

Final Project (10%)

LEARNING OUTCOMES

Upon successful completion of this project, you will have demonstrated the ability to:

1. Convert tables into **Third Normal Form (3NF)** and infer the associated **Entity Relationship Diagram (ERD)** using crow's foot notation.
 2. Implement the **relational database model** by creating tables and constraints on attributes.
 3. Enter data into the database.
 4. Query the database to retrieve information effectively.
-

GROUP WORK & SUBMISSION

- This project must be completed in groups of **2–3 students**.
- Each group must submit **one** final project on Blackboard. **Only one team member should submit the files.**
- **Late submissions** will be penalized by **10% per day**.

Submission Requirements

Submit the following two files:

1. **SQL file:** Project2_GroupX.SQL
2. **Project document:** Project2_GroupX.docx

Group Work Declaration

Add the following declaration at the top of your document:

We, [list all group members' names], declare that the attached assignment is our own work in accordance with Seneca's Academic Policy. No part of this assignment has been copied manually or electronically from any other source (including websites) or distributed to other students.

Member Contributions

Each member must specify their contribution to the project:

Name Task(s) Completed

- 1.
- 2.
- 3.

PROJECT OVERVIEW

The **CheapFare airline reservation system** is a flight booking company handling domestic and international tickets. The company is transitioning to a relational database for better efficiency.

Below is the key information gathered from the airline reservation manager:

1. Each **booking** can be **one-way** or **round-trip** with multiple flights.
2. A **flight** is identified by a flight_num (e.g., AF393), which determines the airline the flight belongs to (e.g. AF). Some flights are operated by a **partner airline** (operatedBy is an airline_code).
3. Each flight is associated with a **specific aircraft** (aircraft_code, description).
4. A **flight number** always departs from the same **departure airport** to the same **arrival airport**, though times and dates vary.
5. A **booking** may generate multiple **e-tickets**, one for each passenger. For example, booking_id 56753365 for a mom and two kids generates 3 different etickets.
6. Each **e-ticket** (eticket_num) belongs to a **traveler** and records meal preferences (optional), ticket price, and tax fees.
7. A **traveler** is identified by a traveler_id, with first_name, last_name, and gender.

BookingRecord				
BookingID	56753365	56753365	56753936	56753936
BookedOn	28-Jan-19	28-Jan-19	25-Jan-19	25-Jan-19
Flight_num	AF393	AF5106	AF393	AF386
Airline_code	AF	AF	AF	AF
Airline_Name	Air France	Air France	Air France	Air France
Aircraft_code	772	332	772	772
Aircraft_desc	BOEING 777 285-305 STD	AIRBUS INDUSTRIE JET 200-345 STD	BOEING 777 285-305 STD	BOEING 777 285-305 STD SEATS
operatedBy	AF	MEA	AF	AF
dep_date	22-Jun-19	23-Jun-19	22-Jun-19	17-Jul-19
dep_time	9:20 PM	1:40 PM	9:20 PM	5:00 PM
dep_airport_cod	YYZ	CDG	YYZ	CDG
dep_airport_name	Toronto Pearson Intl,	Charles de Gaul	Toronto Pearson Intl,	Charles de Gaul
dep_city	Toronto	Paris	Toronto	Paris
arr_date	23-Jun-19	23-Jun-19	23-Jun-19	17-Jul-19
arr_time	10:50 AM	6:55 PM	10:50 AM	7:20 PM
arr_airport_cod	CDG	BEY	CDG	YYZ
arr_airport_name	Charles de Gaul		Charles de Gaul	Toronto Pearson Intl, Ontario
arr_city	Paris	Beirut	Paris	Toronto

TICKET				
eticket_num	573480996631	573480996619	573480996620	573480996621
BookingID	56753936	56753365	56753365	56753365
traveler_id	111	112	113	114
traveler_Fname	Andrew	Mariam	Yasmine	Hasan
traveler_Lname	Smith	Daoud	Ch	Ch
Gender	Male	Female	Female	Male
Meal_preference	Vegetarian	Halal	Vegetarian	Halal
ticket_price	1200	1353	1142	1142
taxes_fees	182	182	180	180

PART I: DATABASE NORMALIZATION & ERD (50%)

1. Normalize all given tables to **3NF**, showing all steps (1NF → 2NF → 3NF) using **arrow notation**.
2. Do NOT include derived attributes. Consider the composite key (bookingID, flight_num) as PK in the bookingRecord table.
3. Combine the tables obtained at the end of the normalization process into one logical design.
4. Create a **logical database design** based on normalization results.
5. Draw an **ERD** using **draw.io** or **Microsoft Visio**, ensuring:
 - No **many-to-many relationships**
 - No **multi-valued attributes**
 - Clear **Primary Keys (PK)** and **Foreign Keys (FK)**
 - Strong and weak relationships labeled correctly
6. Export the ERD as **.jpg** and insert it into your project document.

Evaluation Criteria:

Task	Marks
Normalization process (1NF to 3NF with proper notation)	20%
Logical database design (consistent and correct)	15%
ERD with correct PK, FK, and relationships	15%

PART II: DATABASE IMPLEMENTATION (20%)

Write **SQL scripts** to:

1. **Create tables** with appropriate **PK, FK, and constraints**.
2. Provide a **screenshot** of the object explorer displaying tables, columns, and constraints.
3. Insert sample **data** into tables using INSERT statements.

Evaluation Criteria:

Task	Marks
Correct table creation with constraints	10%
Screenshot of database structure	5%
Accurate data insertion statements	5%

PART III: DATA MANIPULATION (30%)

For the following queries, include: (a) The **SQL command** (in the SQL file and the document)

(b) The **output in text format**

(c) The **number of rows affected**

1. **Retrieve airline, aircraft, and airport details** for flight number AF393:
 - Display: flight_num, airline_code, airline_name, aircraft_code, aircraft_description, departure_airport_code, departure_airport_name, arrival_airport_code, arrival_airport_name.
2. **Retrieve flight details for bookingID 56753365:**
 - Display: booking_id, booking_date, flight_num, departure_airport_code, departure_date_time, arrival_airport_code, arrival_date_time.
 - Order by: departure_date_time.
3. **Retrieve all passengers for bookingID 56753365:**

- Display: bookingID, eticket_num, traveler_first_name, traveler_last_name.
- 4. **Calculate total fees for bookingID 56753365:**
 - Sum of **ticket price** + **tax fees** for all e-tickets.
 - Use SUM() function and WHERE clause.
- 5. **Create a view Book_ticket:**
 - Show list of **all bookings** and number of tickets per booking.

Evaluation Criteria:

Task	Marks
Query correctness and efficiency	20%
Correct use of aggregate functions, joins, and conditions	5%
Creating and using VIEW properly	5%

FINAL REMARKS

- **Test all SQL queries before submission.**
- Ensure **proper formatting** in your document.
- Clearly **label screenshots and code sections.**
- Use **comments in SQL scripts** to explain logic.

Good luck with your project!