A picture containing text, logo, trademark, design

Description automatically generated

**AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)**

**FACULTY OF SCIENCE & TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**INTRODUCTION TO DATABASE**

**Spring 2022-2023**

**Section: M, Group: 01**

**PROJECT ON**

***Air Ticket Booking & Boarding Management System***

**Supervised By**

**KAWSER IROM RUSHEE**

**Submitted By**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Contribution** |
| **1. NILOY, NAFIUR RAHMAN** | **22-46459-1** |  |
| **2. RAHMAN, AZMINUR** | **22-46588-1** |  |
| **3. KUNDU, SAIKOT** | **22-46615-1** |  |
| **4. SARWAR, MD. SAKIB** | **22-46625-1** |  |

Date of Submission: **May 17, 2023**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SL No** | **Topic Name** | **Page No** |
| 01. | Cover Page | 01 |
| 02. | Table of Contents | 02 |
| 03. | Introduction | 03 |
| 04. | Scenario | 03 |
| 05. | ER-DIAGRAM (Screenshot) | 04 |
| 06. | ER-DIAGRAM (Final) | 05 |
| 07. | Normalization | 06-10 |
| 08. | Final Tables | 10 |
| 09. | Table Creation | 10-12 |
| 10. | Table Creation (Screenshot) | 12-16 |
| 11. | Data Insertion | 17-19 |
| 12. | Joining | 20-21 |
| 13. | Subquery | 22-23 |
| 14. | View | 24 |
| 15. | Add Constraint | 25 |



**Introduction:**

‘Uran’ is a company that operates an air ticket booking and boarding management system. The system is managed by managers who oversee ticket agents. Ticket agents sell tickets to customers. Customers can buy tickets for themselves or for other passengers. Passenger must need ticket for boarding in planes. The company also store the route of planes.

**Scenario:**

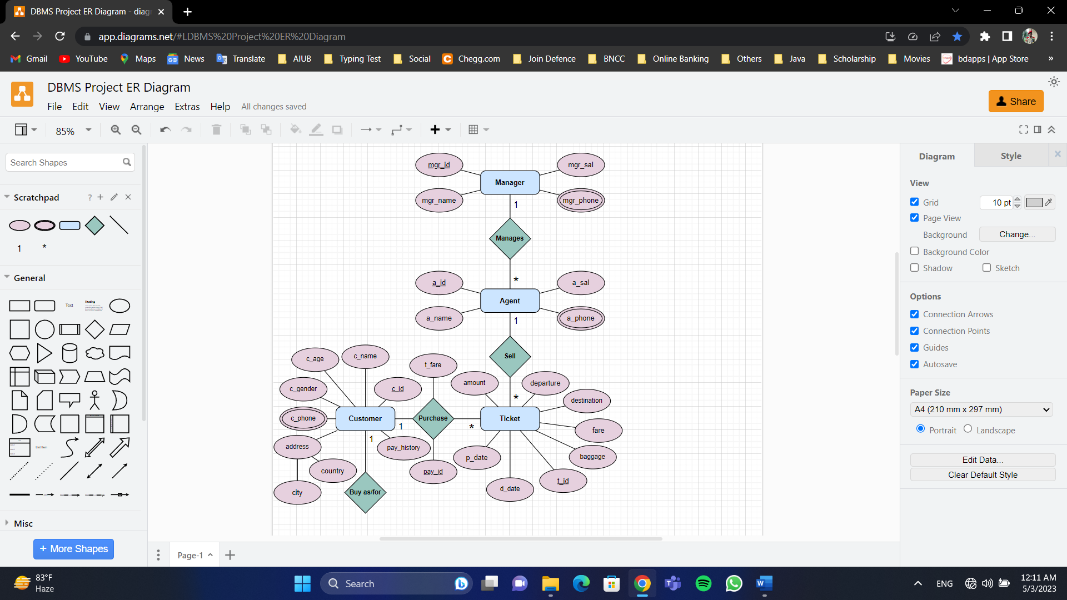
The owners of an air ticket booking and boarding management system ‘Uran’ wants to create a database management system for their company. In this company A manager can **manage** many ticket agent. Managers can be identified using their unique manager ID. Each manager's name, phone number, and salary are stored in the database. One or more ticket agent may work under a manager. A ticket agent can **sell** many ticket. Every agent’s name, phone, salary, as well as a unique agent ID is recorded. Tickets have their fare, amount, departure, destination, departure date, purchase date and baggage stored as records along with a unique ticket ID for ticket. A customer must **purchase** ticket for traveling. During purchasing ticket payment id and total fare are also stored in the database. Customers are identified by their unique Customer ID. Aside from that, a customer's name, address with city & country, phone number, age, gender, and payment history (if available) are also recorded in the database. A customer may be a passenger themselves or **buy as/for** tickets for another passenger. Passengers can be distinctly identified with the help of their Passenger ID. Additionally, a passenger's name, address with city & country, age, phone number, and gender are also stored within the database. Passengers need tickets for traveling. Passengers will use their tickets for **boarding** to plane. Each plane has a distinct plane ID alongside its terminal number, take-off airport, take-off time, destination and capacity. Every passenger **sits on** a seat. Every seat identified by unique seat id alongside seat type. A plane has many seats. A plane **travel in** one route towards their destination. These routes are distinctly marked using a route ID and it’s had distance.

**ER-DIAGRAM (Screenshot):**

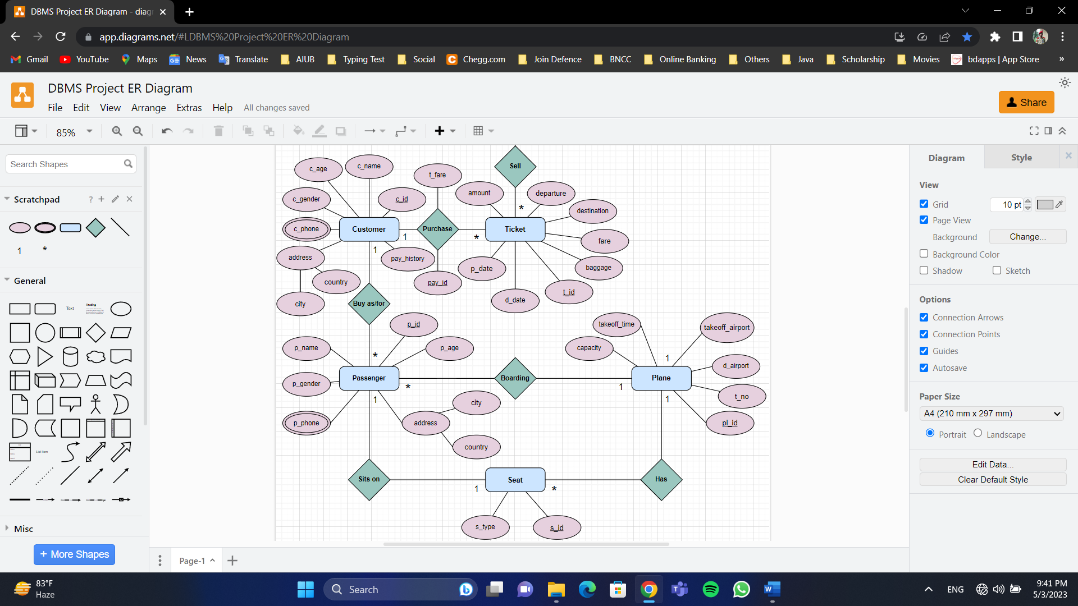
A screenshot of a computer

Description automatically generated

At the beginning



At the Middle



At nearly end

**ER-DIAGRAM (Final):**

A picture containing screenshot, circle, pattern, colorfulness

Description automatically generated

ER-DIAGRAM

**Normalization:**

* **Manages**

**UNF:** mgr\_id, mgr\_name, mgr\_sal, mgr\_phone, a\_id, a\_name, a\_sal, a\_phone

**1NF:** (**mgr\_phone** and **a\_phone** is multivalued attribute)

mgr\_id, mgr\_name, mgr\_sal, mgr\_phone, a\_id, a\_name, a\_sal, a\_phone

**2NF:**

1. mgr\_id, mgr\_name, mgr\_sal
2. mgr\_phone*,* mgr\_id (fk)
3. a\_id, a\_name, a\_sal, mgr\_id (fk)
4. a\_phone*,* a\_id (fk)

**3NF:** (No Transitive Dependency)

1. mgr\_id, mgr\_name, mgr\_sal
2. mgr\_phone*,* mgr\_id (fk)
3. a\_id, a\_name, a\_sal, mgr\_id (fk)
4. a\_phone*,* a\_id (fk)

**Tables:**

1. mgr\_id, mgr\_name, mgr\_sal
2. mgr\_phone*,* mgr\_id (fk)
3. a\_id, a\_name, a\_sal, mgr\_id (fk)
4. a\_phone*,* a\_id (fk)

* **Sell**

**UNF:** a\_id, a\_name, a\_sal, a\_phone, t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage

**1NF:** (**a\_phone**is a multivalued attribute)

a\_id, a\_name, a\_sal, a\_phone, t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage

**2NF:**

1. a\_id, a\_name, a\_sal
2. a\_phone, a\_id (fk)
3. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id (fk)

**3NF:** (No Transitive Dependency)

1. a\_id, a\_name, a\_sal
2. a\_phone, a\_id (fk)
3. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id (fk)

**Tables:**

1. a\_id, a\_name, a\_sal
2. a\_phone, a\_id (fk)
3. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id (fk)

* **Purchase**

**UNF:** t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, c\_id, c\_name, c\_age, c\_gender, city, country, c\_phone, pay\_history, pay\_id, t\_fare

**1NF:** (**c\_phone** is a multivalued attribute)

t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, c\_id, c\_name, c\_age, c\_gender, city, country, c\_phone, pay\_history, pay\_id, t\_fare

**2NF:**

1. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, c\_id (fk)
2. c\_id, c\_name, c\_age, c\_gender, city, country, pay\_history
3. c\_phone, c\_id (fk)
4. pay\_id, t\_fare, c\_id (fk), t\_id (fk)

**3NF:** (Transitive Dependency)

1. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, c\_id (fk)
2. c\_id, c\_name, c\_age, c\_gender, pay\_history
3. city, country, c\_id (fk)
4. c\_phone, c\_id (fk)
5. pay\_id, t\_fare, c\_id (fk), t\_id (fk)

**Tables:**

1. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, c\_id (fk)
2. c\_id, c\_name, c\_age, c\_gender, pay\_history
3. city, country, c\_id (fk)
4. c\_phone, c\_id (fk)
5. pay\_id, t\_fare, c\_id (fk), t\_id (fk)

* **Buy As/For**

**UNF:** c\_id, c\_name, c\_age, c\_gender, city, country, c\_phone, pay\_history, p\_id, p\_name, p\_gender, p\_age, city, country, p\_phone

**1NF:** (**c\_phone**, **p\_phone** is a multivalued attribute)

c\_id, c\_name, c\_age, c\_gender, city, country, c\_phone, pay\_history, p\_id, p\_name, p\_gender, p\_age, city, country, p\_phone

**2NF:**

1. c\_id, c\_name, c\_age, c\_gender, city, country, pay\_history
2. c\_phone, c\_id (fk)
3. p\_id, p\_name, p\_gender, p\_age, city, country, c\_id (fk)
4. p\_phone, p\_id (fk)

**3NF:** (Transitive Dependency)

1. c\_id, c\_name, c\_age, c\_gender, pay\_history
2. city, country, c\_id (fk)
3. c\_phone, c\_id (fk)
4. p\_id, p\_name, p\_gender, p\_age, c\_id (fk)
5. city, country, p\_id (fk)
6. p\_phone, p\_id (fk)

**Table:**

1. c\_id, c\_name, c\_age, c\_gender, pay\_history
2. city, country, c\_id (fk)
3. c\_phone, c\_id (fk)
4. p\_id, p\_name, p\_gender, p\_age, c\_id (fk)
5. city, country, p\_id (fk)
6. p\_phone, p\_id (fk)

* **Boarding**

**UNF:** p\_id, p\_name, p\_gender, p\_age, city, country, p\_phone, pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**1NF:** (**p\_phone** is a multivalued attribute)

p\_id, p\_name, p\_gender, p\_age, city, country, p\_phone**,** pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**2NF:**

1. p\_id, p\_name, p\_gender, p\_age, city, country, pl\_id (fk)
2. p\_phone, p\_id (fk)
3. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**3NF:** (Transitive Dependency)

1. p\_id, p\_name, p\_gender, p\_age, pl\_id (fk)
2. city, country, p\_id (fk)
3. p\_phone, p\_id (fk)
4. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**Table:**

1. p\_id, p\_name, p\_gender, p\_age, pl\_id (fk)
2. city, country, p\_id (fk)
3. p\_phone, p\_id (fk)
4. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

* **Sits On**

**UNF:** p\_id, p\_name, p\_gender, p\_age, city, country, p\_phone, s\_id, s\_type

**1NF:** (**p\_phone** is a multivalued attribute)

p\_id, p\_name, p\_gender, p\_age, city, country, p\_phone**,** s\_id, s\_type

**2NF:**

1. p\_id, p\_name, p\_gender, p\_age, city, country
2. p\_phone, p\_id (fk)
3. s\_id, s\_type, p\_id (fk)

**3NF:** (Transitive Dependency)

1. p\_id, p\_name, p\_gender, p\_age
2. city, country, p\_id (fk)
3. p\_phone, p\_id (fk)
4. s\_id, s\_type, p\_id (fk)

**Table:**

1. p\_id, p\_name, p\_gender, p\_age
2. city, country, p\_id (fk)
3. p\_phone, p\_id (fk)
4. s\_id, s\_type, p\_id (fk)

* **Has**

**UNF:** s\_id, s\_type, pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**1NF:** (No multivalued attribute)

s\_id, s\_type, pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**2NF:**

1. s\_id, s\_type, pl\_id (fk)
2. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**3NF:** (No Transitive Dependency)

1. s\_id, s\_type, pl\_id (fk)
2. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

**Table:**

1. s\_id, s\_type, pl\_id (fk)
2. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity

* **Travels In**

**UNF:** pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity, r\_id, distance

**1NF:** (No multivalued attribute)

pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity, r\_id, distance

**2NF:**

1. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
2. r\_id, distance, pl\_id (fk)

**3NF:** (No Transitive Dependency)

1. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
2. r\_id, distance, pl\_id (fk)

**Table:**

1. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
2. r\_id, distance, pl\_id (fk)

* **Tables**

1. mgr\_id, mgr\_name, mgr\_sal
2. mgr\_phone*,* mgr\_id (fk)
3. a\_id, a\_name, a\_sal, mgr\_id (fk)
4. a\_phone*,* a\_id (fk)
5. a\_id, a\_name, a\_sal
6. a\_phone, a\_id (fk)
7. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id (fk)
8. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, c\_id (fk)
9. c\_id, c\_name, c\_age, c\_gender, pay\_history
10. city, country, c\_id (fk)
11. c\_phone, c\_id (fk)
12. pay\_id, t\_fare, c\_id (fk), t\_id (fk)
13. c\_id, c\_name, c\_age, c\_gender, pay\_history
14. city, country, c\_id (fk)
15. c\_phone, c\_id (fk)
16. p\_id, p\_name, p\_gender, p\_age, c\_id (fk)
17. city, country, p\_id (fk)
18. p\_phone, p\_id (fk)
19. p\_id, p\_name, p\_gender, p\_age, pl\_id (fk)
20. city, country, p\_id (fk)
21. p\_phone, p\_id (fk)
22. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
23. p\_id, p\_name, p\_gender, p\_age
24. city, country, p\_id (fk)
25. p\_phone, p\_id (fk)
26. s\_id, s\_type, p\_id (fk)
27. s\_id, s\_type, pl\_id (fk)
28. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
29. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
30. r\_id, distance, pl\_id (fk)

**Final Tables**

1. mgr\_id, mgr\_name, mgr\_sal
2. mgr\_phone*,* mgr\_id (fk)
3. a\_id, a\_name, a\_sal, mgr\_id (fk)
4. a\_phone*,* a\_id (fk)
5. cc\_id, city, country
6. c\_id, c\_name, c\_age, c\_gender, pay\_history, cc\_id (fk)
7. c\_phone, c\_id (fk)
8. t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id (fk), c\_id (fk)
9. pay\_id, t\_fare, c\_id (fk), t\_id (fk)
10. pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity
11. p\_id, p\_name, p\_gender, p\_age, c\_id (fk), pl\_id (fk), cc\_id (fk)
12. p\_phone, p\_id (fk)
13. s\_id, s\_type, p\_id (fk), pl\_id (fk)
14. r\_id, distance, pl\_id (fk)

**Table Creation**

1. CREATE TABLE manager (mgr\_id NUMBER(3) CONSTRAINT pk\_manager PRIMARY KEY,mgr\_name VARCHAR2(30) NOT NULL,mgr\_sal FLOAT NOT NULL CONSTRAINT manager\_sal check(mgr\_sal between 50000 AND 100000));
2. CREATE TABLE manager\_contact(mgr\_phone NUMBER(11) CONSTRAINT pk\_mgr\_contact PRIMARY KEY, mgr\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_mgr\_contact FOREIGN KEY(mgr\_id) REFERENCES manager(mgr\_id));
3. CREATE TABLE agent(a\_id NUMBER(3) CONSTRAINT pk\_agent PRIMARY KEY, a\_name VARCHAR2(30) NOT NULL, a\_sal FLOAT NOT NULL CONSTRAINT agent\_sal check(a\_sal BETWEEN 20000 AND 50000), mgr\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_agent FOREIGN KEY(mgr\_id) REFERENCES manager(mgr\_id));
4. CREATE TABLE agent\_contact(a\_phone NUMBER(11) CONSTRAINT pk\_agent\_contact PRIMARY KEY, a\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_agent\_contact FOREIGN KEY(a\_id) REFERENCES agent(a\_id));
5. CREATE TABLE address(cc\_id NUMBER(3) CONSTRAINT pk\_address PRIMARY KEY, city VARCHAR2(30) NOT NULL, country VARCHAR2(30) NOT NULL);
6. CREATE TABLE customer(c\_id NUMBER(10) CONSTRAINT pk\_customer PRIMARY KEY, c\_name VARCHAR2(30) NOT NULL, c\_age NUMBER(3) NOT NULL CONSTRAINT c\_age CHECK(c\_age>0), c\_gender VARCHAR2(6) NOT NULL CONSTRAINT c\_gender CHECK (c\_gender IN ('MALE', 'FEMALE', 'OTHERS')), pay\_history NUMBER(10), cc\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_address\_customer FOREIGN KEY (cc\_id) REFERENCES address(cc\_id));
7. CREATE TABLE customer\_contact(c\_phone NUMBER(11) CONSTRAINT pk\_c\_contact PRIMARY KEY, c\_id NUMBER(10) NOT NULL, CONSTRAINT fk\_c\_contact FOREIGN KEY(c\_id) REFERENCES customer(c\_id));
8. CREATE TABLE ticket(t\_id NUMBER(10) CONSTRAINT pk\_ticket PRIMARY KEY, amount NUMBER(2) NOT NULL, fare FLOAT NOT NULL, departure VARCHAR2(100) NOT NULL, destination VARCHAR2(100) NOT NULL, p\_date DATE NOT NULL, d\_date DATE NOT NULL, baggage NUMBER(2) NOT NULL, a\_id NUMBER(3) NOT NULL, c\_id NUMBER(10) NOT NULL, CONSTRAINT fk\_agent\_ticket FOREIGN KEY (a\_id) REFERENCES agent (a\_id),CONSTRAINT fk\_customer\_ticket FOREIGN KEY (c\_id) REFERENCES customer (c\_id));
9. CREATE TABLE payment(pay\_id NUMBER(10) CONSTRAINT pk\_payment PRIMARY KEY, t\_fare FLOAT NOT NULL, c\_id NUMBER(10) NOT NULL, t\_id NUMBER(10) NOT NULL, CONSTRAINT fk\_customer\_payment FOREIGN KEY (c\_id) REFERENCES customer (c\_id), CONSTRAINT fk\_ticket\_payment FOREIGN KEY (t\_id) REFERENCES ticket(t\_id));
10. CREATE TABLE plane(pl\_id NUMBER(3) CONSTRAINT pk\_plane PRIMARY KEY, t\_no NUMBER(3) NOT NULL, takeoff\_airport VARCHAR2(100) NOT NULL, takeoff\_time DATE NOT NULL, d\_airport VARCHAR2(100) NOT NULL, capacity NUMBER(3) NOT NULL);
11. CREATE TABLE passenger(p\_id NUMBER(10) CONSTRAINT pk\_passenger PRIMARY KEY, p\_name VARCHAR2(30) NOT NULL, p\_gender VARCHAR2(6) NOT NULL CONSTRAINT p\_gender CHECK (p\_gender IN ('MALE', 'FEMALE', 'OTHERS')), p\_age NUMBER(3) NOT NULL CONSTRAINT p\_age CHECK(p\_age>0), c\_id NUMBER(10) NOT NULL, pl\_id NUMBER(3) NOT NULL, cc\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_customer\_passenger FOREIGN KEY (c\_id) REFERENCES customer(c\_id), CONSTRAINT fk\_plane\_passenger FOREIGN KEY (pl\_id) REFERENCES plane (pl\_id), CONSTRAINT fk\_address\_passenger FOREIGN KEY (cc\_id) REFERENCES address(cc\_id));
12. CREATE TABLE passenger\_contact(p\_phone NUMBER(11) CONSTRAINT pk\_p\_contact PRIMARY KEY, p\_id NUMBER(10) NOT NULL, CONSTRAINT fk\_p\_contact FOREIGN KEY(p\_id) REFERENCES passenger(p\_id));
13. CREATE TABLE seat(s\_id NUMBER(3) CONSTRAINT pk\_seat PRIMARY KEY, s\_type VARCHAR2(20) NOT NULL, p\_id NUMBER(10) NOT NULL, pl\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_passenger\_seat FOREIGN KEY (p\_id) REFERENCES passenger(p\_id), CONSTRAINT fk\_plane\_seat FOREIGN KEY (pl\_id) REFERENCES plane (pl\_id));
14. CREATE TABLE route(r\_id NUMBER(3) CONSTRAINT pk\_route PRIMARY KEY, distance VARCHAR2(10) NOT NULL, pl\_id NUMBER(3) NOT NULL, CONSTRAINT fk\_plane\_route FOREIGN KEY (pl\_id) REFERENCES plane (pl\_id));

**Table Creation (Screenshot)**

1. **Manager**

A screenshot of a computer

Description automatically generated

1. **Manager Contact**

A screenshot of a computer

Description automatically generated

1. **Agent**

A screenshot of a computer

Description automatically generated

1. **Agent Contact**

A screenshot of a computer

Description automatically generated

1. **Address**

A screenshot of a computer

Description automatically generated

1. **Customer**

A screenshot of a computer

Description automatically generated

1. **Customer Contact**

A screenshot of a computer

Description automatically generated

1. **Ticket**

A screenshot of a computer

Description automatically generated

1. **Payment**

A screenshot of a computer

Description automatically generated

1. **Plane**

A screenshot of a computer

Description automatically generated

1. **Passenger**

A screenshot of a computer

Description automatically generated

1. **Passenger Contact**

A screenshot of a computer

Description automatically generated with medium confidence

1. **Seat**

A screenshot of a computer

Description automatically generated

1. **Route**

A screenshot of a computer

Description automatically generated

**Data Insertion**:

1. **Manager**
2. INSERT INTO manager(mgr\_id, mgr\_name, mgr\_sal) VALUES(100, 'Kawser', 99999);
3. INSERT INTO manager(mgr\_id, mgr\_name, mgr\_sal) VALUES(200, 'Irom', 89999);
4. INSERT INTO manager(mgr\_id, mgr\_name, mgr\_sal) VALUES(300, 'Rushee', 60000);
5. **Manager Contact**
6. INSERT INTO manager\_contact(mgr\_phone, mgr\_id) VALUES(01765238741, 100);
7. INSERT INTO manager\_contact(mgr\_phone, mgr\_id) VALUES(01768738741, 200);
8. INSERT INTO manager\_contact(mgr\_phone, mgr\_id) VALUES(01763438741, 300);
9. **Agent**
   1. INSERT INTO agent(a\_id, a\_name, a\_sal, mgr\_id) VALUES(459, 'Niloy', 30000, 100);
   2. INSERT INTO agent(a\_id, a\_name, a\_sal, mgr\_id) VALUES(588, 'Azminur', 45000, 300);
   3. INSERT INTO agent(a\_id, a\_name, a\_sal, mgr\_id) VALUES(625, 'Sakib', 40000, 200);
   4. INSERT INTO agent(a\_id, a\_name, a\_sal, mgr\_id) VALUES(615, 'Saikot', 35000, 200);
10. **Agent Contact**
11. INSERT INTO agent\_contact(a\_phone, a\_id) VALUES(01763438778, 459);
12. INSERT INTO agent\_contact(a\_phone, a\_id) VALUES(01763438711, 588);
13. INSERT INTO agent\_contact(a\_phone, a\_id) VALUES(01763438767, 625);
14. INSERT INTO agent\_contact(a\_phone, a\_id) VALUES(01763438744, 615);
15. **Address**
16. INSERT INTO address(cc\_id, city, country) VALUES(10, 'Dhaka', 'Bangladesh');
17. INSERT INTO address(cc\_id, city, country) VALUES(20, 'Chittagong', 'Bangladesh');
18. INSERT INTO address(cc\_id, city, country) VALUES(30, 'Rajshahi', 'Bangladesh');
19. INSERT INTO address(cc\_id, city, country) VALUES(40, 'Khulna', 'Bangladesh');
20. INSERT INTO address(cc\_id, city, country) VALUES(50, 'Barishal', 'Bangladesh');
21. INSERT INTO address(cc\_id, city, country) VALUES(60, 'Sylhet', 'Bangladesh');
22. INSERT INTO address(cc\_id, city, country) VALUES(70, 'Rangpur', 'Bangladesh');
23. INSERT INTO address(cc\_id, city, country) VALUES(80, 'Mymensingh', 'Bangladesh');
24. **Customer**
    * 1. INSERT INTO customer(c\_id, c\_name, c\_age, c\_gender, pay\_history, cc\_id) VALUES(22466251, 'Shaon', 22, 'MALE', 9817, 30);
      2. INSERT INTO customer(c\_id, c\_name, c\_age, c\_gender, pay\_history, cc\_id) VALUES(22465881, 'Rahman', 22, 'MALE', 6591, 40);
      3. INSERT INTO customer(c\_id, c\_name, c\_age, c\_gender, pay\_history, cc\_id) VALUES(22464591, 'Nafiur', 69, 'MALE', 23221, 80);
      4. INSERT INTO customer(c\_id, c\_name, c\_age, c\_gender, pay\_history, cc\_id) VALUES(22464592, 'Kundu', 15, 'MALE', 21231, 30);
25. **Customer Contact**
26. INSERT INTO customer\_contact(c\_phone, c\_id) VALUES(01954879621, 22465881);
27. INSERT INTO customer\_contact(c\_phone, c\_id) VALUES(01935489647, 22466251);
28. INSERT INTO customer\_contact(c\_phone, c\_id) VALUES(01821365482, 22464591);
29. INSERT INTO customer\_contact(c\_phone, c\_id) VALUES(01132654791, 22464592);
30. **Ticket**
31. INSERT INTO ticket(t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id, c\_id) VALUES(45698745, 1, 4699, 'DHK', 'JES', TO\_DATE('08-05-2023', 'DD-MM-YYYY'), TO\_DATE('15-05-2023', 'DD-MM-YYYY'), 20, 459, 22466251);
32. INSERT INTO ticket(t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id, c\_id) VALUES(45698565, 2, 9398, 'DHK', 'JES', TO\_DATE('09-05-2023', 'DD-MM-YYYY'), TO\_DATE('15-05-2023', 'DD-MM-YYYY'), 30, 588, 22465881);
33. INSERT INTO ticket(t\_id, amount, fare, departure, destination, p\_date, d\_date, baggage, a\_id, c\_id) VALUES(45698234, 2, 9398, 'DHK', 'JES', TO\_DATE('11-05-2023', 'DD-MM-YYYY'), TO\_DATE('15-05-2023', 'DD-MM-YYYY'), 50, 615, 22464591);
34. **Payment**
35. INSERT INTO payment(pay\_id, t\_fare, c\_id, t\_id) VALUES(1578811645, 4699, 22466251, 45698745);
36. INSERT INTO payment(pay\_id, t\_fare, c\_id, t\_id) VALUES(1578845682, 9398, 22465881, 45698565);
37. INSERT INTO payment(pay\_id, t\_fare, c\_id, t\_id) VALUES(1574568214, 9398, 22464591, 45698234);
38. **Plane**
39. INSERT INTO plane(pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity) VALUES(777, 13, 'DHK', TO\_DATE('15-05-2023 09:30:00', 'DD-MM-YYYY HH24:MI:SS'), 'JES', 150);
40. INSERT INTO plane(pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity) VALUES(787, 10, 'DHK', TO\_DATE('17-05-2023 12:30:00', 'DD-MM-YYYY HH24:MI:SS'), 'CHI', 150);
41. INSERT INTO plane(pl\_id, t\_no, takeoff\_airport, takeoff\_time, d\_airport, capacity) VALUES(666, 02, 'DHK', TO\_DATE('19-05-2023 16:00:00', 'DD-MM-YYYY HH24:MI:SS'), 'RAJ', 150);
42. **Passenger**
43. INSERT INTO passenger(p\_id, p\_name, p\_gender, p\_age, c\_id, pl\_id, cc\_id) VALUES(22466251, 'Shaon', 'MALE', 22, 22466251, 777, 30);
44. INSERT INTO passenger(p\_id, p\_name, p\_gender, p\_age, c\_id, pl\_id, cc\_id) VALUES(22465881, 'Rahman', 'MALE', 22, 22465881, 777, 40);
45. INSERT INTO passenger(p\_id, p\_name, p\_gender, p\_age, c\_id, pl\_id, cc\_id) VALUES(22464591, 'Nafiur', 'MALE', 69, 22464591, 777, 80);
46. INSERT INTO passenger(p\_id, p\_name, p\_gender, p\_age, c\_id, pl\_id, cc\_id) VALUES(16989898, 'Neela', 'FEMALE', 18, 22464591, 777, 50);
47. INSERT INTO passenger(p\_id, p\_name, p\_gender, p\_age, c\_id, pl\_id, cc\_id) VALUES(51897848, 'Saikot', 'MALE', 25, 22465881, 777, 70);
48. **Passenger Contact**
49. INSERT INTO passenger\_contact(p\_phone, p\_id) VALUES(01935489647, 22466251);
50. INSERT INTO passenger\_contact(p\_phone, p\_id) VALUES(01954879621, 22465881);
51. INSERT INTO passenger\_contact(p\_phone, p\_id) VALUES(01821365482, 22464591);
52. INSERT INTO passenger\_contact(p\_phone, p\_id) VALUES(01334567890, 16989898);
53. INSERT INTO passenger\_contact(p\_phone, p\_id) VALUES(01434567890, 51897848);
54. **Seat**
55. INSERT INTO seat(s\_id, s\_type, p\_id, pl\_id) VALUES(101, 'Standard', 22466251, 777);
56. INSERT INTO seat(s\_id, s\_type, p\_id, pl\_id) VALUES(102, 'Business', 22465881, 777);
57. INSERT INTO seat(s\_id, s\_type, p\_id, pl\_id) VALUES(503, 'Economy', 22464591, 777);
58. INSERT INTO seat(s\_id, s\_type, p\_id, pl\_id) VALUES(504, 'Economy', 16989898, 777);
59. INSERT INTO seat(s\_id, s\_type, p\_id, pl\_id) VALUES(103, 'Business', 51897848, 777);
60. **Route**
61. INSERT INTO route(r\_id, distance, pl\_id) VALUES(100, '144 KM', 777);
62. INSERT INTO route(r\_id, distance, pl\_id) VALUES(200, '227 KM', 787);
63. INSERT INTO route(r\_id, distance, pl\_id) VALUES(300, '192 KM', 666);

**Joining**

1. **Equijoin**

**Question:** "Find customers who have passengers with the same gender as themselves."

**Query:**

SELECT c.c\_name, c.c\_gender, p.p\_name, p.p\_gender FROM customer c, passenger p WHERE c.c\_id = p.c\_id AND c.c\_gender = p.p\_gender;

**Screenshot:**

A screenshot of a computer

Description automatically generated

1. **Outer Join**

**Left:**

**Question:** List all customers and their corresponding tickets, including those customers who do not have any tickets.

**Query:**

SELECT c.c\_name, t.t\_id FROM customer c, ticket t WHERE c.c\_id = t.c\_id (+);

**Screenshot:**

**A screenshot of a computer

Description automatically generated**

**Right:**

**Question:** List all tickets and their corresponding customers, including those tickets that are not associated with any customer.

**Query:**

SELECT c.c\_name, t.t\_id FROM ticket t, customer c WHERE c.c\_id (+) = t.c\_id;

**Screenshot:**

A screenshot of a computer

Description automatically generated

1. **Self-Join**

**Question:** Retrieve the agent names and their corresponding manager names from the agent table where the agent is managed by another agent.

**Query:**

SELECT a1.a\_name AS "Agent Name", a2.a\_name AS "Manager Name"

FROM agent a1, agent a2

WHERE a1.mgr\_id = a2.a\_id;

**Screenshot:**

**A screenshot of a computer

Description automatically generated**

**Subquery**

1. **Question:** "Retrieve the agent name & salary who has the highest salary among all agents."

**Query:** SELECT a\_name AS "AGENT NAME", a\_sal AS "SALARY" FROM agent WHERE a\_sal = (SELECT MAX(a\_sal) FROM agent);

**Screenshot:**

**A screenshot of a computer

Description automatically generated**

1. **Question:** "Find the customer’s name who have purchased tickets with a fare greater than the average fare across all tickets."

**Query:** SELECT c\_name AS "CUSTOMER NAME" FROM customer WHERE c\_id IN (SELECT c\_id FROM ticket WHERE fare > (SELECT AVG(fare) FROM ticket));

**Screenshot:**

**A screenshot of a computer

Description automatically generated**

1. **Question:** Find the names of customers whose names end with the letter 'n' and who have booked tickets with an amount greater than or equal to the average amount of tickets booked by agents whose names start with 'A'.

**Query:**

SELECT c.c\_name AS "CUSTOMER NAME"

FROM customer c

WHERE c.c\_name LIKE '%n' AND c.c\_id IN (

SELECT t.c\_id

FROM ticket t

WHERE t.amount >= (

SELECT AVG(amount)

FROM ticket

WHERE a\_id IN (

SELECT a\_id

FROM agent

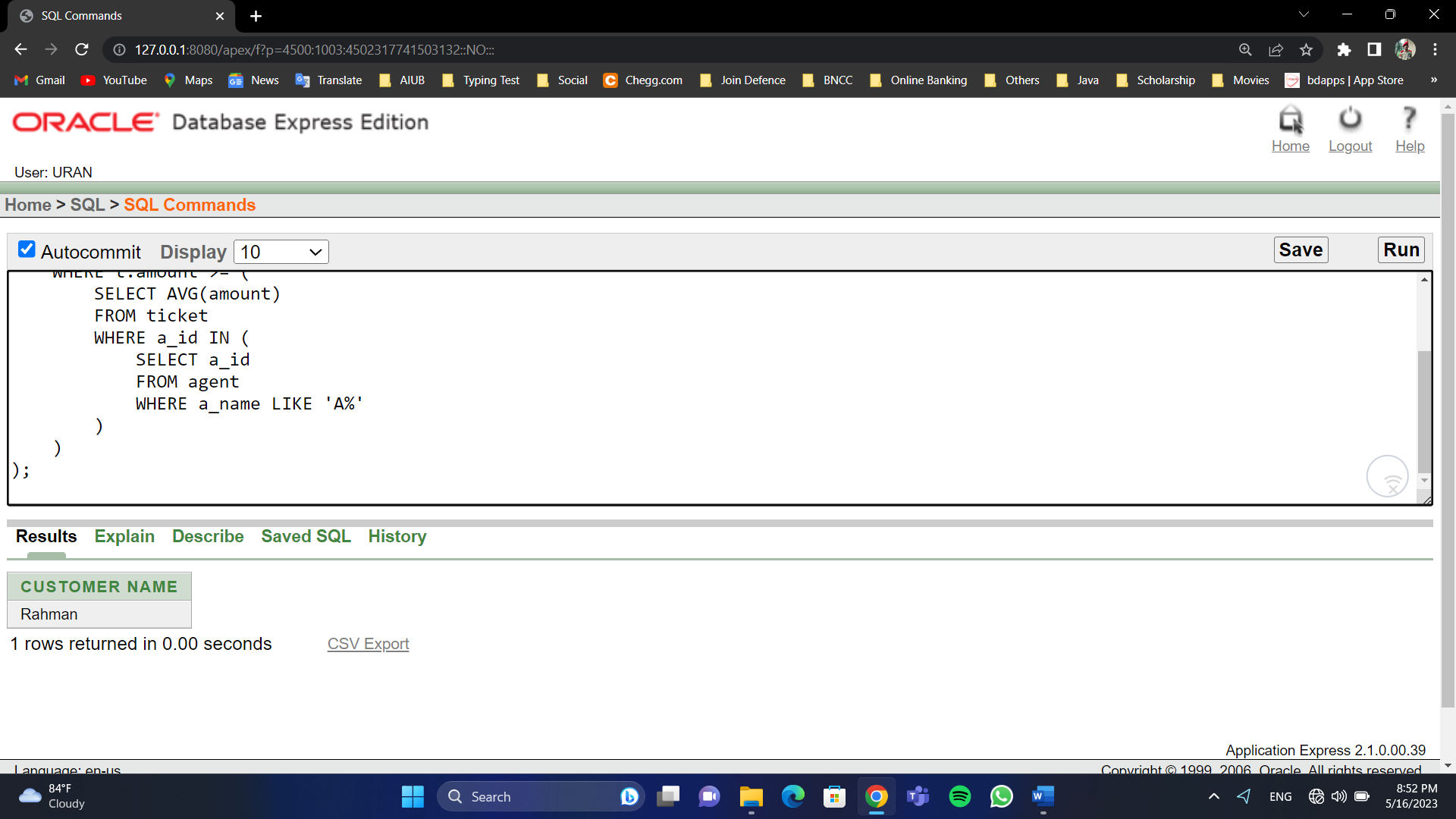
WHERE a\_name LIKE 'A%'

)

)

);

**Screenshot:**

****

**View**

1. **Complex View**

**Question:** Create a complex view that combines information from multiple tables to provide a comprehensive overview of customers and their corresponding ticket details.

**Query:**

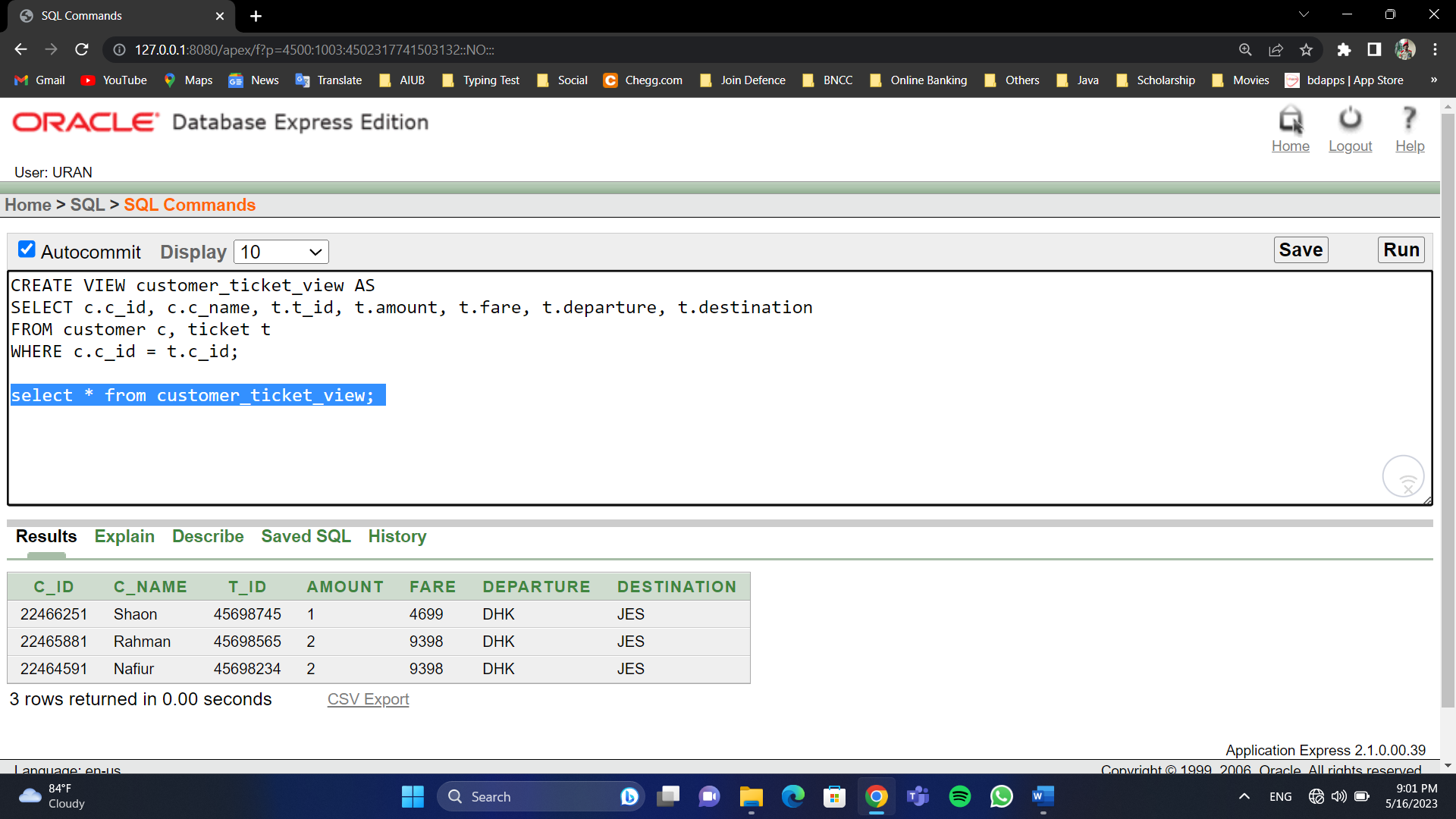
CREATE VIEW customer\_ticket\_view AS

SELECT c.c\_id, c.c\_name, t.t\_id, t.amount, t.fare, t.departure, t.destination

FROM customer c, ticket t

WHERE c.c\_id = t.c\_id;

**Screenshot:**

****

1. **Simple View**

**Question:** Create a simple view that displays the names and salaries of managers.

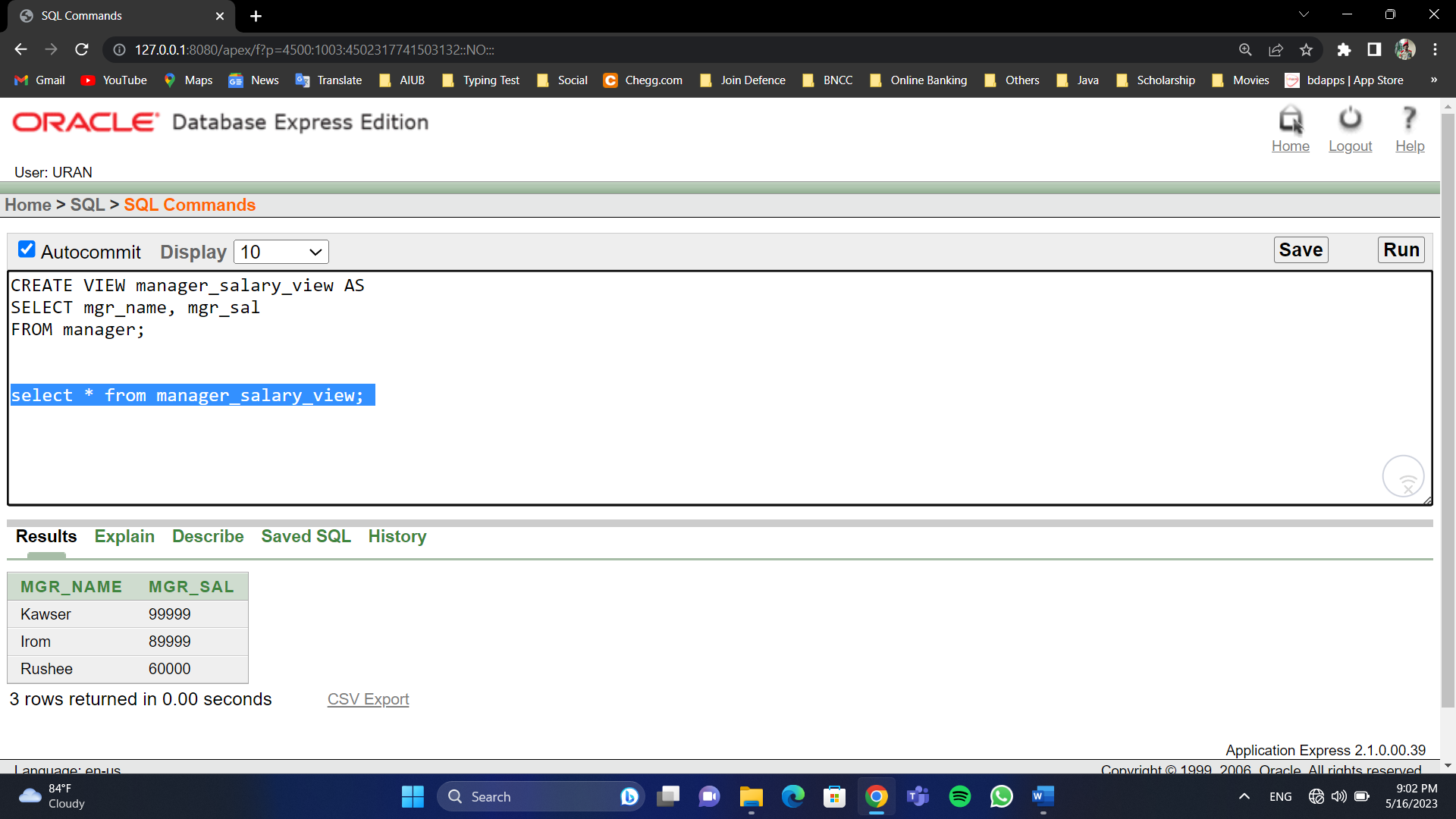
**Query:**

CREATE VIEW manager\_salary\_view AS

SELECT mgr\_name, mgr\_sal

FROM manager;

**Screenshot:**

****

**Add Constraint**

**Question:** Add constraint in customer table payment history and it must be positive or 0 and not null.

**Query:**

ALTER TABLE customer

ADD CONSTRAINT check\_pay\_history\_positive

CHECK (pay\_history >= 0 AND pay\_history IS NOT NULL);

**Screenshot:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**