

AIRPORT MANAGEMENT SYSTEM

Wing Lee

Student ID: 920558688

GitHub: ayanamesu

Checkpoint #	Date Submitted
Checkpoint 1	02/20/2024
Checkpoint 2	03/05/2024
Checkpoint 3	04/01/2024

San Francisco State University

CSC 675-775-database-systems

Table of Contents

Project Description.....	2
Functional Database Requirements.....	3
Non-Functional Database Requirements.....	7
Entity Relationship Diagram (ERD).....	9
Entity Description.....	10
Entity Establishment Relationship Diagram (EER).....	13
Constraints Descriptions.....	14

Project Description

The Airport Management Database System mission is to make the process of managing flight scheduling, baggage handling, security, staff, Location and ticket handling to make this process more efficient and practical. One major task in the Airport management system is managing staff, airline and where belongings and tickets go to which terminal and which gate. An online database for Airport Management systems would help organize things where they should be and this would provide a more efficient way of functioning since we need this important data to track users and flights and terminals before planes depart. This would also allow airport management to keep track of staff and basic functions of the airport and provide necessary resources if needed. This database would require many staff (employees) because in a single airport there are multiple airlines and different companies operating so that we would require many staff roles such as engineers, security, managers for airlines, stores and restaurants. Keeping track and dividing these roles into specific categories would make this process more efficient and more organizable. Each employee would need to display their work status under each specific category because they could either work under an airline or just general airport staff. Passengers would need a unique identifier in this case would be the ticket since each passenger can only possess one ticket. Similar industry practices would be ADB safegate. They take note of flight , flight path and managing flight updates, they would benefit because this includes flight and specifying specific airlines with each passenger connected to it. It helps build a bridge connecting the airport and the flight so the process would be faster.

Another industry practice is Indra. This could mainly benefit from this because they keep track of billing and flights and real time operation management with this database. They could benefit from ticket to store billing to flights which they already somewhat have implemented.

Functional Database Requirements

1. General User

1.1 A user shall be able to create only one account by providing email and password.

2. Account

2.1 An account shall have one user

2.2 An account shall be a passenger or staff (employee)

2.3 An account shall only have one email

2.4 An account shall have payment information.

3. Passenger

3.1. A passenger shall create only one account

3.2. A passenger shall have multiple tickets for different flights.

3.3. A passenger shall have one or many luggage.

3.4. A passenger shall only board in one flight

4. Ticket

8.1. A ticket shall be associated with one passenger

8.2. A ticket shall have one flight.

5. Terminal

2.1. A terminal shall have many Airlines

2.2. A terminal shall have multiple gates.

2.3. A terminal shall have many Staffs

2.4. A terminal shall have multiple managers

2.5. A terminal shall have one or many Baggage Handlers, Airline Staff, Security, Store Staff and Engineers

2.6. A terminal shall have zero or many Restaurants.

2.7. A terminal shall have many Lounges.

2.8. A terminal shall have many Stores

6. Gate

3.2. A gate can have multiple flights.

3.3. Each gate shall be located in only one terminal.

7. Flight (Plane)

7.1. A flight shall have one gate.

7.2. A flight shall be operated by one airline.

7.3. A flight shall have zero or many tickets booked.

7.4. A flight shall have multiple Engineers

7.5. A flight shall have multiple passengers

8. Airline

4.1. An airline can have multiple flights.

4.2. An airline shall have zero or many Staff

4.3. An airline shall have only one Lounge.

4.4. An airline shall have only one terminal.

9. Staff (Employee)

11.1. A Staff shall consist of Airline, Manager ,Security, Engineer and Baggage Handler ,Restaurant and Stores.

11.2. A Staff Shall have only one account.

11.3. A Staff shall work for at least 1 Airline

11.4. A Staff shall have a name.

11.5. A Staff shall work in zero or many terminal

11.6. A Staff shall work in one Restaurant.

10. Security

12.1. A Security shall be assigned to one or many terminals

12.2. A Security is a Staff.

12.3. A Security shall have a name.

11. Engineer

11.1. Engineers shall maintain multiple flights

11.2 An Engineer is a Staff

11.3. An Engineer shall have a name.

12. Manager

12.1. A Manager shall oversee multiple terminals

12.2. A manager shall consists of restaurant manager, Store manager and Airline manager

12.3. A Manager is a Staff

12.4. A Manager shall work in at most one store

12.5. A Manager shall have a name.

13. Baggage Handler

13.1. A baggage handler shall be in multiple terminals

13.2. A Baggage Handler is a Staff.

13.3. A Baggage Handler shall have a name

14. Restaurant

14.1. A Restaurant shall only have one Manager

14.2. A Restaurant shall have multiple Staff.

14.3. A Restaurant shall be in one or many terminals.

14.4. A Restaurant shall have at most one Lounge

15. Luggage

15.1. A Luggage shall be assigned to one passenger

16. Lounge

16.1. A Lounge shall be in only one terminals

16.2. A Lounge shall have at least Airline

16.3. A Lounge shall have many Restaurants.

17. Stores

17.1. A store shall have one or many Staffs.

17.2. Store shall have one terminal.

17.3. A Store has one manager.

17.4. A Store shall receive zero or many passengers

Non-Functional Database Requirements

1. Performance

1.1. modifying data would be fast

1.2. The database shall update with the user within an efficient time

2. Storage

2.1. Database tables will be assigned 10 mb of memory

3. Security

3.1. Passwords shall be encrypted in the database.

4. Compatibility

The database system shall be supported by MySQL

The database system shall be compatible with browsers

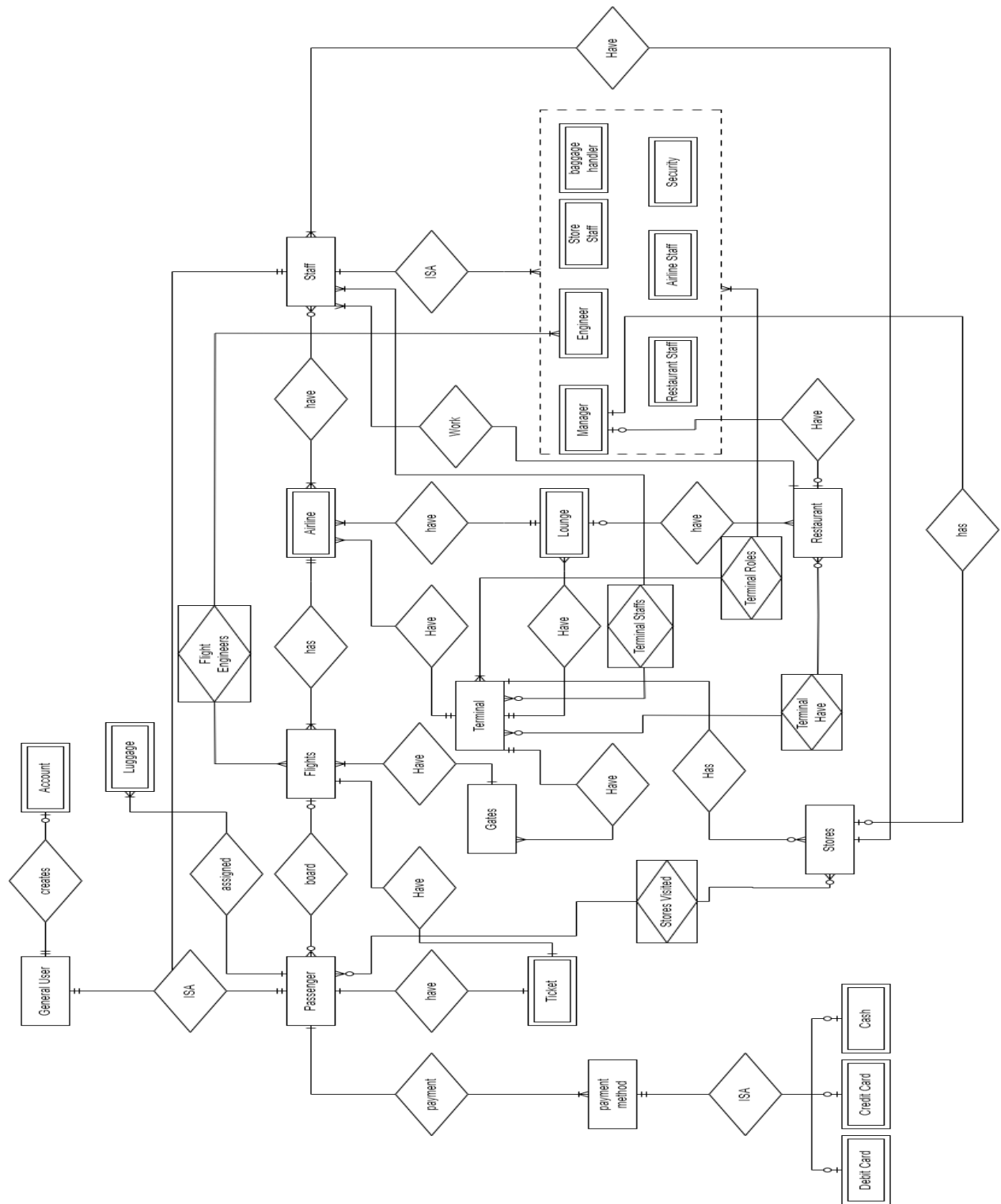
5. Scalability

5.1. This database system will support many users

6. Data backup/ media storage

6.1. The database's information shall be backed up and saved every day at 11:59 pm.

Entity Relationship Diagram (ERD)



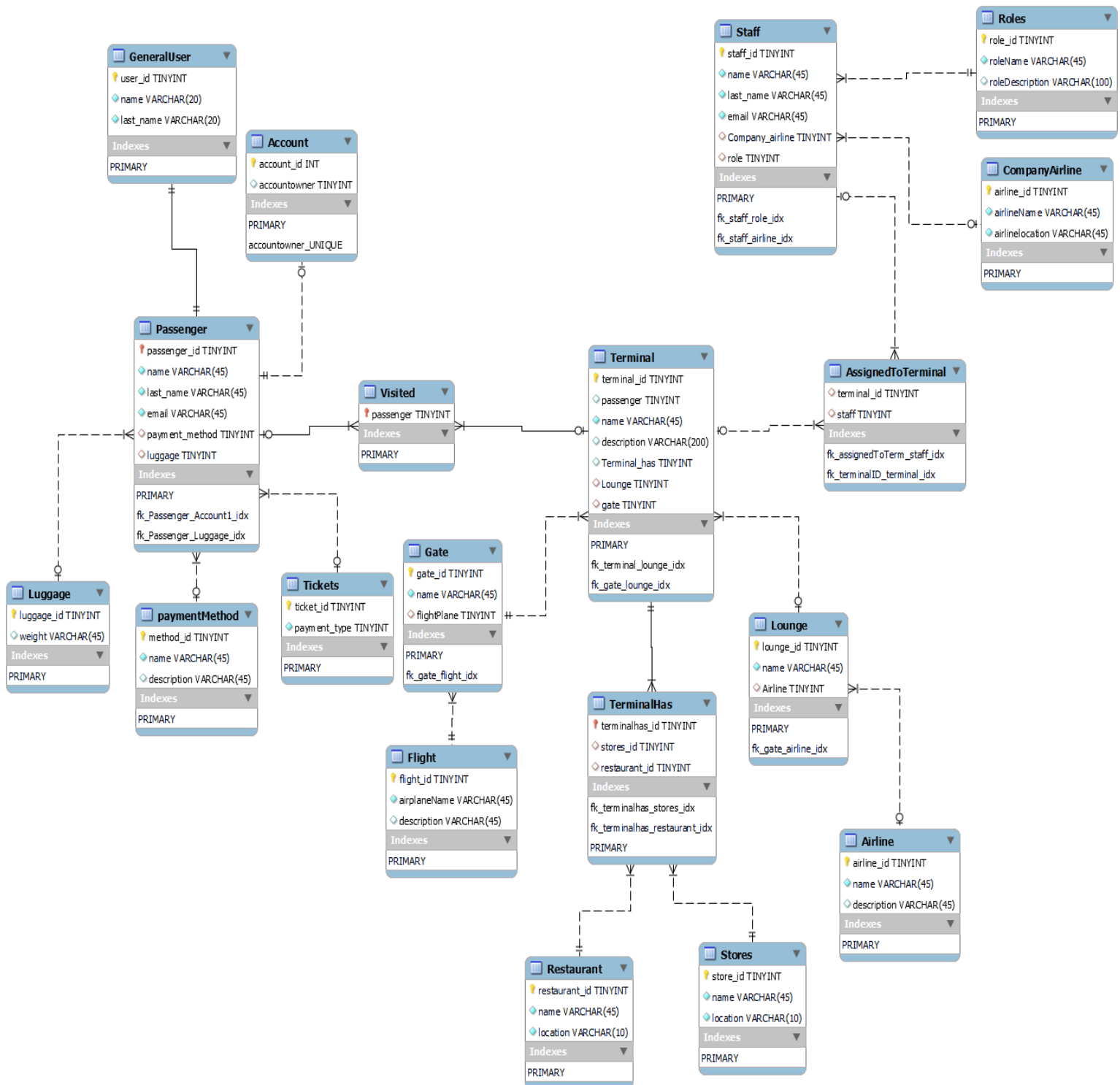
Entity Description

1. General User (Strong)
 - a. user_id: (key, numeric)
 - b. name: (composite, alphanumeric)
 - c. Last_name: (multivalue, alphanumeric)
2. Account (Weak)
 - a. Account id: (key, strong, numeric)
 - b. Email: (key, alphanumeric)
 - c. Created: (composite, multivalue, date)
3. Passenger (Strong)
 - a. Passenger id: (key, strong, numeric)
 - b. Fullname: (composite, alphanumeric)
 - c. Email: (key, alphanumeric)
4. Ticket (Weak)
 - a. Ticket_id: (key, numeric)
 - b. Payment_type: (weak, key, numeric)
 - c. passenger_id: (weak, key, numeric)
5. Terminal (Strong)
 - a. Terminal_id: (key, numeric)
 - b. Num_stores: (numeric)
 - c. Gate_number: (numeric)
 - d. Staff_id: (numeric)
6. Gate (Strong)
 - a. Gate_id: (key, numeric)
 - b. Airline Assigned: (alphanumeric)
 - c. Terminal_id: (numeric)
7. Flight (Strong)
 - a. Flight_id: (key, Strong, numeric)
 - b. Tickets_Booked: (multivalue, numeric)
 - c. Passengers: (alphanumeric)
 - d. Engineers: (alphanumeric)
 - e. Airline_id: (numeric)
 - f. Gate_number: (numeric)

8. Airline (Weak)
 - a. Airline_id: (key, alphanumeric)
 - b. Airline_staff: (multivalued, alphanumeric)
 - c. Airline_flights: (multivalued, alphanumeric)
 - d. Airline_Lounge: (multivalued, numeric)
 - e. Airline_name: (composite, alphanumeric)
 - f. Airline_company: (composite, alphanumeric)
9. Staff (Strong)
 - a. Staff_id: (key, alphanumeric)
 - b. Staff_roles: (multivalued, alphanumeric)
 - c. Staff_schedule: (composite, alphanumeric)
 - d. Company_airline: (composite, alphanumeric)
10. Security (Weak)
 - a. Security_id: (key, alphanumeric)
 - b. Fullname: (composite, alphanumeric)
 - c. Security_task: (alphanumeric)
11. Engineers (Weak)
 - a. Engineer_id: (key, alphanumeric)
 - b. Full name: (composite, alphanumeric)
 - c. Email: (composite, alphanumeric)
12. Manager(weak)
 - a. Manager_id: (key, alphanumeric)
 - b. Manager_role: (multivalued, alphanumeric)
 - c. Manager_department: (alphanumeric)
13. Baggage Handler (weak)
 - a. Baggagehandler_id: (key, alphanumeric)
 - b. Handler_name: (alphanumeric)
 - c. Airline_company: (alphanumeric)
14. Restaurant (Strong)
 - a. Restaurant_id: (key, alphanumeric)
 - b. Restaurant_name: (composite, alphanumeric)
 - c. Restaurant_location: (alphanumeric)

15. Luggage (weak)
 - a. Luggage_id: (key, alphanumeric)
 - b. Luggage_weight: (numeric)
 - c. Luggage_status: (alphanumeric)
16. Lounge (Weak)
 - a. Lounge_id: (key alphanumeric)
 - b. Lounge_company: (composite, alphanumeric)
 - c. Lounge_name: (alphanumeric)
17. Stores (Strong)
 - a. Store_id: (key, numeric)
 - b. Store_name: (alphanumeric)
 - c. Store_location: (composite, alphanumeric)
18. Payment method (Strong)
 - a. Method_id: (key, numeric)
 - b. Method_name: (composite, alphanumeric).
 - c. Description: (composite, alphanumeric)

Entity Establishment Relationship Diagram (EER)



Constraints Descriptions

Table	FK	ON DELETE	ON UPDATE	Comment
Passenger	user_id	CASCADE	CASCADE	If a general user were to delete their account, their user_id would go too.
Passenger	Payment Method	SET NULL	CASCADE	If payment method were to be deleted, then the user would have to pick another one. It'll be set to null in the meantime
Account	Accountowner	CASCADE	CASCADE	If Passengers were to delete their account. Then their account would be deleted as well.
Passenger	Tickets	SET NULL	CASCADE	If Passenger were to delete their ticket it would be SET NULL first until the user inserts a ticket in if not then it would be null
Passenger	Luggage	SET NULL	CASCADE	If Passenger were to delete this it would not affect the passenger it just means they bring no luggage unless added back in.
Visited	passenger	SET NULL	CASCADE	If a terminal were to lose a passenger, there would be the option to have another passenger
Visited	Passenger	CASCADE	CASCADE	If a general user were to delete their account, their user_id would go too.
Terminal	Lounge	NO ACTION	CASCADE	If the lounge were to be deleted, the terminal will keep its old lounge location. Must be changed manually.

Terminal	Gate	NO ACTION	CASCADE	If Gate were to be deleted must be changed manually
Gate	Flight plane	SET NULL	CASCADE	If flight were to be deleted, then flights will be set to null and would wait for another flight
TerminalHas	terminal has	CASCADE	CASCADE	If the terminal were to be deleted then all of the things the terminal has within would also be deleted (ex. Stores, restaurants).
TerminalHas	Store	SET NULL	CASCADE	If a store were to be deleted, then it would be SET NULL until a new store were to be inserted
TerminalHas	Restaurant	SET NULL	CASCADE	If a Restaurant were to be deleted, then it would be SET NULL until a new Restaurant were to be inserted
Staff	role	SET NULL	CASCADE	If a role were to be deleted, then staff will have a null role so it'll need to be updated.
Staff	Company Airline	SET NULL	CASCADE	If a Company were to be deleted, then staff will have a null company so it'll need to be updated.
AssignedTo Terminal	terminal	SET NULL	CASCADE	If a Terminal were to be deleted, then the staff will have null and will have to be assigned to another Terminal.

AssignedTo Terminal	Staff	SET NULL	CASCADE	If a staff were to be deleted, then they would need to be replaced with another staff to work on the terminal.
Lounge	Airline	SET NULL	CASCADE	If the airline were to be deleted then it would be SET NULL waiting for another airline to be inserted or take its place.