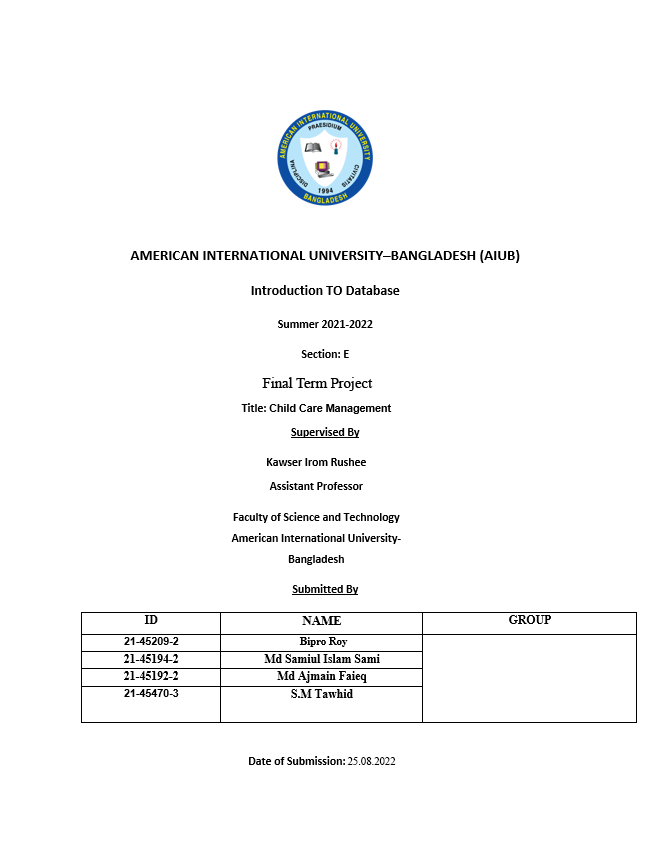
****

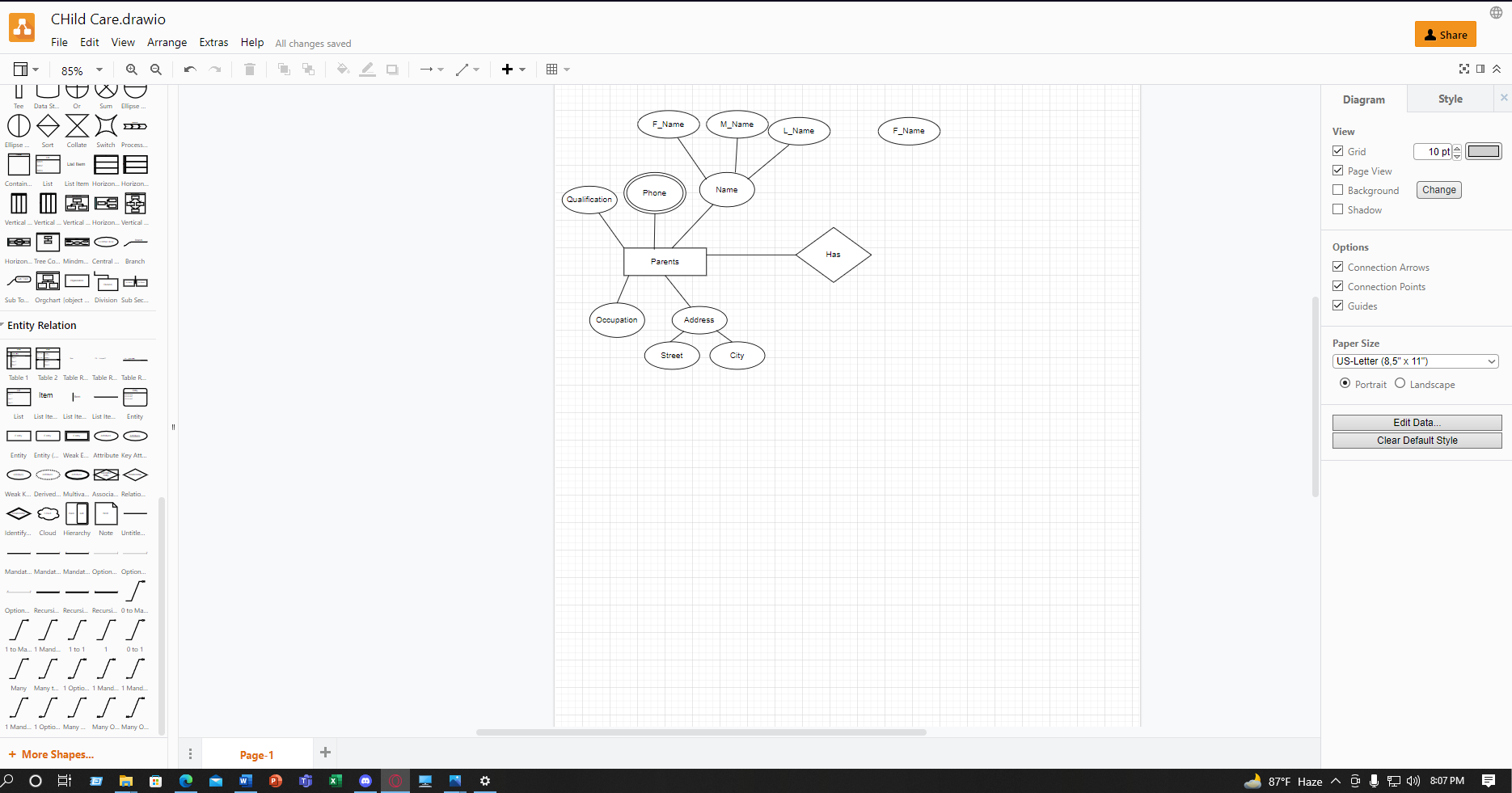
**Child Care Management**

**Case Study:**

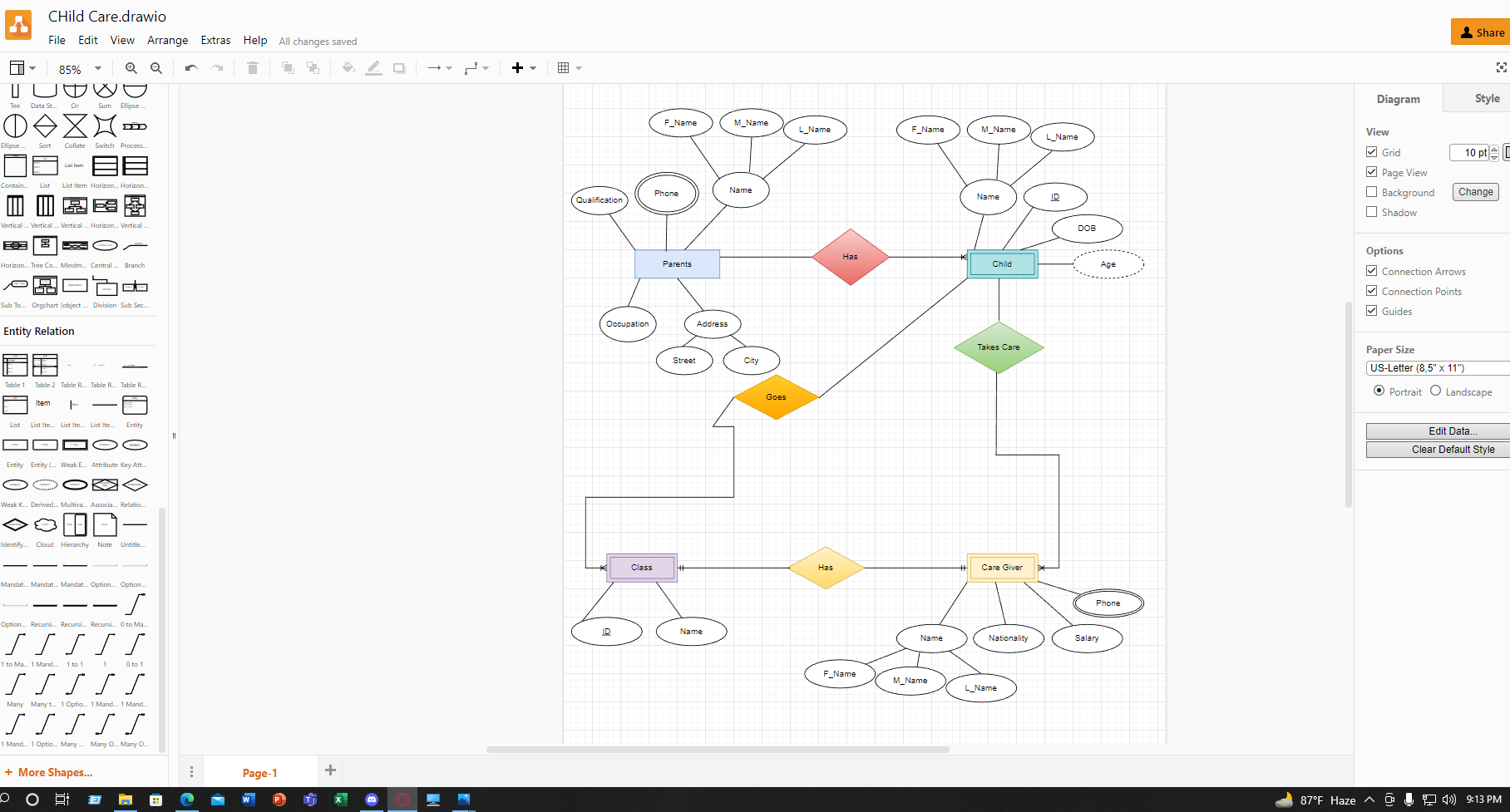
ER diagram represents mainly Four entities: Parents, Child, babysitter, and class. Parents entity contains 5 main attributes P\_Id, P\_name, P\_Phone, Location and occupation. Here P\_Id is the primary key and P\_Phone is multi values which attribute means attribute may have more than one value associated with the key and name. Location attribute consists of sub-attributes L\_ID, City, Division. Parents entity shows one to many relations with child entity means 1 parents can have more than one child or one child also. Child entity consists CH\_name, CH\_ID, Age attributes. Here CH\_ID is the primary key that helps in unique value filtration and joining of two columns. Dimond's shape explained the relationship between two entities. In the above ER Diagram Child entity is connected with two other entities one is with class and another one with a babysitter, with it shows that one or many children go to one class and one babysitter takes care of one or more child. In a Babysitter, the entity consists of four main attributes BS\_ID, BS\_Name, Salary, BS\_Phone. Here BS\_ID is primary key and BS\_Phone is multivalued attribute. In class entity, it includes two attributes one C\_ID and a C\_name here C\_ID is the primary key. Between class and babysitter entity shows one relationship means one class has only one Babysitter.

.

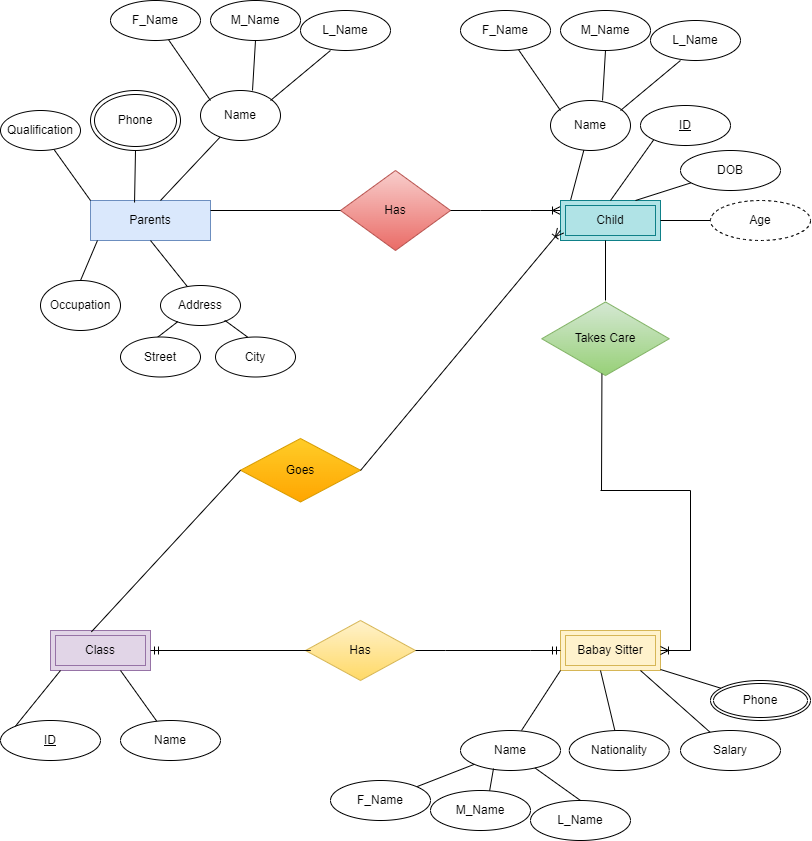
ER DIAGRAM STARTING TIME:



ER DIAGRAM ENDING TIME:



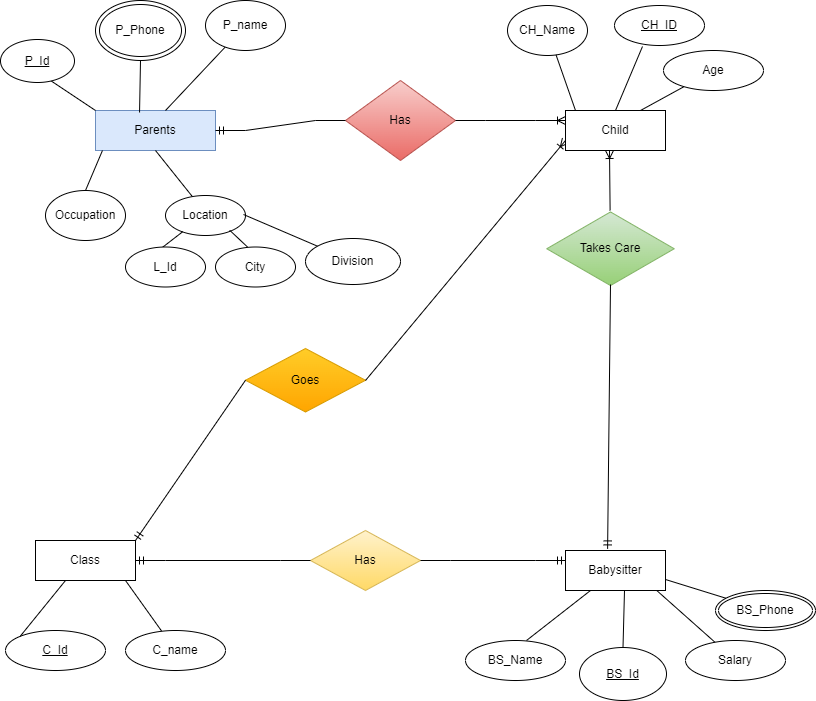
**ER-DIAGRAM OF MID-TERM:**



**Changes of ER-Diagram after Midterm :**

* Changed some Attributes for simplify
* Added Composite Attribute Division and Location Id in Location , Parents Id in Parents Entity , Babysitter Id in Babysitter Entity
* Removed DOB , Qualification, Nationality and Some Composite Attributes
* Defined Clearly Cardinality Relation In ER\_Diagram

**Final ER Diagram**

****

**Normalization**

*Has Relation: ( 1st )*

Parents ( P\_ID, Pname, Occupation, L\_ID, City, Division, P\_Phone)

Child ( CH\_ID, CH\_name, Age)

**1NF:** P\_ID, Pname, Occupation, L\_ID, City, Division, P\_Phone , CH\_ID, CH\_name, Age

**2NF:** 1st: P\_ID, P\_Phone, Pname, Occupation, L\_ID, City, Division

2nd : CH\_ID, CH\_name, age, P\_ID

**3NF:** 1st : P\_ID, P\_Phone, Pname, Occupation, L\_ID

2nd : CH\_ID, CH\_name, age, P\_ID

3rd : L\_ID, City, Division

*Takes Care Relation:*

Child ( CH ID, CH\_name, age)

BabySitter( Bs ID, Bs\_name, Salary, Bs\_Phone)

**1NF:** CH ID, CH\_name, age, Bs ID, Bs\_name, Salary, Bs\_Phone

**2NF:**  1st : CH ID, CH\_name, age, Bs\_ID

2nd : Bs ID, Bs\_Phone , Bs\_name, Salary,

**3NF:** 1st : CH ID, CH\_name, age, Bs\_ID

2nd : Bs ID, Bs\_Phone , Bs\_name, Salary

*Has Relation: (2nd)*

Class ( C ID , C\_name )

BabySitter( Bs ID, Bs\_name, Salary, Bs\_Phone)

**1NF:** C ID , C\_name, Bs ID, Bs\_name, Salary, Bs\_Phone

**2NF :** 1st : C ID , C\_name , Bs\_ID

2nd : Bs ID, Bs\_Phone , Bs\_name, Salary

**3NF:** 1st : C ID , C\_name , Bs\_ID

2nd : Bs ID, Bs\_Phone , Bs\_name, Salary

*Goes Relation:*

Class ( C ID , C\_name )

Child ( CH\_ID, CH\_name, age)

**1NF:** C ID , C\_name , CH\_ID, CH\_name, age

**2NF:** 1st : C ID , C\_name

2nd : CH\_ID, CH\_name, age, C\_ID

**3NF:** 1st : C ID , C\_name

2nd : CH\_ID, CH\_name, age, C\_ID

**FINAL TABLE :**

**Parents:** P\_ID, P\_Phone , Pname, Occupation, L\_ID

**Child\_Parents :** CH\_ID, CH\_name, age, P\_ID

**Location :** L\_ID, City, Division

**BabySitter :** Bs ID, Bs\_Phone, Bs\_name, Salary

**Child\_BabySitter :** CH ID, CH\_name, age, Bs\_ID

**Class :** C ID , C\_name

**Class\_BabySitter** : C ID , C\_name , Bs\_ID

**Child\_Class :** CH\_ID, CH\_name, age, C\_ID

**Table Creation With Constraints**

|  |  |  |
| --- | --- | --- |
| **Parents** | | |
| **Column Name** | **Data type** | **Constraints** |
| P\_Id | Number(5) | Primary Key |
| P\_Phone | Varchar2(11) | Unique |
| P\_Name | Varchar2(30) | Not Null |
| Occupation | Varchar2(21) | Not Null |
| L\_Id | Number (5) | Not Null |

**Create Table Parents (P\_Id Number(5) CONSTRAINT pidpk PRIMARY KEY,**

**P\_Name VARCHAR2(30) CONSTRAINT pnmcons NOT NULL,**

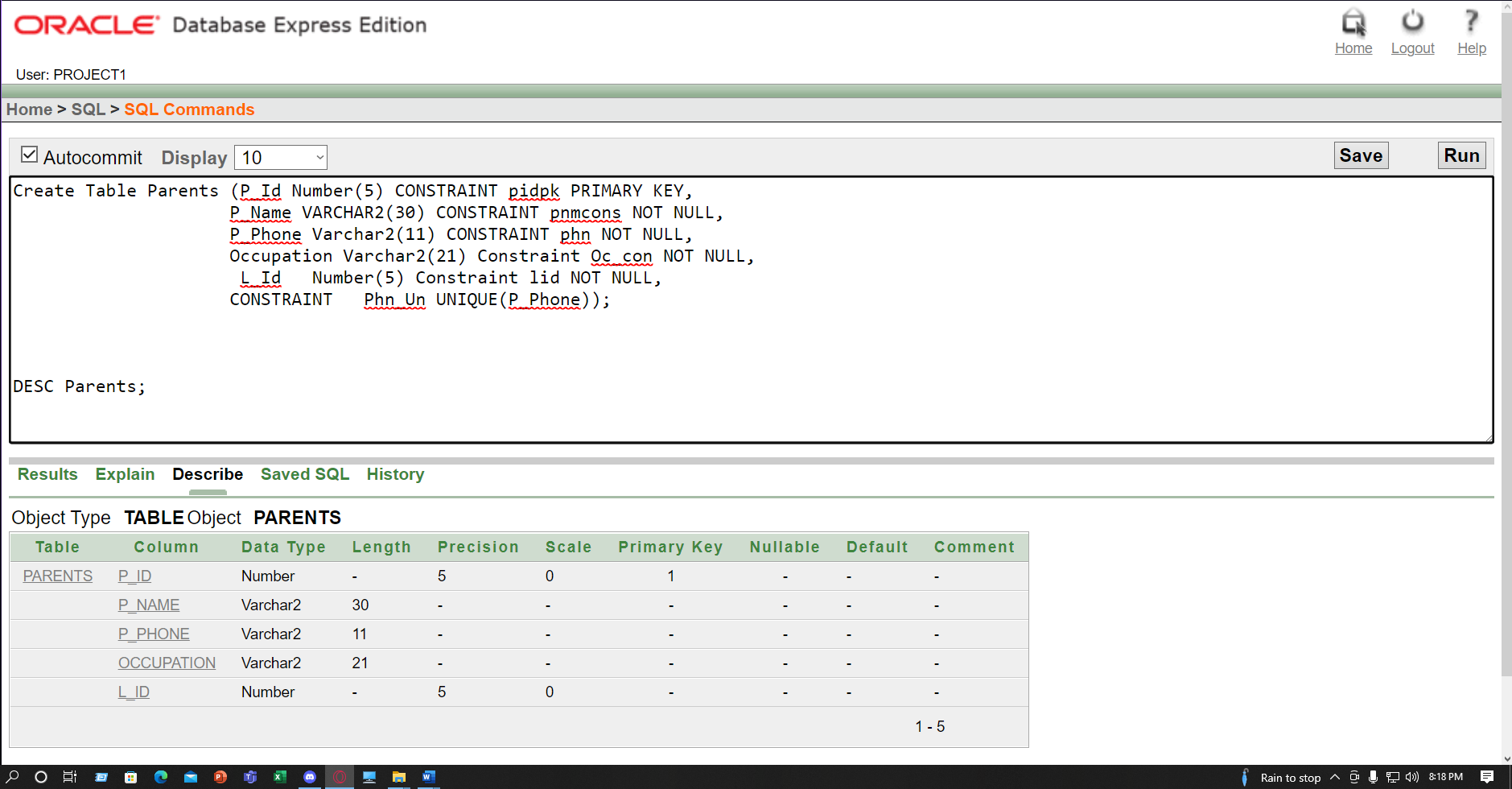
**P\_Phone Varchar2(11) CONSTRAINT phn NOT NULL,**

**Occupation Varchar2(21) Constraint Oc\_con NOT NULL,**

**L\_Id Number(5) Constraint lid NOT NULL,**

**CONSTRAINT Phn\_Un UNIQUE(P\_Phone));**

**DESC Parents;**

****

|  |  |  |
| --- | --- | --- |
| **Child\_Parents** | | |
| **ColumnName** | **Data type** | **Constraints** |
| CH\_Id | Number(5) | Primary Key |
| CH\_Name | Varchar2(32) | NOT NULL |
| Age | Number(2) | NOT NULL |
| P\_Id | Number(5) | Foreign Key |

**Create Table Child\_Parents (CH\_ID Number(5) CONSTRAINT chpk1 PRIMARY KEY, CH\_Name Varchar2(32) Constraint chn2 NOT NULL,Age Number(2) Constraint agcon1 NOT NULL , P\_Id number (5),**

**Constraint amm2 Foreign Key (P\_Id) REFERENCES Parents(P\_Id) );**

**DESC Child\_Parents;**

****

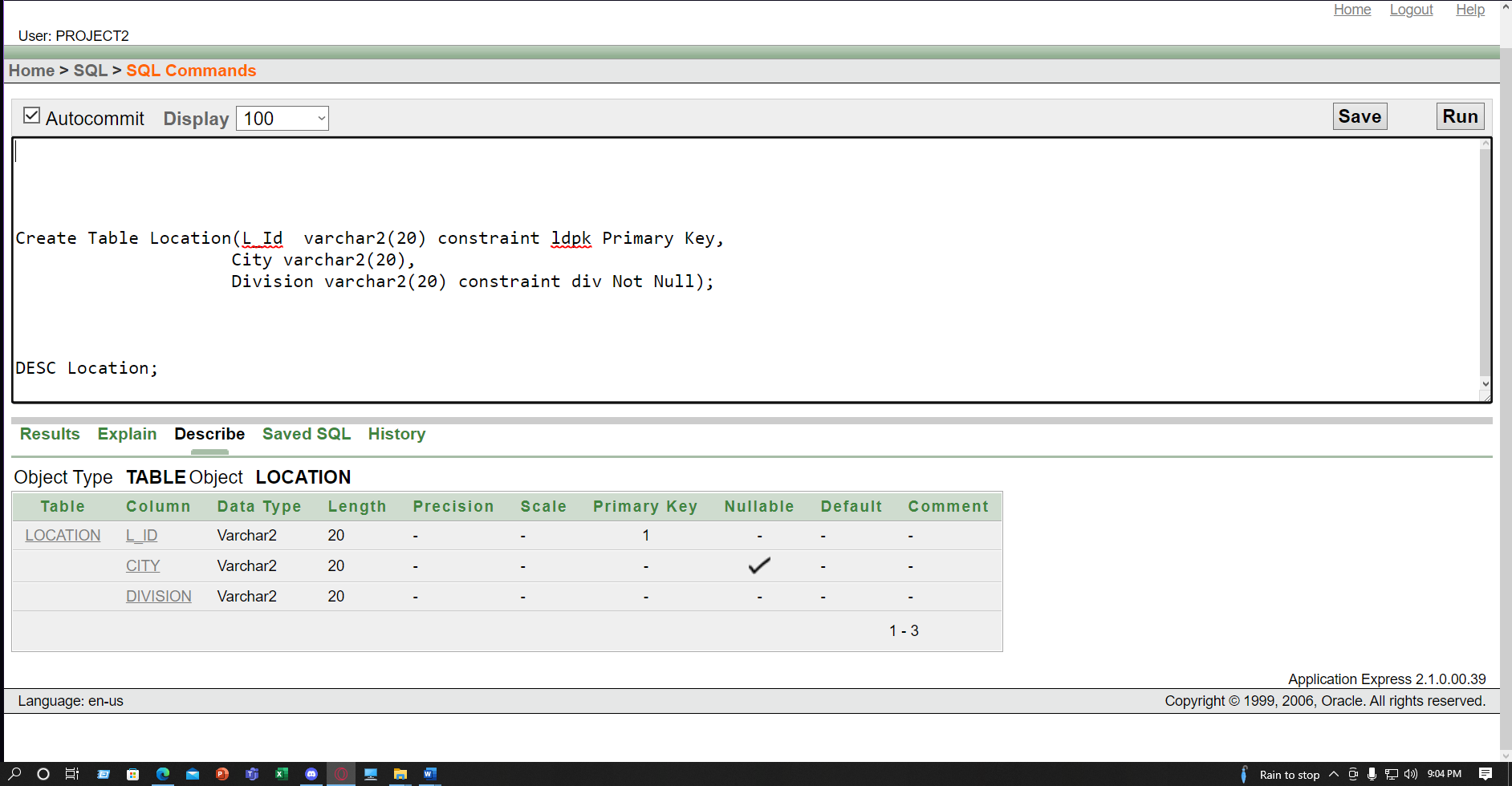
|  |  |  |
| --- | --- | --- |
| **Location** | | |
| **Column Name** | **Data type** | **Constraints** |
| L\_Id | Varchar2(20) | Primary Key |
| City | Varchar2(20) |  |
| Division | Varchar2(20) | Not Null |

**Create Table Location(L\_Id varchar2(20) constraint ldpk Primary Key,**

**City varchar2(20),**

**Division varchar2(20) constraint div Not Null);**

**DESC Location;**



|  |  |  |
| --- | --- | --- |
| **Babysitter** | | |
| Column **Name** | **Data type** | **Constraints** |
| BS\_Id | Number (5) | Primary Key |
| BS\_Phone | Varchar2(11) | Unique |
| BS\_Name | Varchar2(30) | Not Null |
| Salary | Number(7) | Not Null |

**Create Table Babysitter( BS\_Id number(5) constraint bspk PRIMARY KEY,**

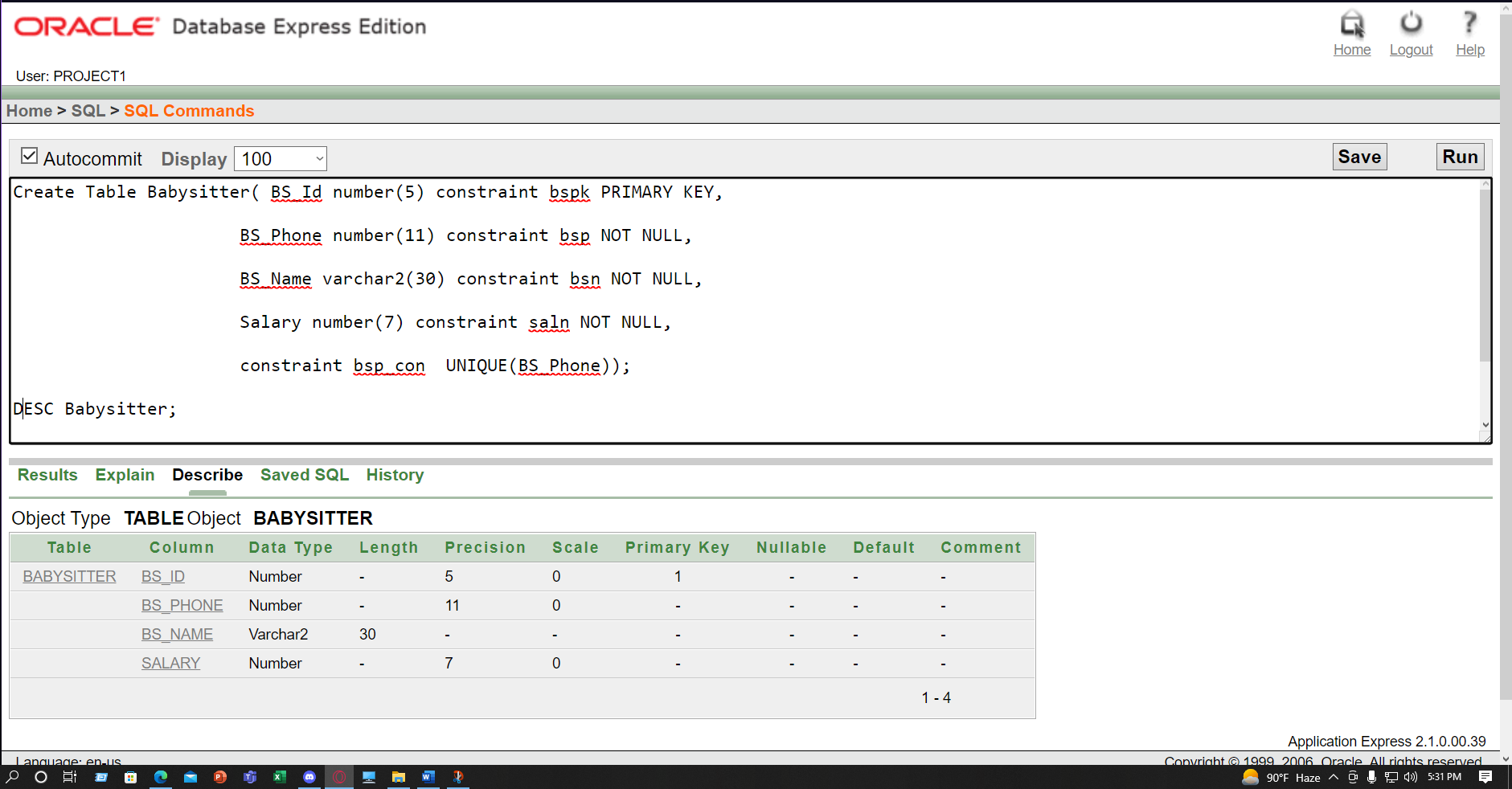
**BS\_Phone number(11) constraint bsp NOT NULL,**

**BS\_Name varchar2(30) constraint bsn NOT NULL,**

**Salary number(7) constraint saln NOT NULL,**

**constraint bsp\_con UNIQUE(BS\_Phone) );**

**DESC Babysitter;**

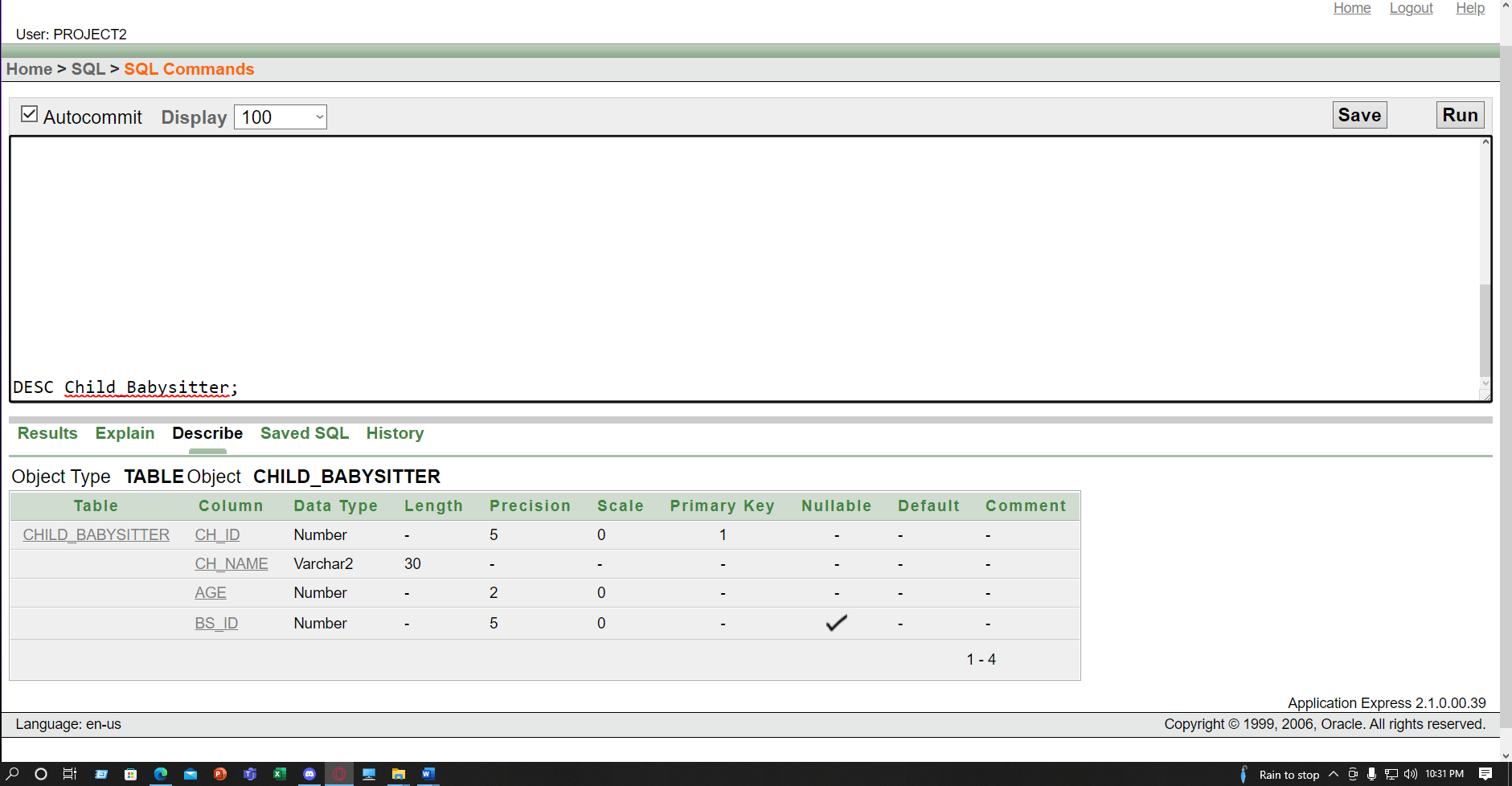
****

|  |  |  |
| --- | --- | --- |
| **Child\_Babysitter** | | |
| Column **Name** | **Data type** | **Constraints** |
| CH\_ID | Number(5) | Primary Key |
| CH\_Name | Varchar2(30) | Not Null |
| Age | Varchar2(2) | Not Null |
| BS\_Id | Number (5) | Foreign Key |

**Create Table Child\_Babysitter(CH\_ID number(5) constraint chpk22 Primary Key,**

**CH\_Name Varchar2(30) constraint chn21 Not Null,Age number(2) constraint ag212 Not Null ,Bs\_Id number (5), Constraint anm22 Foreign Key (BS\_Id) REFERENCES Babysitter(BS\_Id) );**

**DESC Child\_Babysitter;**

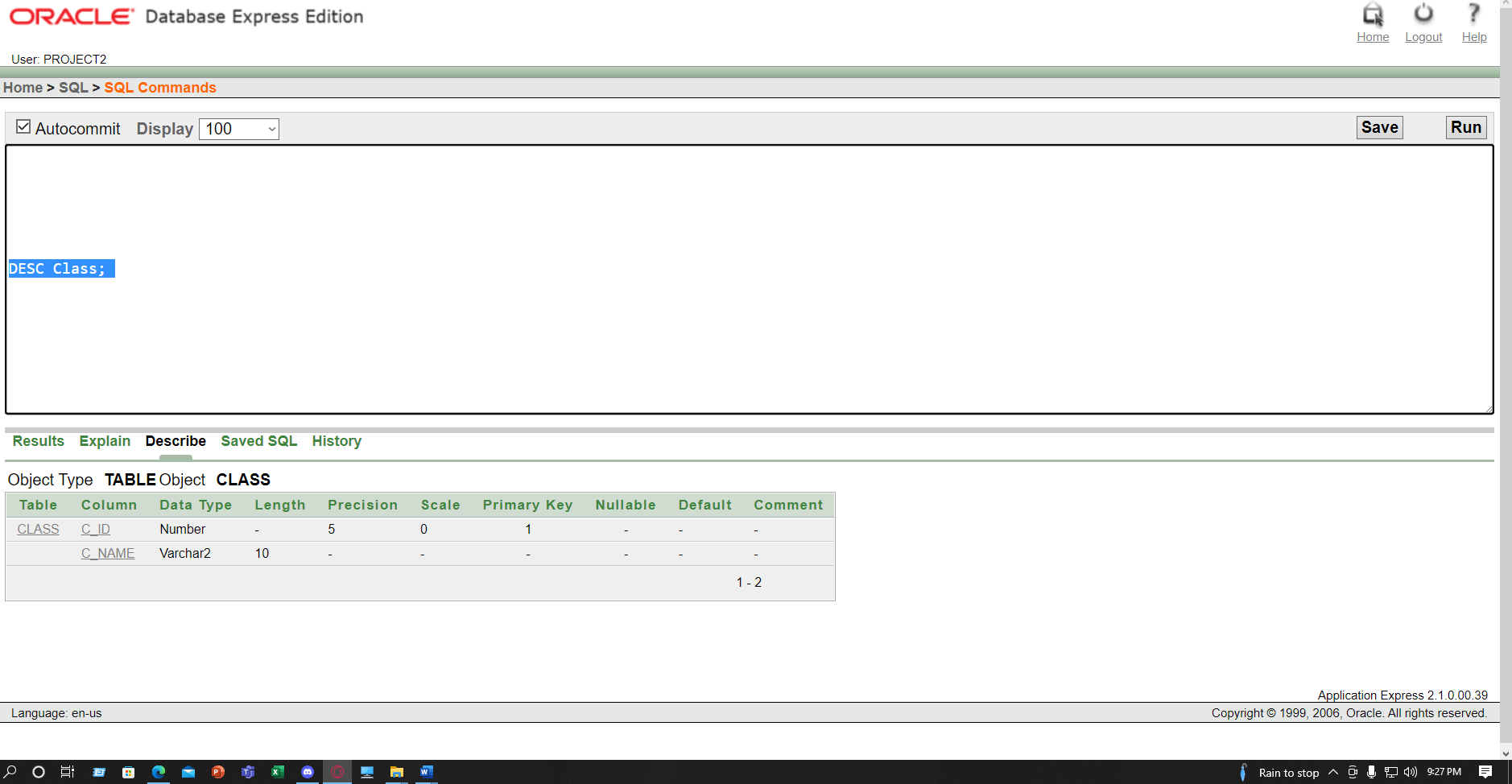
****

|  |  |  |
| --- | --- | --- |
| **Class** | | |
| Column **Name** | **Data type** | **Constraints** |
| C\_Id | Number(5) | Primary Key |
| C\_Name | Varchar2(10) | Not NULL |

**Create Table Class( C\_Id number(5) constraint clpk PRIMARY KEY,**

**C\_Name varchar(10) constraint cln NOT NULL );**

**DESC Class;**

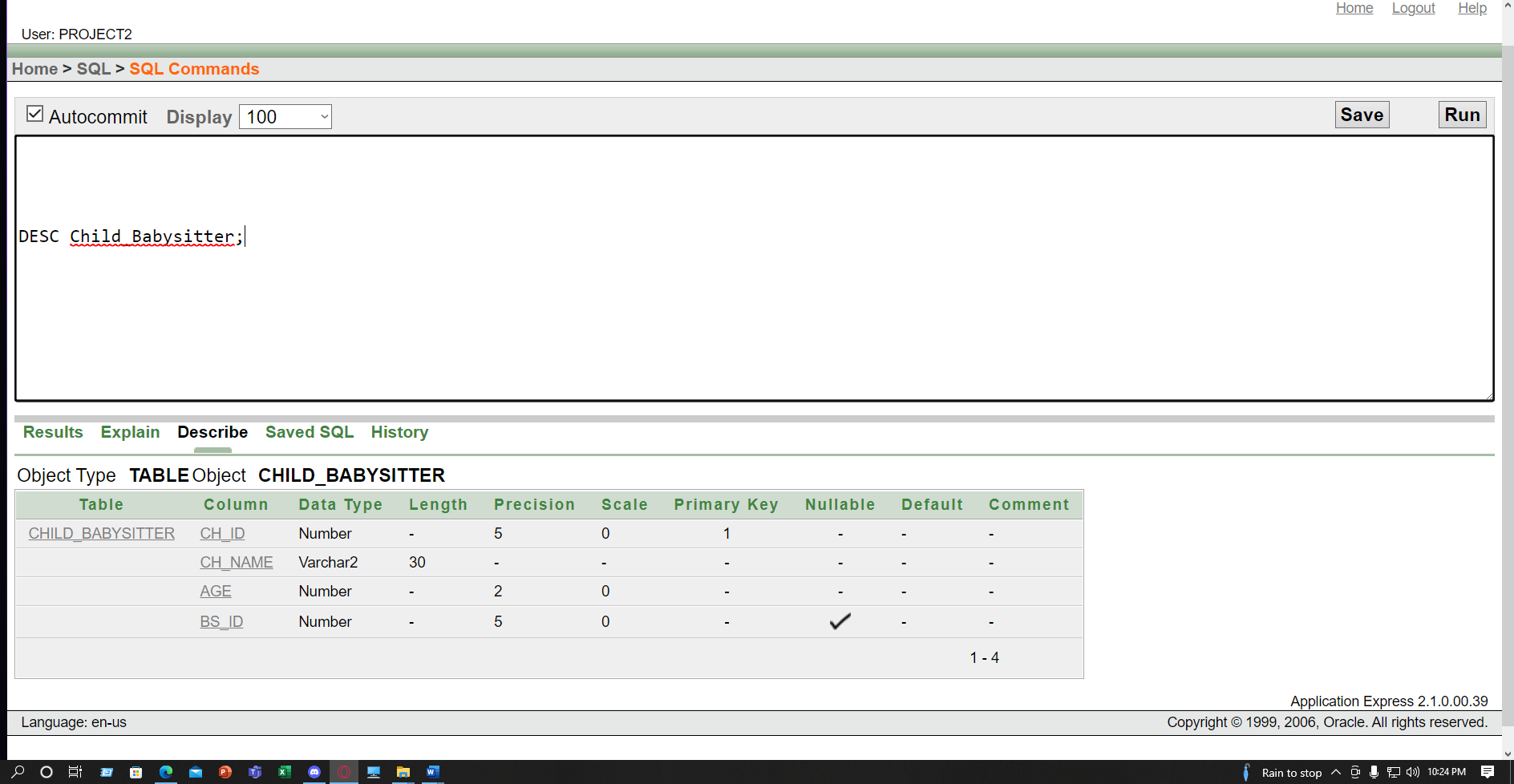
****

|  |  |  |
| --- | --- | --- |
| **Class\_Babysitter** | | |
| Column **Name** | **Data type** | **Constraints** |
| C\_Id | Number(5) | Primary Key |
| C\_Name | Varchar2(10) | Not Null |
| BS\_Id | Number(5) | Foreign Key |

**Create Table Class\_Babysitter( C\_Id number(5) constraint cdi11 Primary Key, C\_Name varchar2(10) constraint cnam22 Not Null ,BS\_Id number(5),**

**Constraint bsid20 Foreign Key(BS\_Id) REFERENCES Babysitter(Bs\_Id) );**

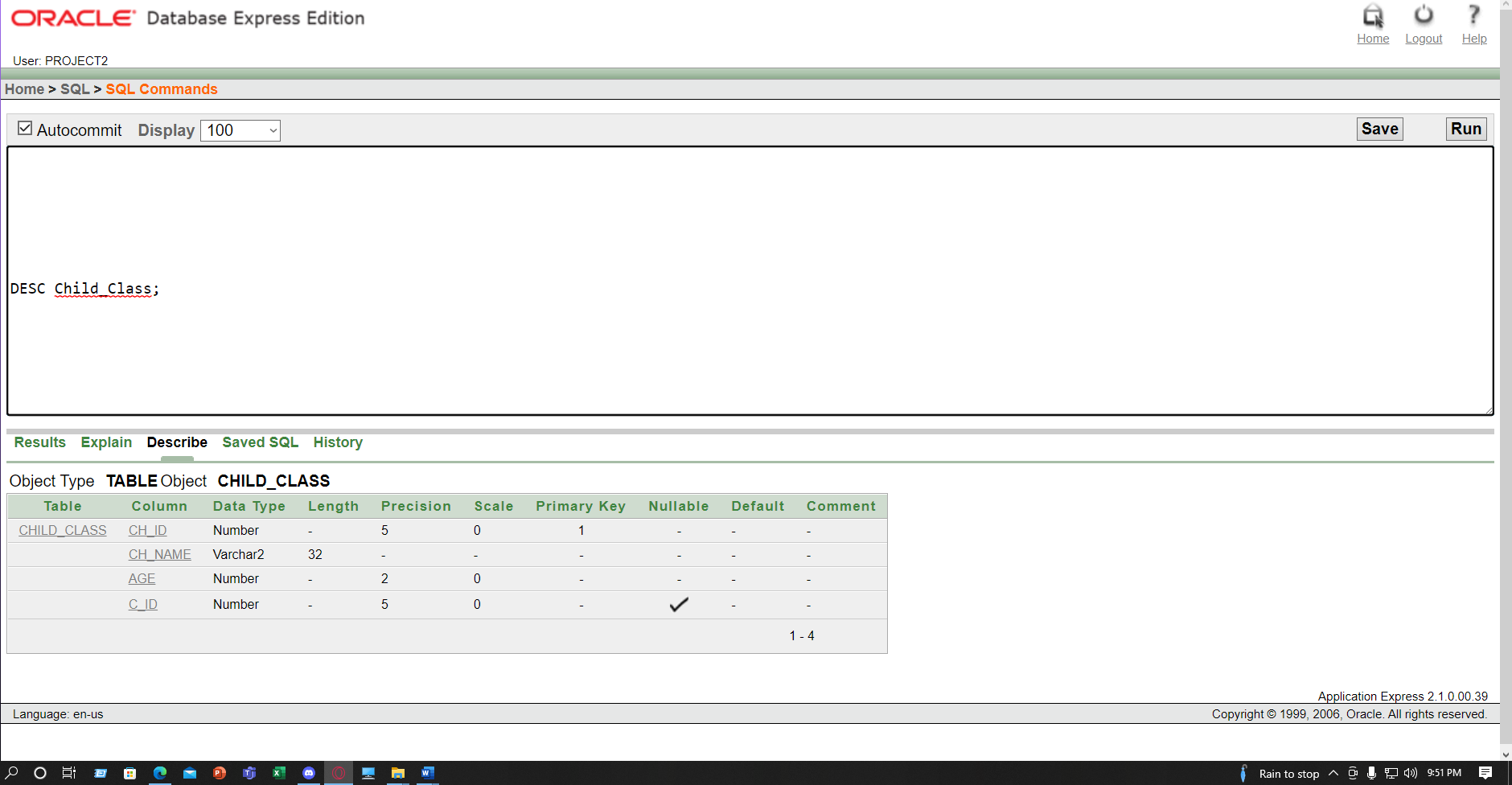
**DESC Class\_Babysitter ;**

****

|  |  |  |
| --- | --- | --- |
| **Child\_Class** | | |
| Column **Name** | **Data type** | **Constraints** |
| CH\_Id | Number(5) | Primary Key |
| CH\_Name | Varchar2(32) | Not Null |
| Age | Number (2) | Not Null |
| C\_Id | Number(5) | Foreign Key |

**Create Table Child\_Class ( CH\_ID number(5) constraint chd32 primary key , CH\_Name varchar2(32) Constraint chnm21 Not Null,Age number(2) Constraint ag32 Not Null,C\_Id Number (5), Constraint cd232 Foreign Key(C\_Id) REFERENCES Class (C\_Id));**

**DESC Child\_Class;**

****

**Data Insertion:**

Table**: Parents**

INSERT INTO Parents VALUES (21451,'Jason Roy','01688354400','Businessman' ,12135 );

INSERT INTO Parents VALUES (21452,'Peter Parkur', '01688354411','Journalist' ,12146);

INSERT INTO Parents VALUES (21453, 'Marry Jen', '01688354422','Teacher' ,12156 );

INSERT INTO Parents VALUES (21454,'Osborn Harry','01688354433','Engineer' , 12167);

INSERT INTO Parents VALUES (21455, 'Octavius', '01688354444','Doctor' ,12178 );

INSERT INTO Parents VALUES (21456,'Frankestine','01688354466','Engineer' ,12198);

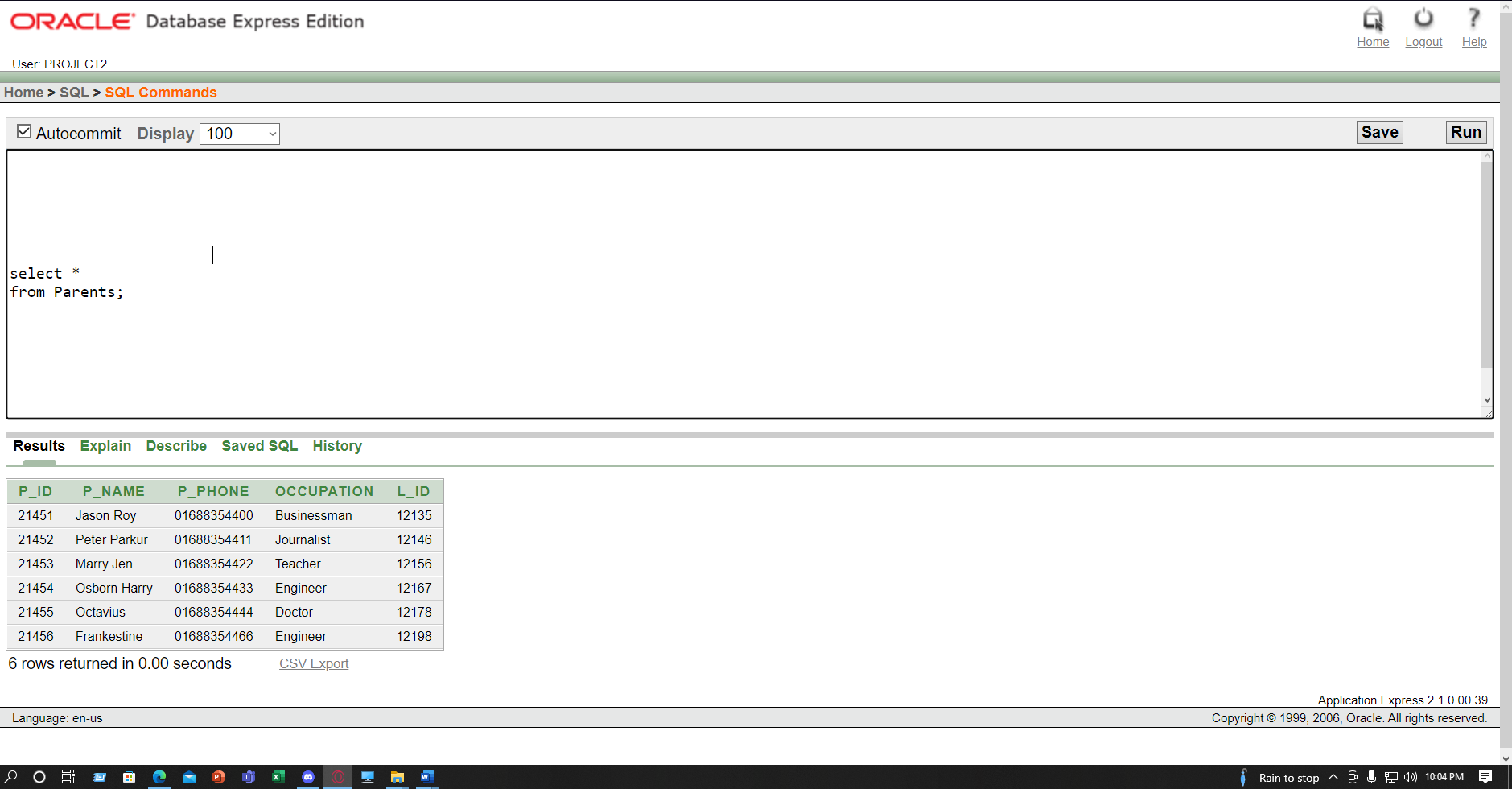


Table : **Child\_Parents**

INSERT INTO Child\_Parents VALUES (20451, 'Richard',6 , 21451 );

INSERT INTO Child\_Parents VALUES (20452, 'Philip',8 , 21452 );

INSERT INTO Child\_Parents VALUES (20453, 'Miles',9 , 21452 );

INSERT INTO Child\_Parents VALUES (20454, 'Dapr',6, 21453);

INSERT INTO Child\_Parents VALUES (20455, 'Shroud',7, 21454 );

INSERT INTO Child\_Parents VALUES (20456, 'Tenz',10, 21455);

INSERT INTO Child\_Parents VALUES (20457, 'Zellsis',9, 21456 );

INSERT INTO Child\_Parents VALUES (20458, 'Zombs',9, 21456);

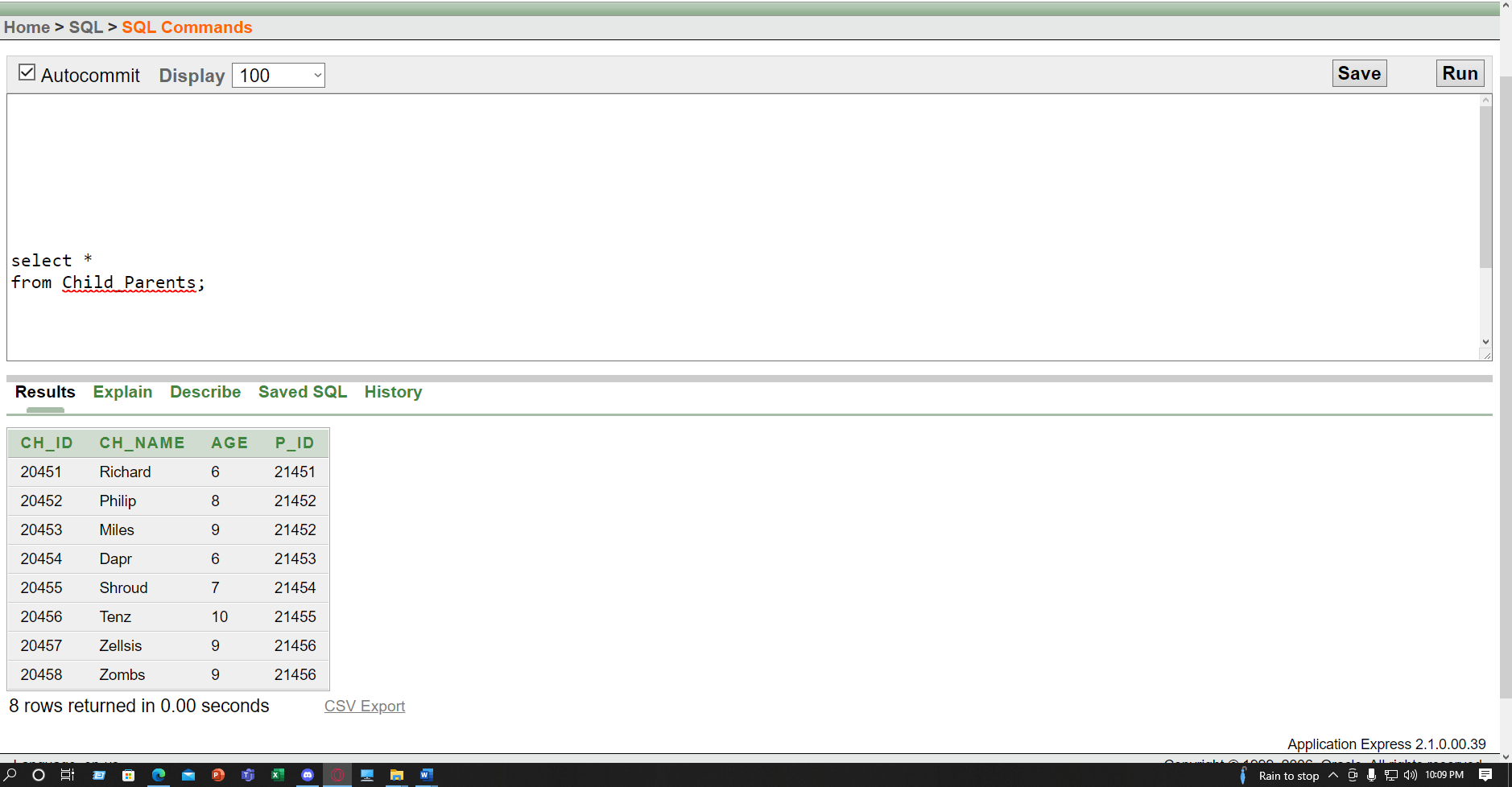


Table:**Location**

INSERT INTO Location VALUES (12135, 'Banani','Dhaka');

INSERT INTO Location VALUES (12146, 'Tejgaon','Dhaka');

INSERT INTO Location VALUES (12156, 'Sylhet','Sylhet');

INSERT INTO Location VALUES (12167 , 'Khulna','Khulna');

INSERT INTO Location VALUES (12178, 'Barisal','Barisal');

INSERT INTO Location VALUES (12198, 'Rajshahi','Rajshahi' );

INSERT INTO Location VALUES (12111, 'Noakhali','Chittagong' );

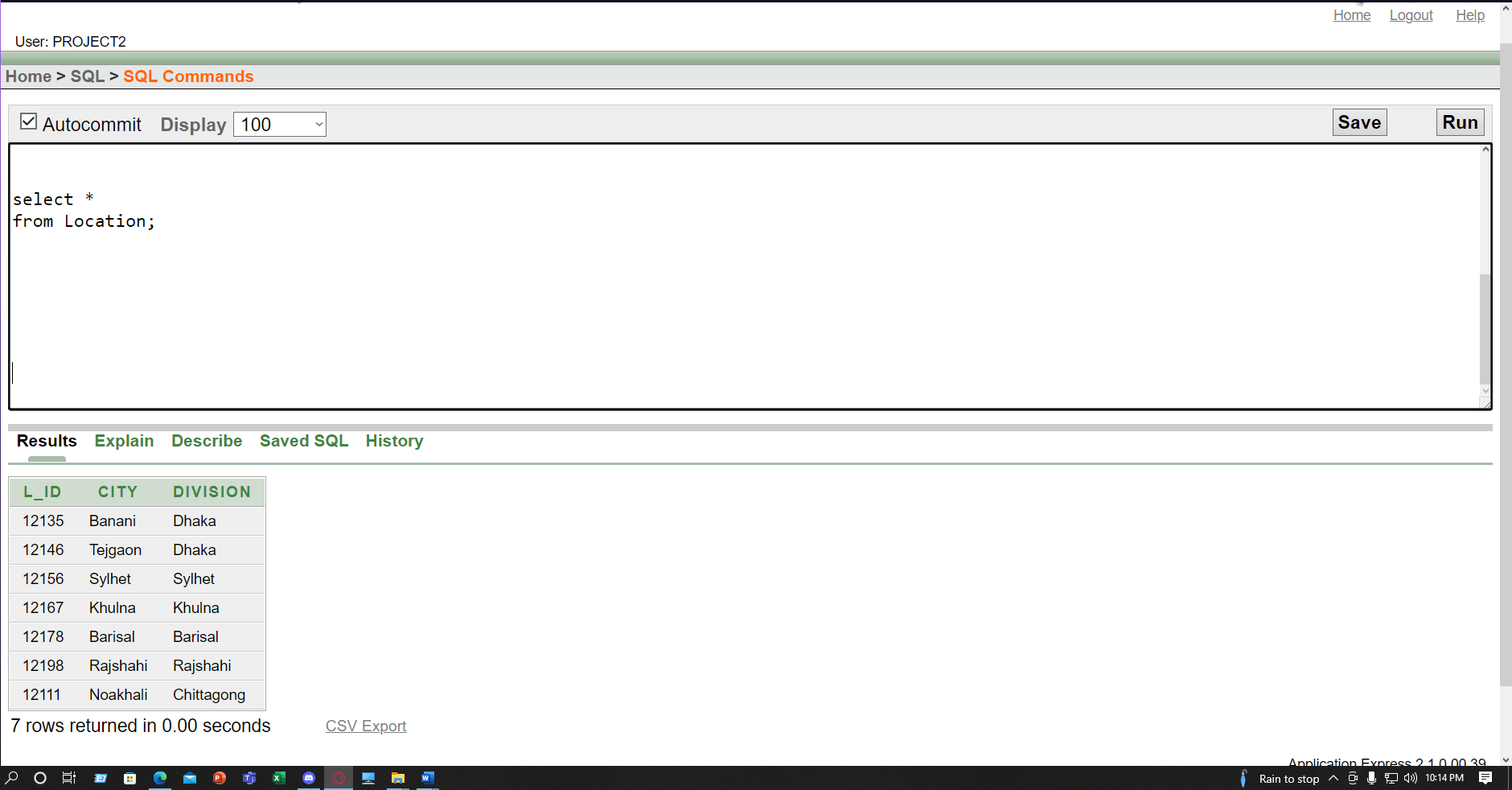


Table: **Babysitter**

INSERT INTO Babysitter VALUES (10451, '01715023511','Sophia' ,50000 );

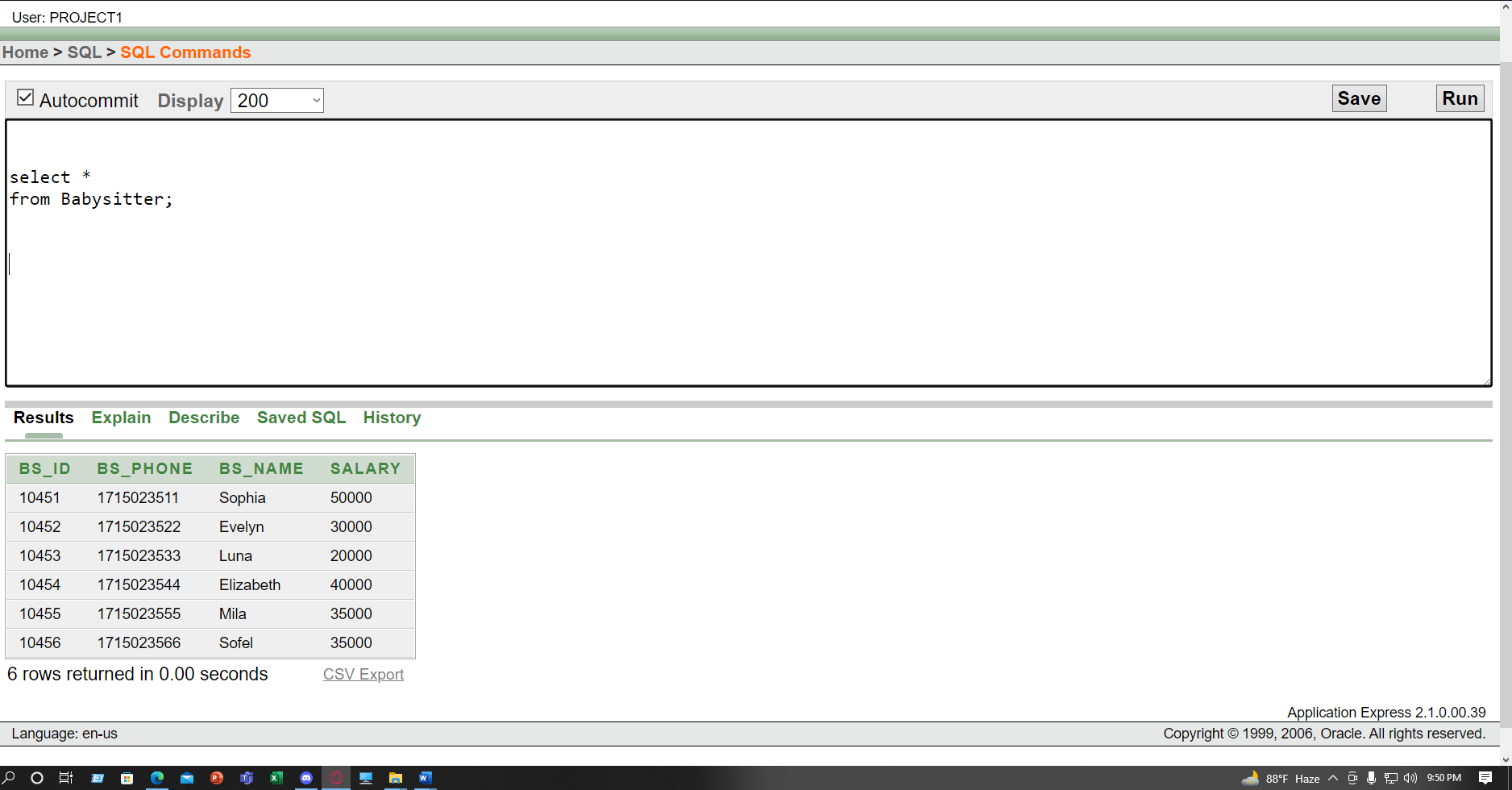
INSERT INTO Babysitter VALUES (10452, '01715023522','Evelyn' ,30000 );

INSERT INTO Babysitter VALUES (10453, '01715023533','Luna' ,21000 );

INSERT INTO Babysitter VALUES (10454, '01715023544','Elizabeth' ,40000);

INSERT INTO Babysitter VALUES (10455, '01715023555','Mila' ,35000 );

INSERT INTO Babysitter VALUES (10456, '01715023566','Sofel' ,35000 );



Table**: Child\_Babysitter**

INSERT INTO Child\_Babysitter VALUES (20451, 'Richard',6 , 10451);

INSERT INTO Child\_Babysitter VALUES ( 20452, 'Philip',8 , 10451);

INSERT INTO Child\_Babysitter VALUES (20453, 'Miles',9 , 10452);

INSERT INTO Child\_Babysitter VALUES (20454, 'Dapr',6, 10452);

INSERT INTO Child\_Babysitter VALUES (20455, 'Shroud',7, 10453);

INSERT INTO Child\_Babysitter VALUES (20456, 'Tenz',10, 10453);

INSERT INTO Child\_Babysitter VALUES (20457, 'Zellsis',9, 10454);

INSERT INTO Child\_Babysitter VALUES (20458, 'Zombs',9, 10455);

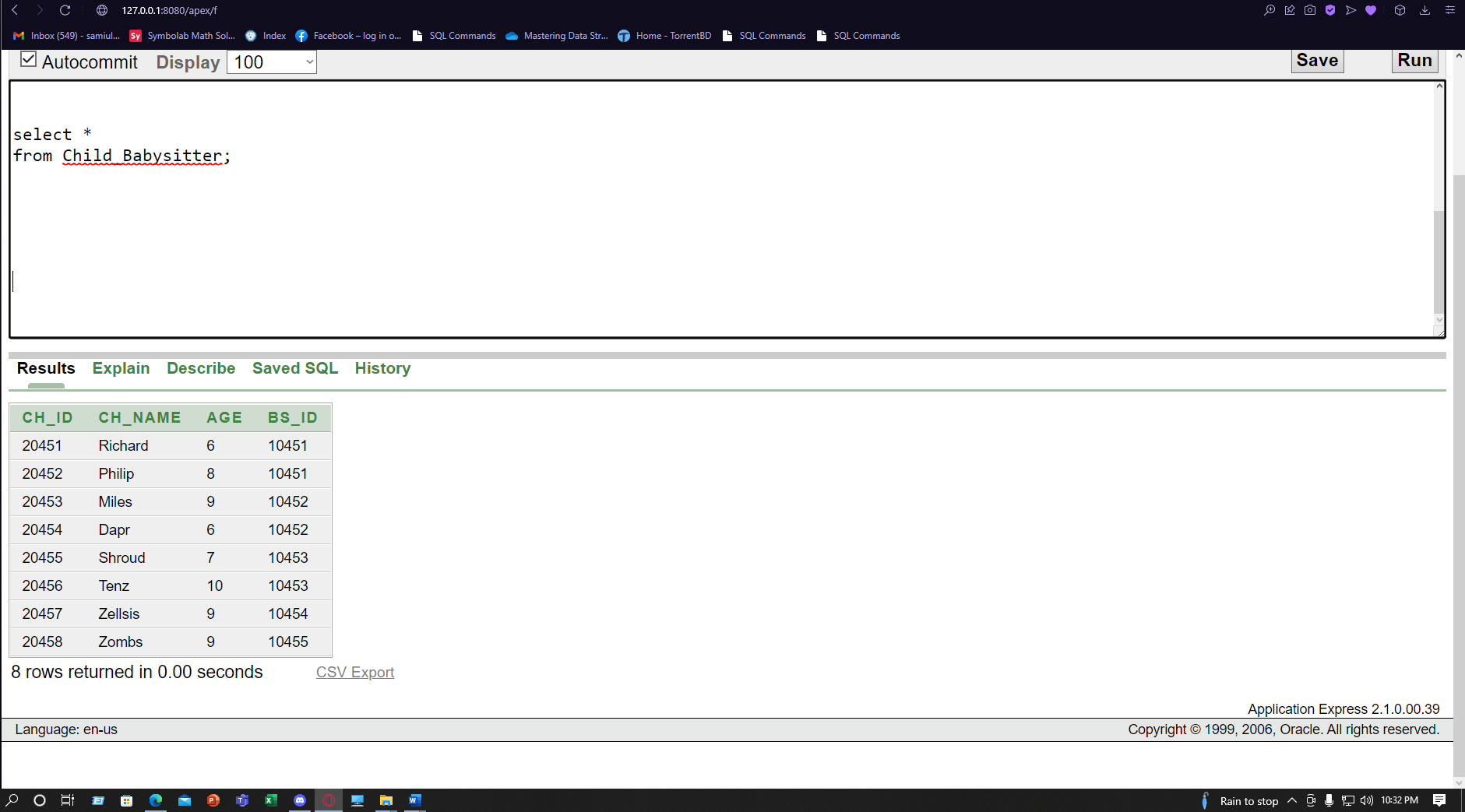


Table:**Class**

INSERT INTO Class VALUES (40451, 'Tiger');

INSERT INTO Class VALUES (40452, 'Lion');

INSERT INTO Class VALUES (40453, 'Lotus');

INSERT INTO Class VALUES (40454, 'Rose');

INSERT INTO Class VALUES (40455, 'Lily');

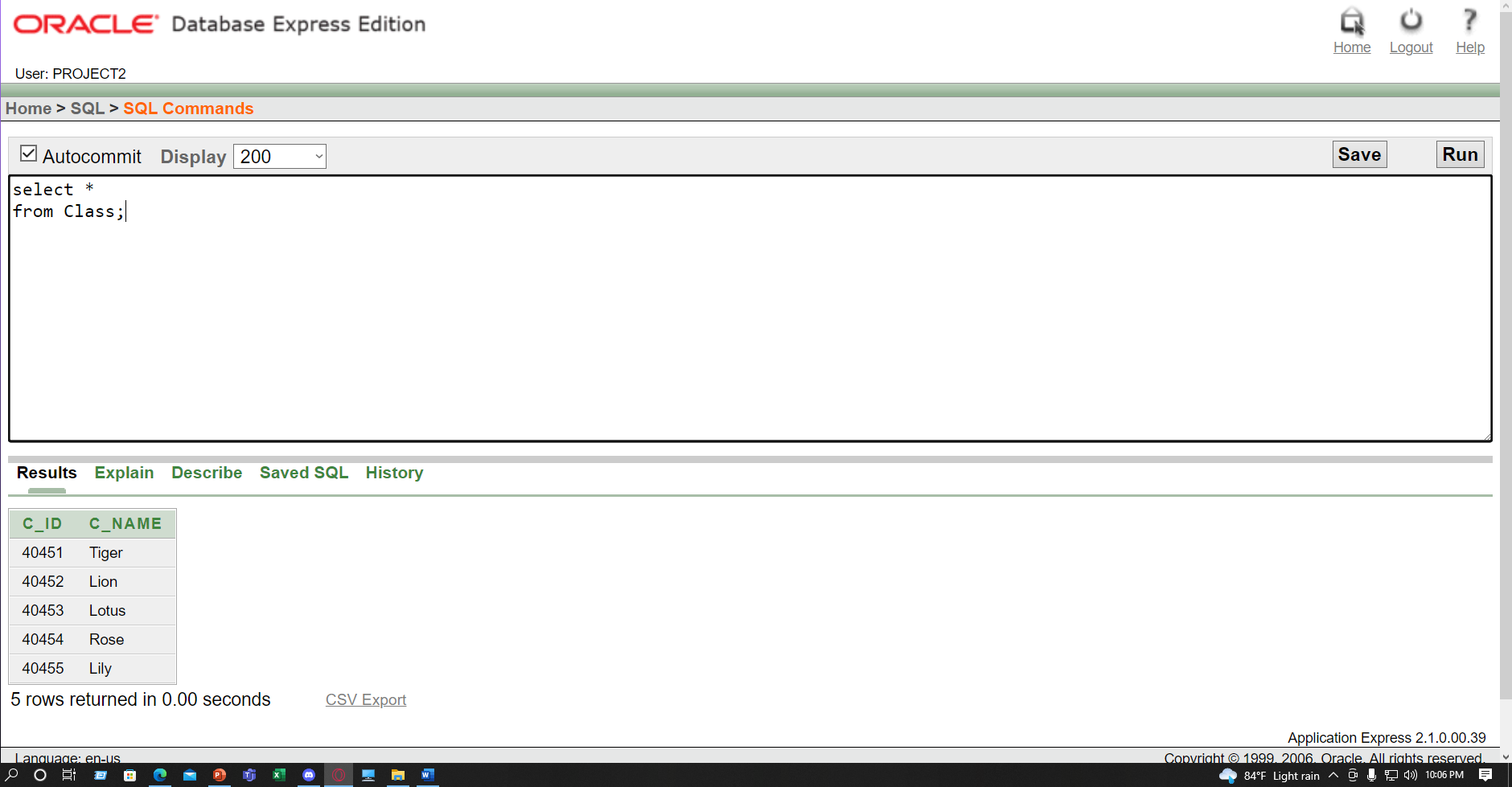


Table:**Class\_Babysitter**

INSERT INTO Class\_Babysitter VALUES (40451, 'Tiger',10451);

INSERT INTO Class\_Babysitter VALUES (40452,'Lion',10452 );

INSERT INTO Class\_Babysitter VALUES (40453, 'Lotus',10453 );

INSERT INTO Class\_Babysitter VALUES (40454,'Rose',10454 );

INSERT INTO Class\_Babysitter VALUES (40455, 'Lily',10455 );

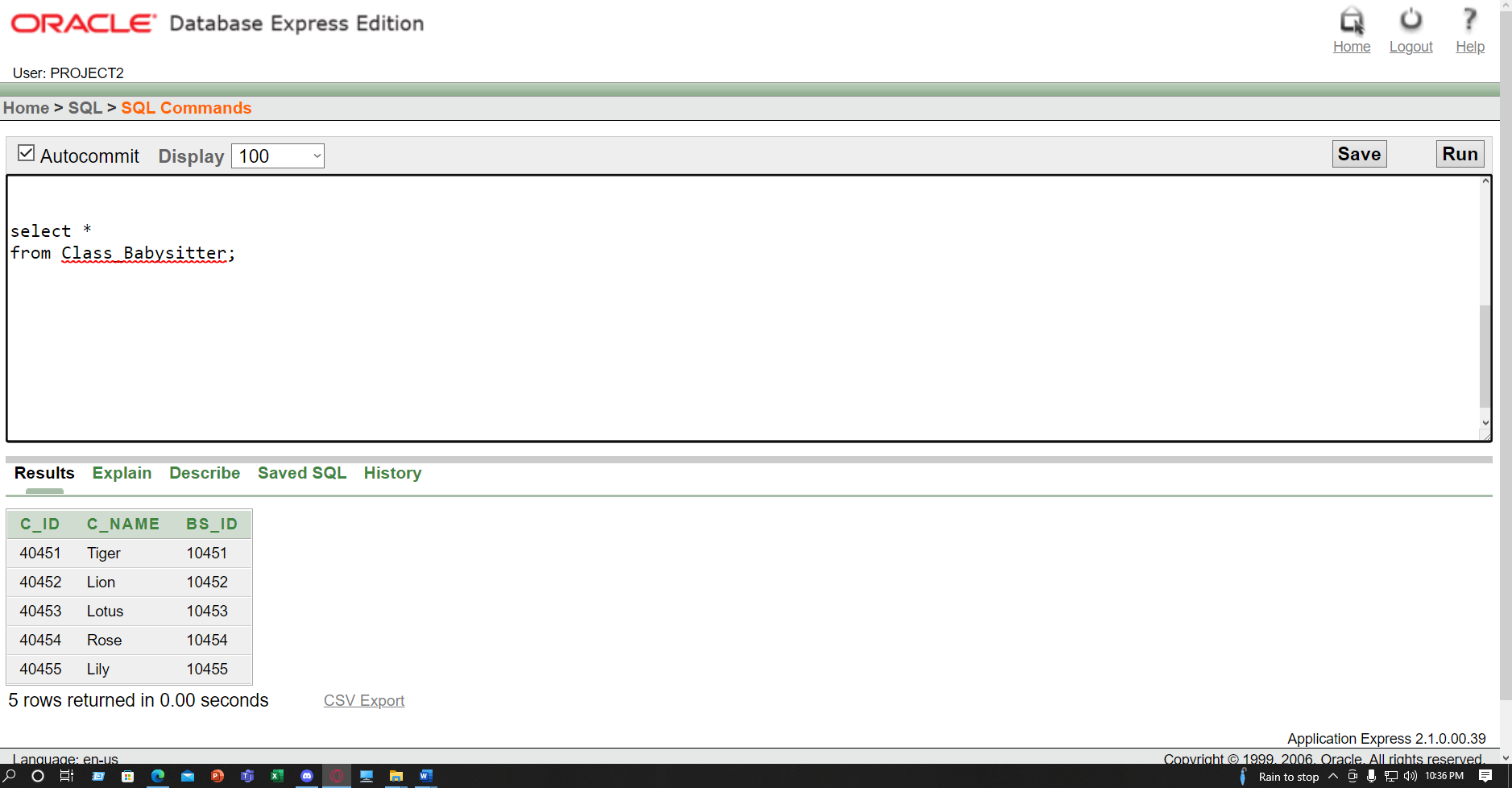


Table: **Child\_Class**

INSERT INTO Child\_Class VALUES (20451, 'Richard',6 , 40451);

INSERT INTO Child\_Class VALUES (20452, 'Philip',8 , 40451);

INSERT INTO Child\_Class VALUES (20453, 'Miles',9 ,40452);

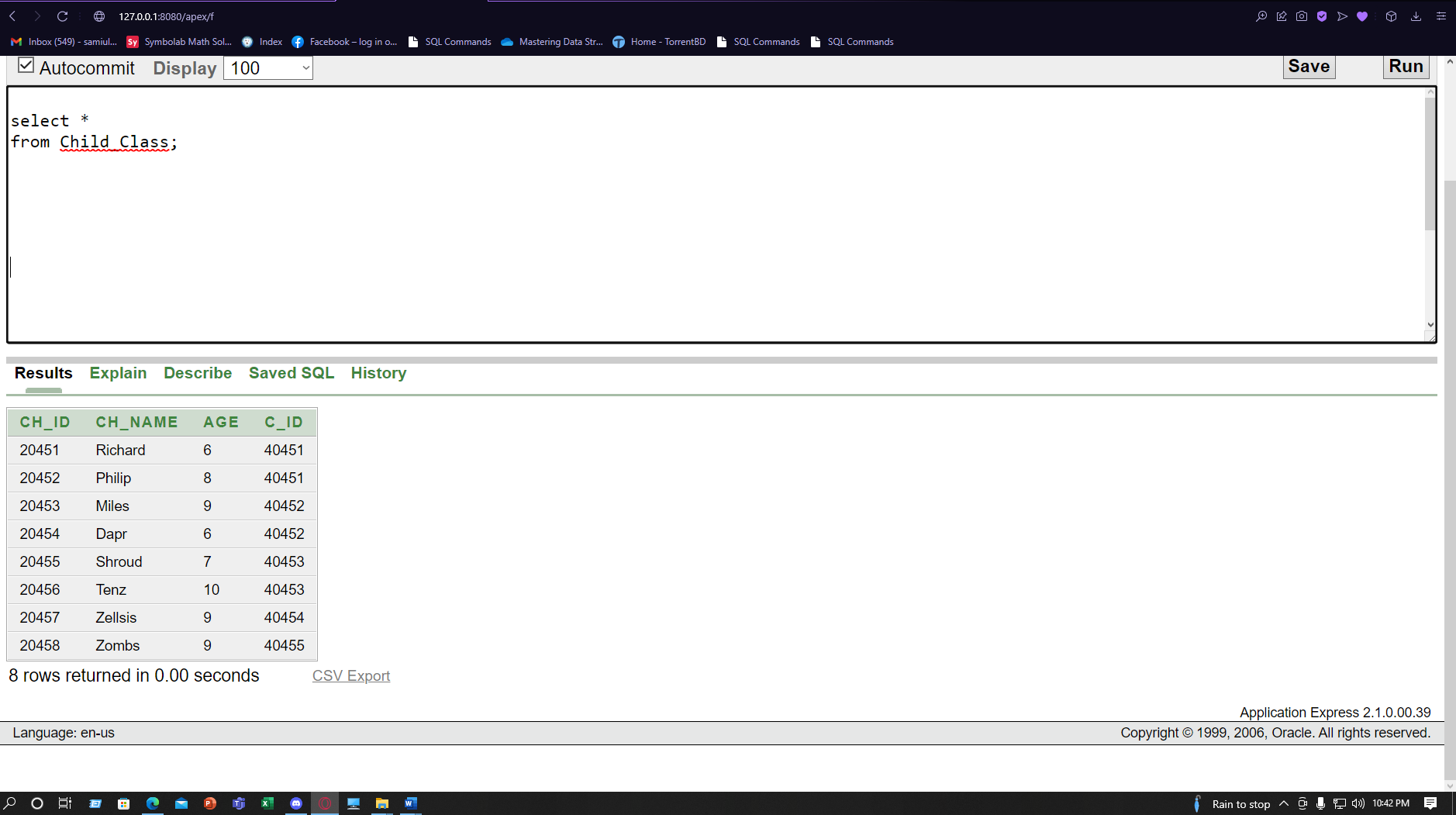
INSERT INTO Child\_Class VALUES (20454, 'Dapr',6, 40452);

INSERT INTO Child\_Class VALUES (20455, 'Shroud',7, 40453);

INSERT INTO Child\_Class VALUES (20456, 'Tenz',10, 40453);

INSERT INTO Child\_Class VALUES (20457, 'Zellsis',9, 40454);

INSERT INTO Child\_Class VALUES (20458, 'Zombs',9,40455);



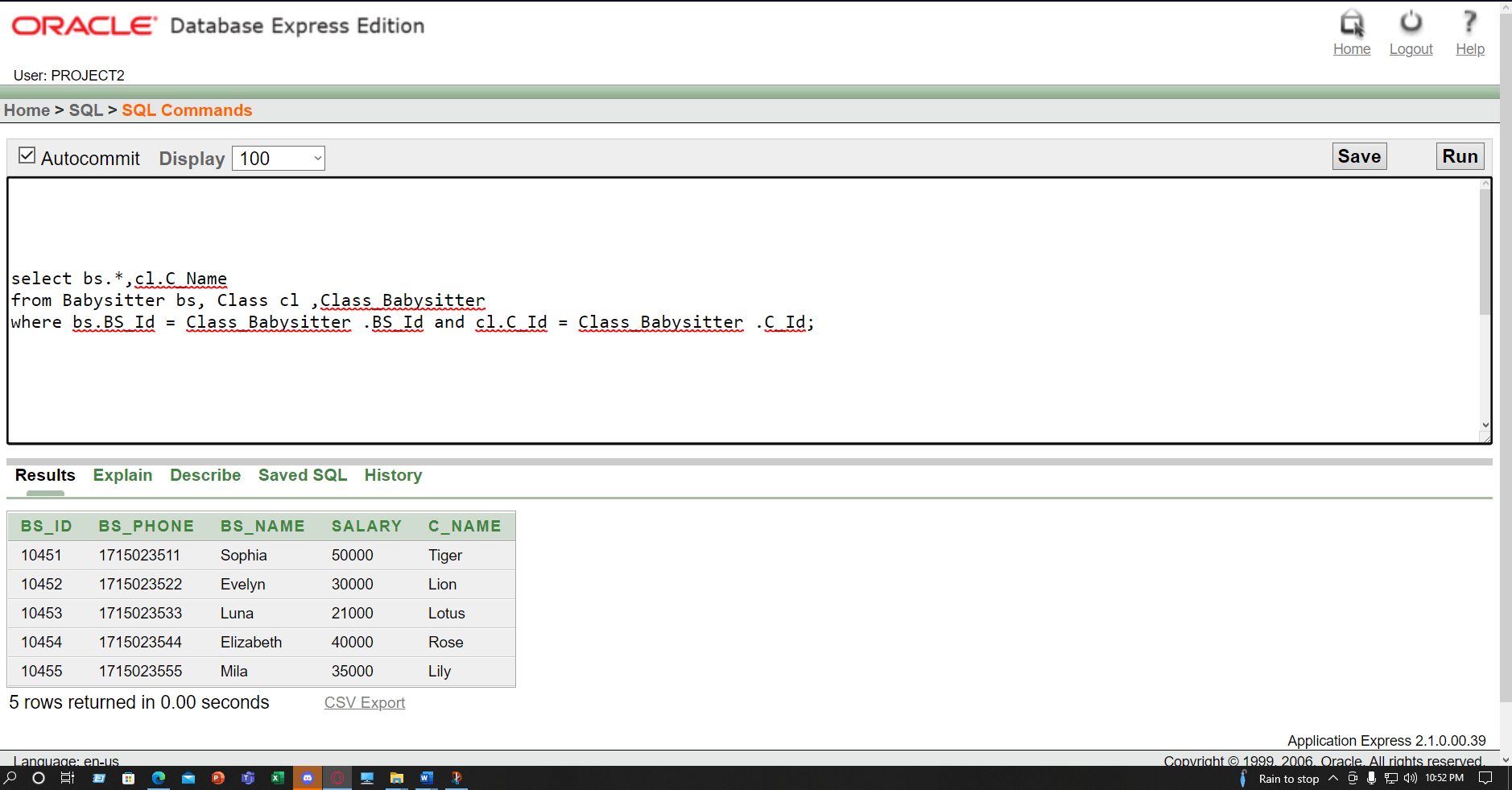
Joining**: Equijoin**

* **Show Babysitter information who are assigned with Class and Show also Class Name?**

select bs.\*,cl.C\_Name

from Babysitter bs, Class cl ,Class\_Babysitter

where bs.BS\_Id = Class\_Babysitter .BS\_Id and cl.C\_Id = Class\_Babysitter .C\_Id;

****

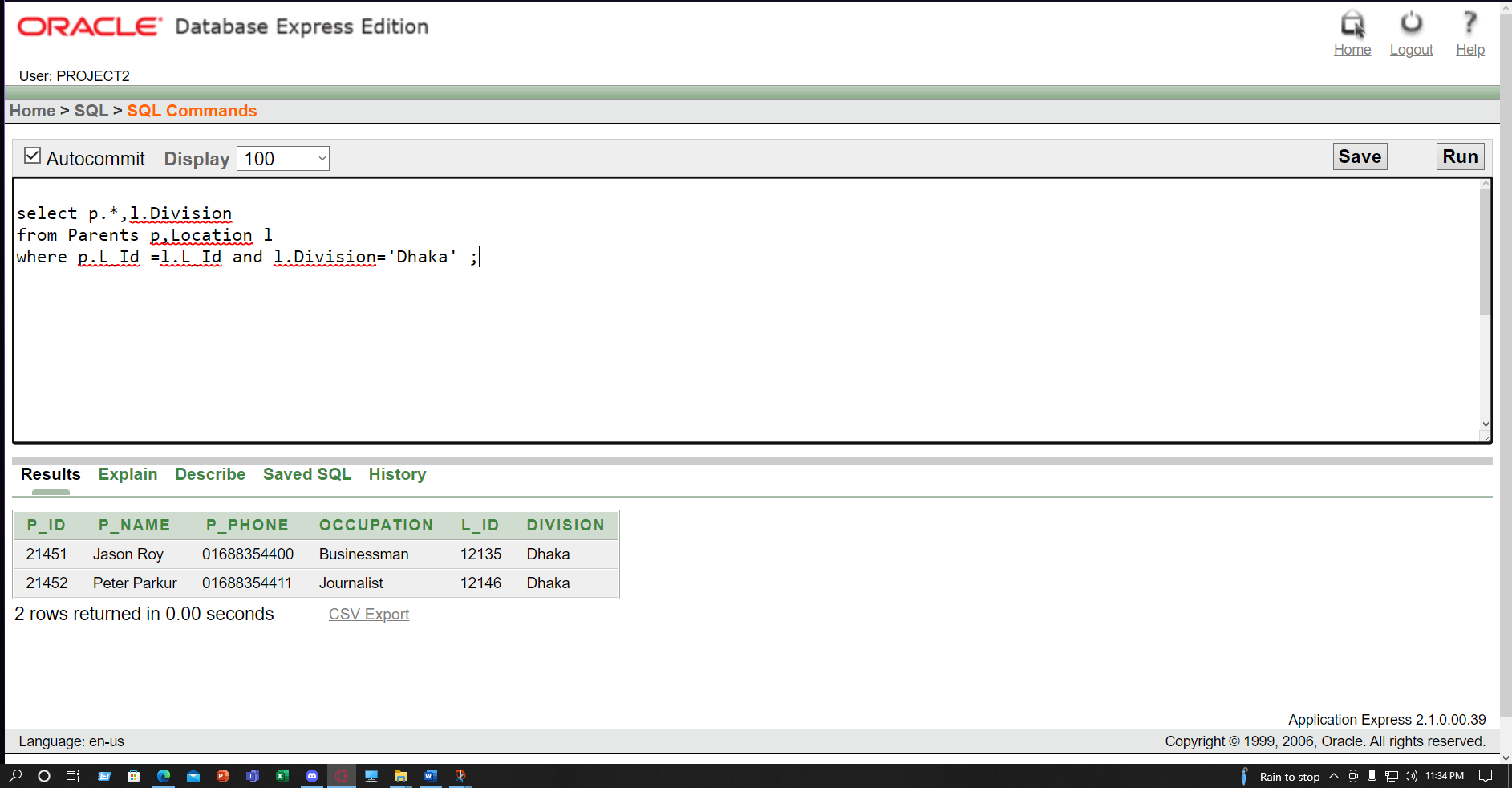
Joining**: Equijoin**

* **Show Parents Information who are in Dhaka Division?**

select p.\*,l.Division

from Parents p,Location l

where p.L\_Id =l.L\_Id and l.Division='Dhaka' ;



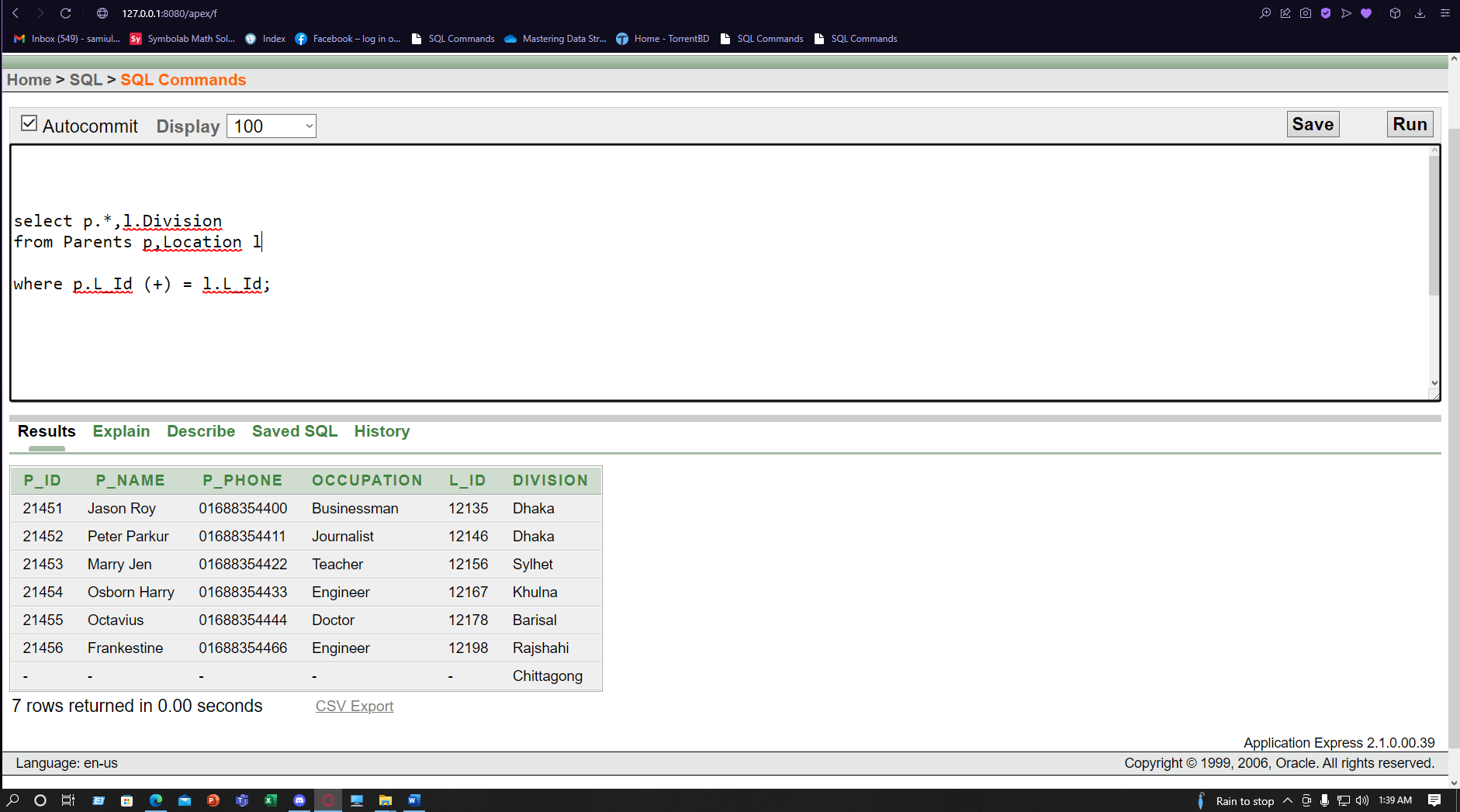
Joining: **Outer join**

* **Show Parents Id,Name,Phone Location-wise ?**

**select p.\*,l.Division**

**from Parents p,Location l**

**where p.L\_Id (+) = l.L\_Id;**



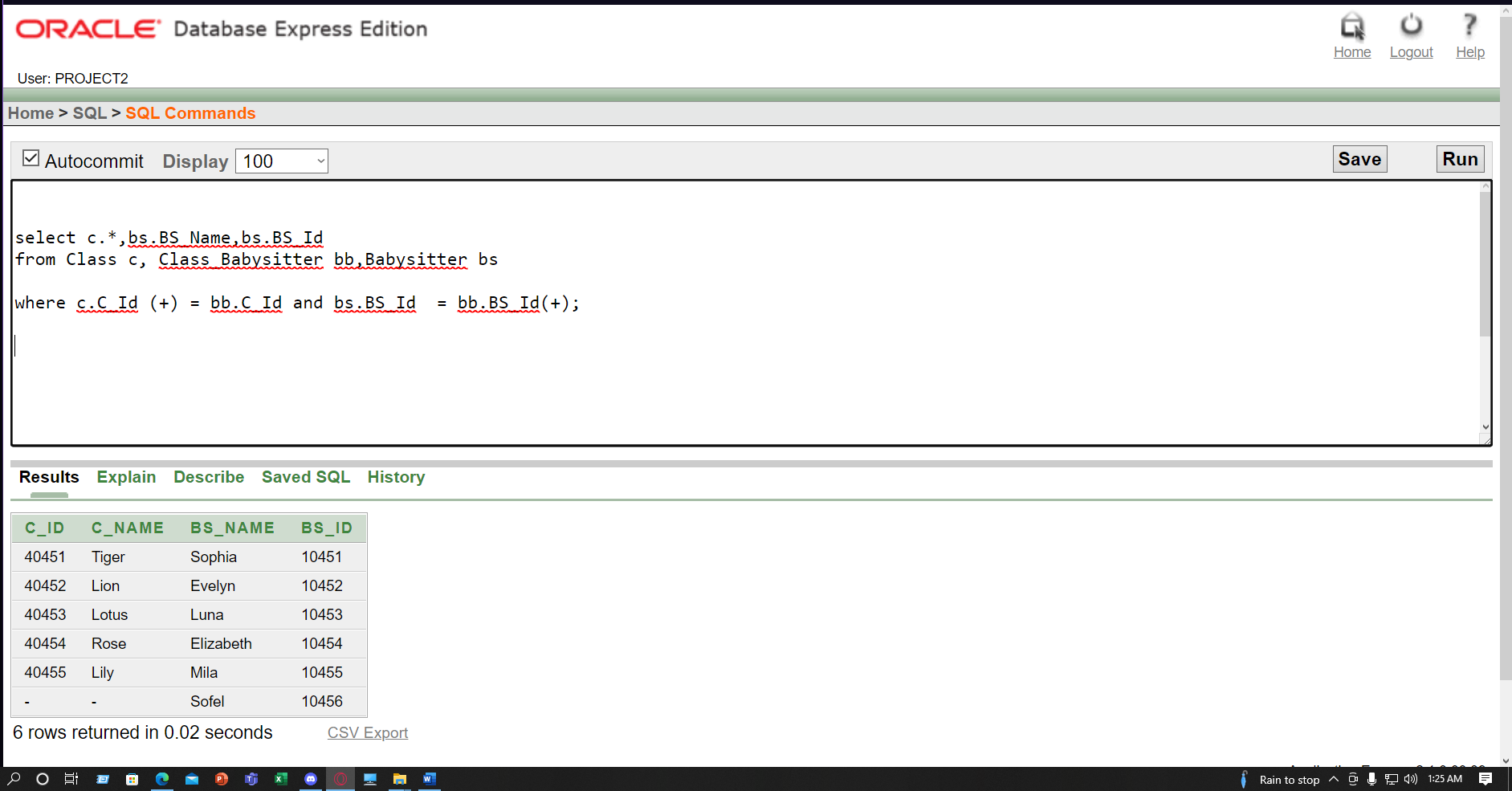
Joining: **Outer join**

* **Show Class Information and Show all Babysitter Name ,Id**

select c.\*,bs.BS\_Name,bs.BS\_Id

from Class c, Class\_Babysitter bb , Babysitter bs

where c.C\_Id (+) = bb.C\_Id and bs.BS\_Id = bb.BS\_Id(+);



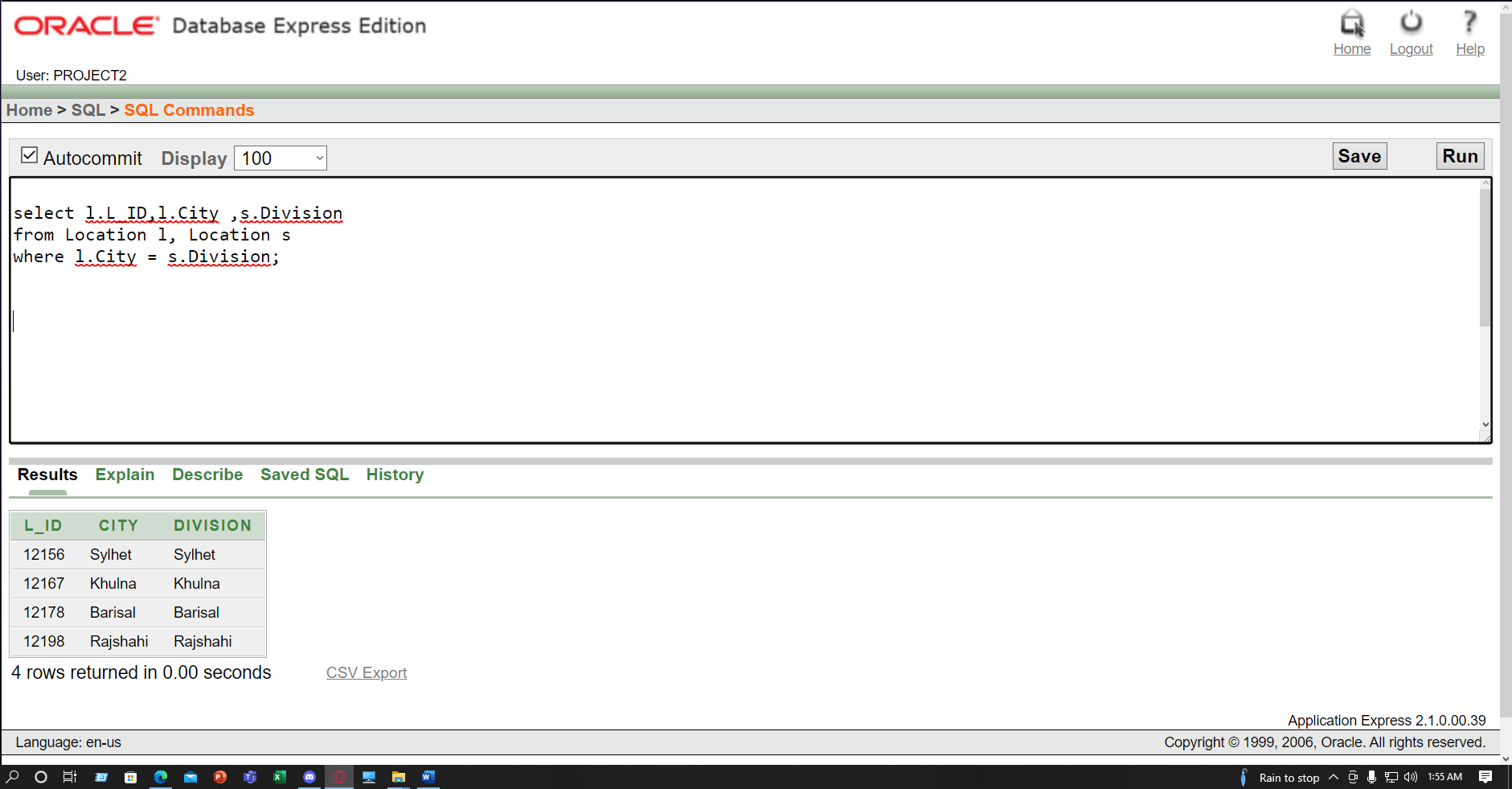
Joining: **Self Joining**

* **Show Location Information where City and Division is same**

select l.L\_ID, l.City , s.Division

from Location l, Location s

where l.City = s.Division;



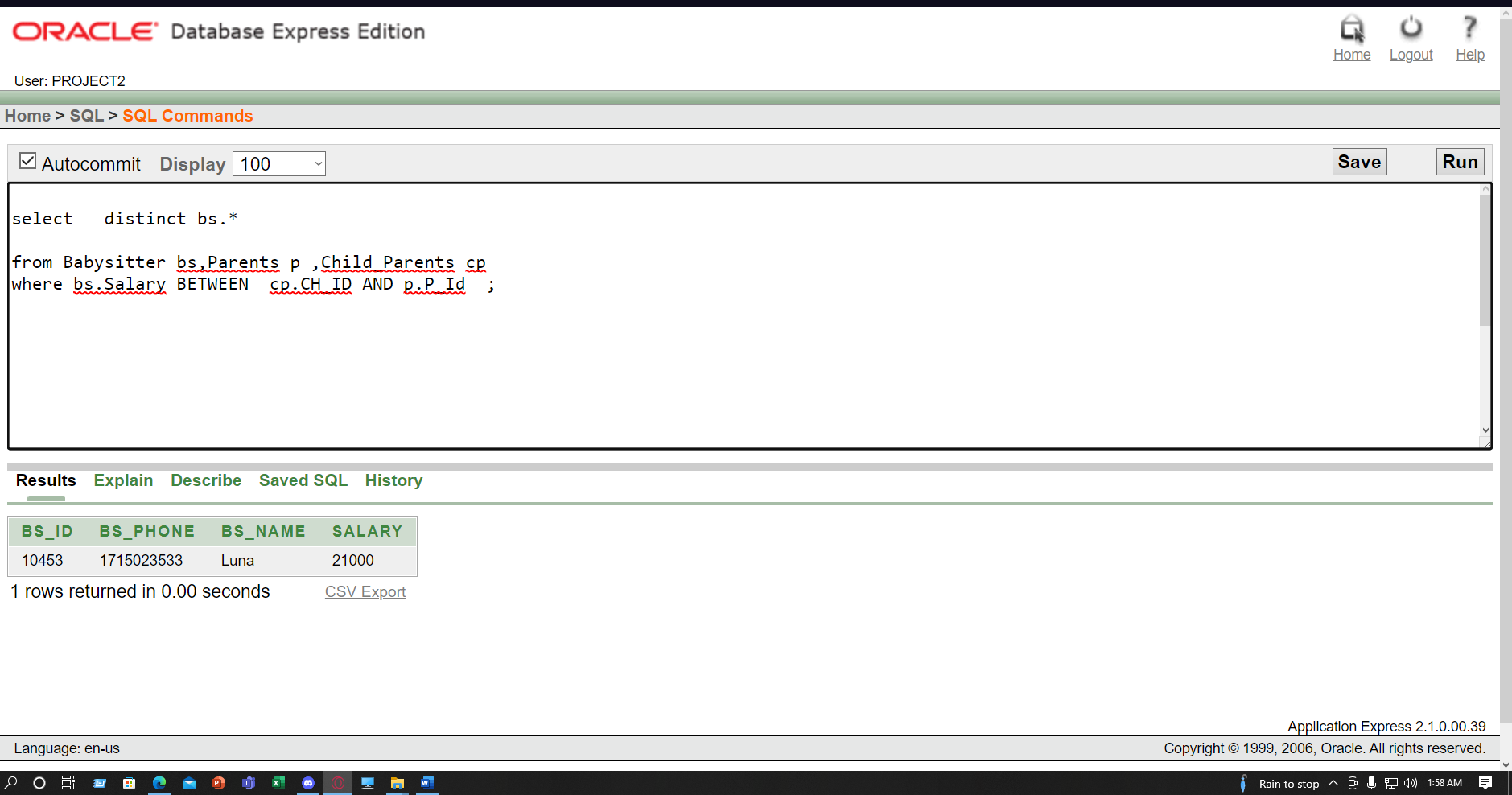
**Joining: Non-equijoin**

* **Show the Babysitter information where Babysitter's salary is between Child Id and Parents Id**

select distinct bs.\*

from Babysitter bs,Parents p ,Child\_Parents cp

where bs.Salary BETWEEN cp.CH\_ID AND p.P\_Id ;



**Sub-query:**

* **Display All the child's Name whose age are greater than Shroud' Age?**

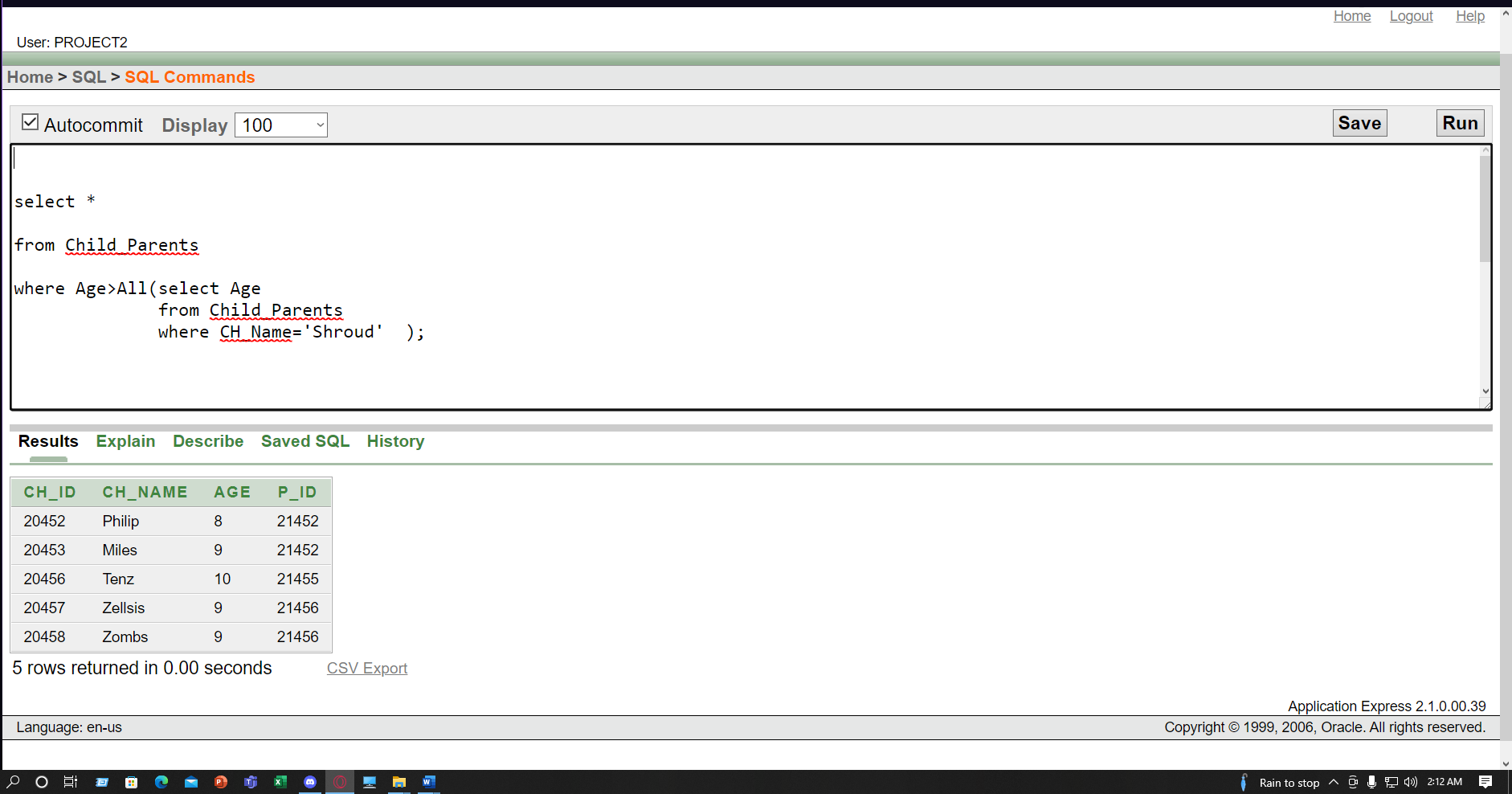
select \*

from Child\_Parents

where Age>All(select Age

from Child\_Parents

where CH\_Name='Shroud' );

****

* **Show All the Parents information Who Stay in Dhaka Division?**

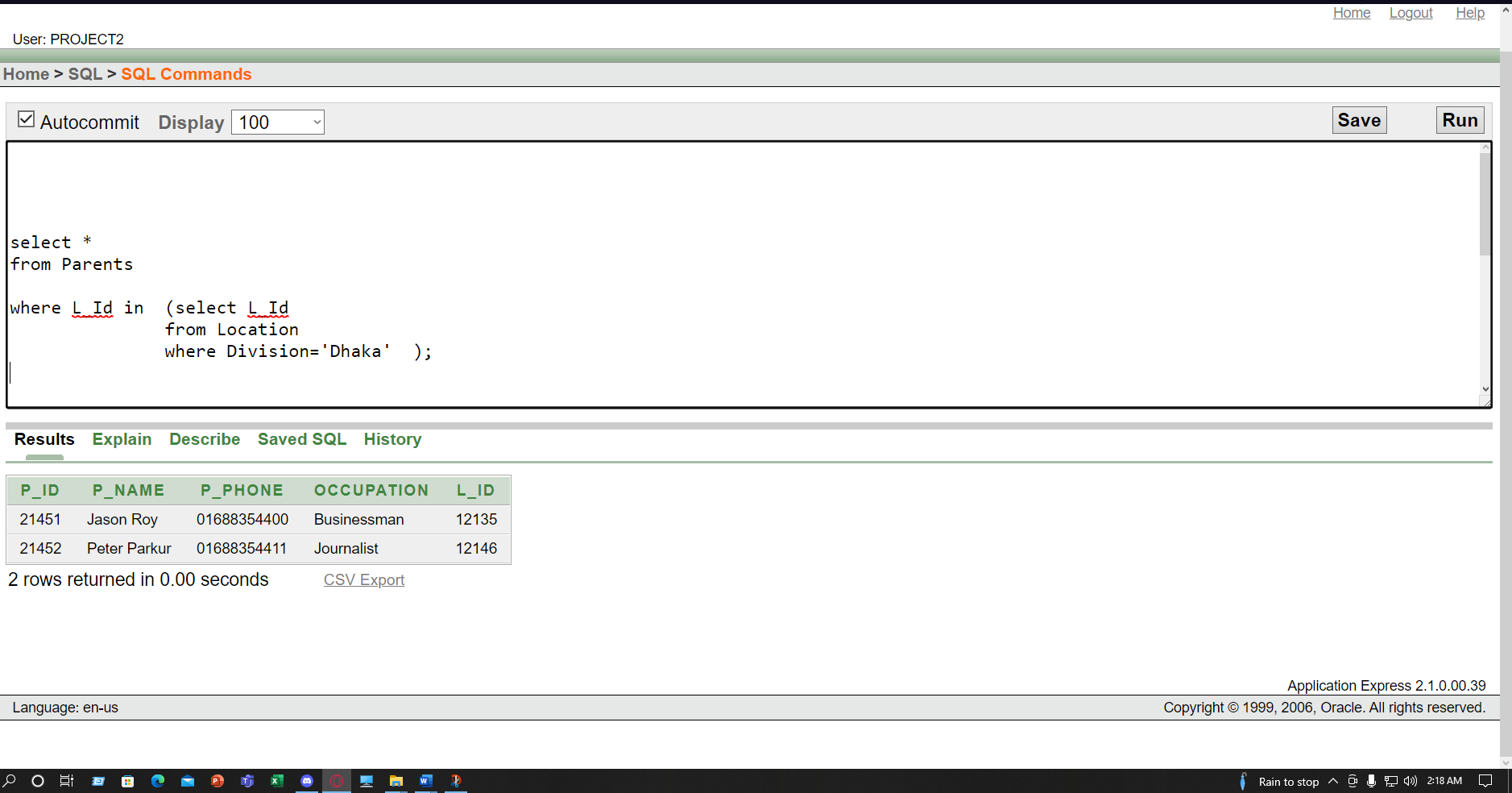
select \*

from Parents

where L\_Id in (select L\_Id

from Location

where Division='Dhaka' );

****

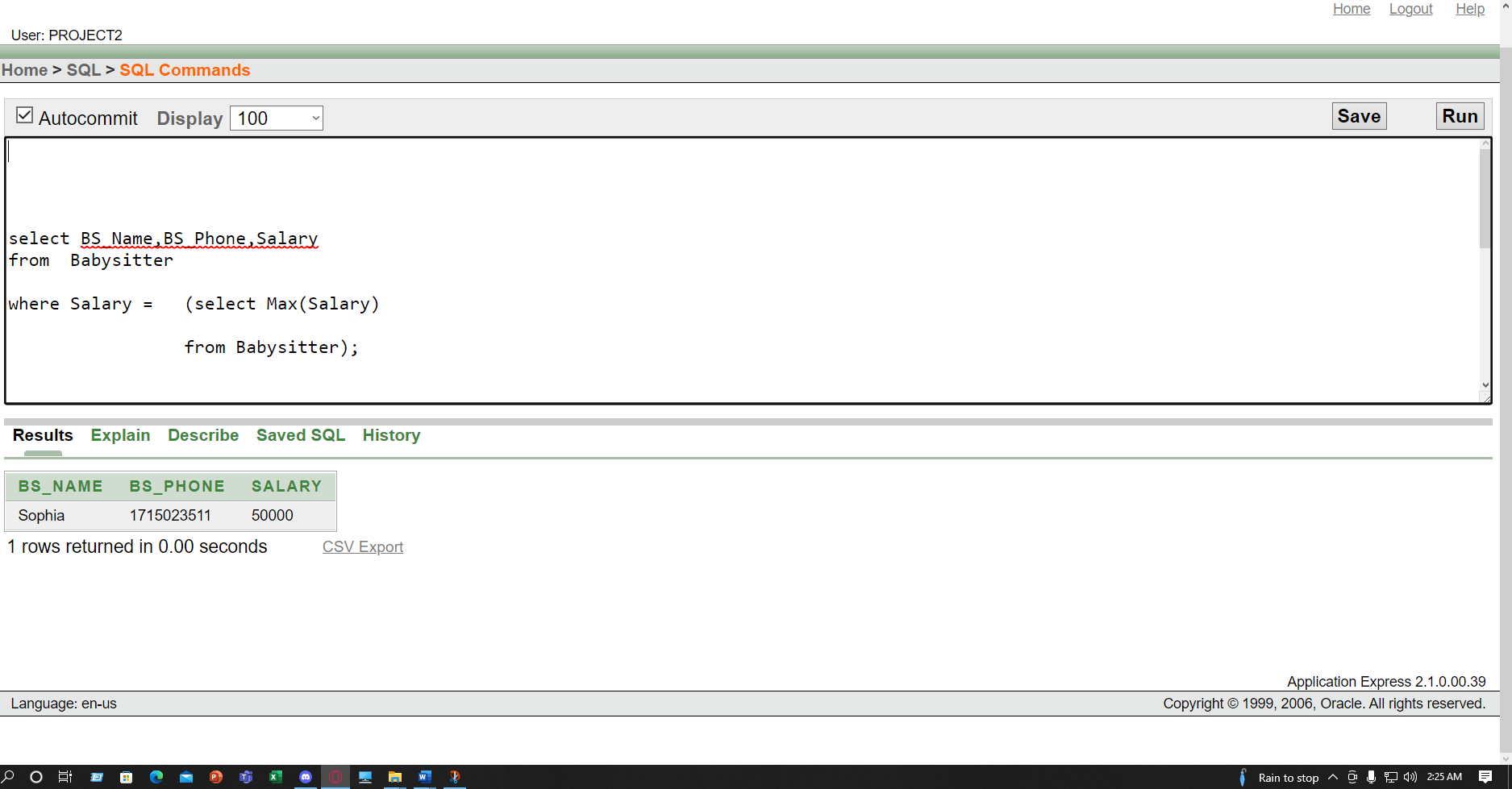
* **Display Maximum Salary of Babysitter**

select BS\_Name,BS\_Phone,Salary

from Babysitter

where Salary = (select Max(Salary)

from Babysitter);

****

* **Show All Child's Name whose Age is greater than 8 and which Parents is Engineer**

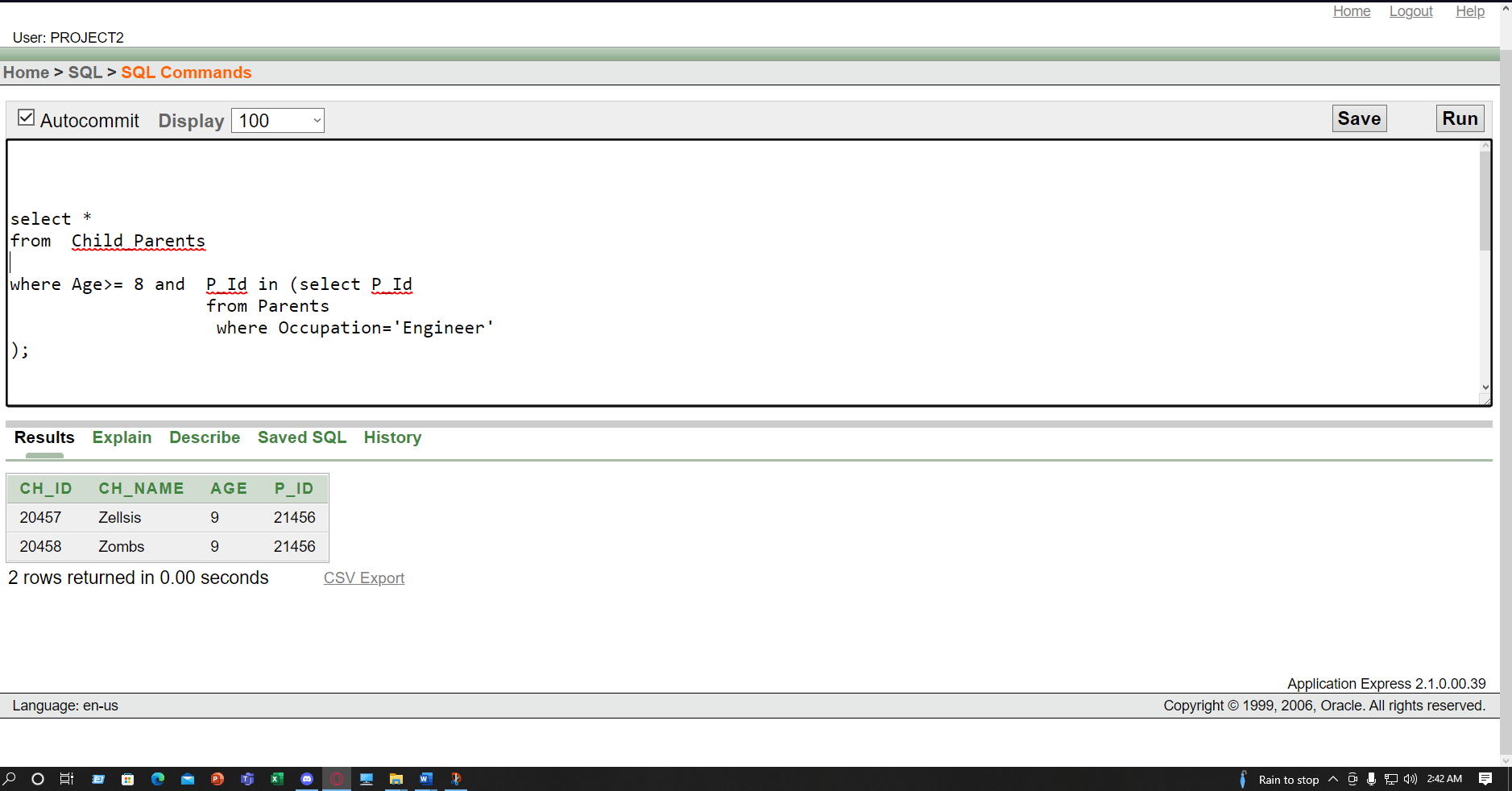
select \*

from Child\_Parents

where Age>= 8 and P\_Id in (select P\_Id

from Parents

where Occupation='Engineer' );



**Simple View:**

* **Create View All the child's Name whose age are Less than 9**

Create View child\_9

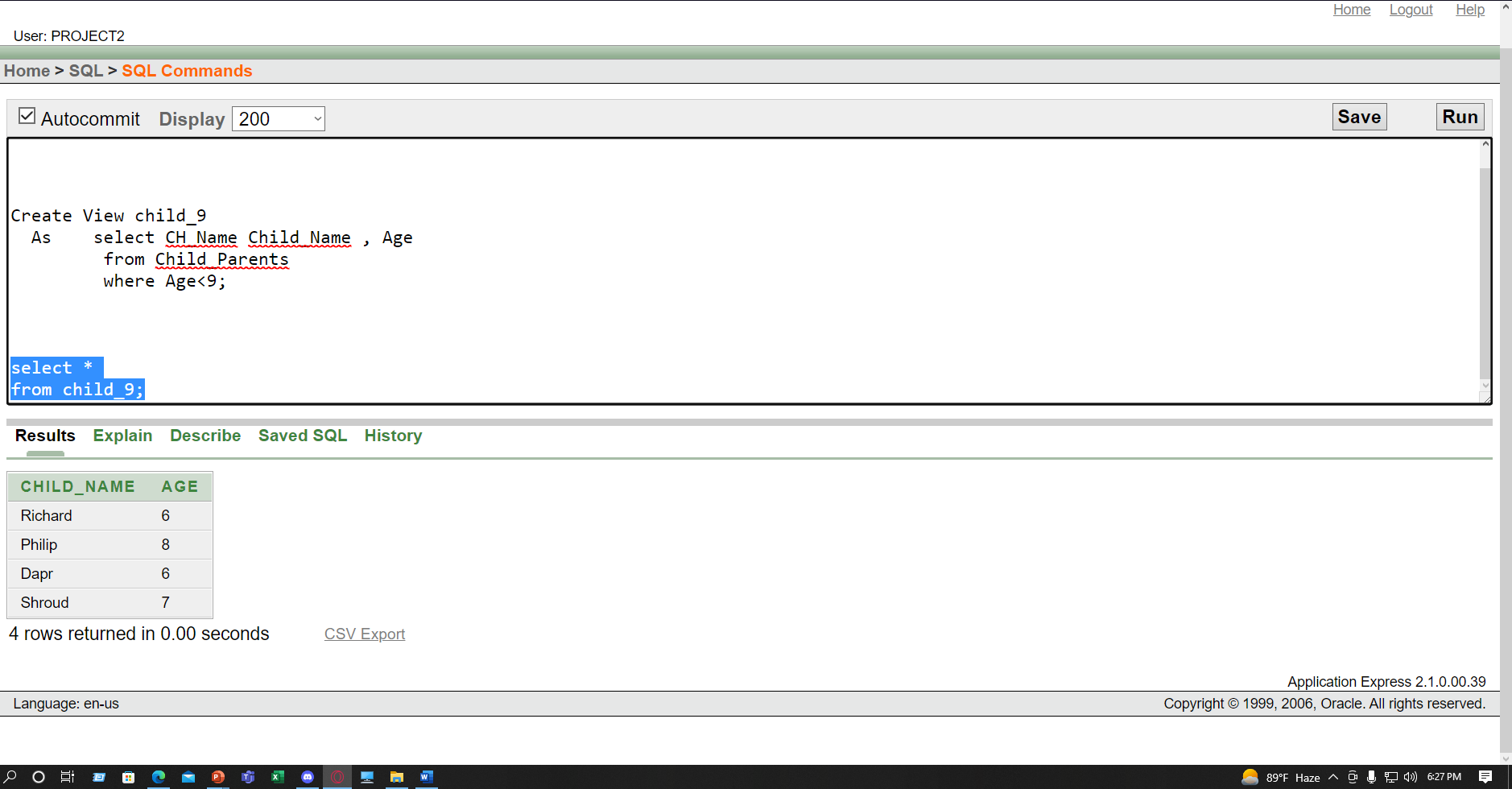
As select CH\_Name Child\_Name , Age

from Child\_Parents

where Age<9;

select \*

from child\_9;



**Complex View:**

* **Show a view of number of Child Age-wise**

create or replace view child\_count ("Child Age","Number of Child")

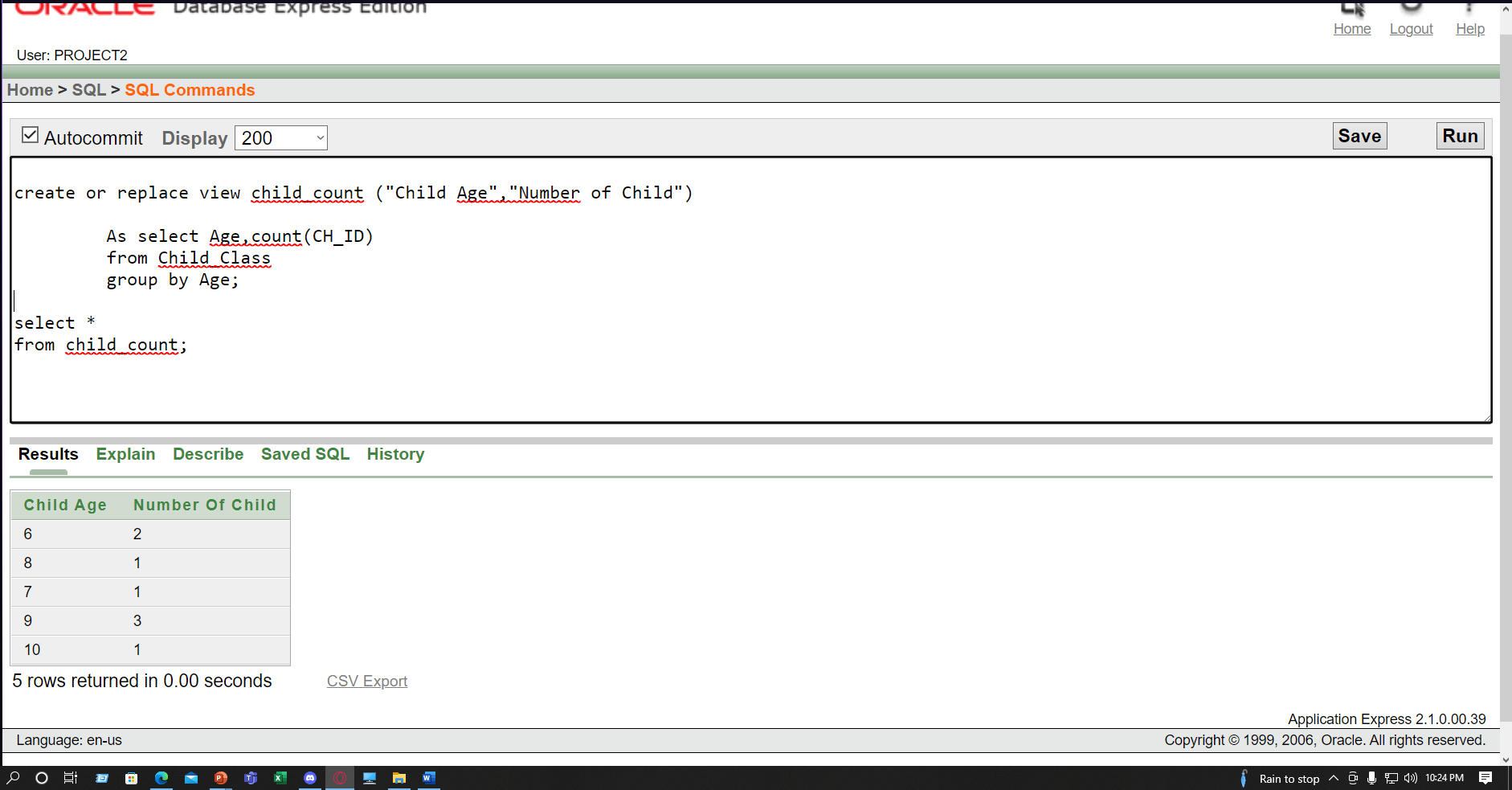
As select Age,count(CH\_ID)

from Child\_Class

group by Age;

select \*

from child\_count**;**

****

**Adding constraint in table**

* Constraint is already added in all tables but one additional constraint is added in Location table.

**Alter Table Location Modify City Not Null;**

