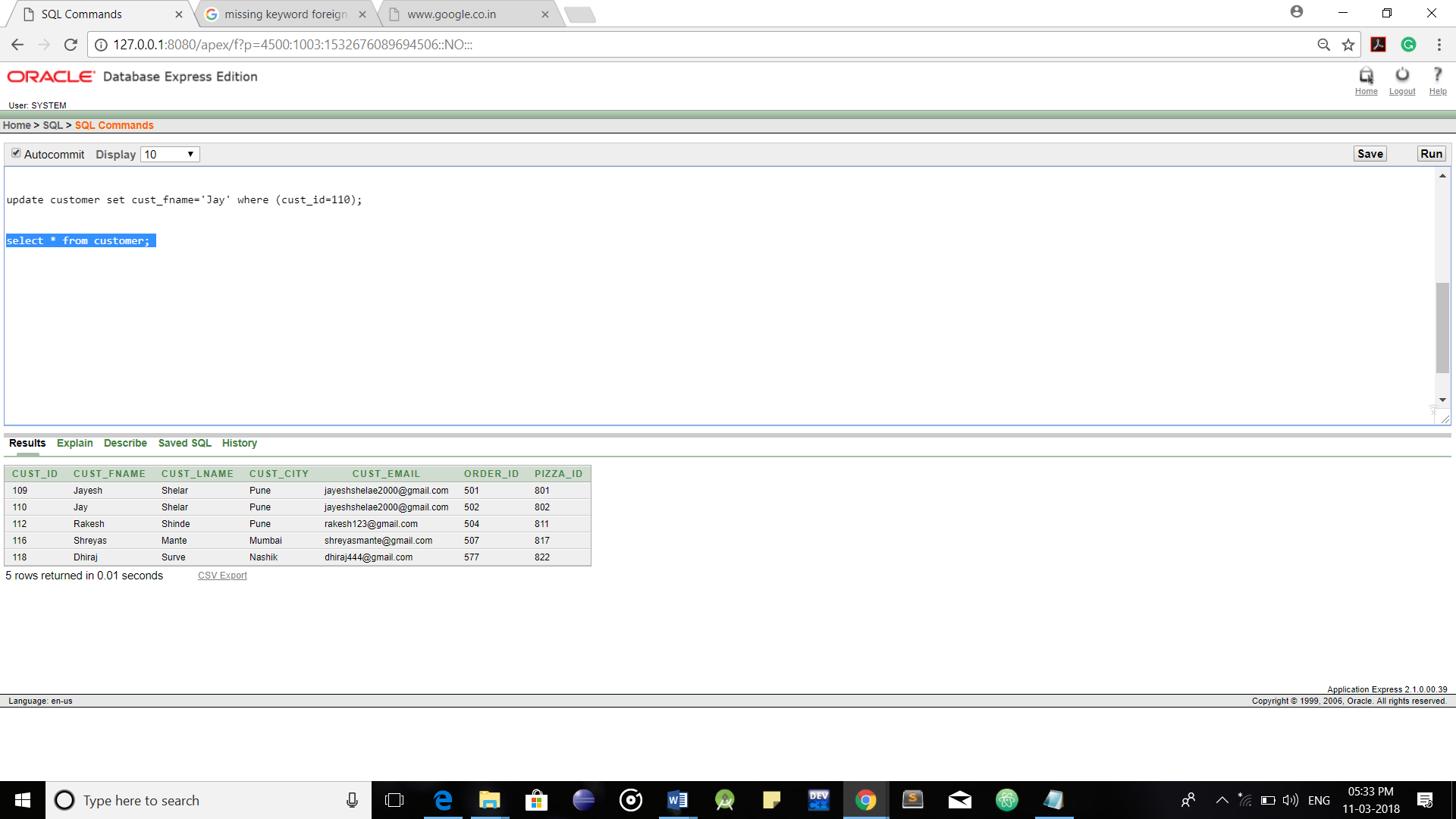
**Inserting Data into Tables:**



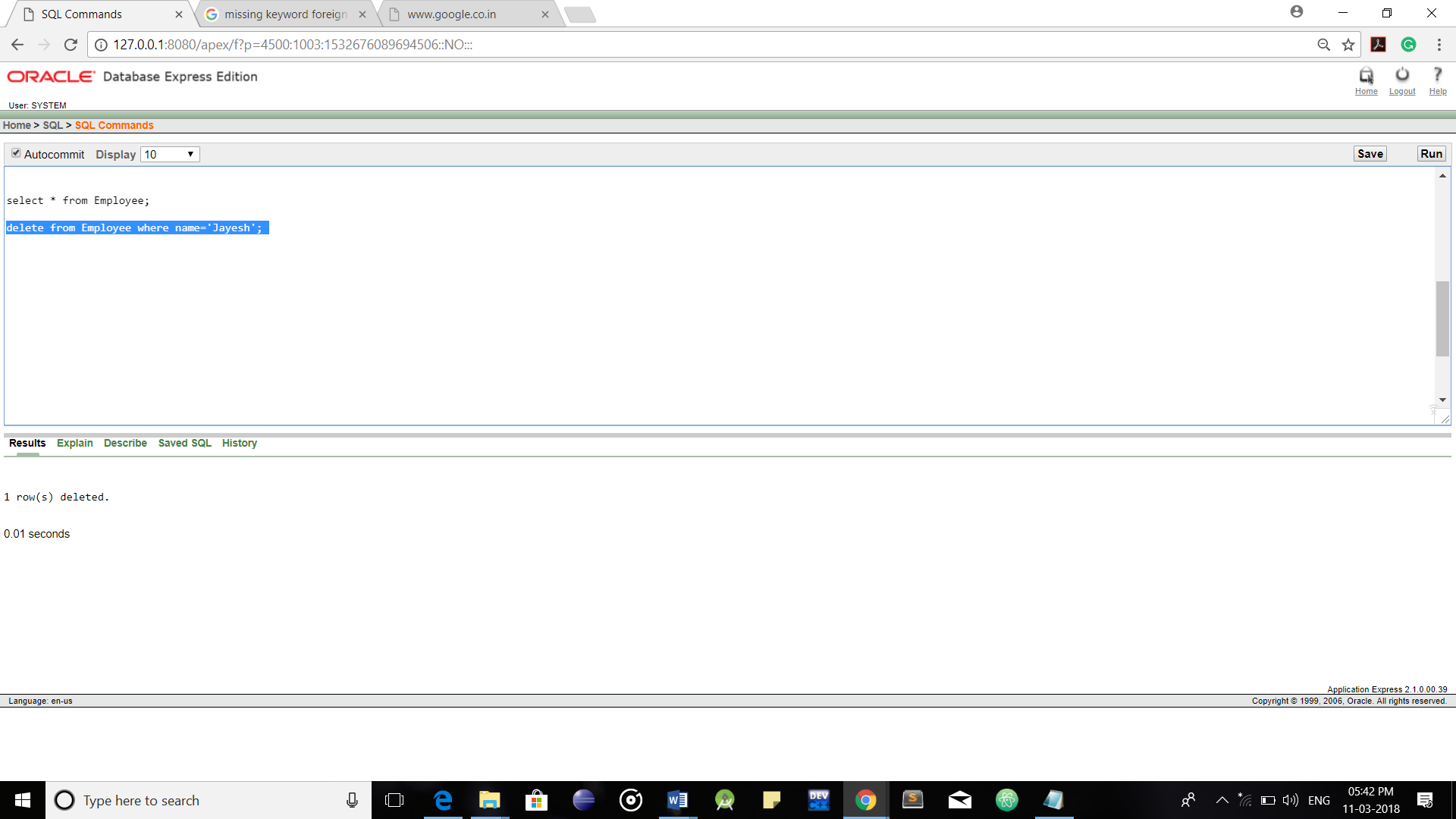




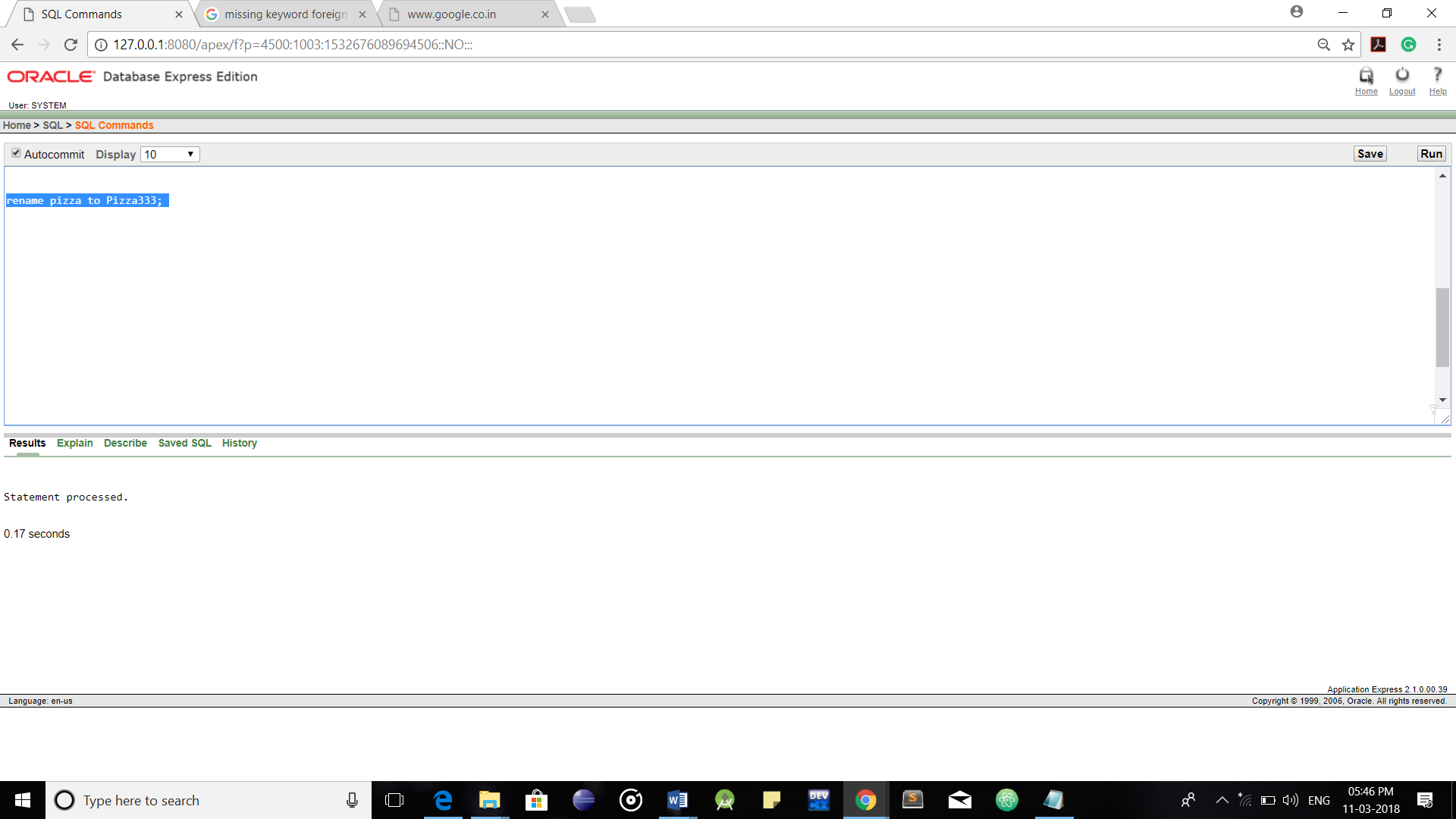
**Updating Contents of Table:**



**Deleting contents from Table:**



**Renaming Table Name:**



**Constraints:**

**Constraints**

Not Null Constraints

Foreign Key Constraints

Unique Key Constraints

Primary Key Constraints

Key Constraints

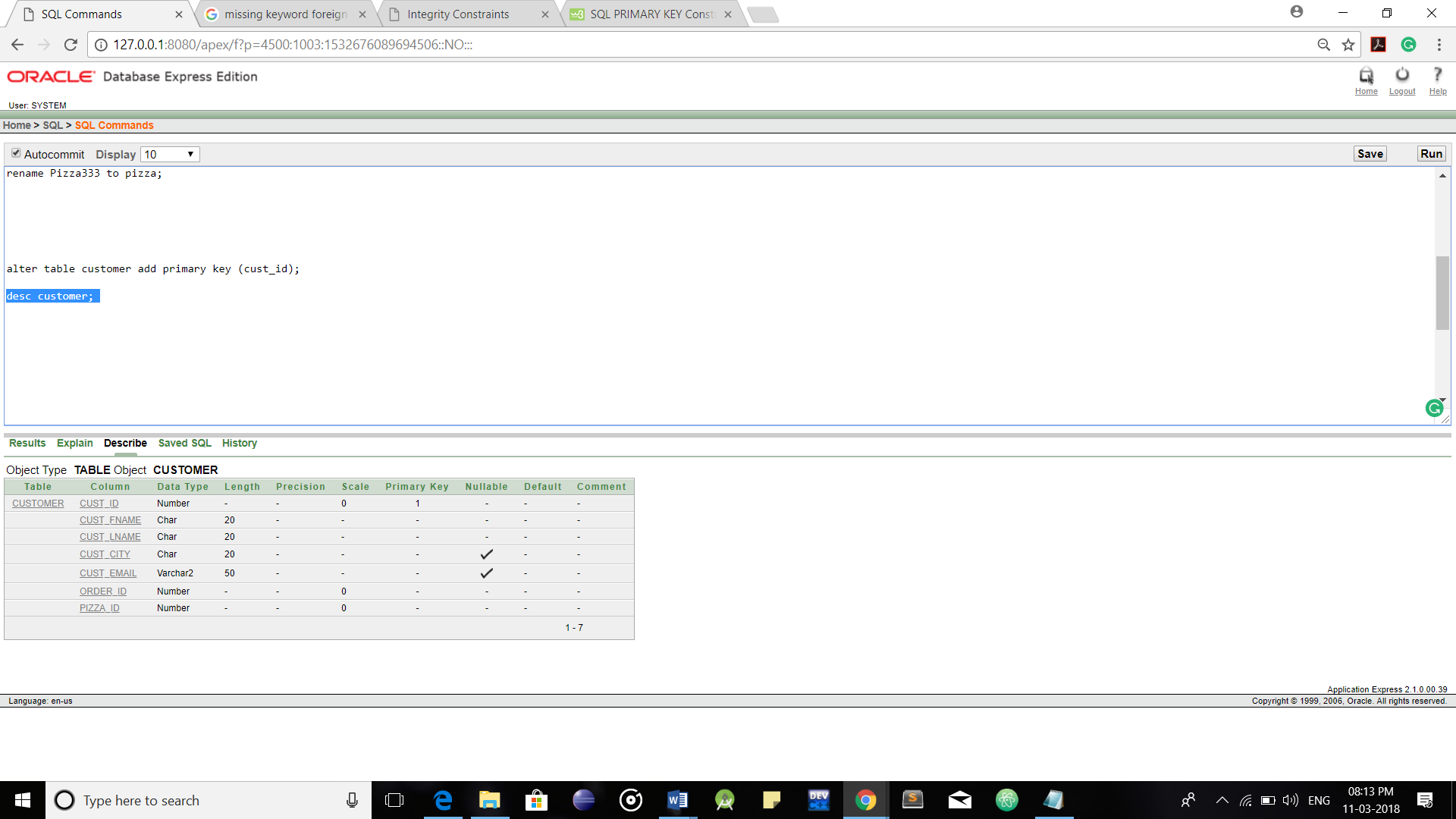
Referential Integrity Constraints

Entity Integrity Constraints

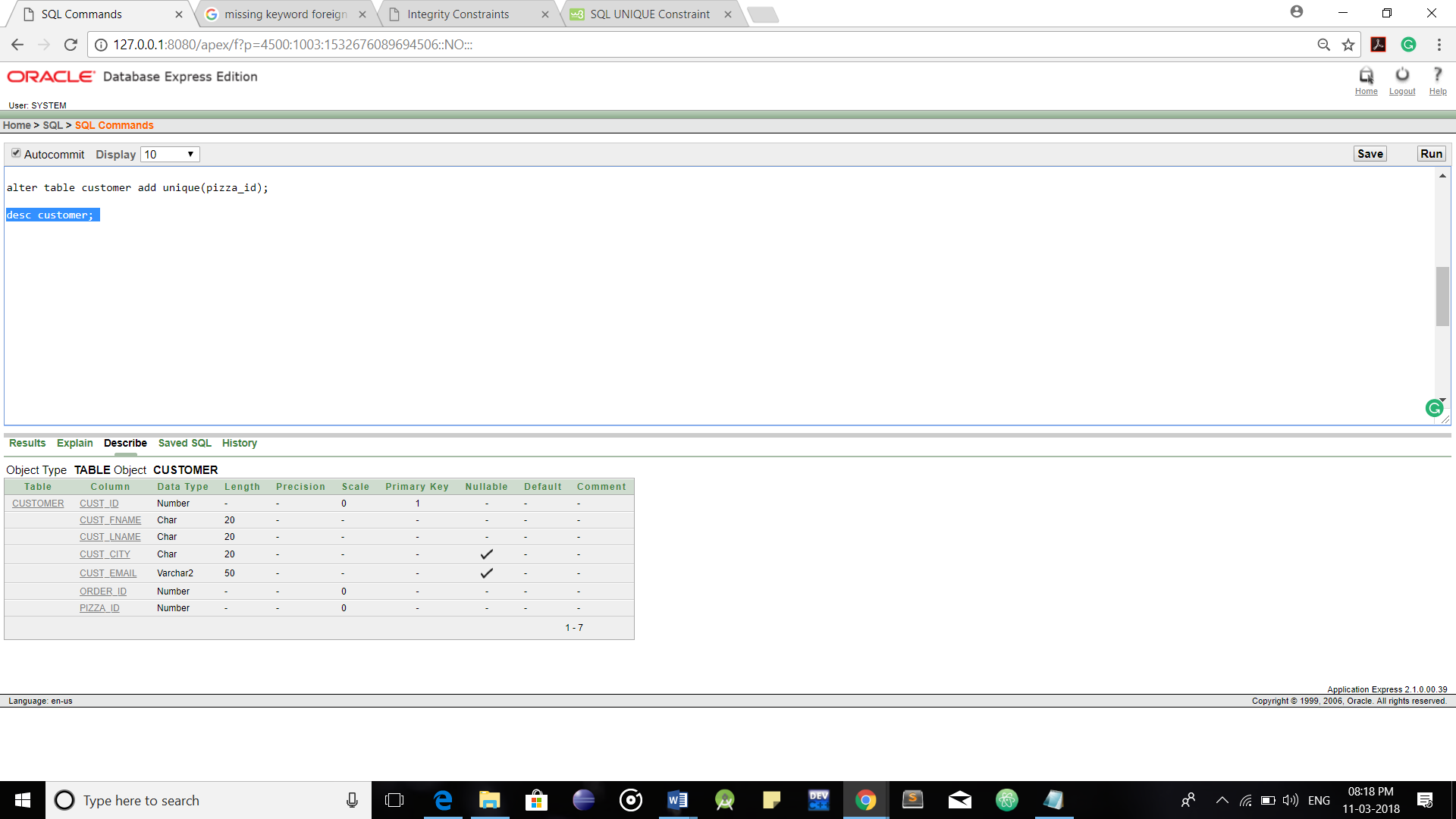
Domain Integrity

**Key Constraints:**

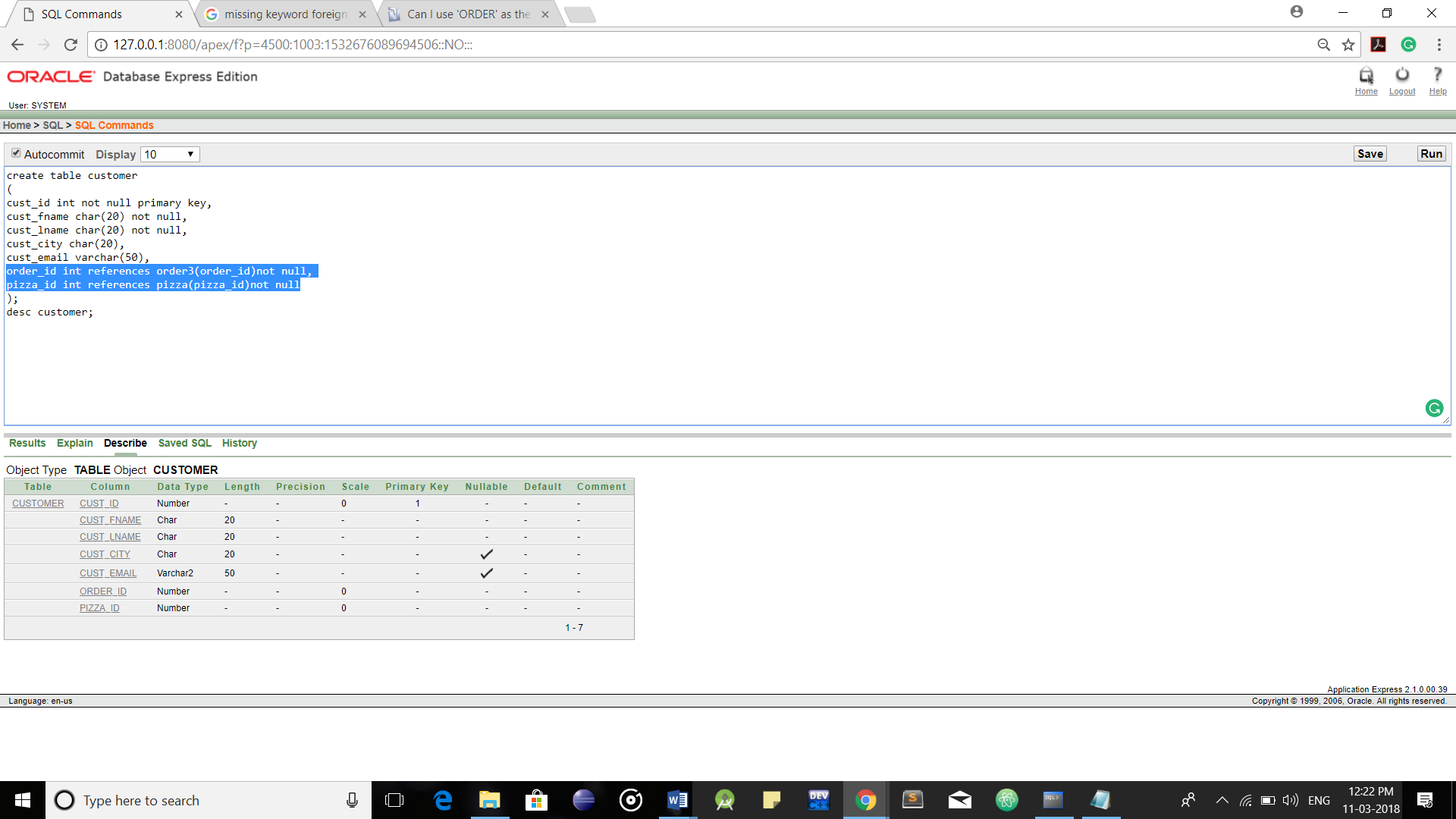
**1.Primary Key Constraints:**



**2.Unique Key Constraints:**



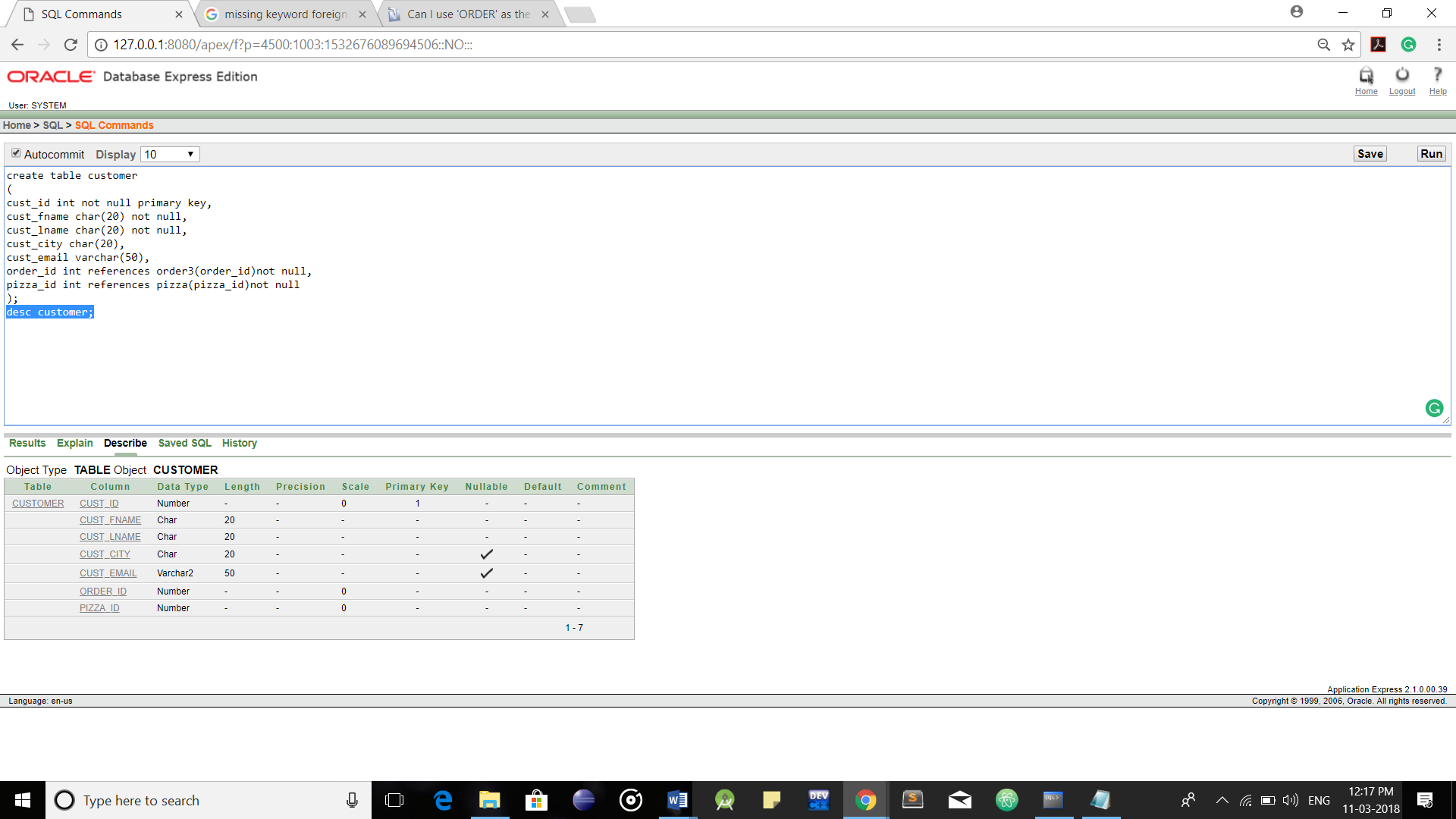
**3.Foreign Key Constraint:**



Foreign Keys

**4.Not Null Constraint:**

**Tip:** If the table has already been created, you can add a NOT NULL constraint to a column with the [ALTER TABLE](https://www.w3schools.com/sql/sql_alter.asp) statement.



Not Null Constraints

**Operators in SQL:**

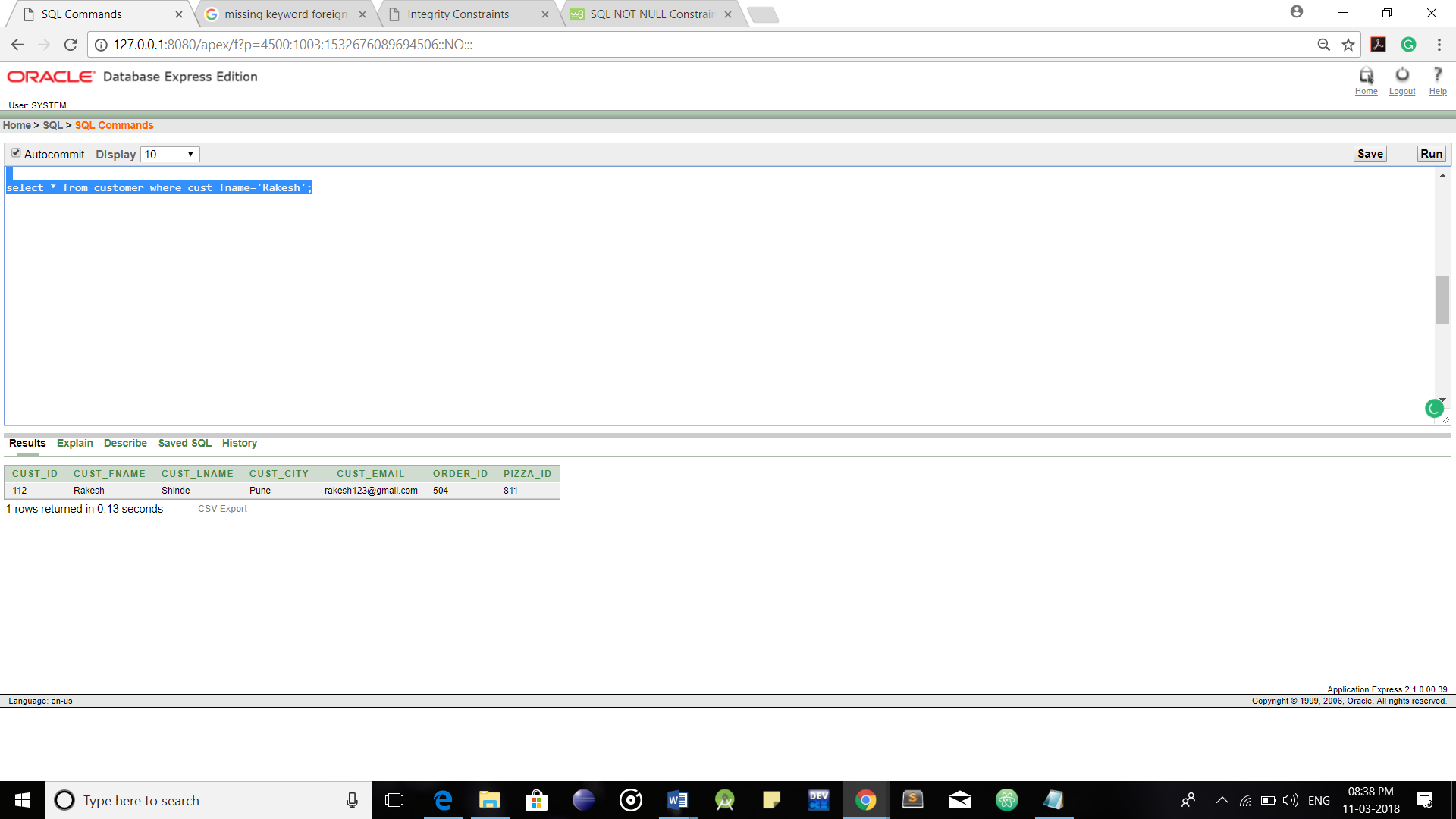
**1.Comparison/Relational Operators**

**2.Logical & Conjunctive Operators**

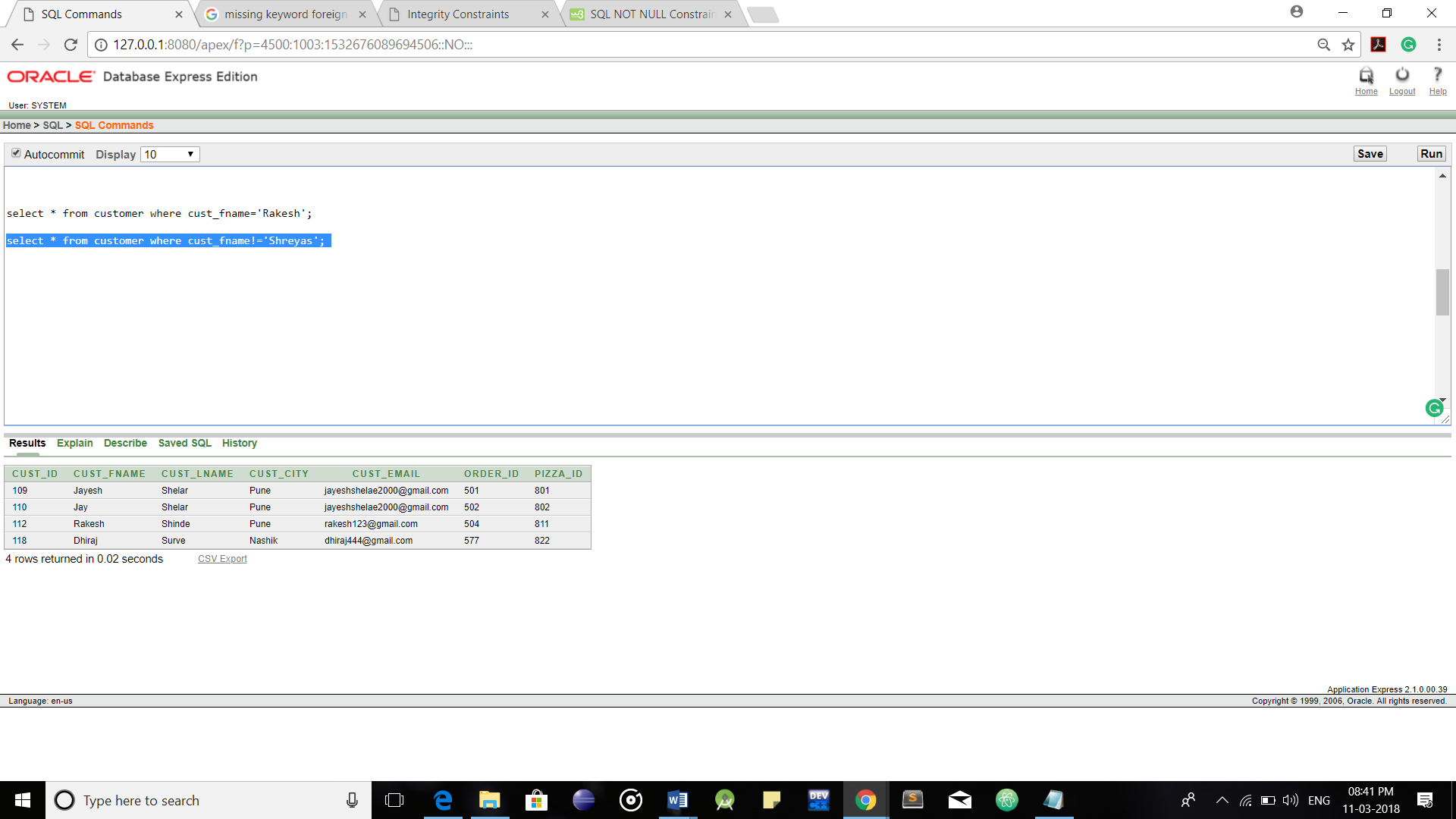
**3.Operators for Negating Conditions**

**4.Arithmetic Operators**

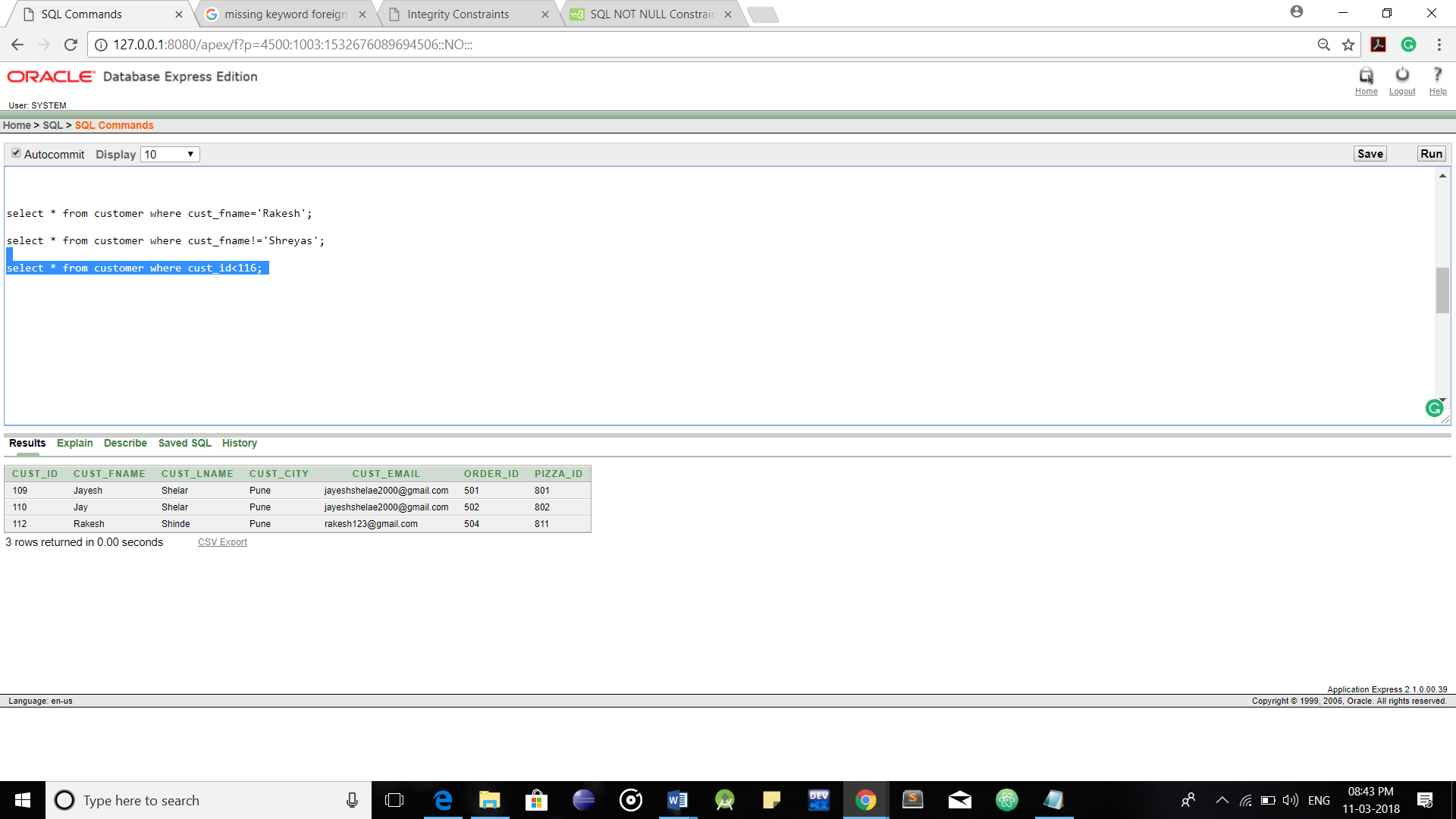
**1.Comparison/Relational Operators:**



Equality Operator

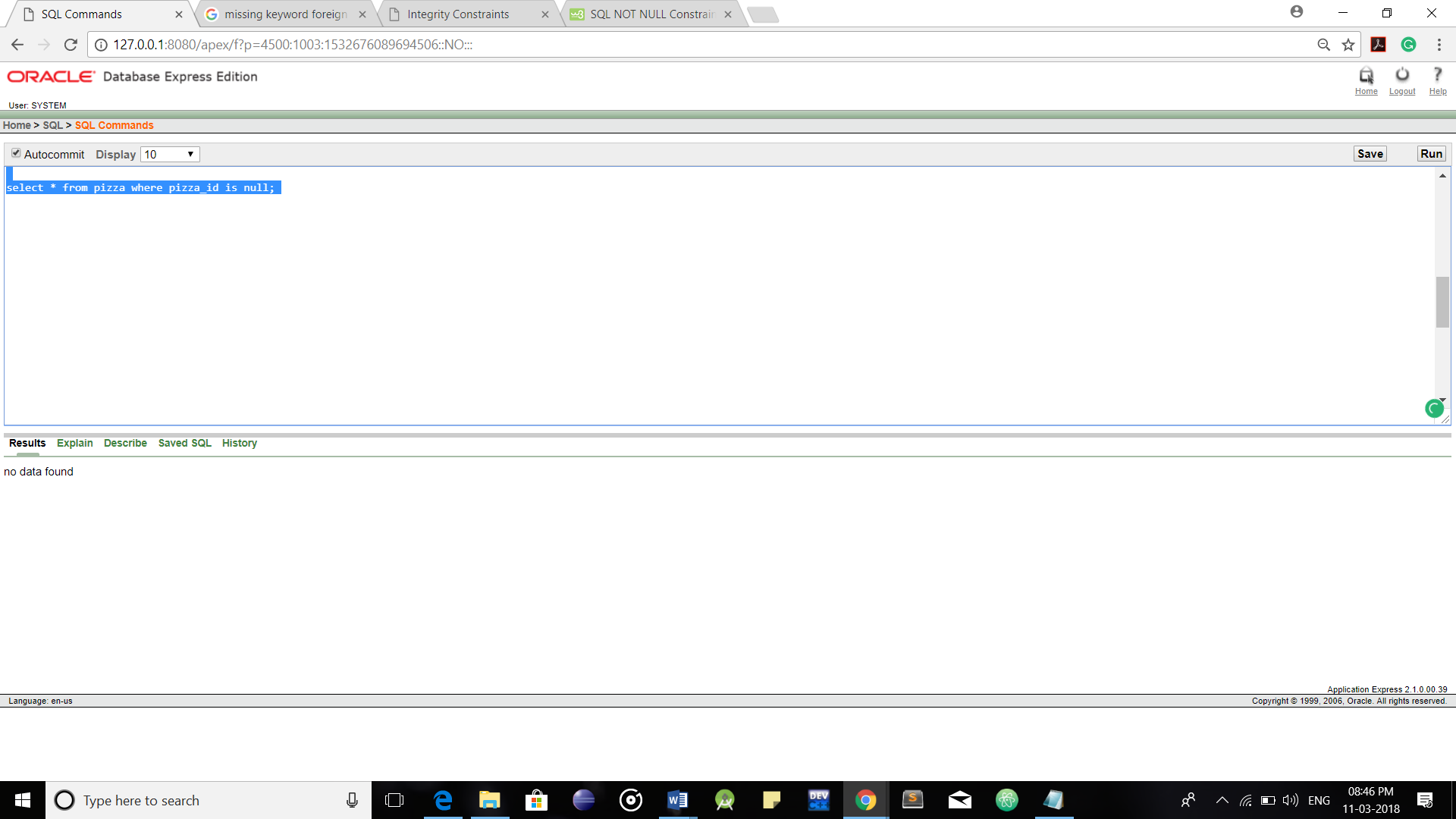


Non-Equality Operator

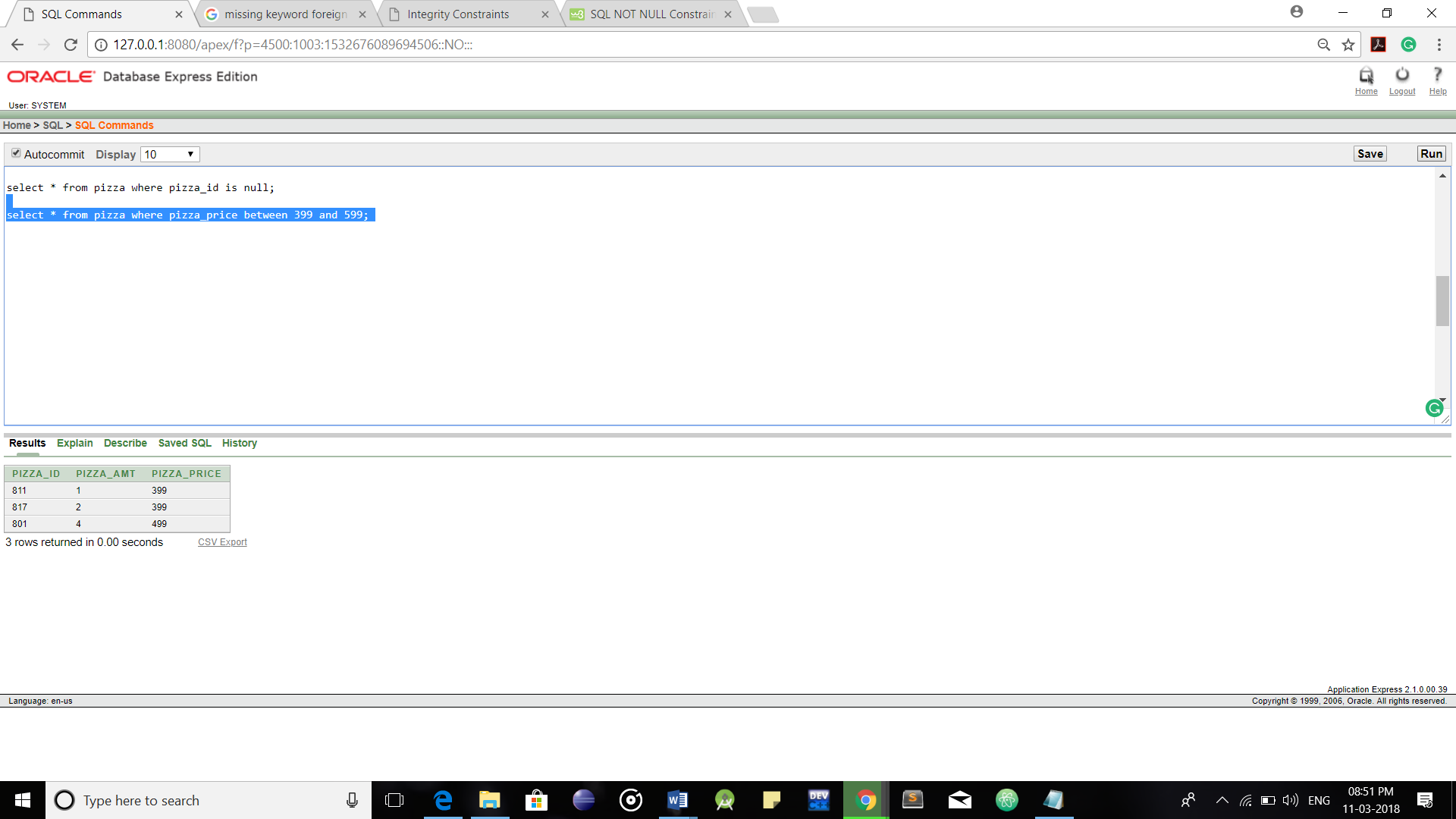


Less-than Operator

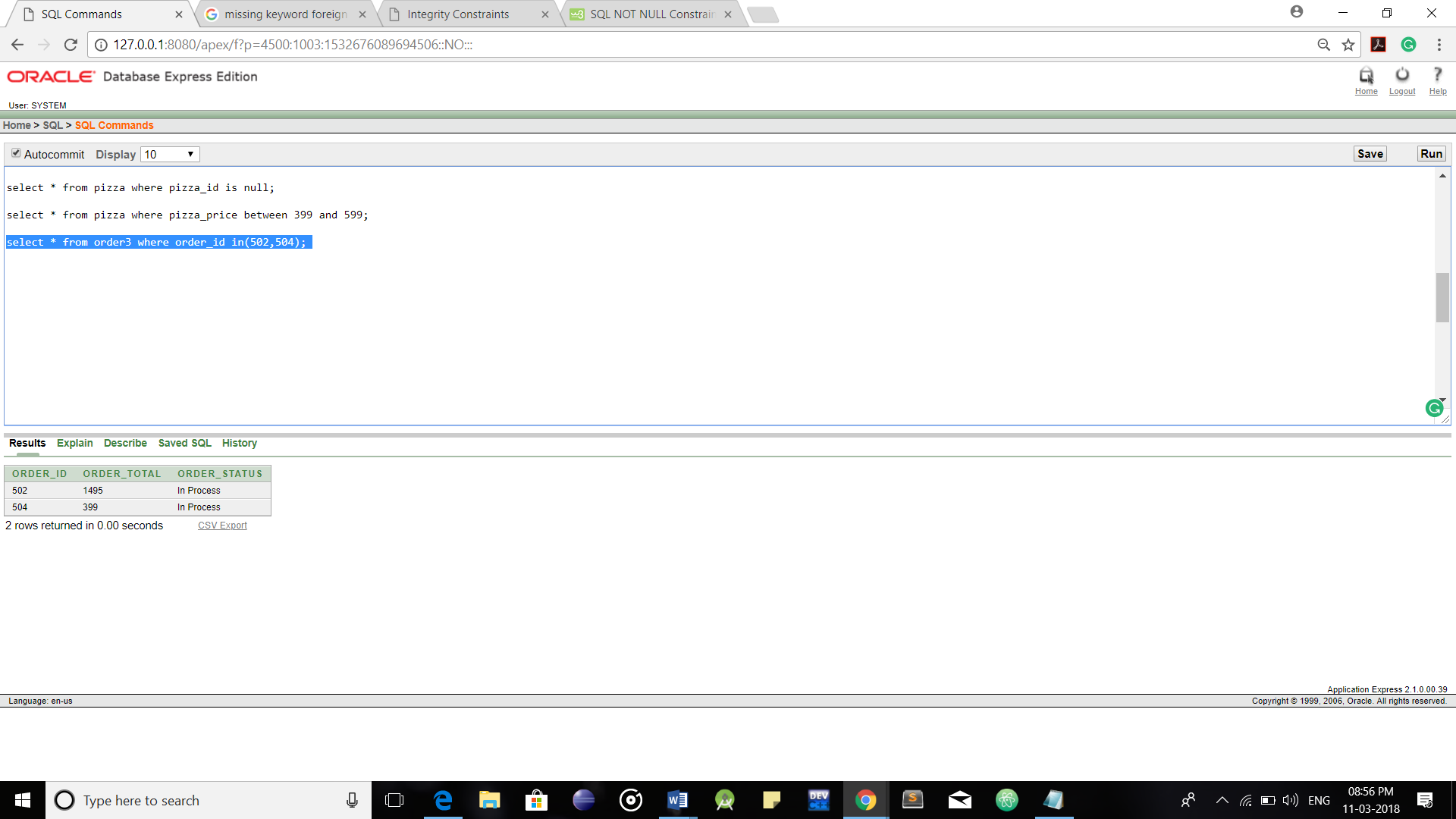
**2.Logical Operators:**



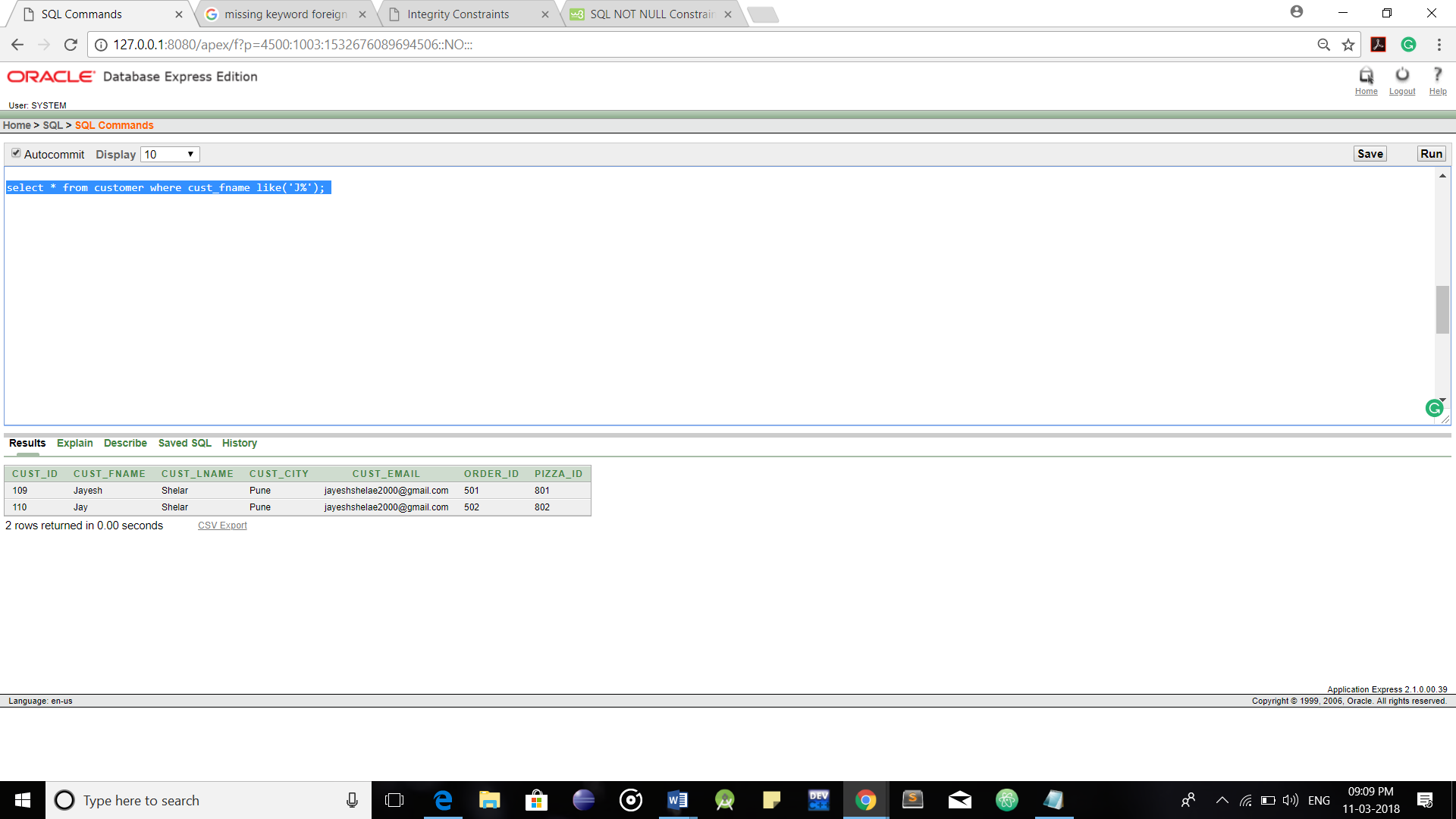
Is-Null Operator



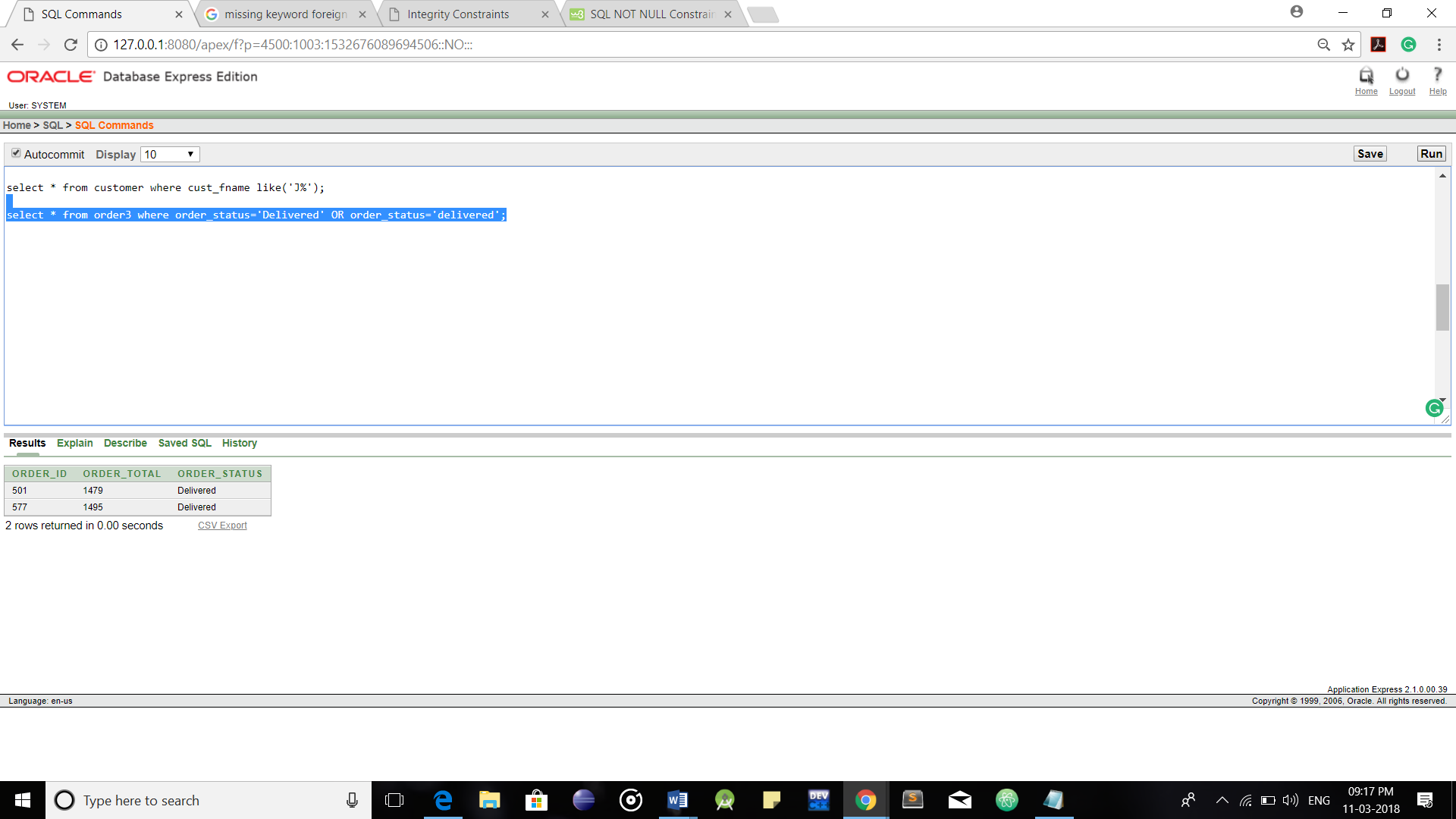
Between Operator



IN Operator

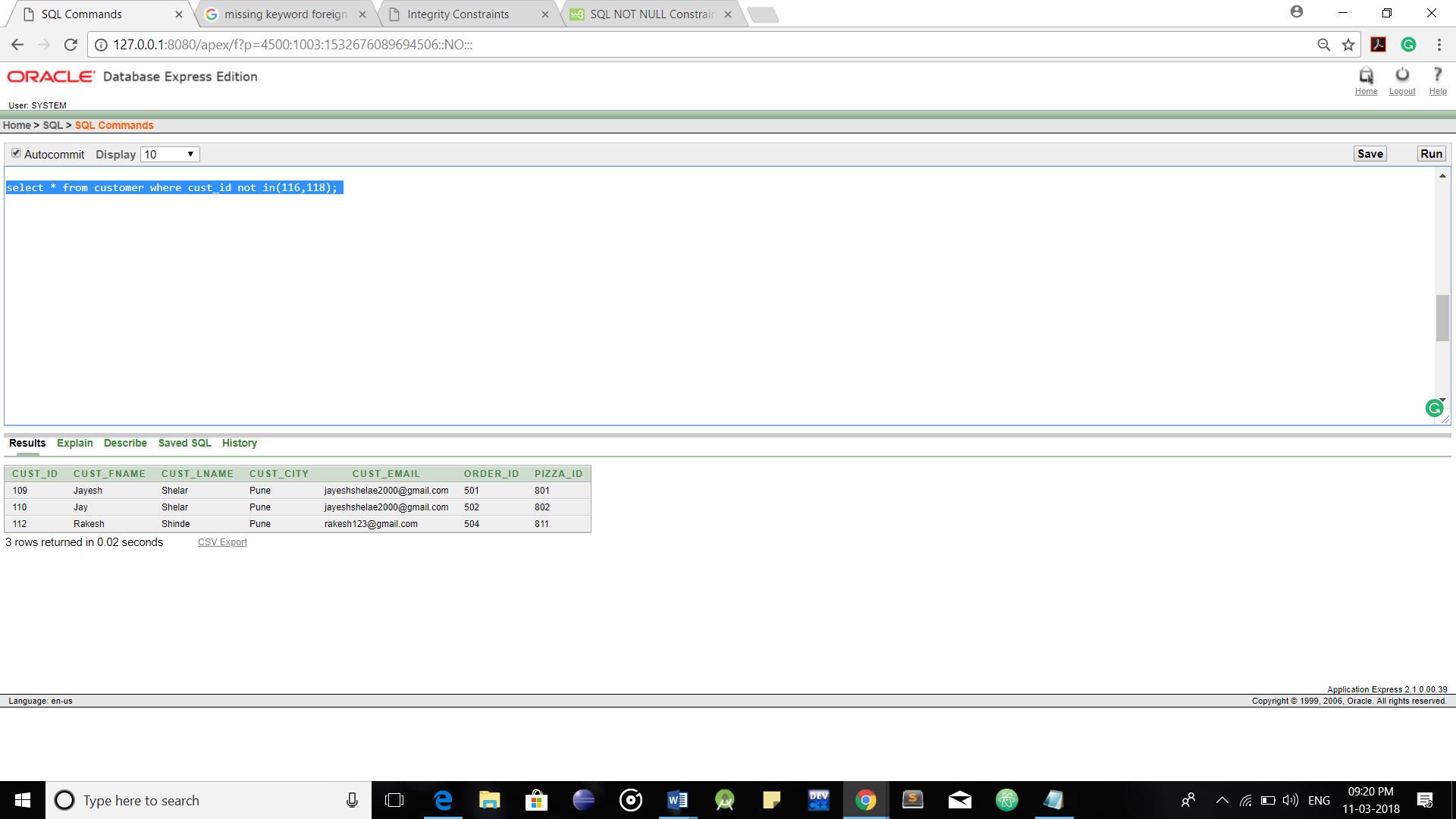


Like Operator



Conjunctive OR Operator

**3.Operators for Negating Conditions:**

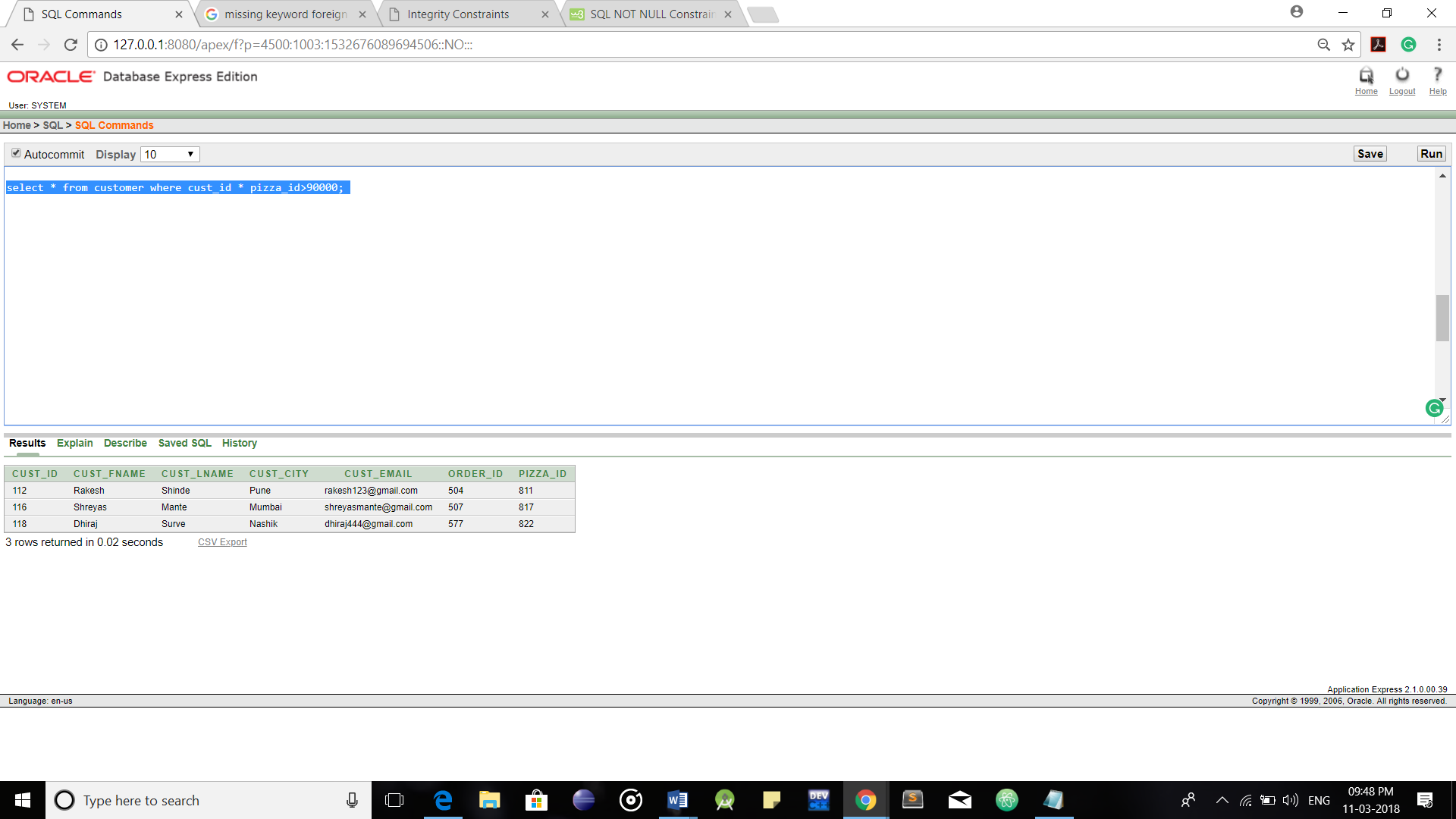


Negating NOT Operator

**4.Arithmetic Operators:**



Addition Operator

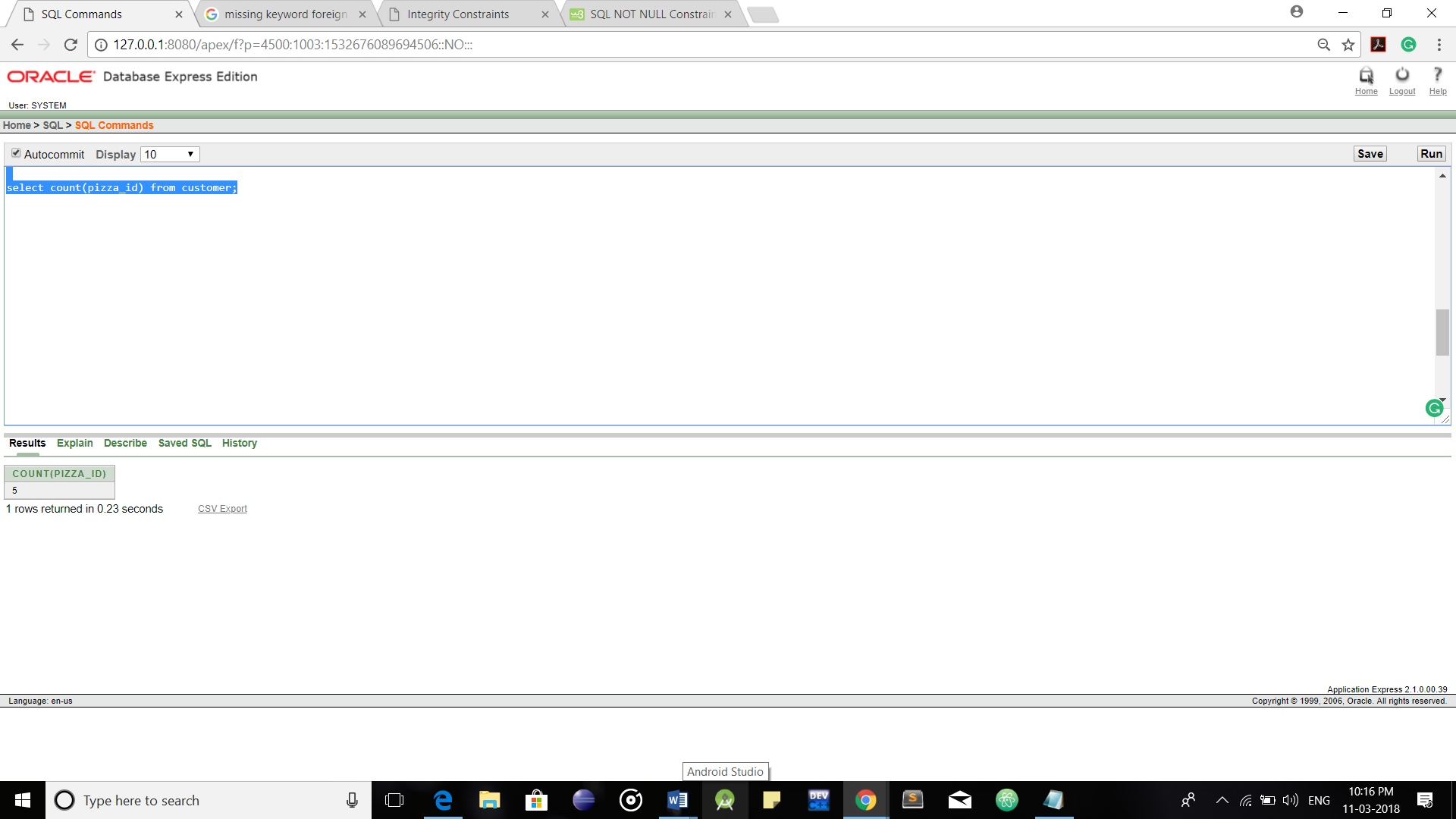


Multiplication Operator

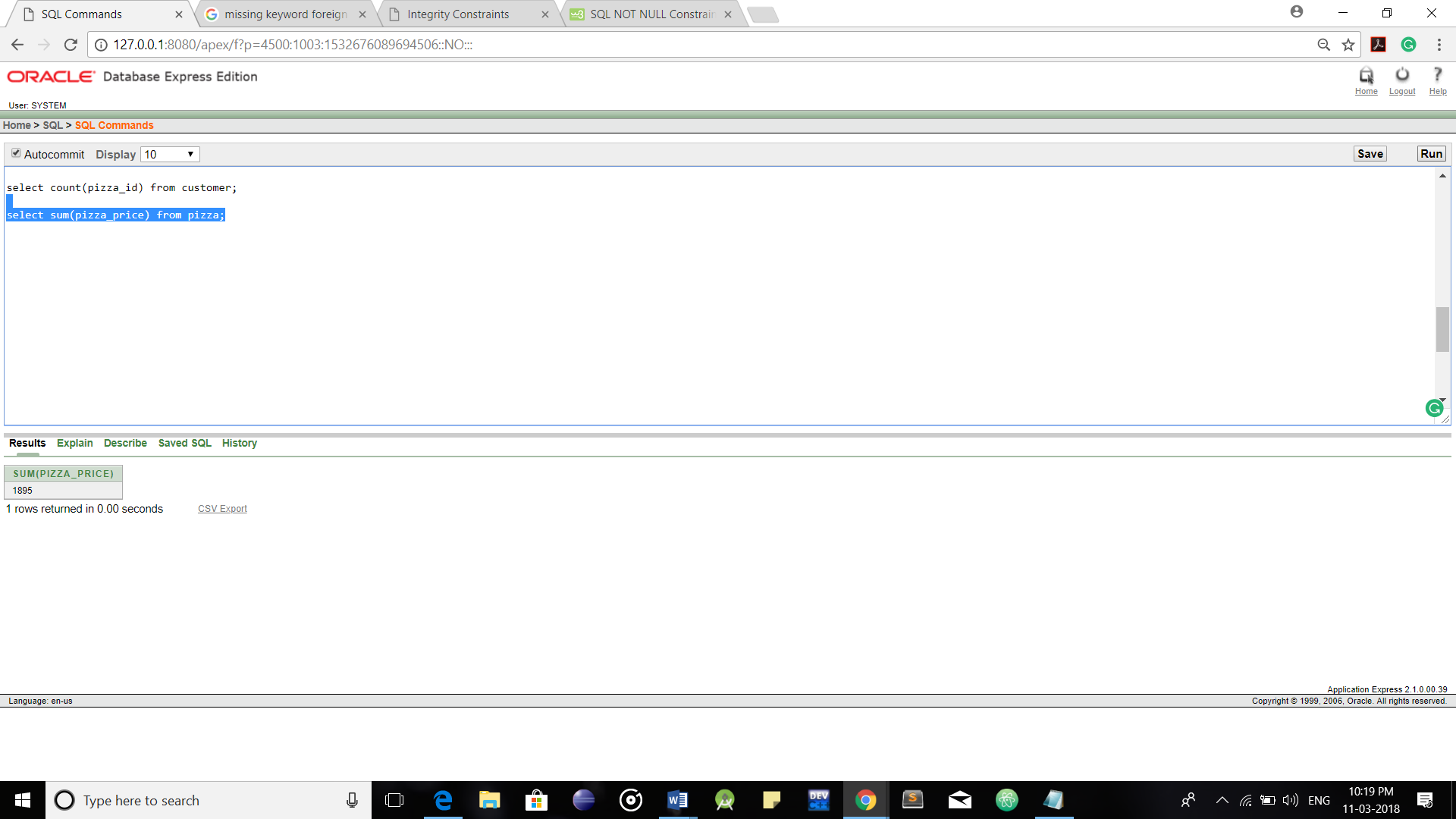
**Functions in SQL:**

**1.Aggregate Functions:**

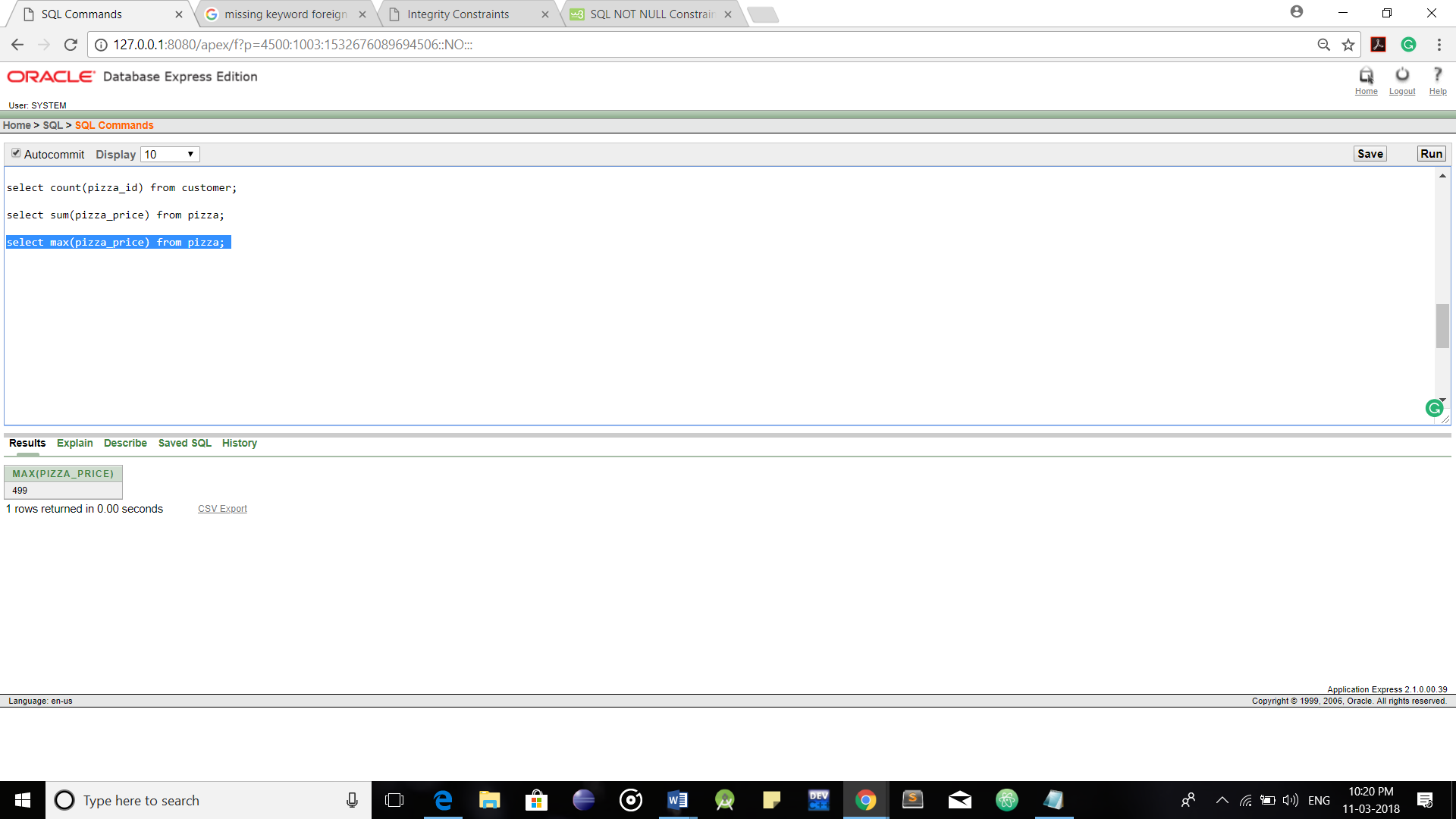
**1)Count Function:**



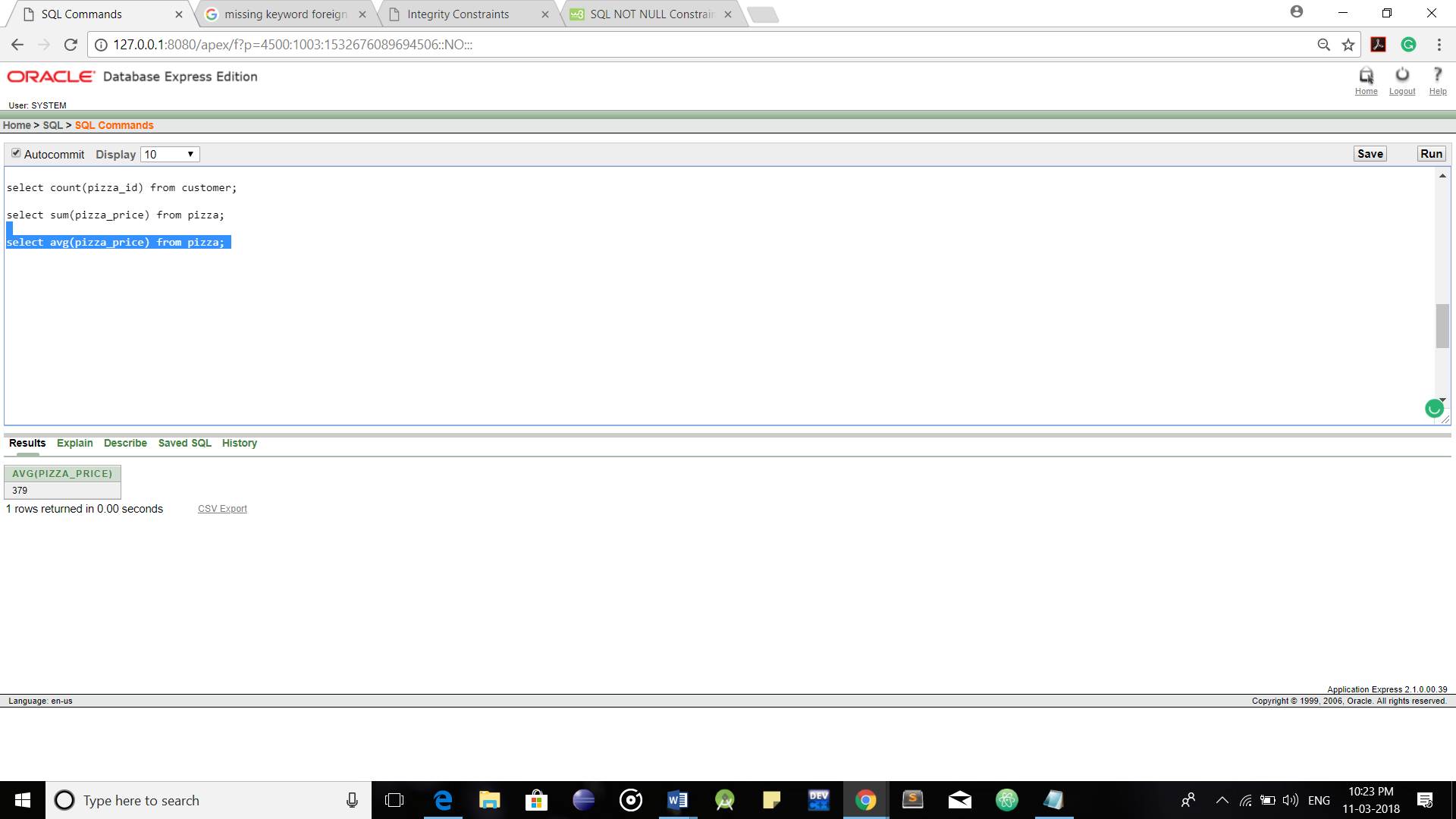
**2)Sum Function:**



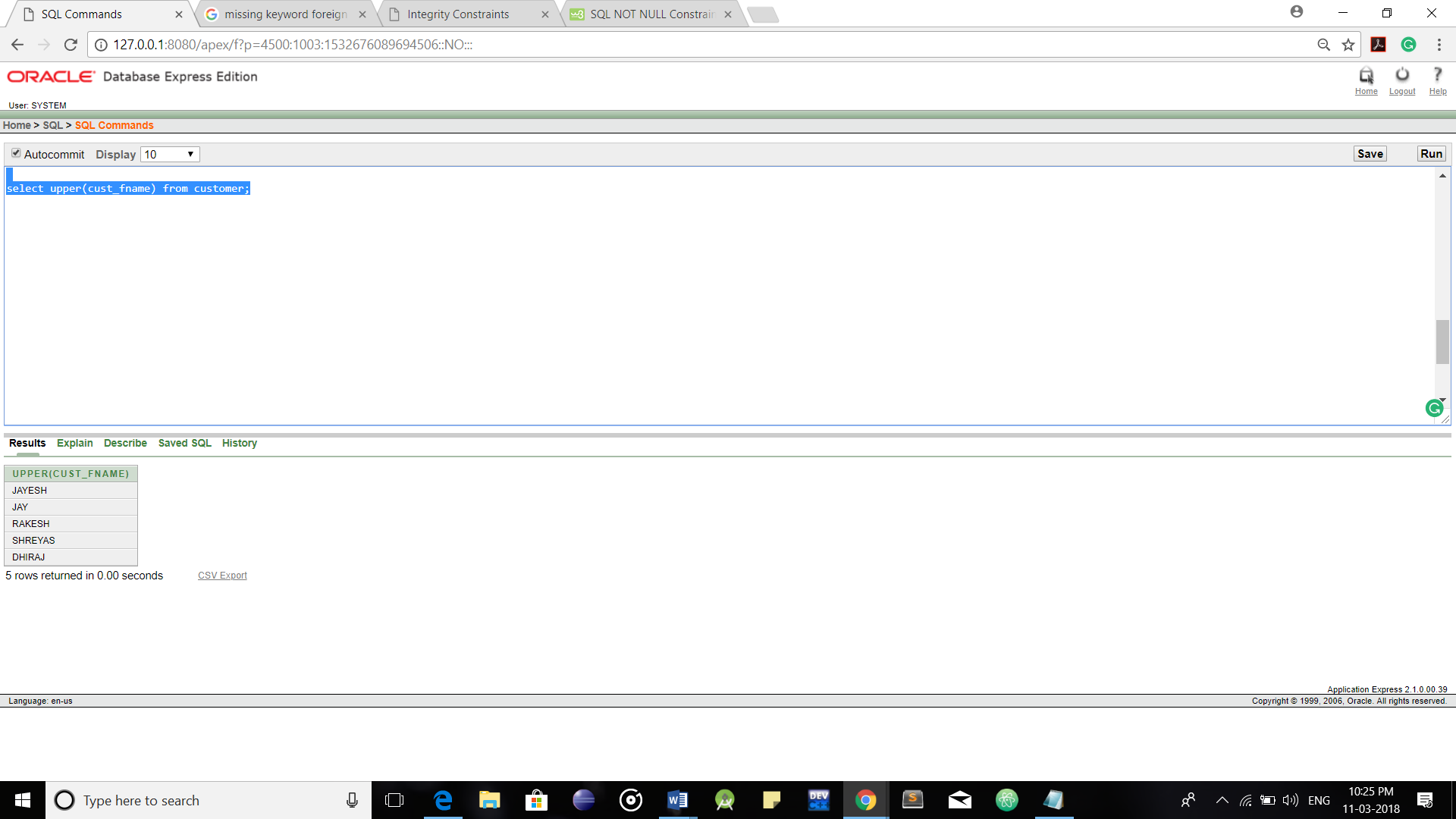
**3)Max/Min:**



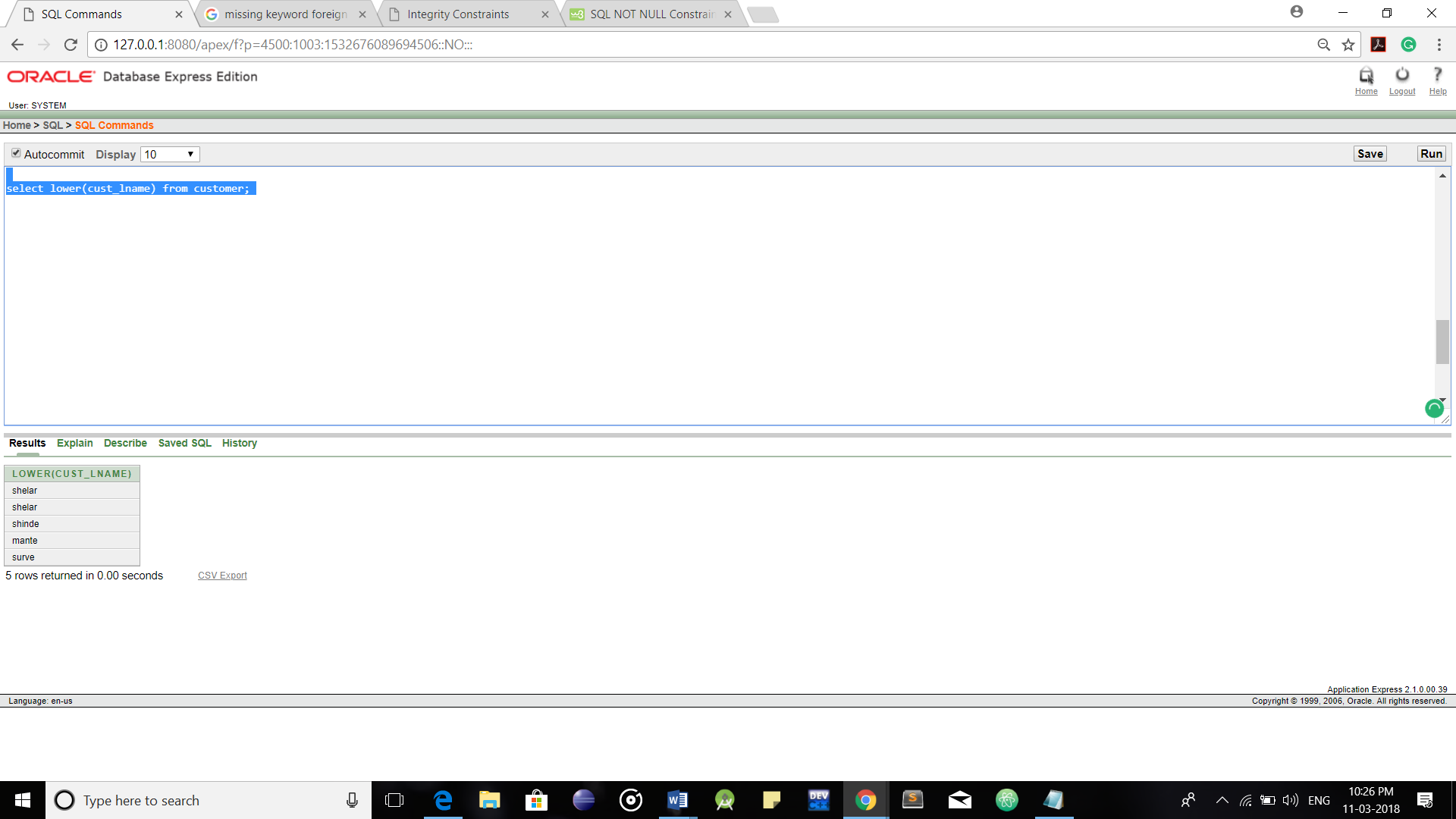
**4)Average:**



**2.String Functions:**



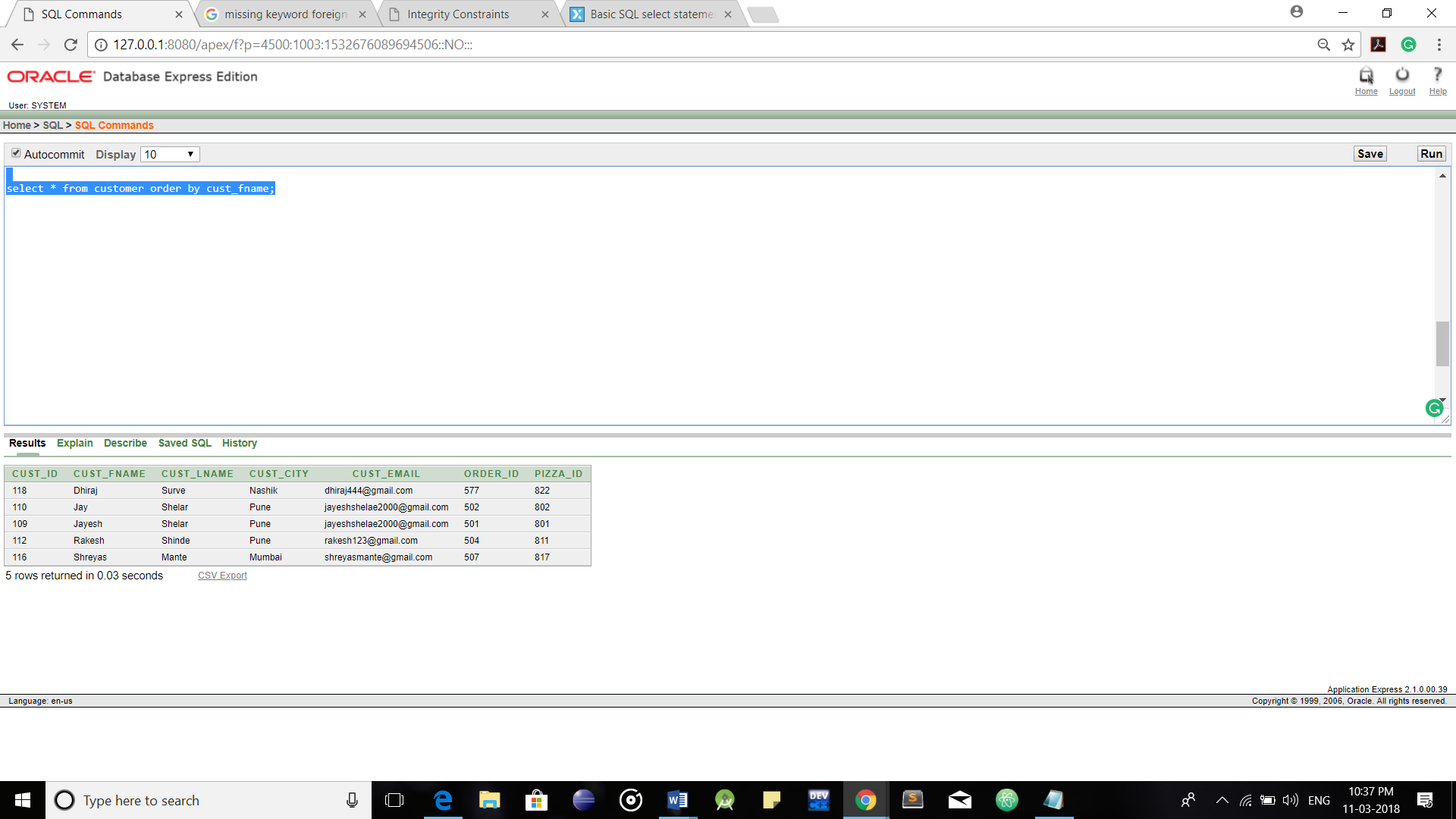
Upper Case



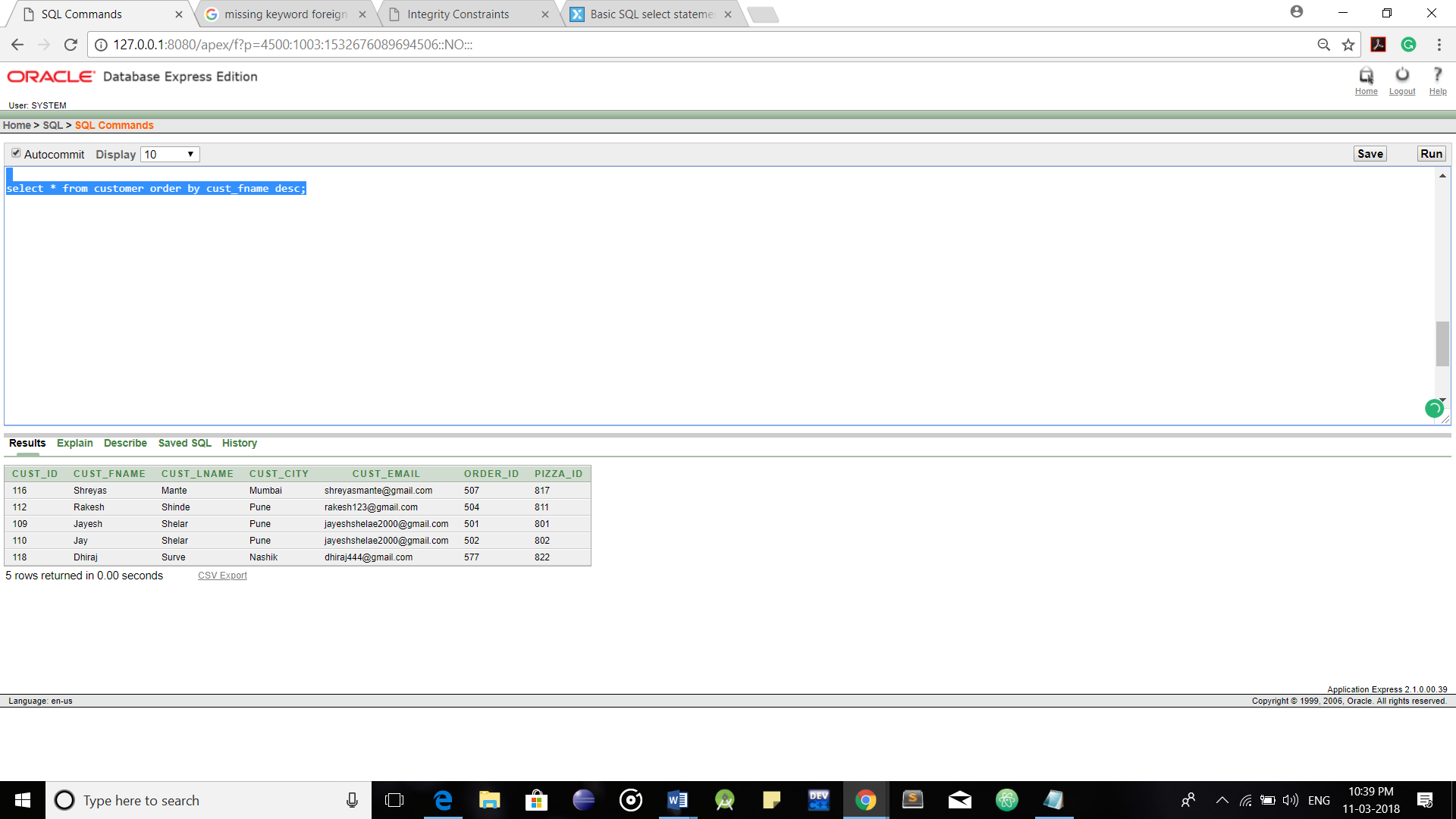
Lower Case

**Clauses:**

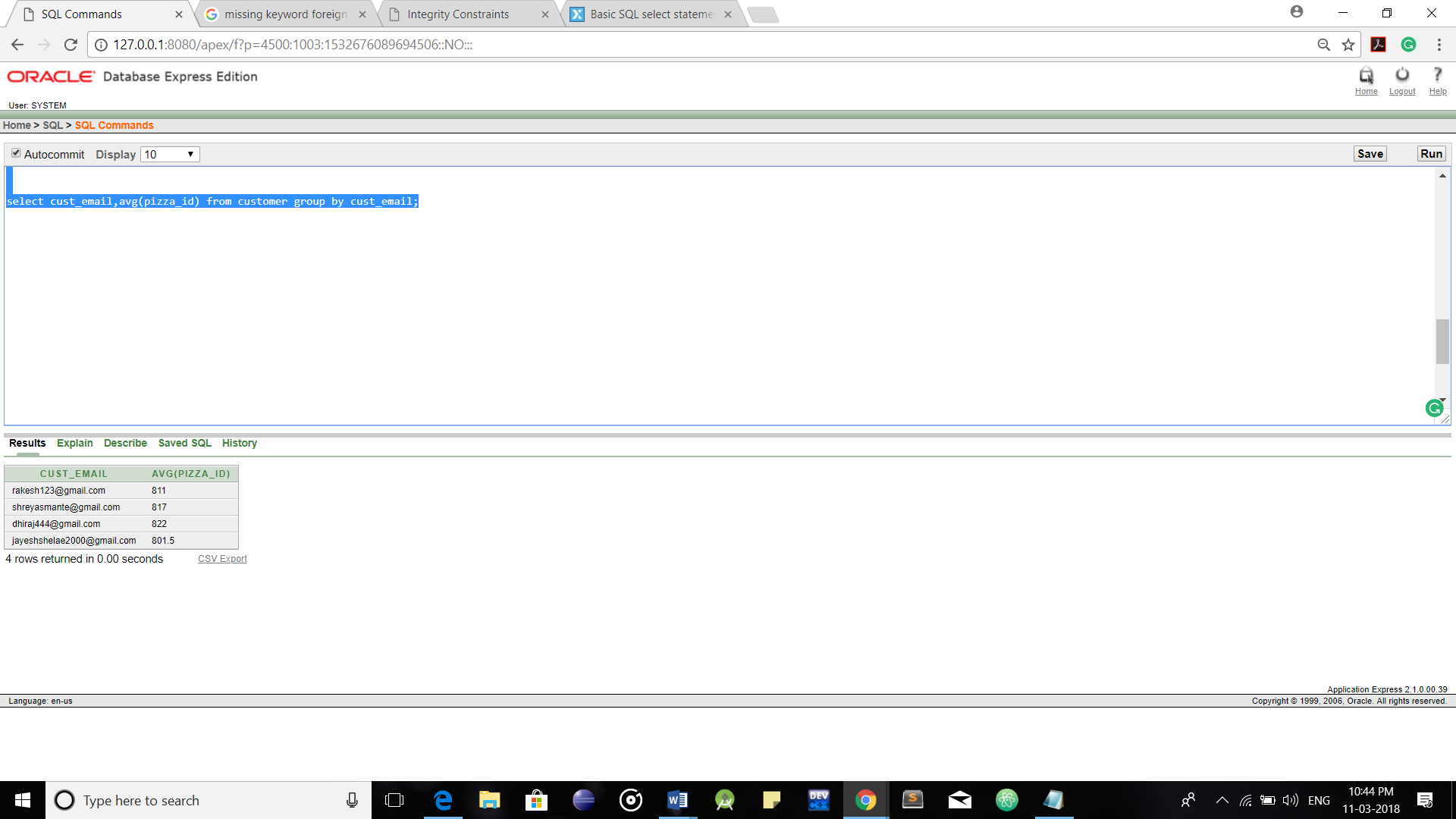
**1.Order by Clause:**



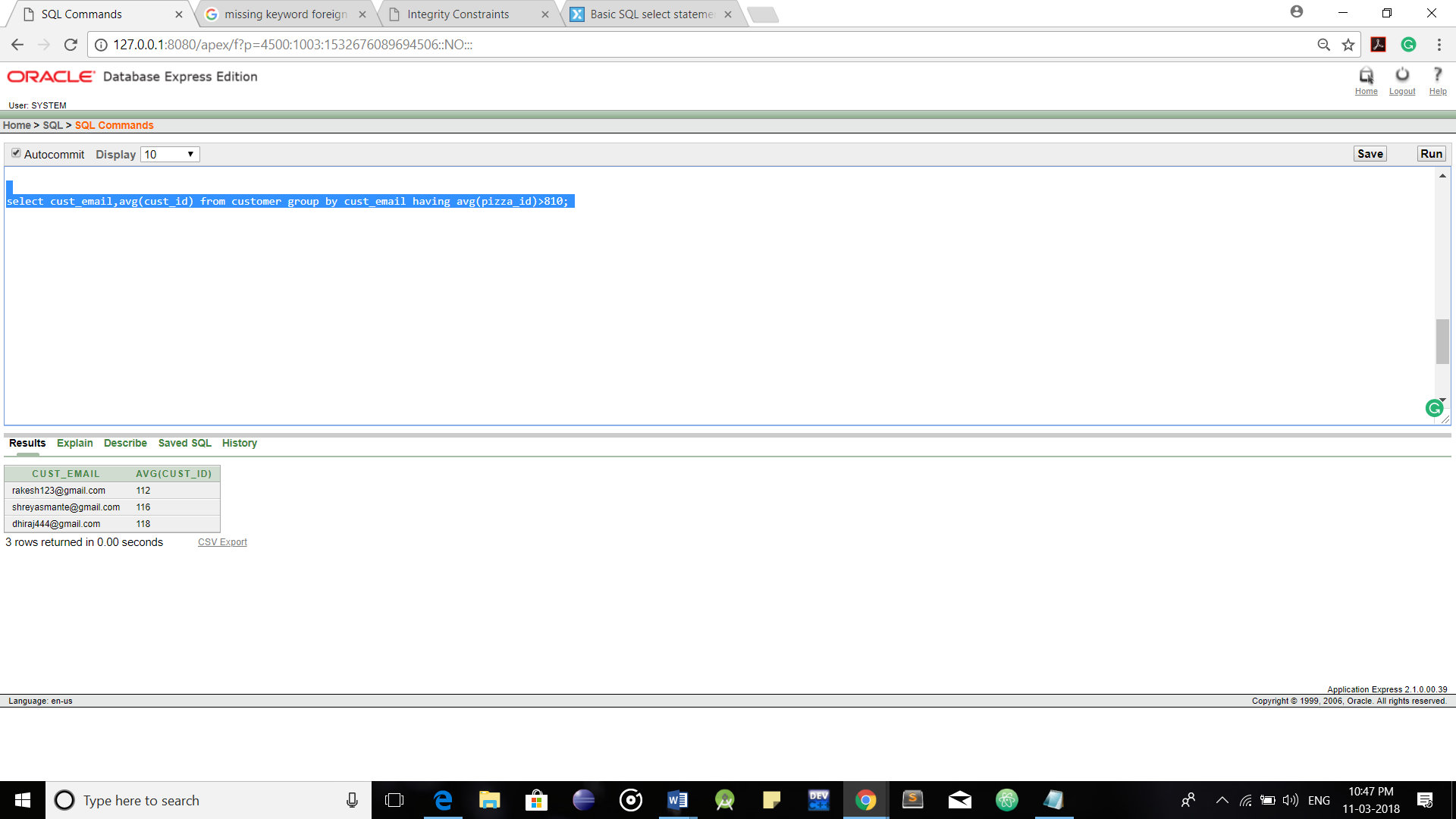
**2.Order by(desc):**



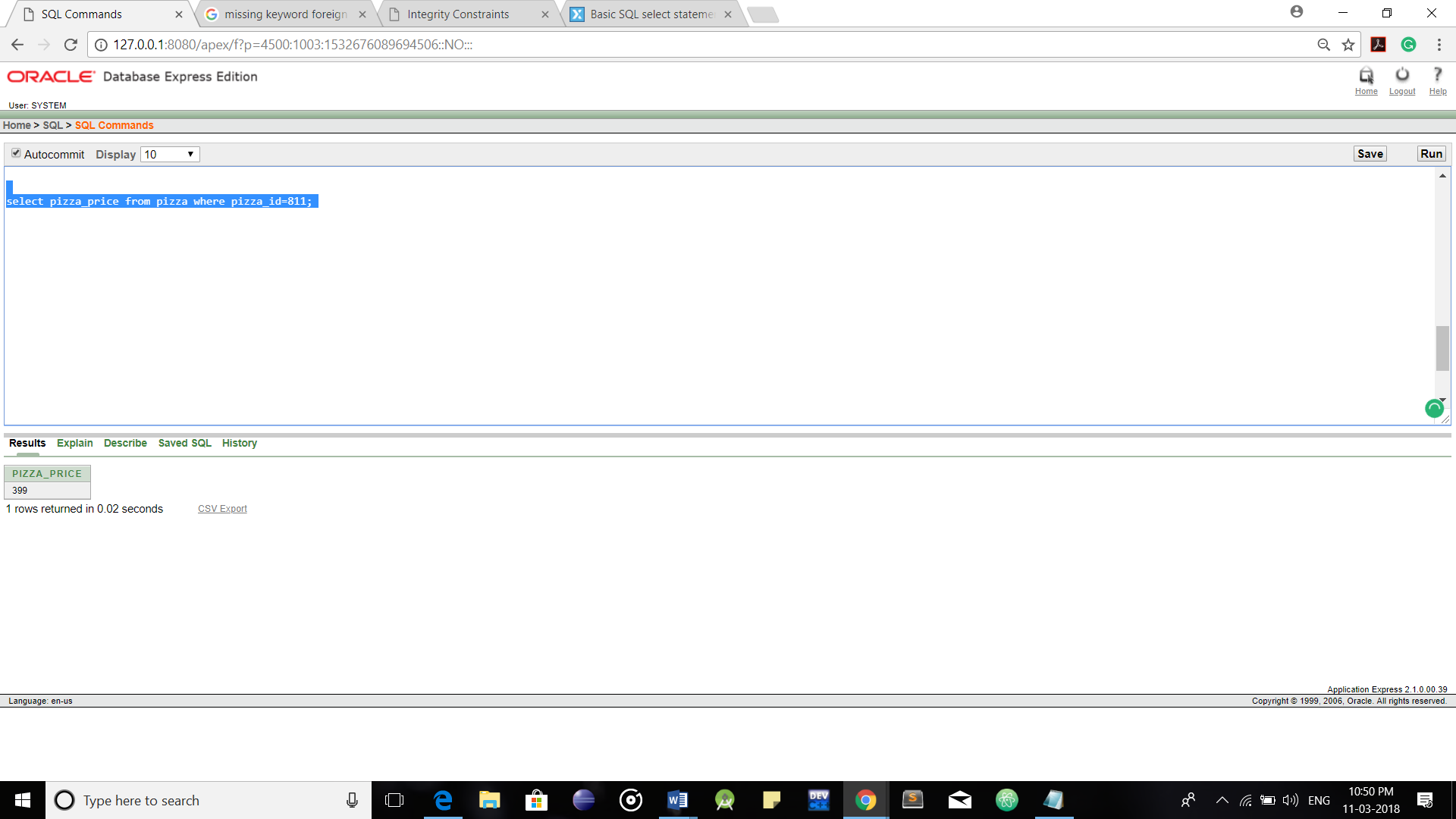
**3.Group by Clause:**



**4.Having Clause:**

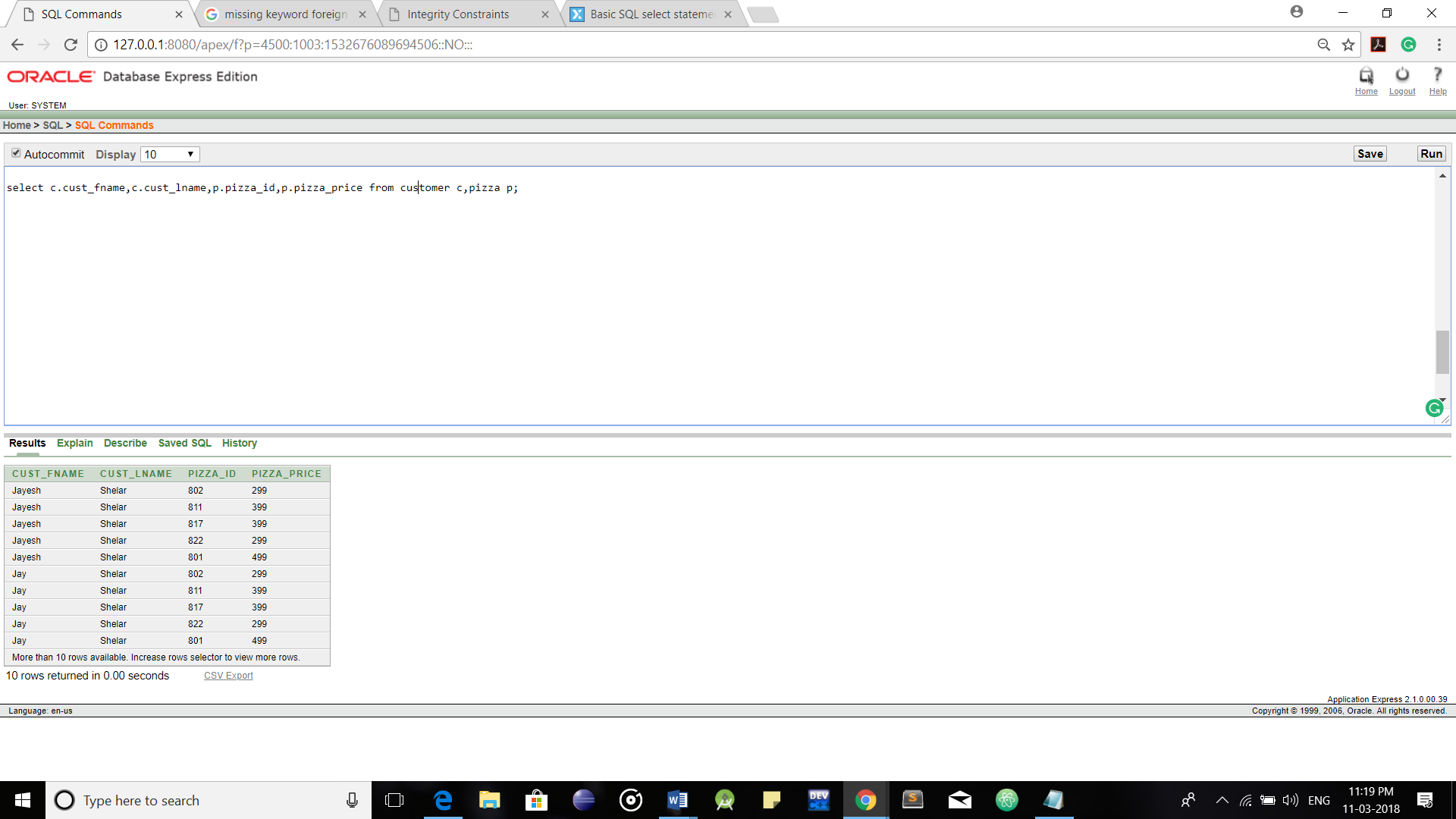


**5.Where Clause:**

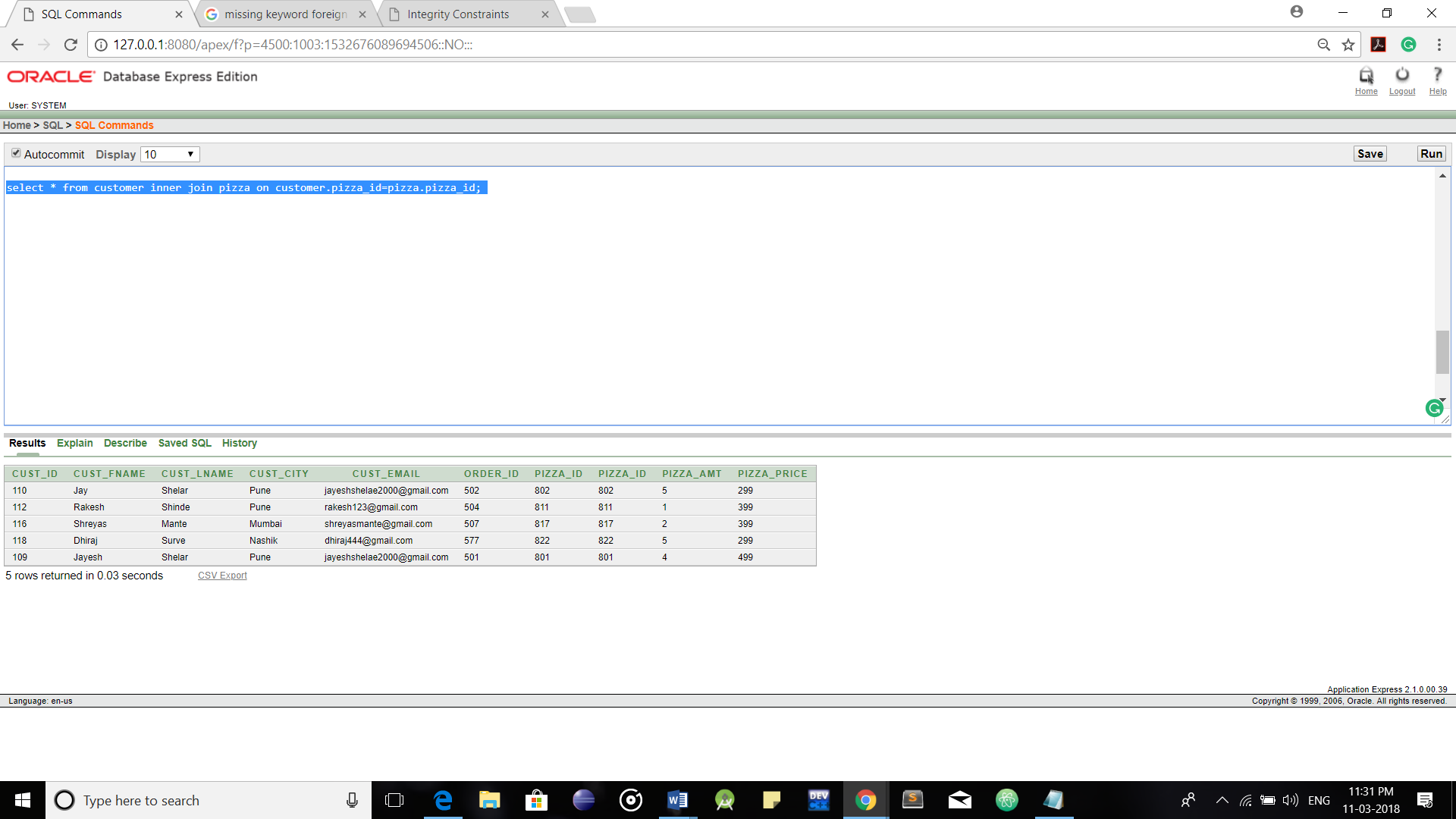


**SQL Joins:**

**1.Cross Join:**



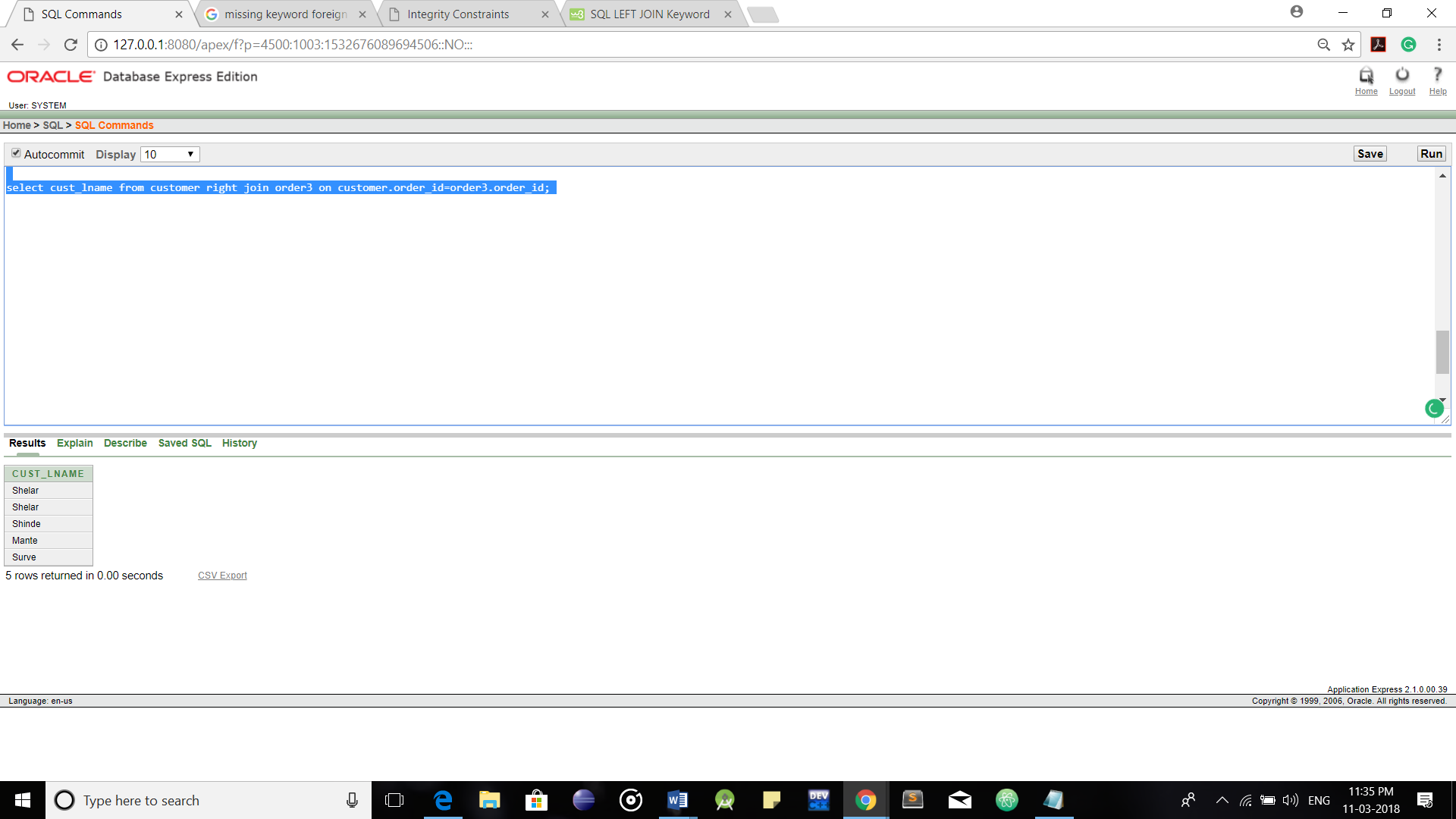
**2.Inner Join:**



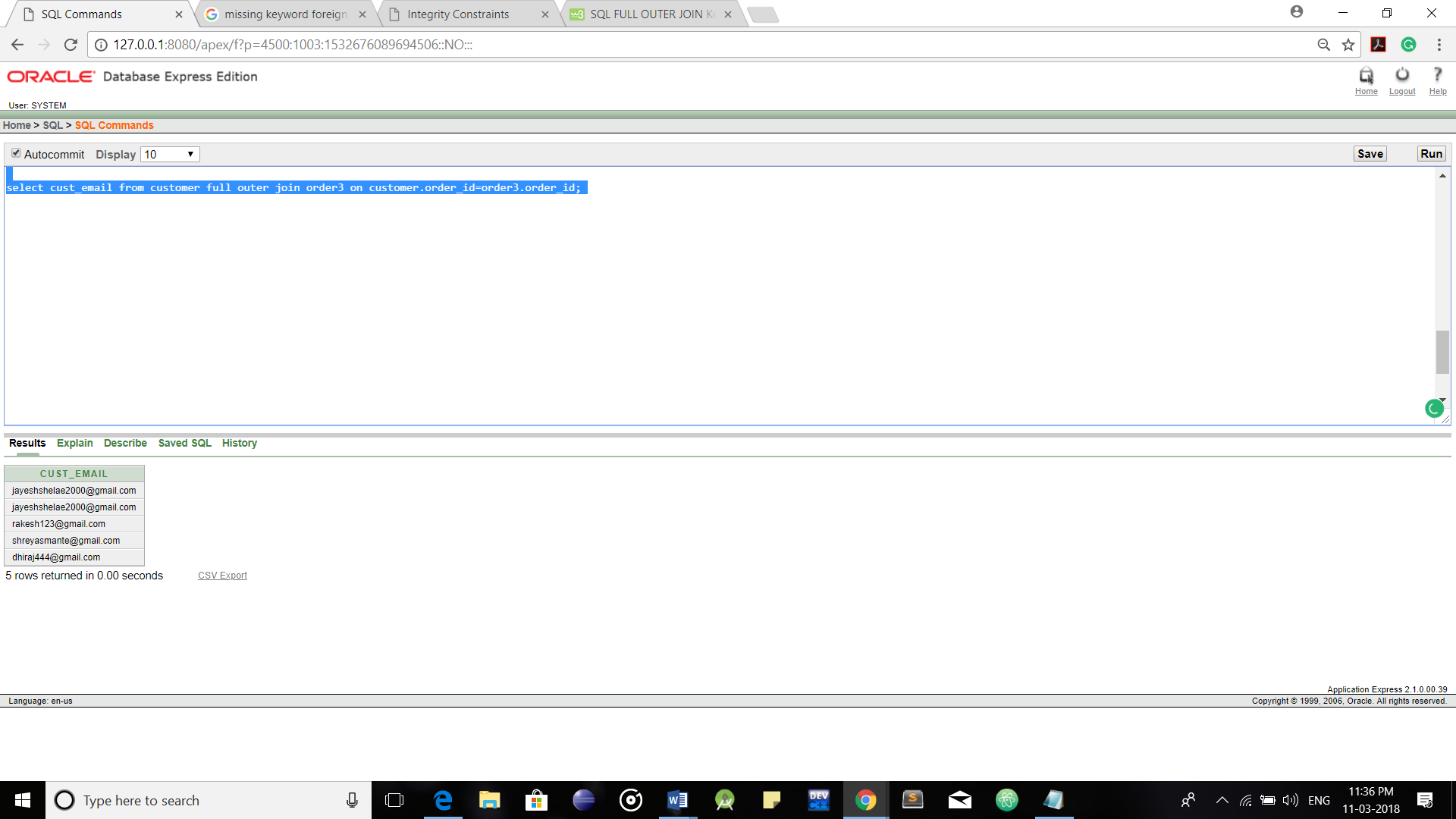
**3.Outer Join(Left):**



**4.Outer Join(Right):**

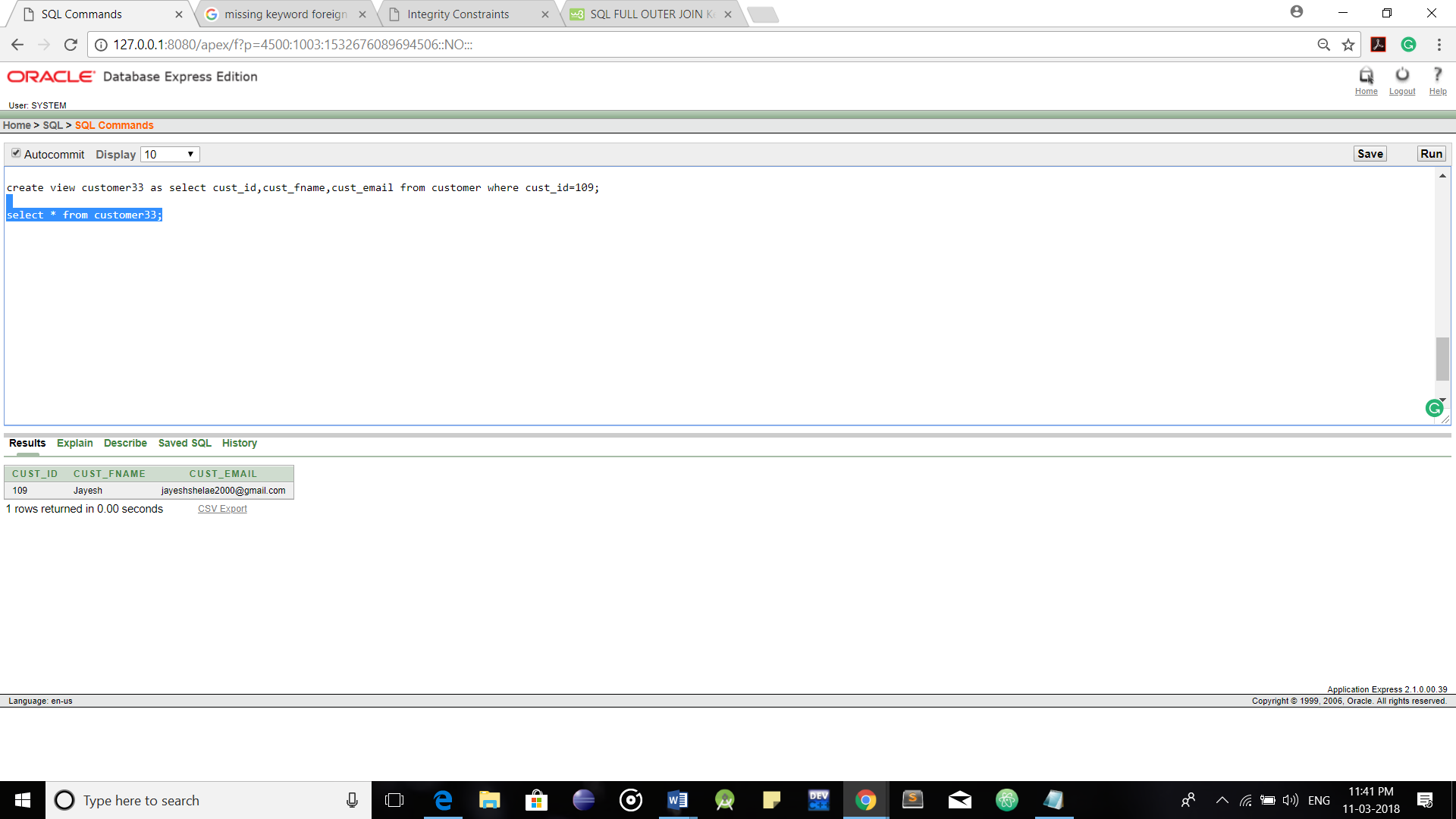


**5.Outer Join(Full):**

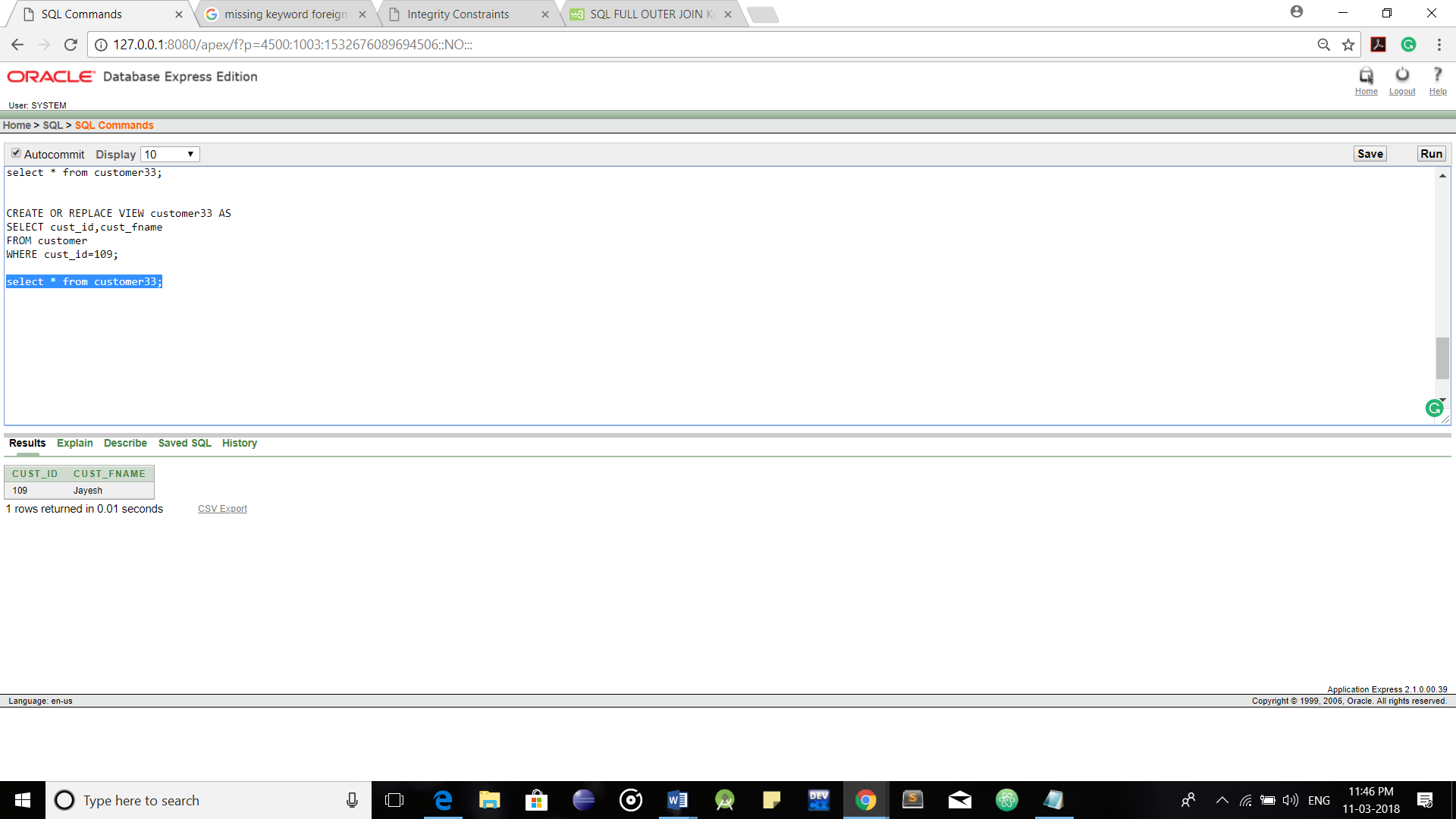


**View:**

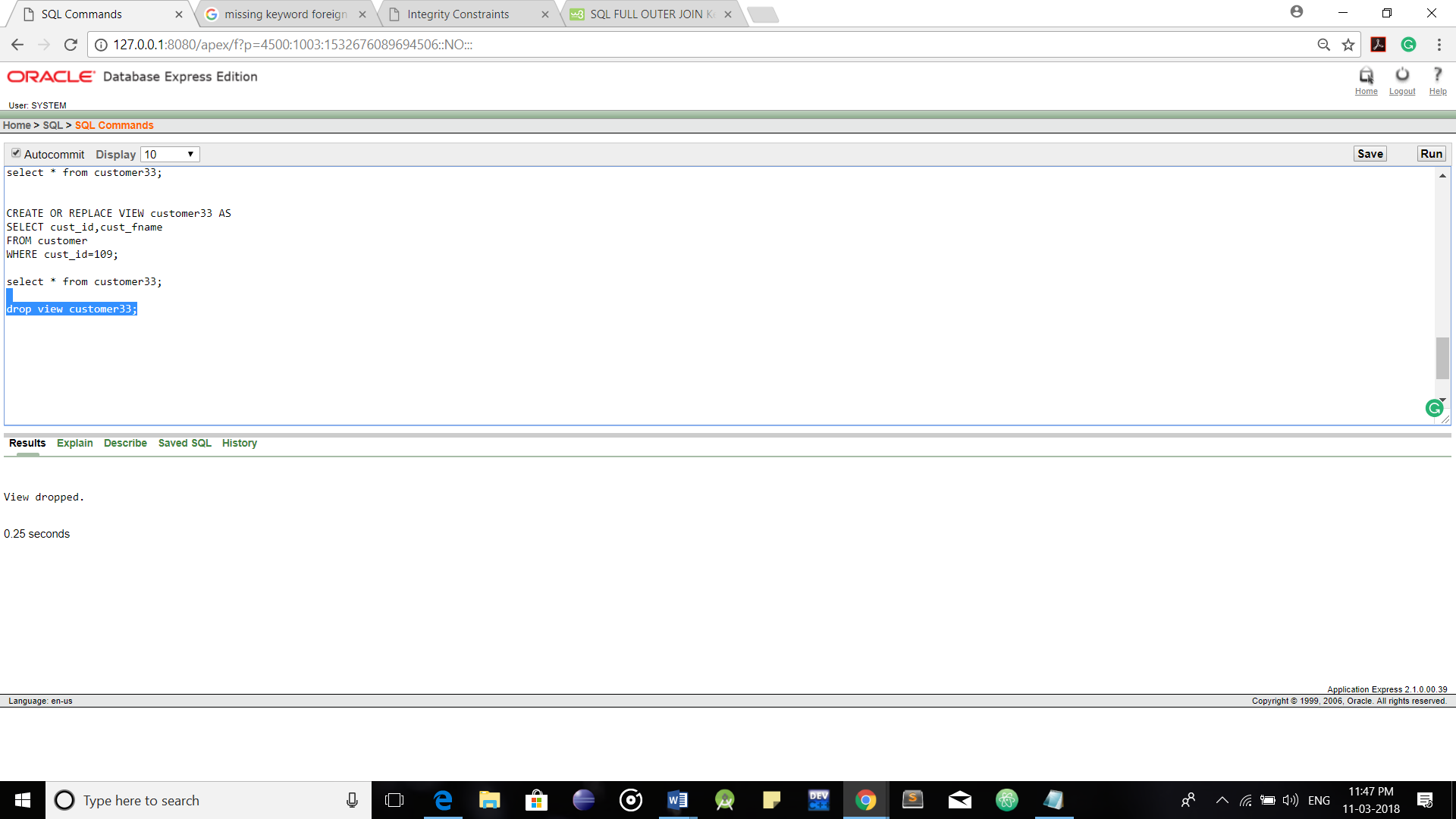
**Creating a View:**



**Updating a View:**

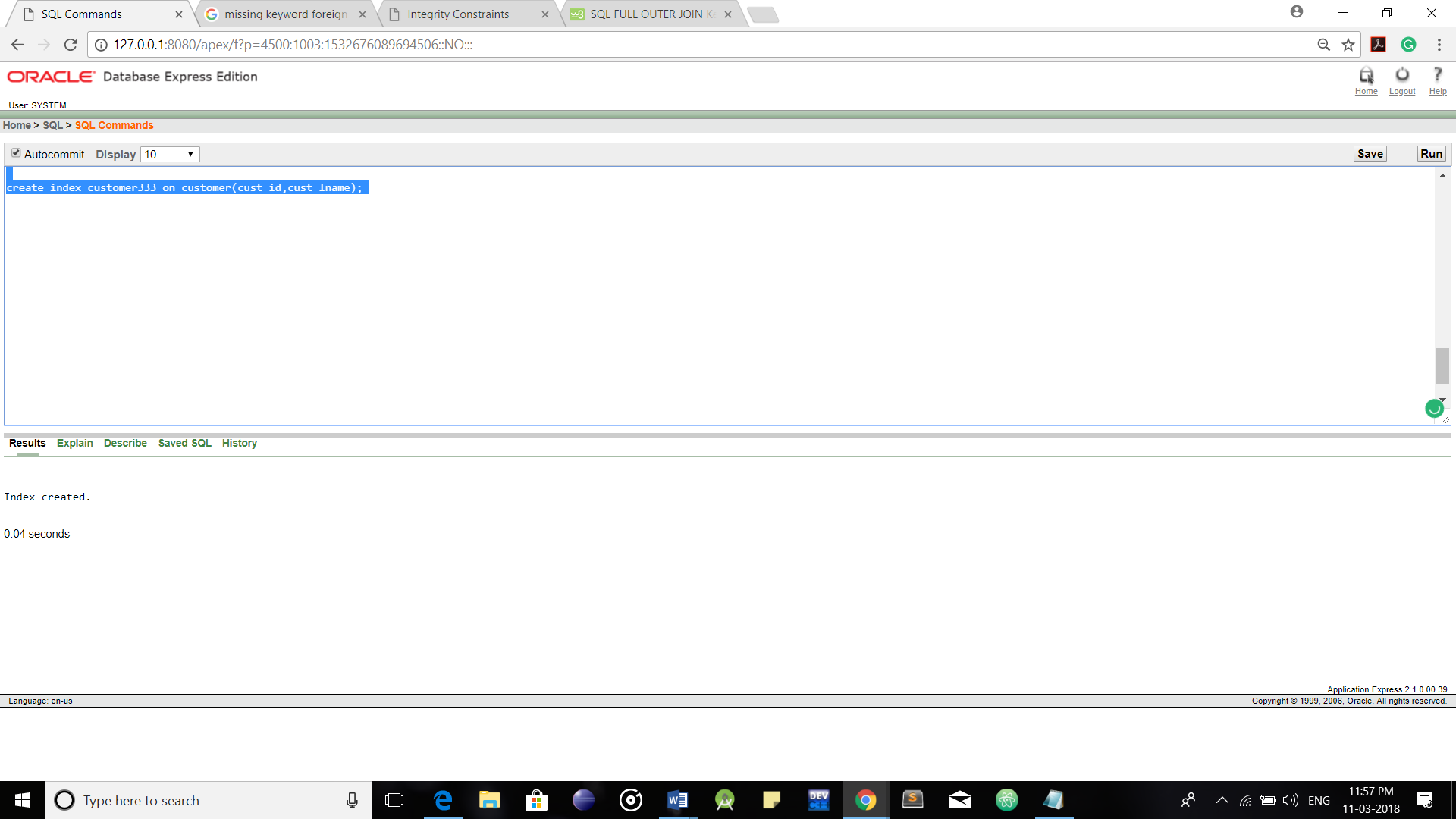


**Dropping a View:**

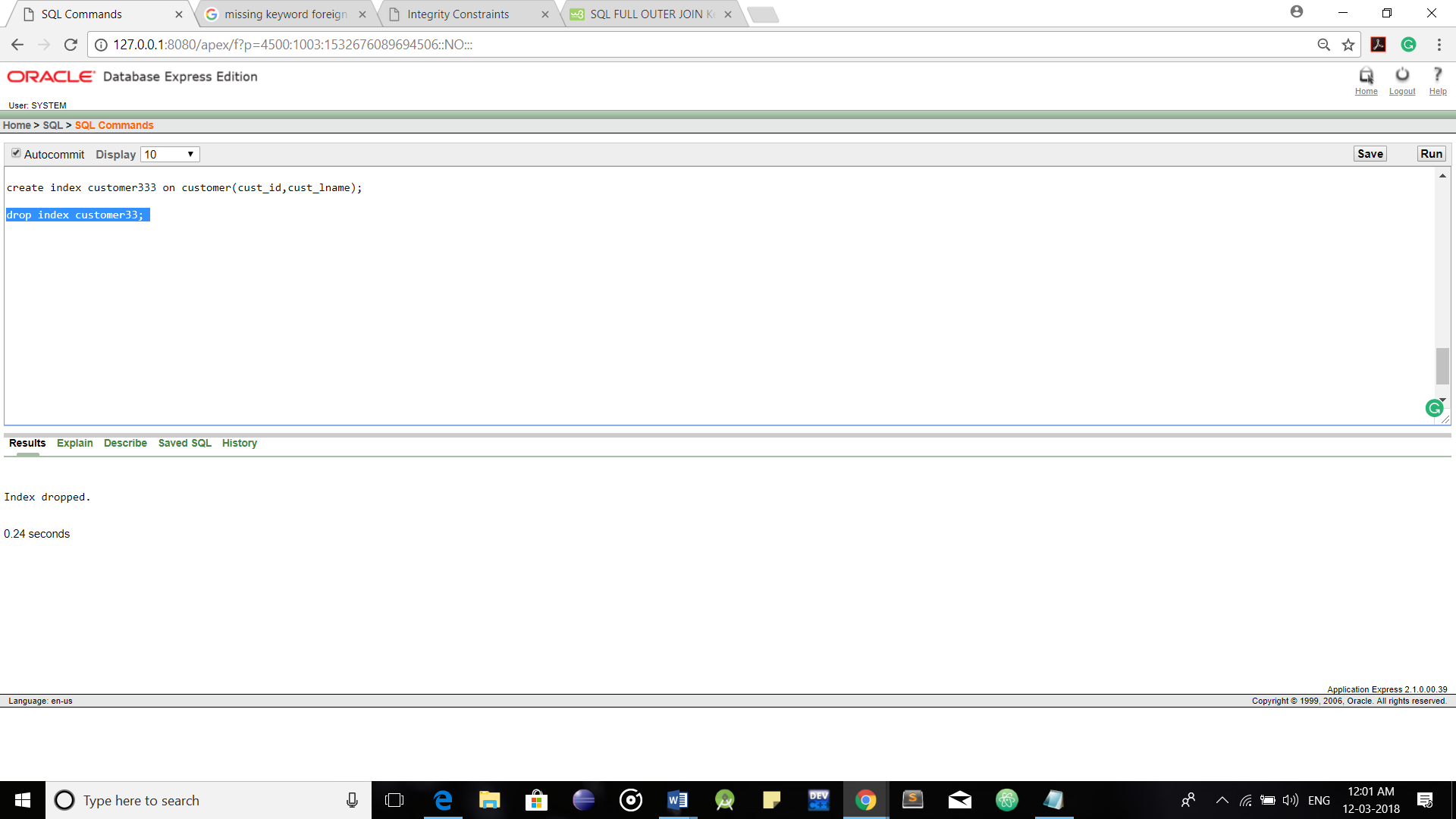


**Index:**

**Creating an Index:**



**Dropping an Index:**



**Sequence:**

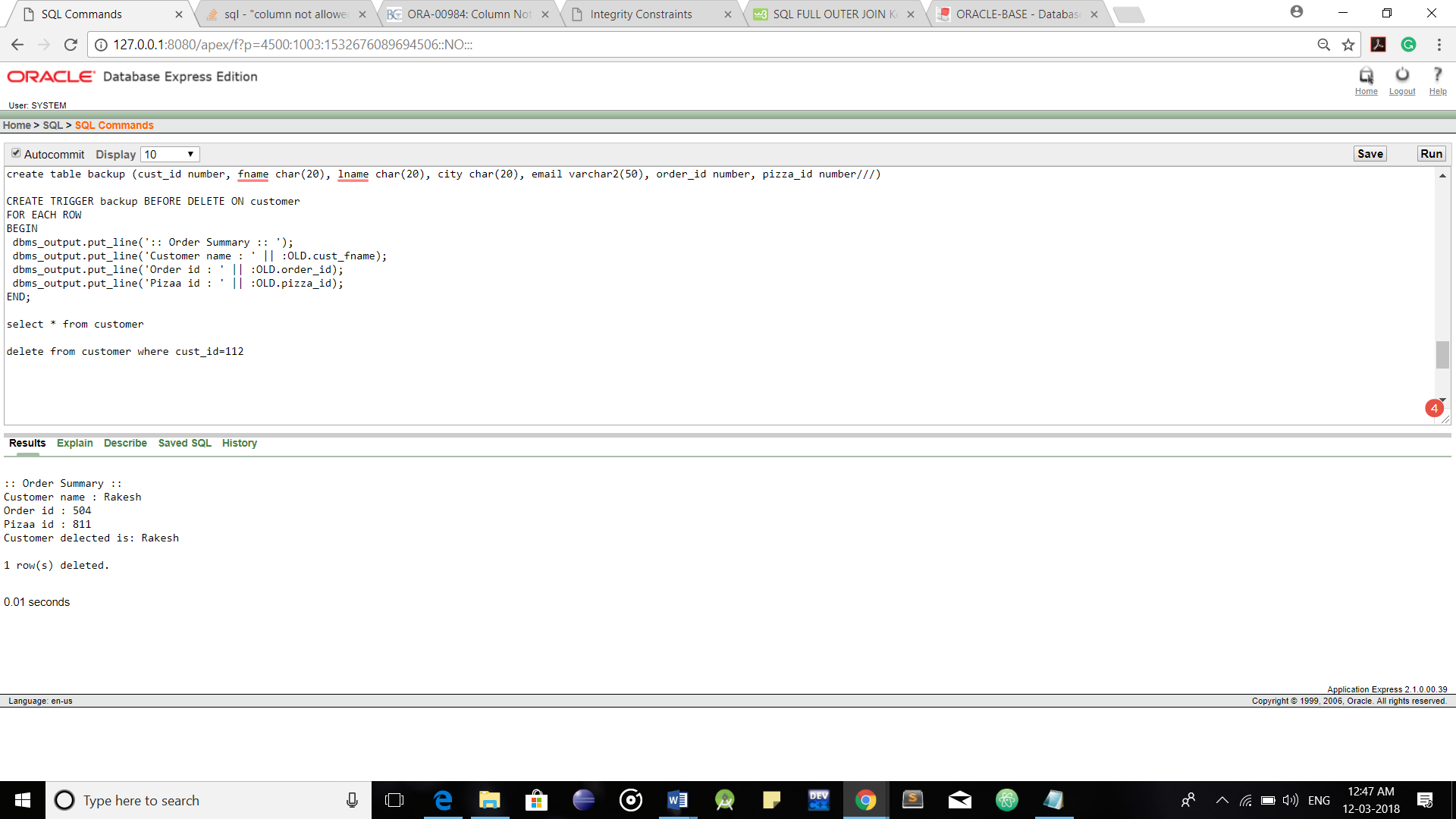
**Creating a Sequence:**



**Trigger:**

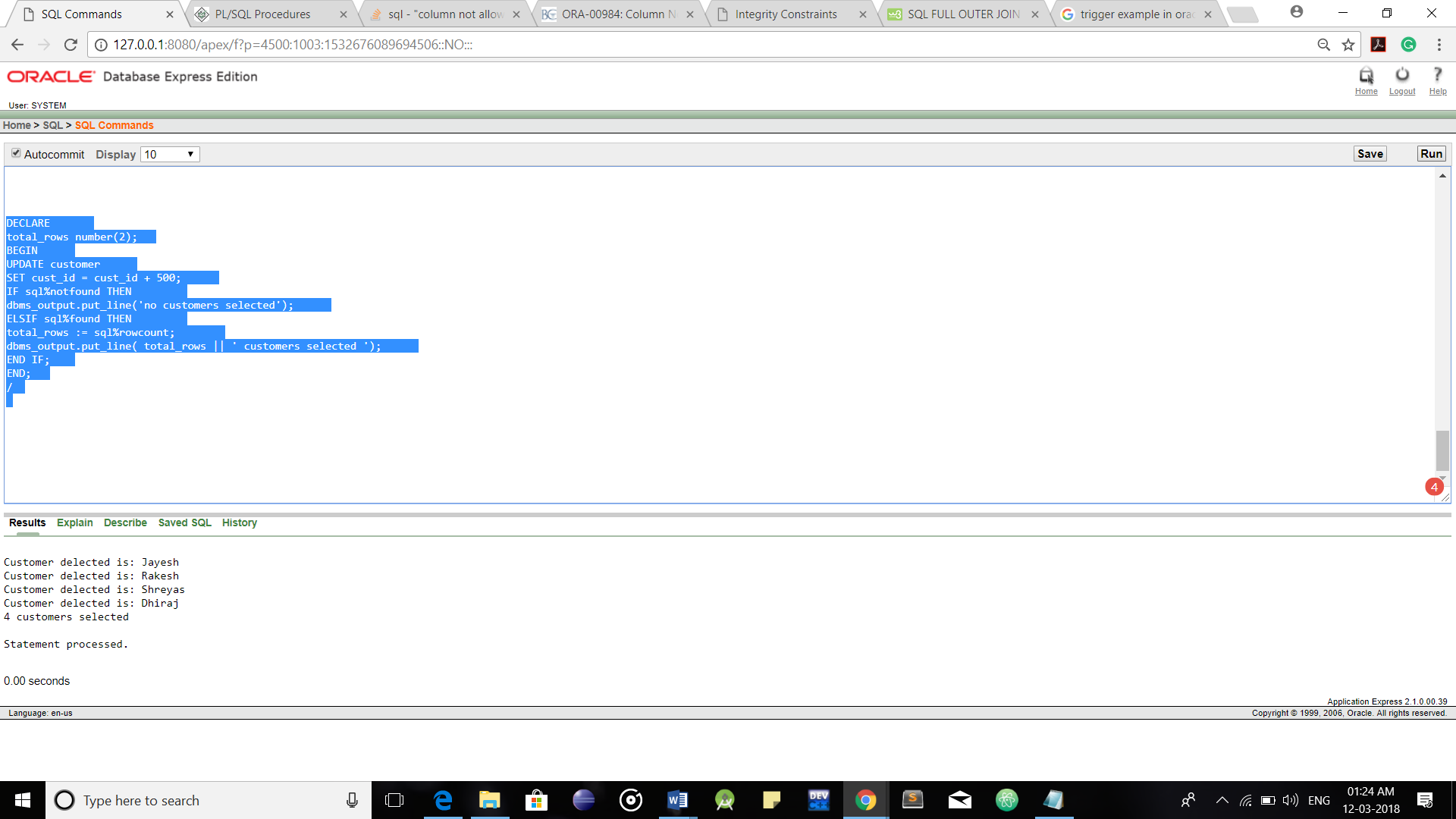
Triggers can be defined on the table, view, schema, or database with which the event is associated.

Here, in the table below, when a deletion statement is encountered, the trigger displays the following message i.e the order details of the customer being deleted.

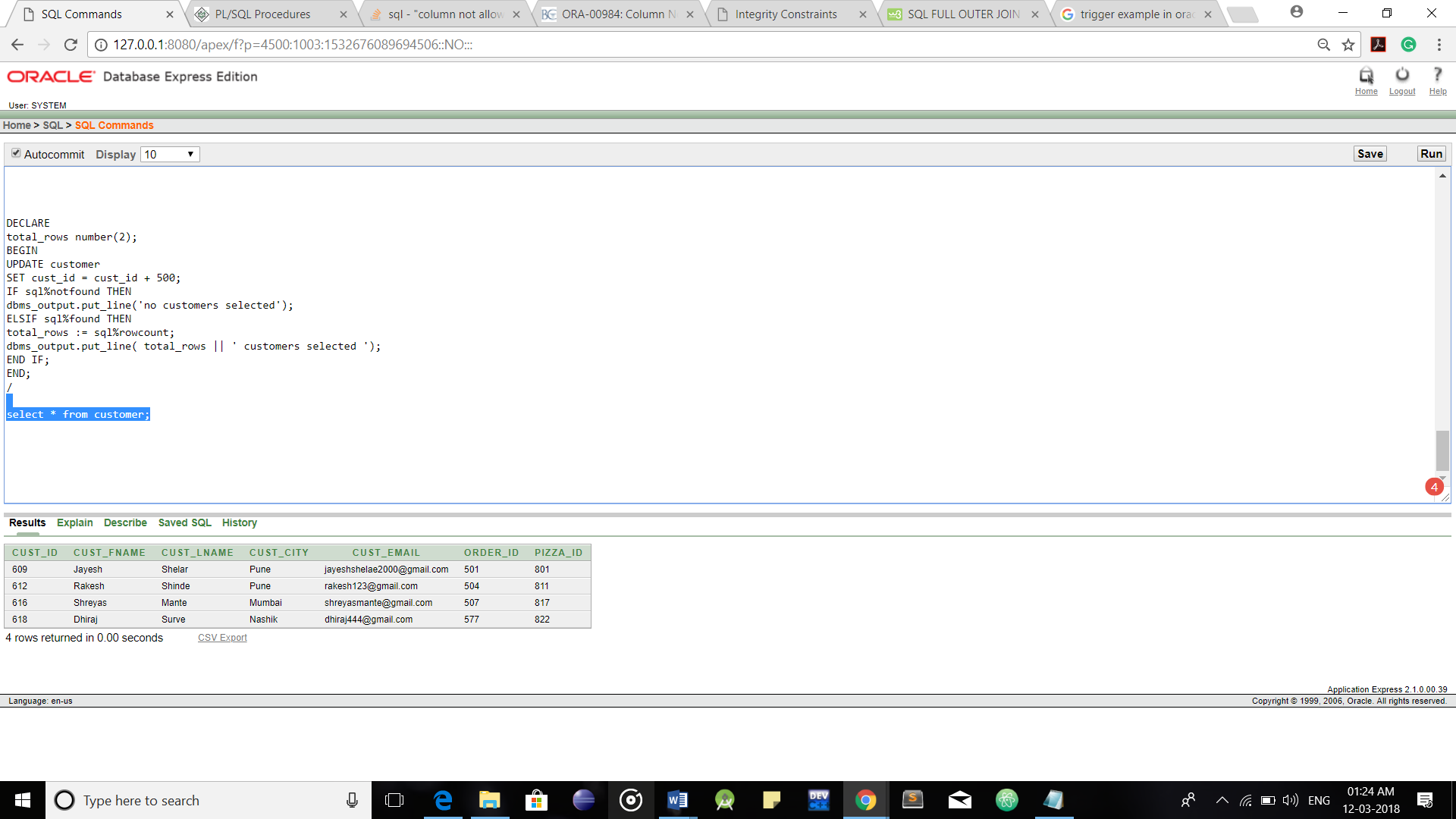


Trigger before a delete operation.

**Cursor and Control Structure:**



Adding 500 to cust\_id if some condition occurs. Cursor, Control Structure are used.



Here you can see that 500 are added to the cust\_id in Customer Table.