A blue circle with text and images

Description automatically generated

**American International University-Bangladesh**

Faculty of Science & Technology (CSE)

**Introduction to Database**

Summer 2022-23

**Section: J, Group: 04**

**Bangladesh Railway e-ticket System**

**Supervised by**

**Kawser Irom Rushee**

|  |  |
| --- | --- |
| **Group Members** | **ID** |
| **MD. IBTIHAZZAMAN** | **22-49153-3** |
| **MD. SAZZAD KHAN** | **22-49172-3** |
| **SAIFUL ISLAM ONI** | **22-49196-3** |

**ER Diagram Scenario**

In Bangladesh Railway e-ticketing system, a passenger may Purchase many tickets. One ticket must be Purchased by exactly one passenger. A passenger is identified by ab email & password. The system also stores passenger’s name, NID, phone number, balance, and security question for security purpose . A ticket is identified by route number & price. While purchasing a ticket, the date of issuing ticket and ticket ID is also stored. A passenger may have ticket history. Ticket history contains ticket ID, train name, departure station, arrival station, issue time & price. Admin manages tickets. Admin has an email address & a password. Admin also operates many trains. A train is identified by train ID & train name. A train has many routes, and a route has also many trains. A route is identified by route ID & route information.

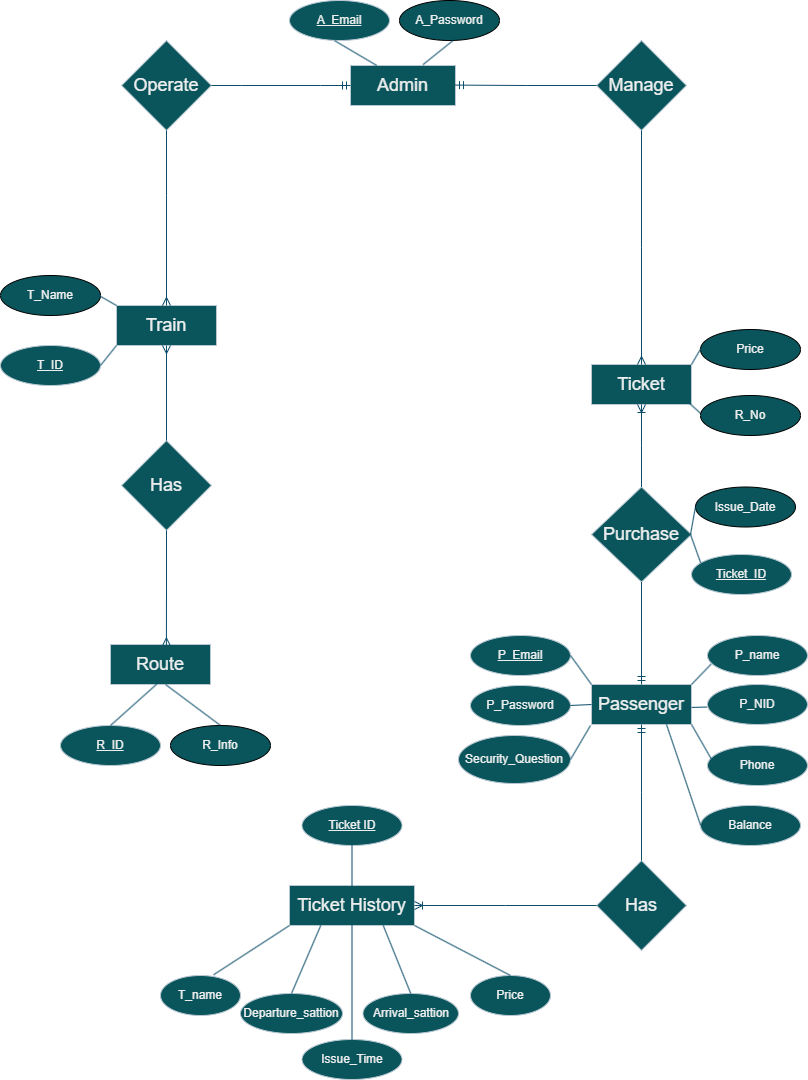
A screenshot of a computer

Description automatically generated**ER Diagram Making**

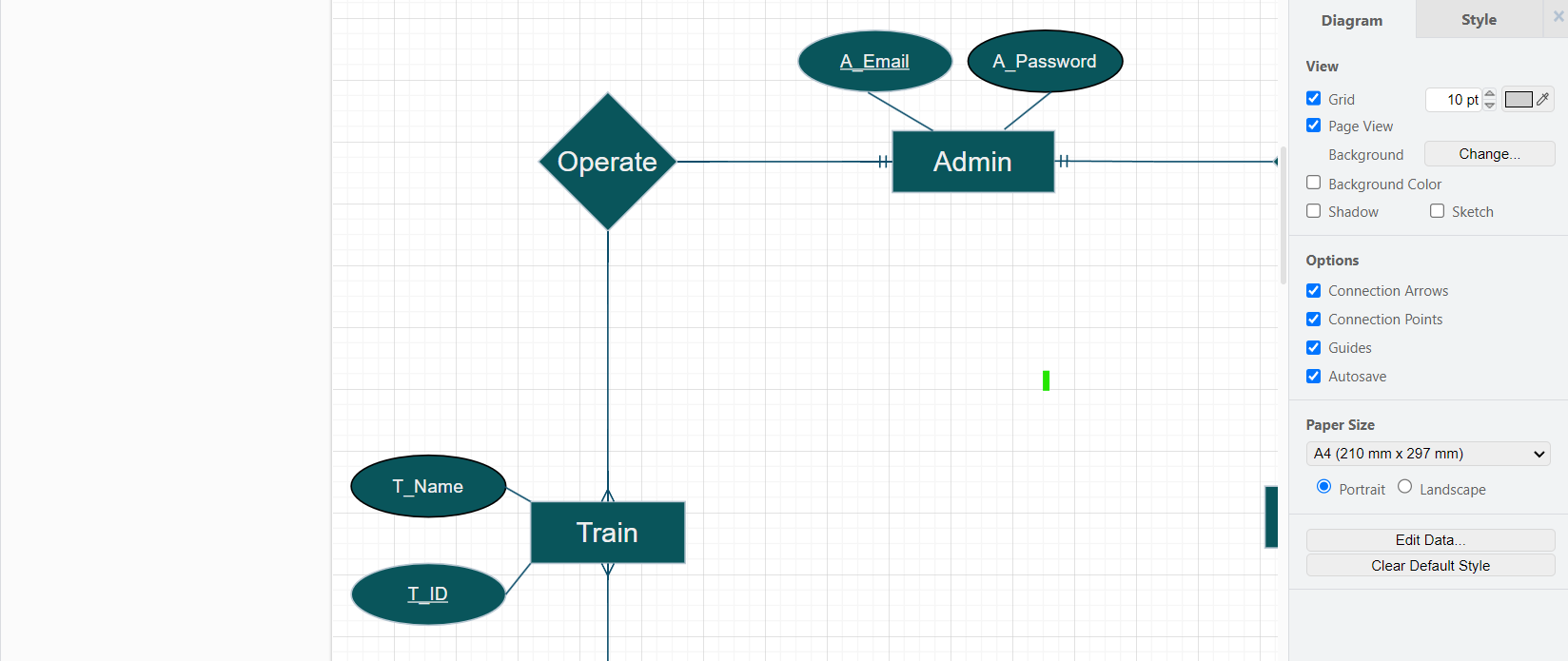
A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Final ER Diagram**

**Normalization**

1. **Operate (**Admin **Operate** Train**)**

**UNF : A\_Email**, A\_Password, **T\_ID**, T\_Name

**1NF : A\_Email**, **T\_ID**, A\_Password, T\_Name

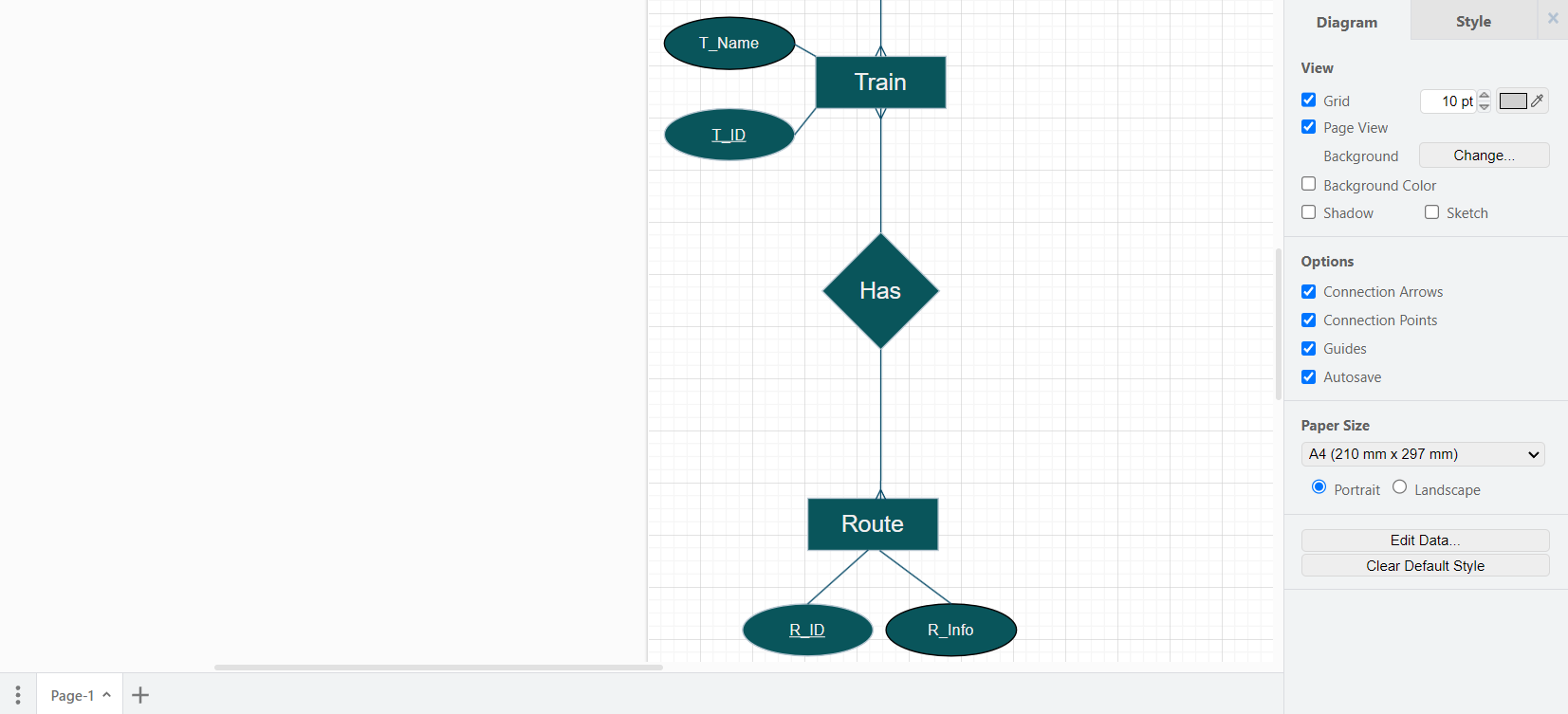
**2NF : 1st : A\_Email**, A\_Password

**2nd :** **T\_ID**, T\_Name, **A\_Email**

**3NF : 1st : A\_Email**, A\_Password

**2nd :** **T\_ID**, T\_Name, **A\_Email**

1. **Has (**Train **has** Route**)**



**UNF : T\_ID**, T\_Name, **R\_ID**, R\_Info

**1NF : T\_ID**, T\_Name, **R\_ID**, R\_Info

**2NF : 1st : T\_ID**, T\_Name

**2nd :** **R\_ID**, R\_Info

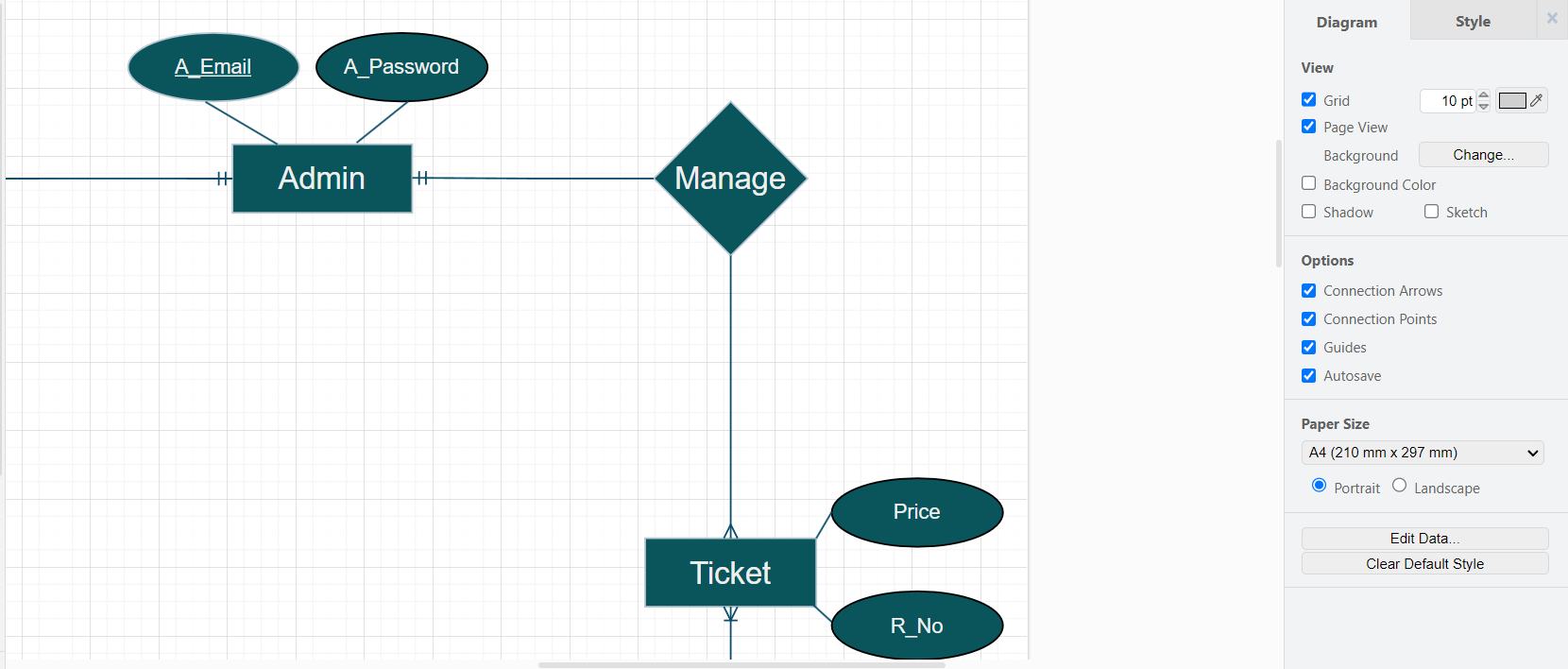
**3rd : T\_ID**, **R\_ID**

**3NF : 1st : T\_ID**, T\_Name

**2nd :** **R\_ID**, R\_Info

**3rd : T\_ID**, **R\_ID**

1. **Manage (**Admin **Manage** Ticket**)**



**UNF : A\_Email**, A­\_Password, R\_No, Price

**1NF : A\_Email**, A­\_Password, R\_No, Price

**2NF :**

**1st : A\_Email**, A\_Password

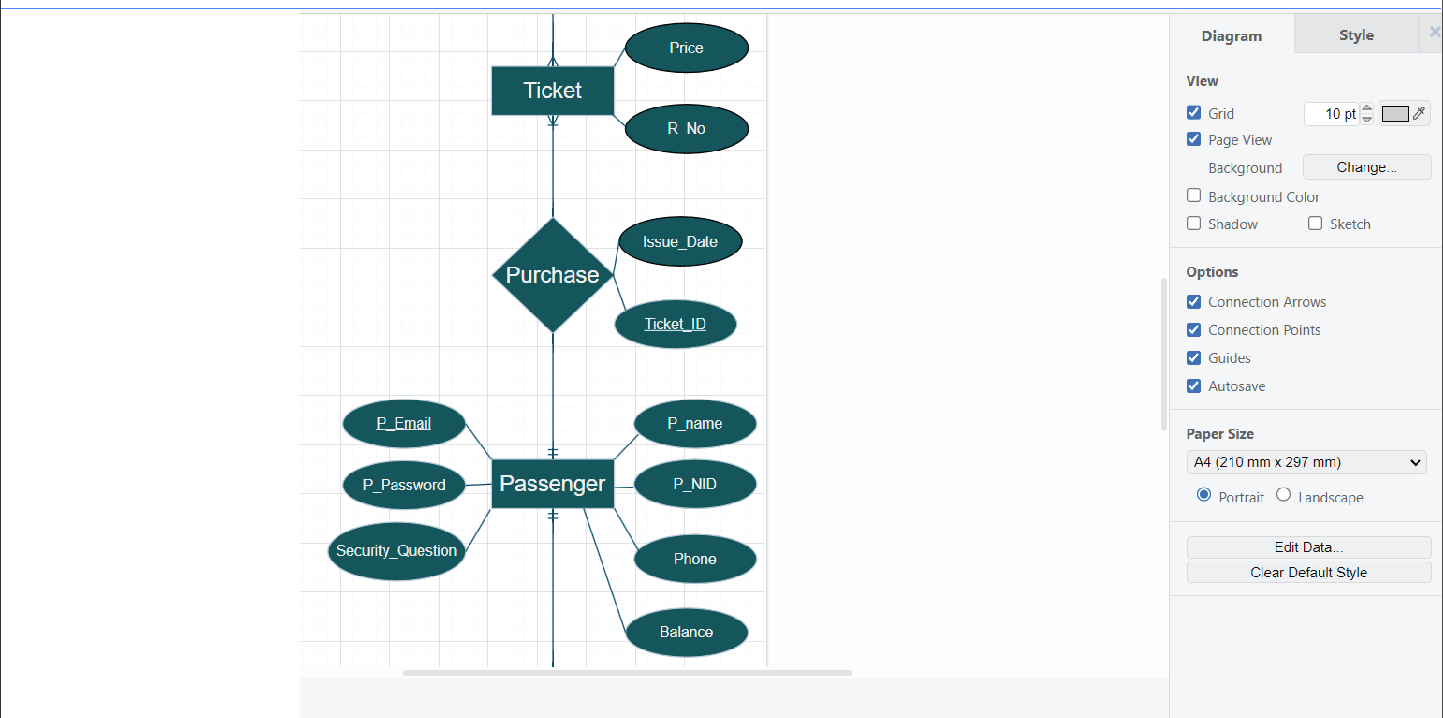
**2nd :** R\_No, Price, **A\_Email**

**3NF :**

**1st : A\_Email**, A­\_Password

**2nd :** R\_No, Price, **A\_Email**

1. **Purchase (**Passenger **Purchase** Ticket**)**



**UNF :** R\_No, Price, **Ticket\_ID**, Issue\_Date, **P\_email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**1NF : Ticket\_ID**, Issue\_Date, R\_No, Price, **P\_email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**2NF :**

**1st : Ticket\_ID**, **P\_email**, R\_No, Price, Issue\_Date

**2nd :** **P\_email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**3NF :**

**1st : Ticket\_ID**, **P\_email**, R\_No, Price, Issue\_Date

**2nd :** **P\_email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

1. **Has (**Passenger **Has** Ticket History**)**

A diagram of a ticket history

Description automatically generated

**UNF : Ticket­\_ID**, T\_Name, Departure\_satation, Arrival\_station, Issue\_Time, Price, **P\_Email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**1NF : Ticket­\_ID**, T\_Name, Departure\_satation, Arrival\_station, Issue\_Time, Price, **P\_Email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**2NF :**

**1st : Ticket­\_ID**, T\_Name, Departure\_satation, Arrival\_station, Issue\_Time, Price, **P\_Email**

**2nd :** **P\_Email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**3NF :**

**1st : Ticket­\_ID**, T\_Name, Departure\_satation, Arrival\_station, Issue\_Time, Price, **P\_Email**

**2nd :** **P\_Email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

**Finalization**

1st : **A\_Email**, A\_Password

2nd **:** **T\_ID**, T\_Name, **A\_Email**

3rd : **T\_ID**, T\_Name

4th : **R\_ID**, R\_Info

5th : **T\_ID,** **R\_ID**

6th : **A\_Email**, R\_No, Price

7th : **Ticket\_ID**, **P\_email**, R\_No, Price, Issue\_Date

8th : **P email**, P\_Password, P\_Name, P\_NID, Phone, Balance, Security\_Question

9th : **Ticket­\_ID**, **P\_Email** , T\_Name, Departure\_satation, Arrival\_station, Issue\_Time, Price

**Schema Creation**

create user BANGLADESHRAILWAY identified by BANGLADESHRAILWAY;

grant connect, resource, unlimited tablespace to BANGLADESHRAILWAY;

ALTER USER BANGLADESHRAILWAY DEFAULT TABLESPACE USERS;

ALTER USER BANGLADESHRAILWAY TEMPORARY TABLESPACE TEMP;

Screens screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

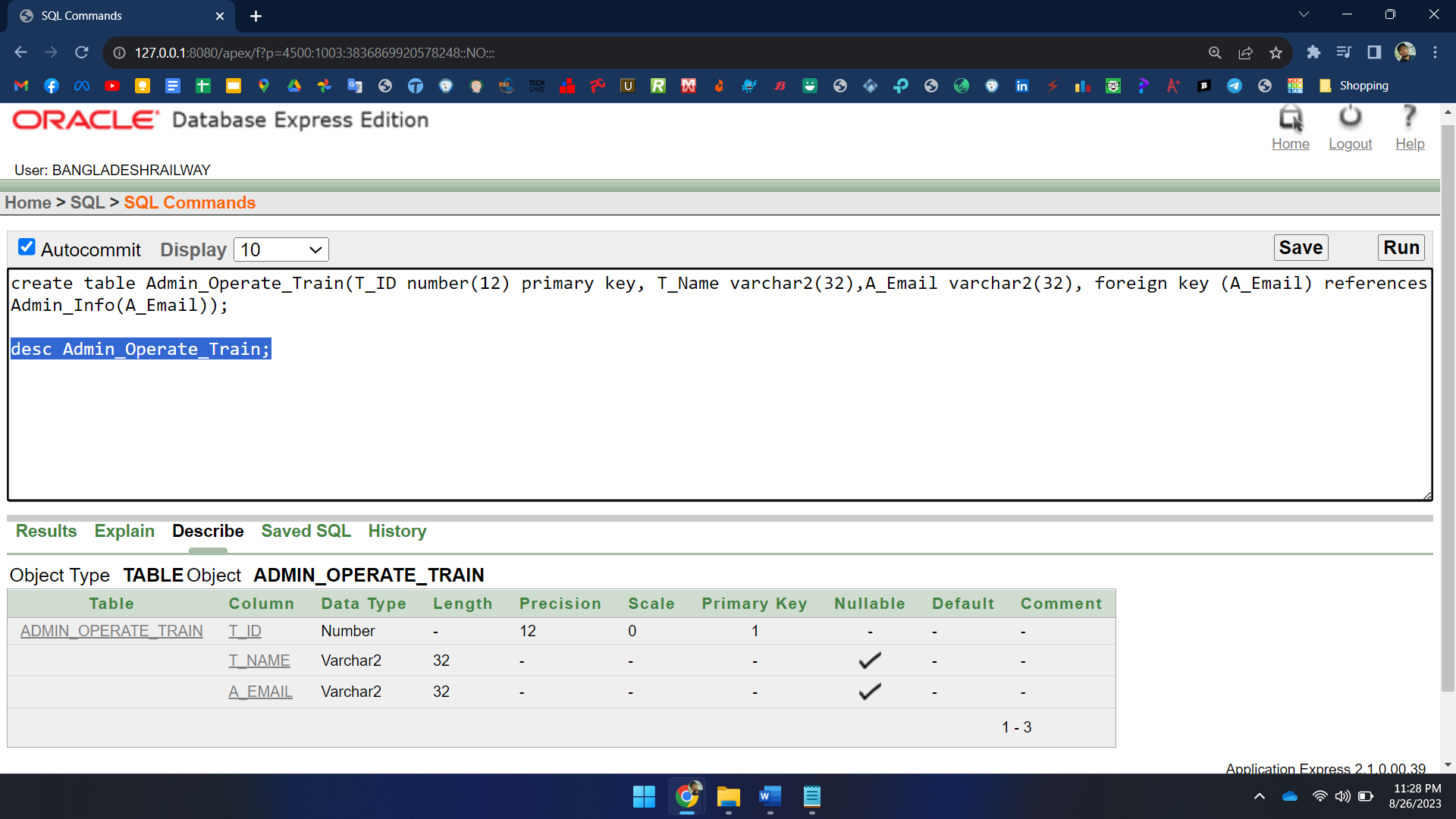
**Table Creation**

1. create table Admin\_Info (A\_Email varchar2(32) primary key, A\_Password varchar (32));

A screenshot of a computer

Description automatically generated

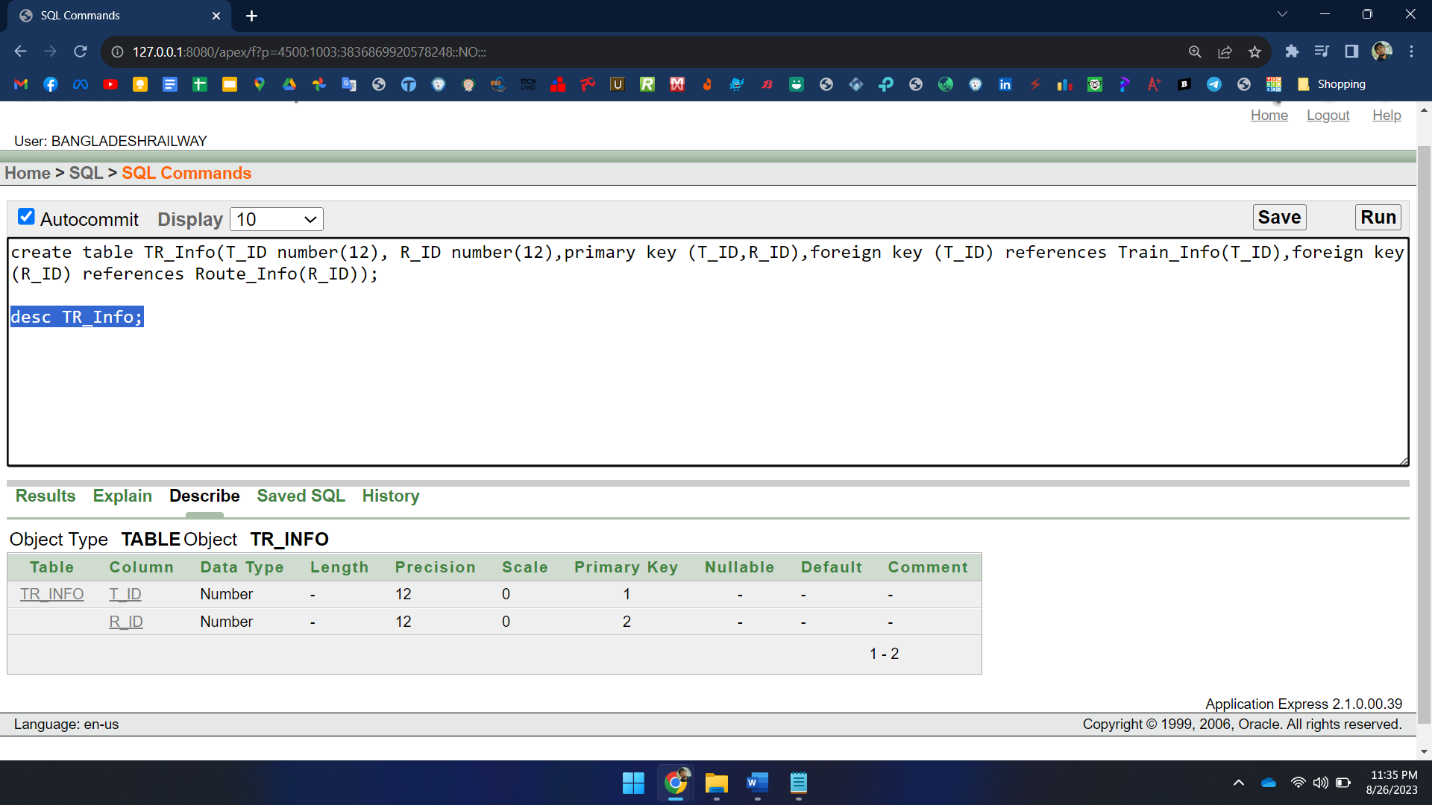
1. create table Admin\_Operate\_Train(T\_ID number(12) primary key, T\_Name varchar2(32),A\_Email varchar2(32), foreign key (A\_Email) references Admin\_Info(A\_Email));



1. A screenshot of a computer

   Description automatically generatedcreate table Train\_Info (T\_ID number (10) primary key, T\_Name varchar2(35) Not Null);
2. A screenshot of a computer

   Description automatically generatedcreate table Route\_Info (R\_ID number (10) primary key, R\_Info varchar2(35) Not Null);
3. create table TR\_Info (T\_ID number (12), R\_ID number (12), primary key (T\_ID, R\_ID), foreign key (T\_ID) references Train\_Info(T\_ID),foreign key (R\_ID) references Route\_Info(R\_ID));



1. A screenshot of a computer

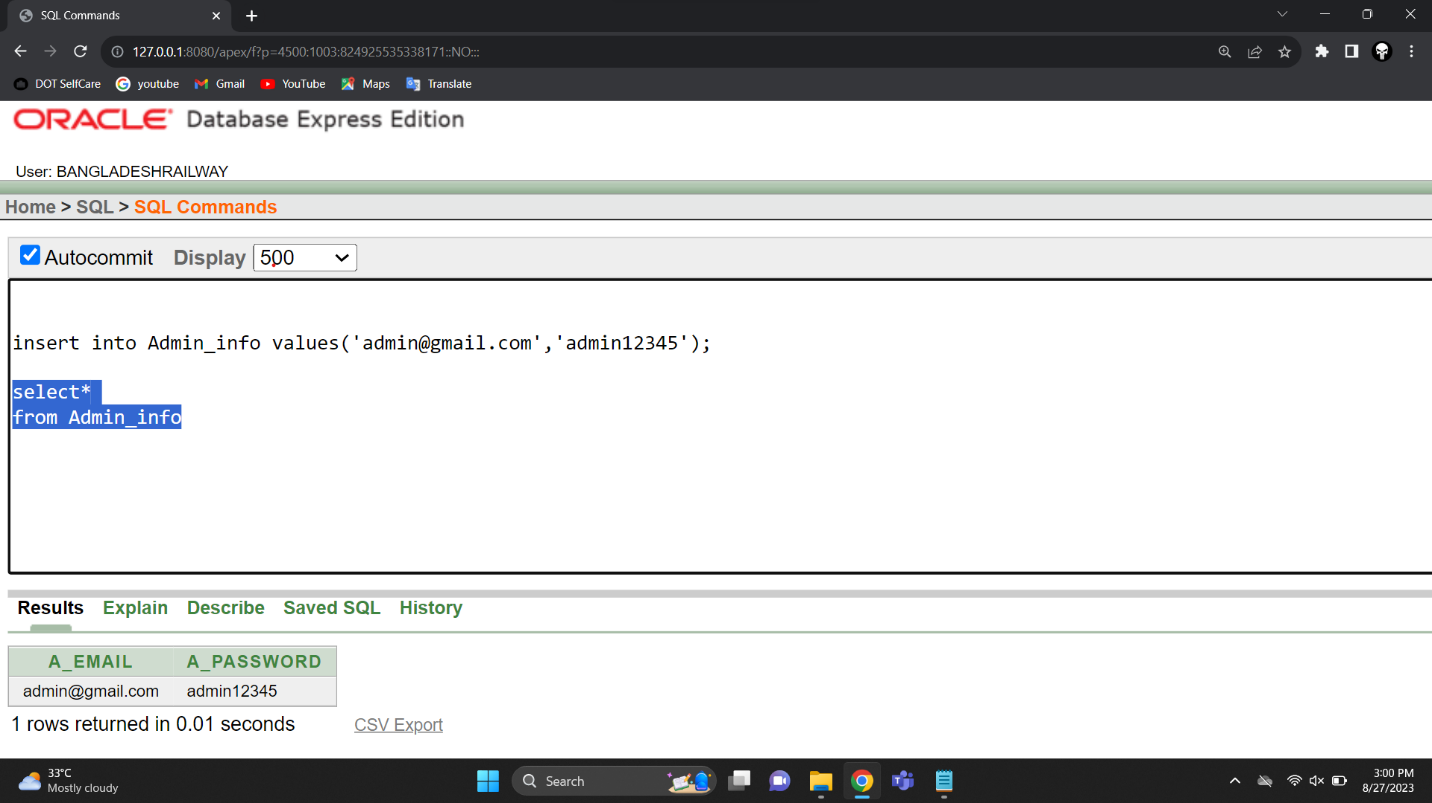
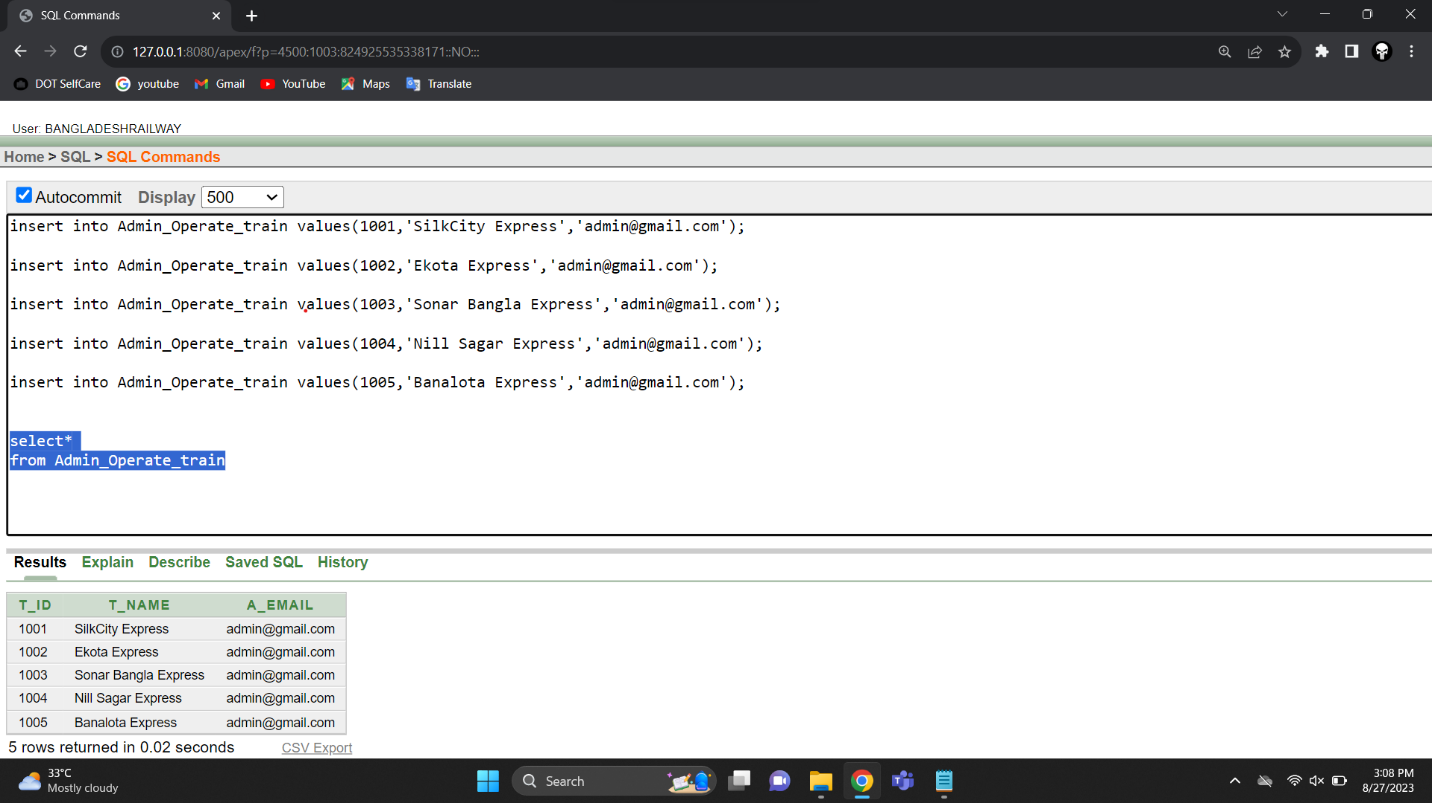
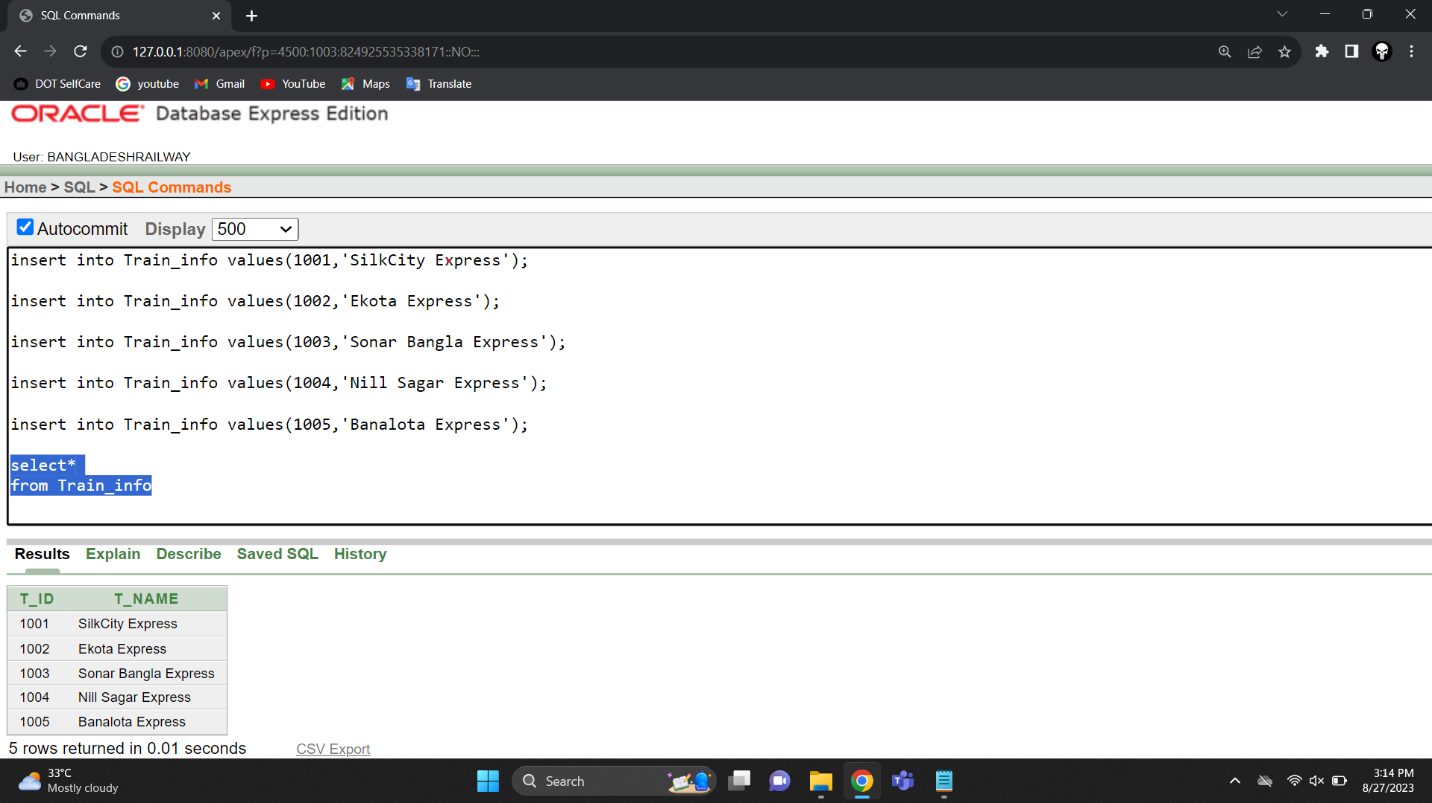
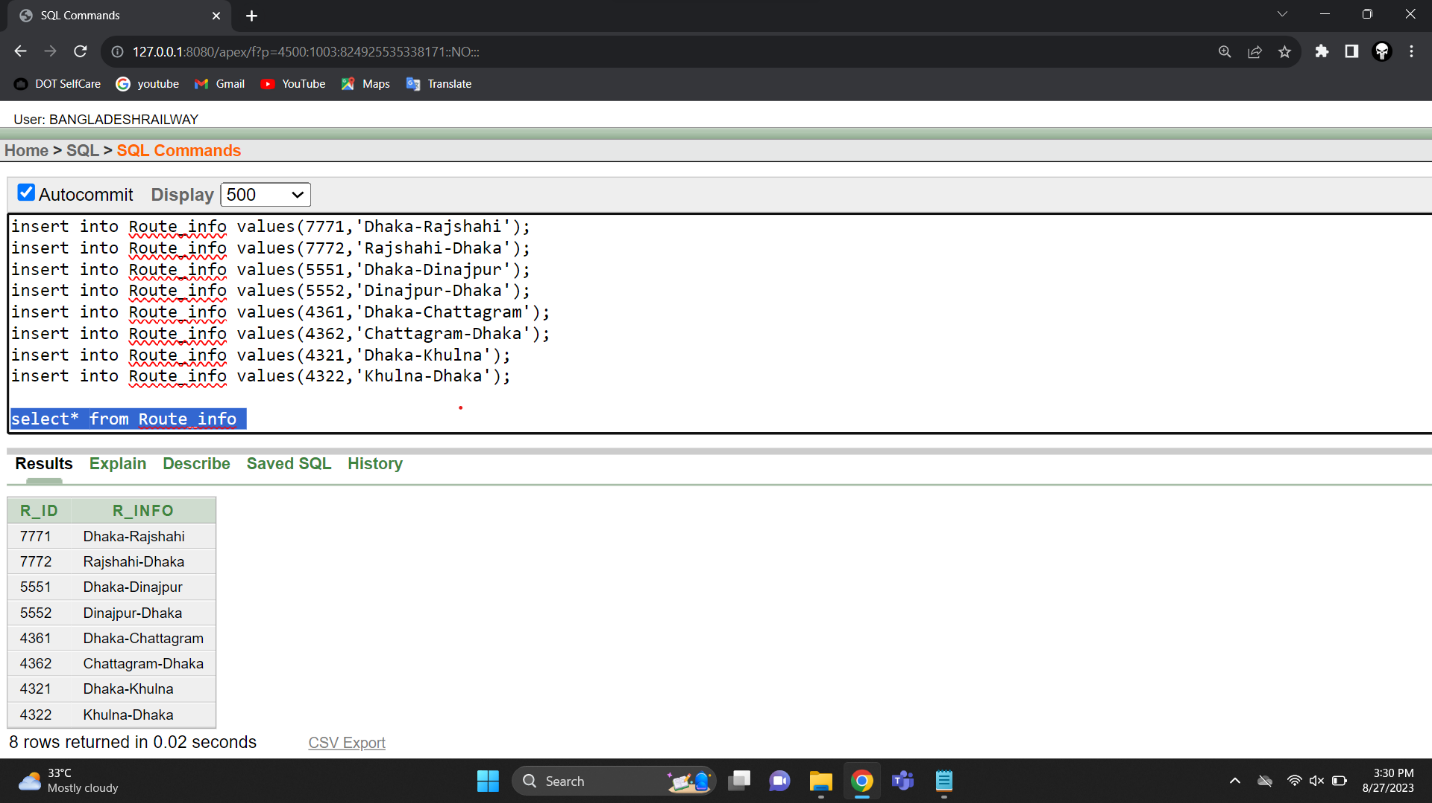
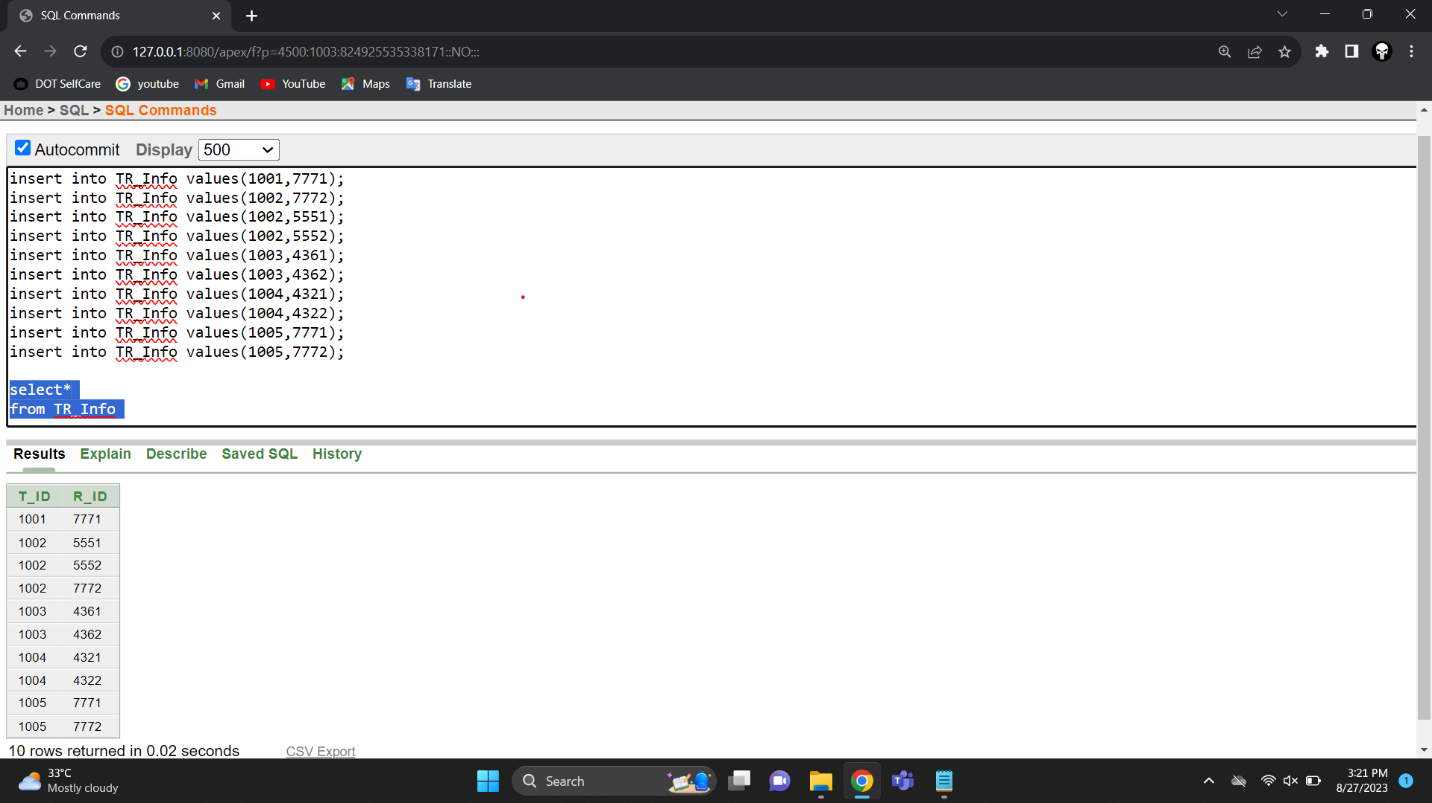
   Description automatically generatedcreate table Admin\_Manage\_Ticket (R\_No number (10), price float, A\_Email varchar2(32), foreign key (A\_Email) references Admin\_Info(A\_Email));
2. A screenshot of a computer

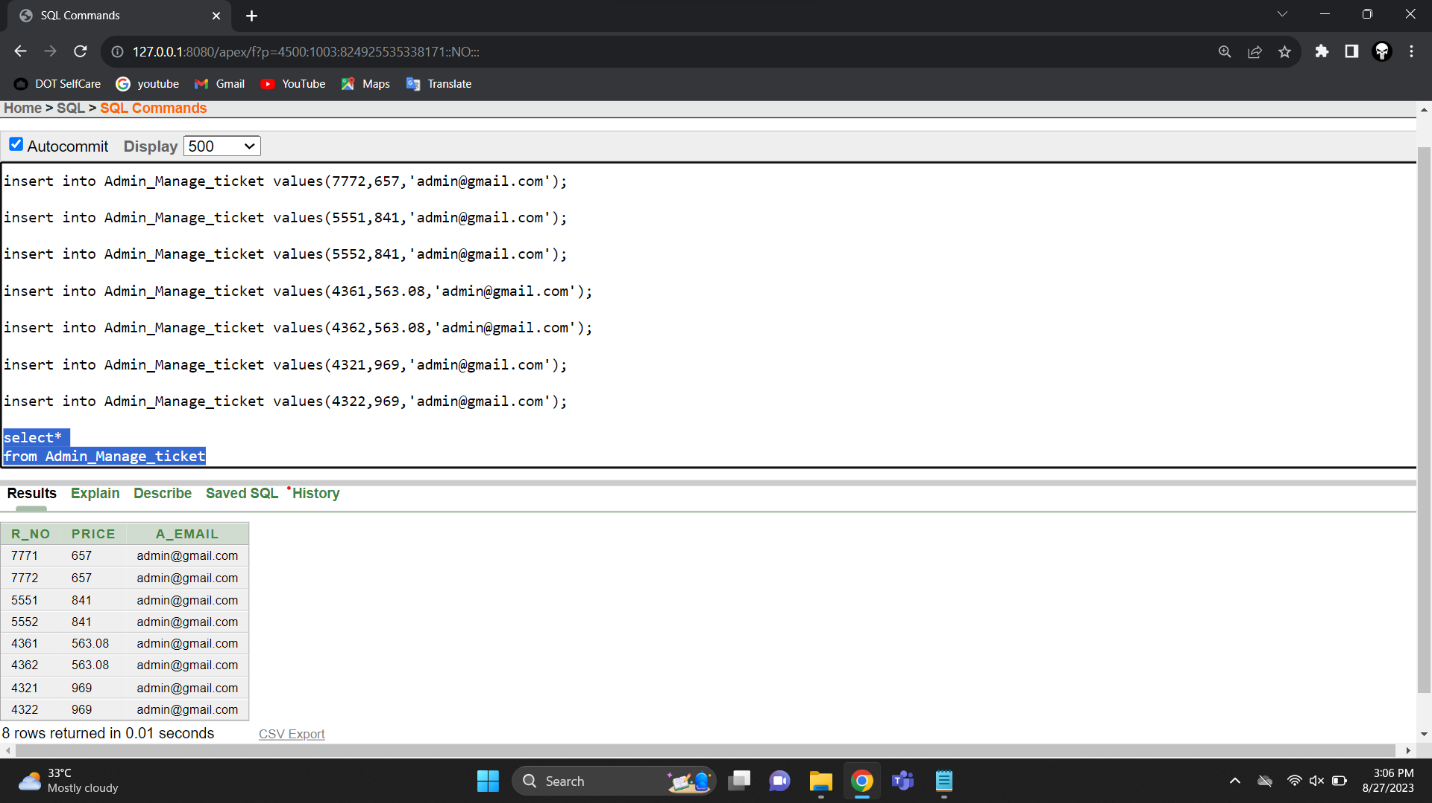
   Description automatically generatedcreate table Ticket\_Info (R\_No number (12) Not Null, Ticket\_ID number (12) primary key, Price float, Issue\_Date varchar2(32), P\_Email varchar2(32), foreign key (P\_Email) references Passenger\_Details(P\_Email));
3. A screenshot of a computer

   Description automatically generatedcreate table Passenger\_Details (P\_Email varchar2(32) primary key, P\_Name varchar2(30) Not Null, P\_NID number (15) Not Null, P\_Password varchar2(30) Not Null, Security\_Question varchar2(50), Balance float, Phone number(12));
4. A screenshot of a computer

   Description automatically generatedcreate table Ticket\_History (Ticket\_ID number (15) primary key, T\_Name varchar2(32), Departure\_sattion varchar2(32), Arrival\_station varchar2(32), Issue\_time varchar2(32), Price float, P\_Email varchar (32), foreign key (P\_Email) references Passenger\_Details(P\_Email));

**Data Insertion**

1. **Admin\_Info**
2. **Admin\_Operate\_Train**
3. **Train\_Info**
4. **Route\_Info**
5. **TR\_Info**
6. **Admin\_Manage\_Ticket**

****

1. **Passenger\_Details**

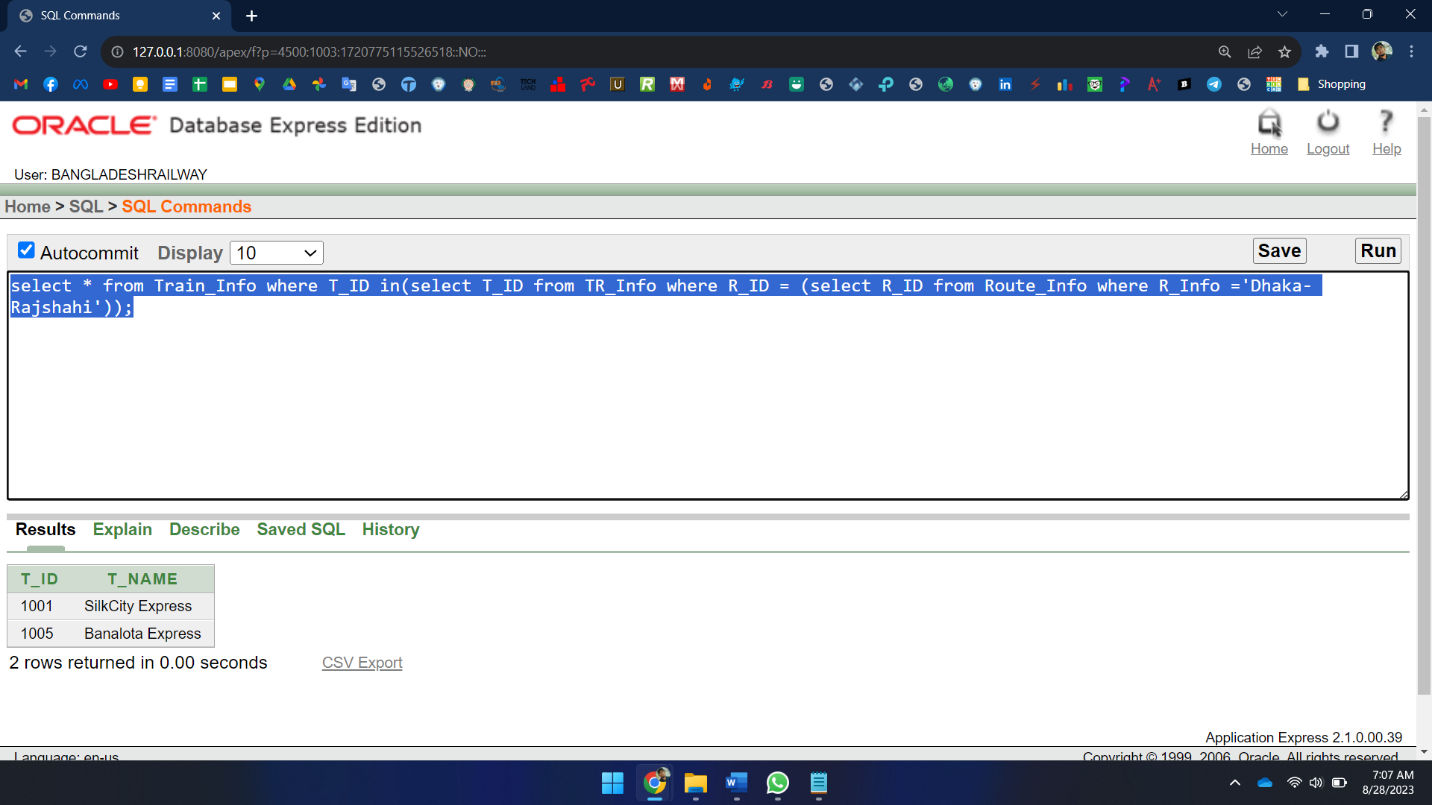
A screenshot of a computer

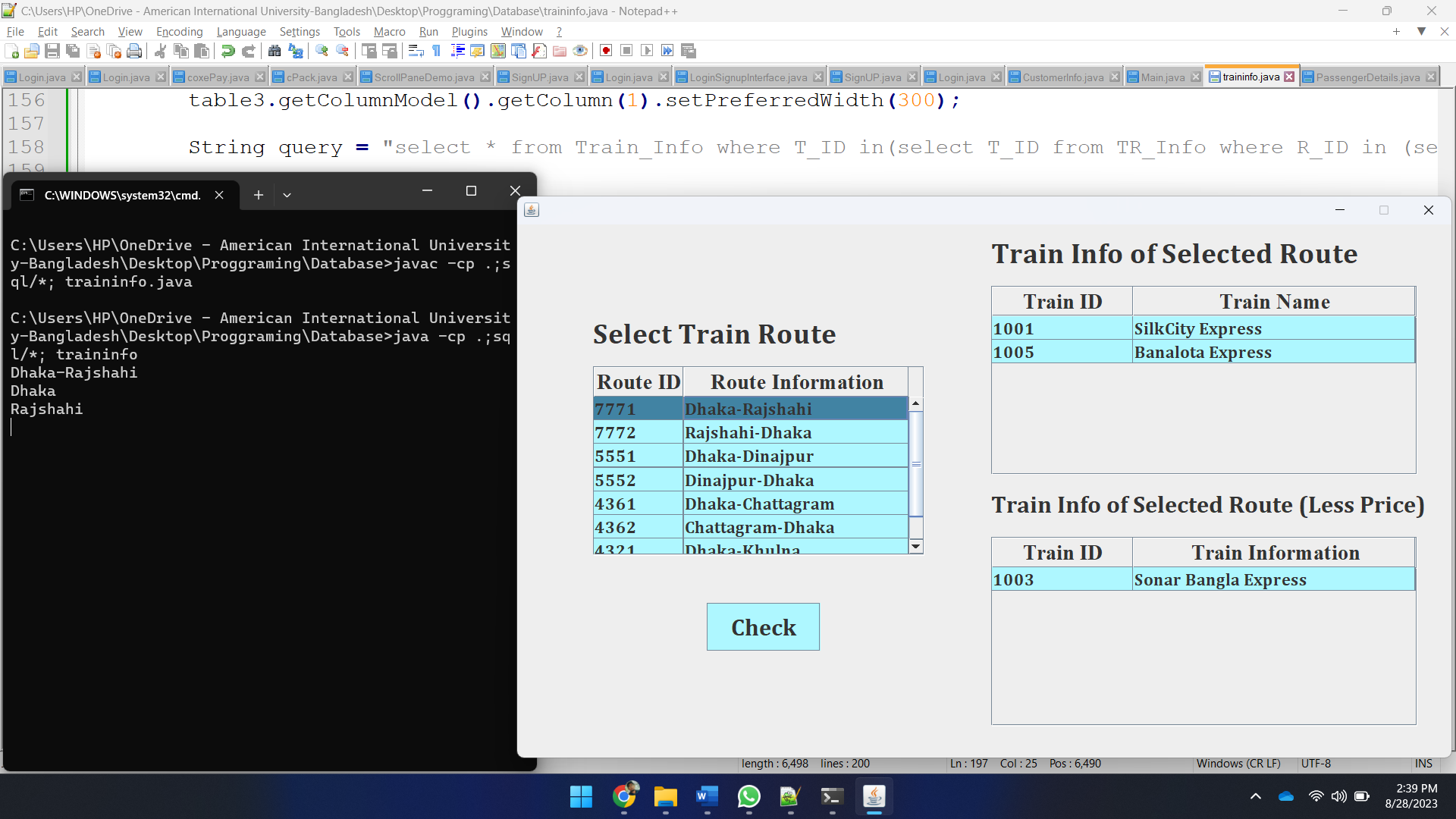
Description automatically generated

**Subquery**

1. **Show all the train Information that has a route of Dhaka to Rajshahi.**

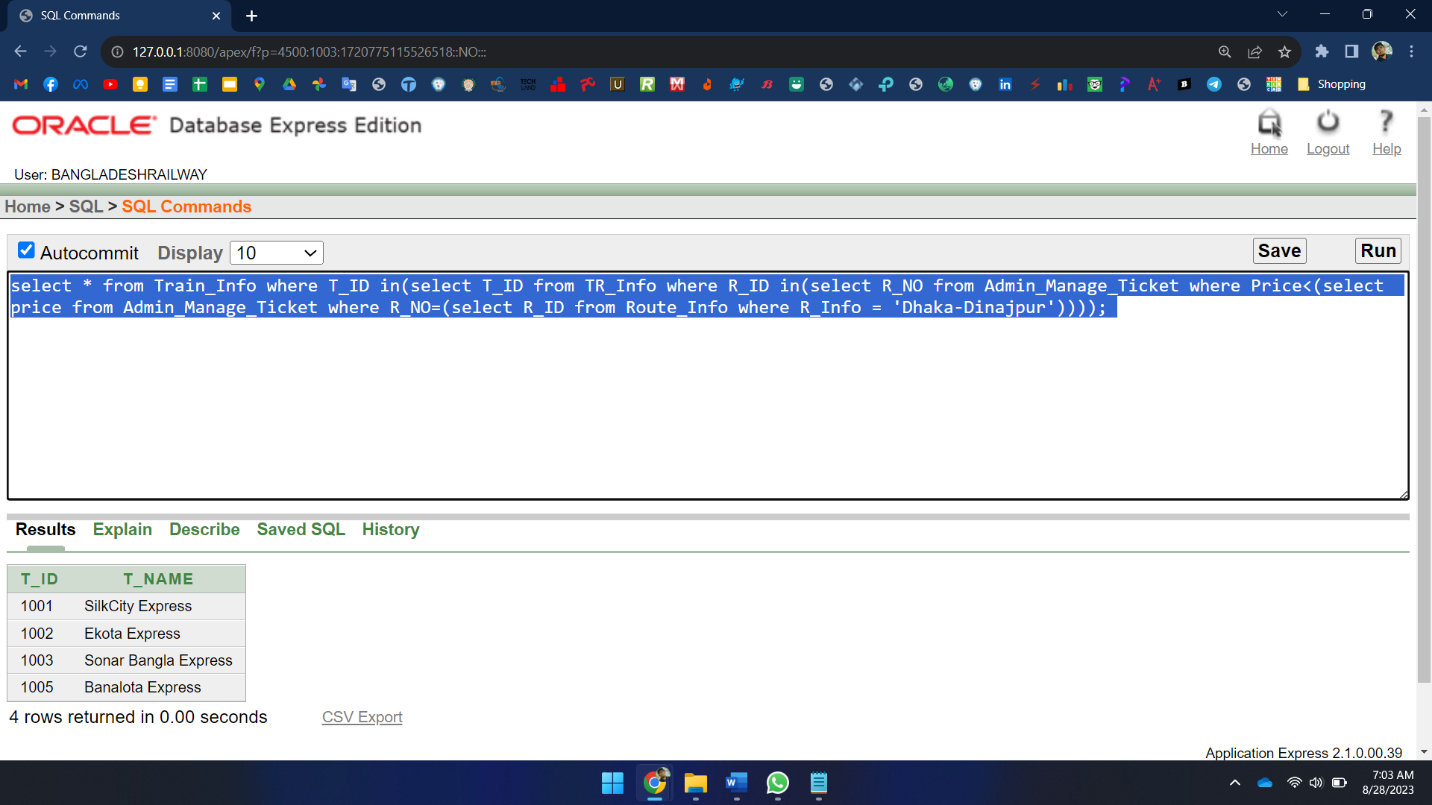
select \* from Train\_Info where T\_ID in(select T\_ID from TR\_Info where R\_ID = (select R\_ID from Route\_Info where R\_Info ='Dhaka-Rajshahi'));





1. **Show all the Train information those who have less fare price than Dhaka to Dinajpur route.**

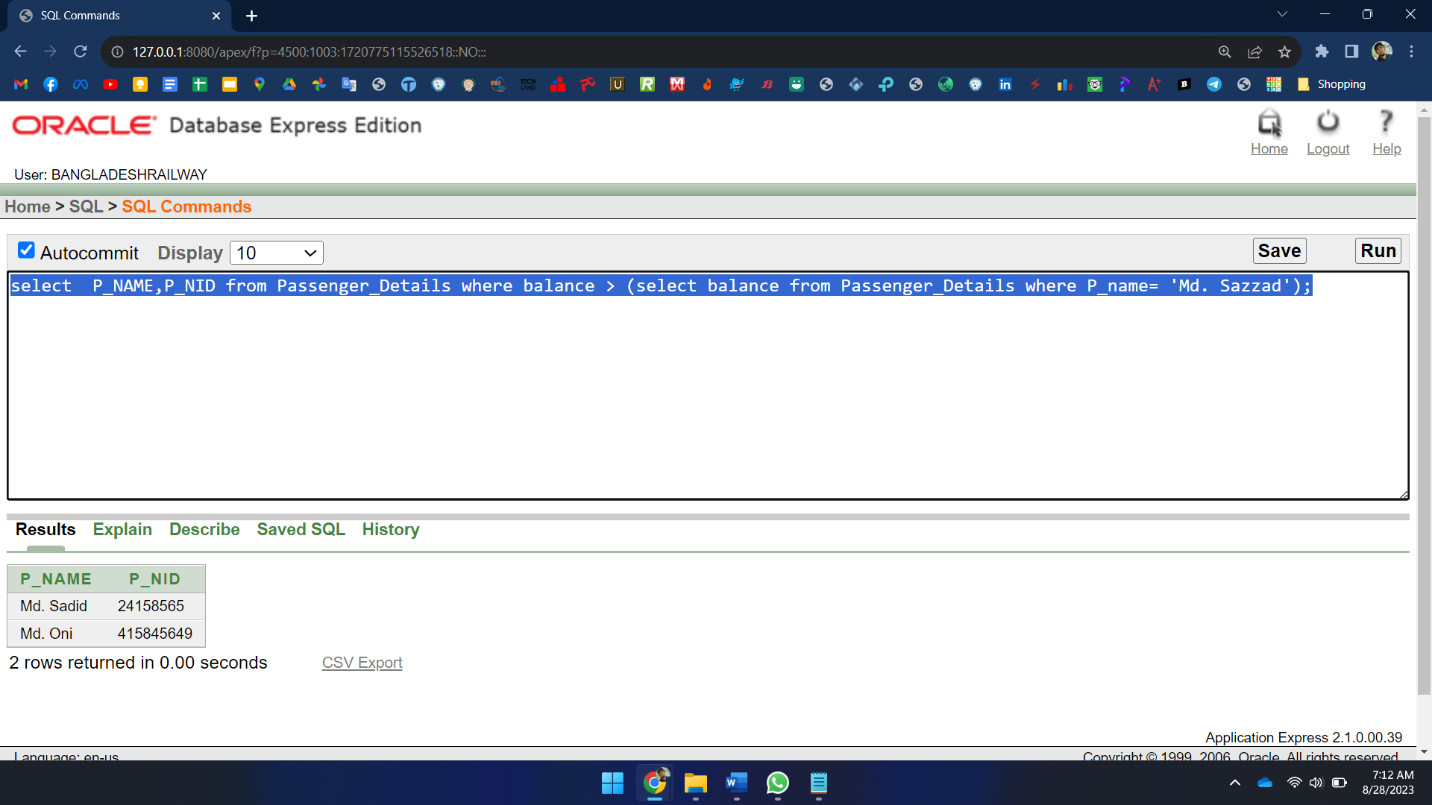
select \* from Train\_Info where T\_ID in(select T\_ID from TR\_Info where R\_ID in (select R\_NO from Admin\_Manage\_Ticket where Price<(select price from Admin\_Manage\_Ticket where R\_NO=(select R\_ID from Route\_Info where R\_Info = 'Dhaka-Dinajpur'))));

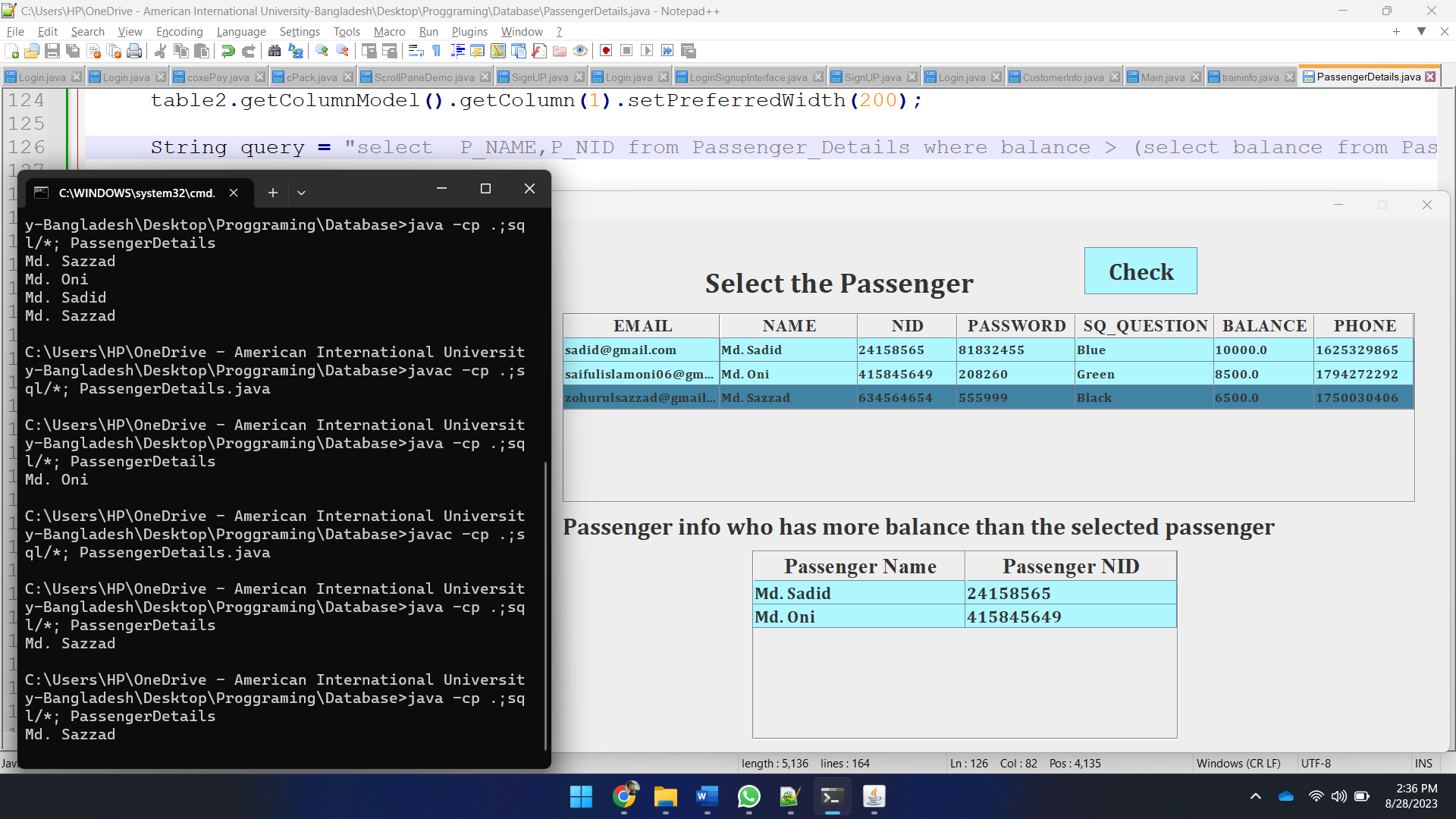


A screenshot of a computer

Description automatically generated

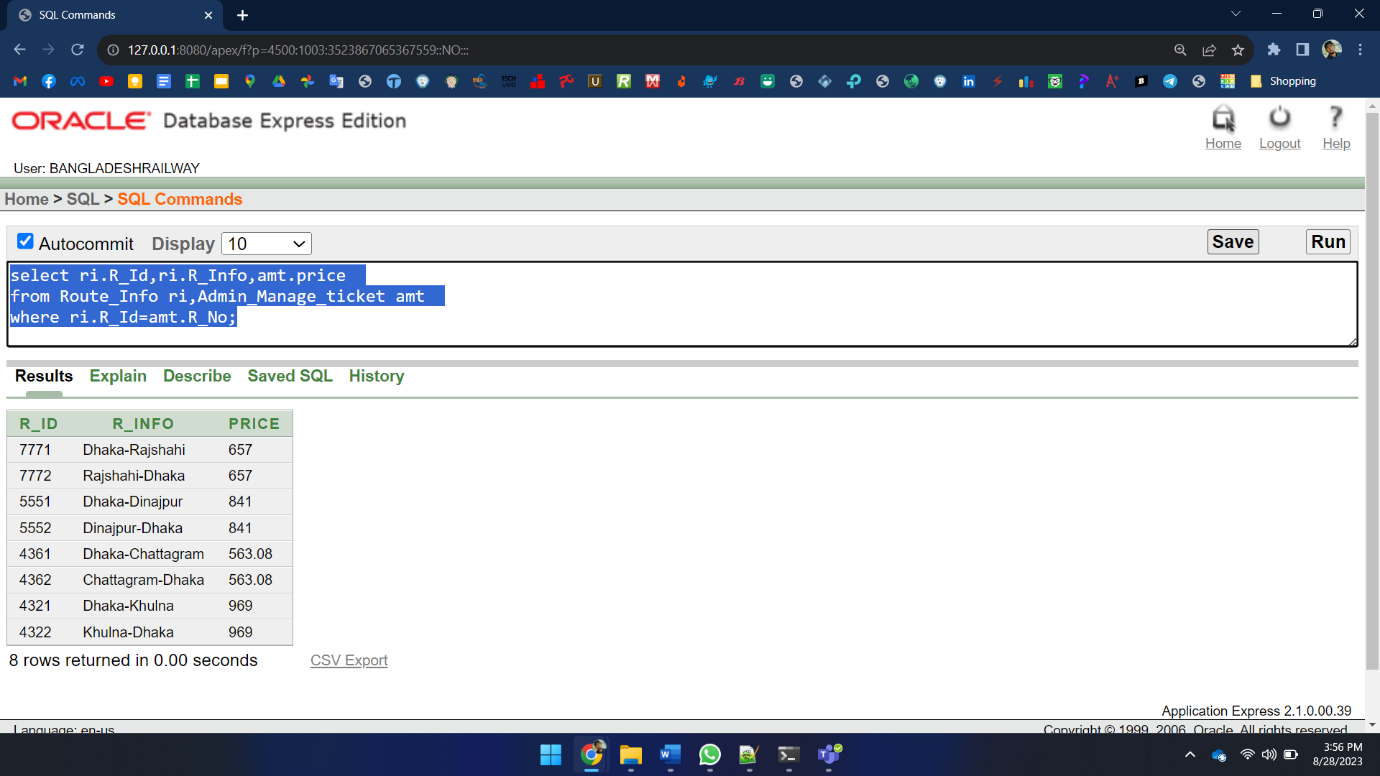
1. **Show all the user names & NID who have more balance than Md. Sazzad.**

 select P\_NAME,P\_NID from Passenger\_Details where balance > (select balance from Passenger\_Details where P\_name= 'Md. Sazzad');



**Joining Query**

1. **Equijoin**
   1. **Show the Route information with price.**

select ri.R\_Id,ri.R\_Info,amt.price from Route\_Info ri,Admin\_Manage\_ticket amt where ri.R\_Id=amt.R\_No;

1. **Outer join** 
   1. **Show all the train information with route ID for all the existing trains.**

select t.\*, tr.r\_id from train\_info t, TR\_Info tr where t.t\_id=tr.t\_id(+);

A computer screen with a white screen

Description automatically generated

* 1. **Show all the route information with Train ID for all the existing routes.**

A computer screen shot of a computer

Description automatically generatedselect r.\*, tr.t\_id from route\_info r, tr\_info tr where tr.r\_id(+)=r.r\_id order by r.r\_id;

1. **Self-join** 
   1. **Show the names who has more balance than whom. And rename the column as “Person1 has more balance than Person2”.**

select a.P\_NAME ||' '||'Has more balance than'||' '|| b.P\_NAME

from PASSENGER\_DETAILS a, PASSENGER\_DETAILS b

where a.BALANCE>b.BALANCE;

A screenshot of a computer

Description automatically generated