****

**Airport System**

**Database**

**10204282**

**A/618/7400**

**Section (2)**

**Submitted by**

Hamza AL-Risheq

**Submitted to**

Dr.Rami Ibrahim

**summer 2022 – 2023**

Table of Contents

[Part 1: Database Requirements: 3](#_Toc142243493)

[1.1 Scenario 3](#_Toc142243494)

[1.2 Data Requirements 4](#_Toc142243495)

[1.3 User and System Requirements 4](#_Toc142243496)

[Part 2: Database Design: 6](#_Toc142243497)

[2.1 Conceptual Design: 6](#_Toc142243498)

[2.1.1 Definition of conceptual design 6](#_Toc142243499)

[2.1.2 The Goal of conceptual design 6](#_Toc142243500)

[2.1.3 Main components of conceptual design 6](#_Toc142243501)

[2.1.4 the conceptual design of my system 6](#_Toc142243502)

[2.2 Schema after Mapping 8](#_Toc142243503)

[2.3 Normalization 9](#_Toc142243504)

[2.3.1: 1st NF 9](#_Toc142243505)

[2.3.2: 2nd NF 10](#_Toc142243506)

[2.3.3: 3rd NF 12](#_Toc142243507)

[2.4 Logical Design 14](#_Toc142243508)

[2.4.1 Definition of logical design 14](#_Toc142243509)

[2.4.2 The goal of logical design 14](#_Toc142243510)

[2.4.3 logical design components 15](#_Toc142243511)

[2.4.4 Logical design figure 16](#_Toc142243512)

[2.5 physical design 17](#_Toc142243513)

[2.5.1 Definition of physical design 17](#_Toc142243514)

[2.5.2 the goal of physical design 17](#_Toc142243515)

[2.5.3 physical design components 17](#_Toc142243516)

[2.5.4 Physcial desgin figure 18](#_Toc142243517)

[2.6 Effectiveness of the Design 19](#_Toc142243518)

[References: 21](#_Toc142243519)

# Part 1: Database Requirements:

## 1.1 Scenario

Amman Civil Airport was unable to handle the increasing demand for airport space, therefore Queen Alia International Airport (QAIA) was constructed in 1983. At the time, despite ongoing construction and improvement, passenger traffic was growing at a rate that was higher than the global average, recording annual growth rates of 25–30% and putting significant strain on airport infrastructure. Over 2.3 million people traveled through transit, arrived, and left in 1981.

A new international airport with the capacity to meet demand In the near future Is being built by the Jordanian Ministry of Transportation. The estimated final cost of QAIA’s construction was JOD 84 million Facilities for passengers were built to accommodate 3.5 million passengers annually.

Since then, QAIA has developed as the kingdom’s main international gateway and a Middle Eastern layover location for foreign planes. By 2012, QAIA was providing service to 40 international airlines and an average of more than 6 million passengers.

The airport is divided into departments such as the department of labor and wages, flight safety and quality department, department of information system support, passenger transportation service, fuel and lubricants services, department of accounting and reporting. Each department has a department ID, department name, one manager, many employees, and number of employees. Each employee is registered in one department. Each employee has a name which consists of their first name and last name, department ID, employee email, job title and employee phone number. And they can have multiple phone numbers and one email.

In the airport there are many airline companies, which are hosting many flights. Each airline has a unique ID, name, country, email, and phone number. Each airline has one email and phone number. Each flight has a unique ID, departure date, departure time, arrival time, arrival date, arrival airport, and arrival country. The one who is responsible for registering the flights is one of the employees.

Flights can park in a specific aircraft stand, each one has a unique number, status, and size.

The flights are assigned to a specific plane gate from the airport so the passengers can go through it from the airport to the plane. Each gate has a unique ID, and status.

Each flight can have many passengers which each passenger has a unique ID, passenger name which consists of first name and last name, birth date, phone number, passport ID, nationality, and email. Each passenger can have multiple nationalities, and phone numbers.

Each passenger can reserve many flights and for each passenger there are a specific number of bags, and for each passenger has a seat number.

## 1.2 Data Requirements

* The Airport is divided into many departments. Department has a department ID, department name, one manager, many employees, and number of employees.
* Each employee is registered with one department. So, each employee has a name which consists of their first name and last name, department ID, employee email, job title and employee phone number. And they can have multiple phone numbers.
* The airport contains many airline companies each company has. Each airline has a unique ID, name, country, email (one work email for each employee), and phone number (each airline has one phone number to contact them).
* The flights in the airport are responsible by the airline companies. Each flight has a unique ID, departure date, departure time, arrival time, arrival date, arrival airport, and arrival country.
* The flights are registered by one employee.
* In there are many aircraft stands. So, many flights can park at the aircraft stand. Each aircraft stand has a unique number, status, and size.
* The passengers can pass to the flight through the plane gate so many flights are assigned to one plane gate. Each plane gate has a unique ID, and status.
* Each passenger can reserve many flights. And each passenger has a unique ID, passenger name which consists of first name and last name, birth date, phone number (can have multiple phone numbers), passport ID, nationality (can have multiple nationalities), and email (one email).

## 1.3 User and System Requirements

* Manager of the department:
  + Responsibilities:
    - Monitoring and reviewing the work of employees within the department.
    - managing the department's hiring and onboarding of new workers.
    - ensuring the efficient and regular execution of every departmental activity.
    - monitoring the department's overall performance and guaranteeing effectiveness
  + Interaction with the system:
    - logging in securely to the system to gain access.
    - seeing department performance metrics and reports.
    - Keeping track of employee assignments and records.
* Employees:
  + They responsible for:
    - performing the department's allocated tasks and responsibilities.
    - observing the department manager's directions and recommendations.
    - working together to accomplish departmental goals as a team.
    - letting the department manager know of any problems or worries.
  + Interaction with system:
    - using a secure login to gain access to the system.
    - seeing department performance reports and statistics.
    - managing the assignments and records of employees.
    - Registering the flights.
* Passengers:
* Responsibilities:
  + - acquiring boarding tickets and checking in for their flights.
    - completing immigration and customs formalities, if necessary.
    - getting support and data from airport employees as required.
    - boarding the plane and according to flight safety procedures.
    - Checking their baggage weight.
  + Interaction with system:
* use smartphone applications or self-service kiosks for check-in.
* obtaining flight notifications and updates by SMS/email.
* accessing information on airport facilities and maps.
* Airline company:
* Responsibilities:
  + - organizing and planning flights.
    - coordinating gate allocations and other logistics with airport administration.
    - ensuring that the resources, including staff and aircraft, are available.
    - managing the processes for handling tickets, luggage, and passenger reservations.
  + Interaction with system:
    - Accessing the system through airline-specific accounts.
    - Managing flight schedules, availability, and capacity.
    - Processing passenger reservations and ticketing.

# Part 2: Database Design:

## 2.1 Conceptual Design:

2.1.1 Definition of conceptual design is the first level of logical database design methods. At this level, the primary objective is to create a database that remains independent of any database software or physical constraints. A conceptual data model that details the key data entities, relationships, and constraints of a certain problem domain is the process' output. So, we can have a high-level preview for the database.

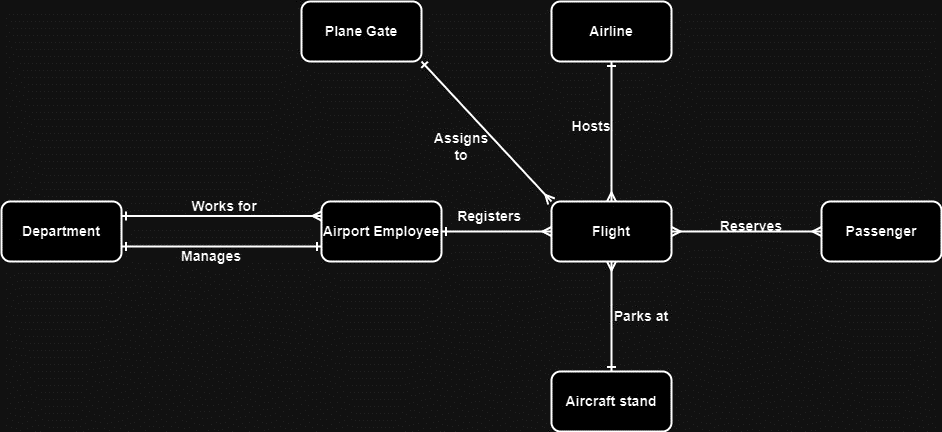
### 2.1.2 The Goal of conceptual design

* Evaluating Requirements: To determine which data needs to be managed and maintained in the database, collect and examine the requirements from users and stakeholders.
* The aim is to promote effective communication and collaboration among designers, developers, and stakeholders by presenting a transparent and easily comprehensible database structure.
* Build the Stage for Logical Design: The conceptual model will be further developed and transformed into a more exact and detailed representation during the ensuing logical design process.

### 2.1.3 Main components of conceptual design

* Entities.
* Relationship between entities: we have three types of relations:
  + One to one 1:1.
  + One to many 1: M.
  + Many to many M: N.

### 2.1.4 the conceptual design of my system

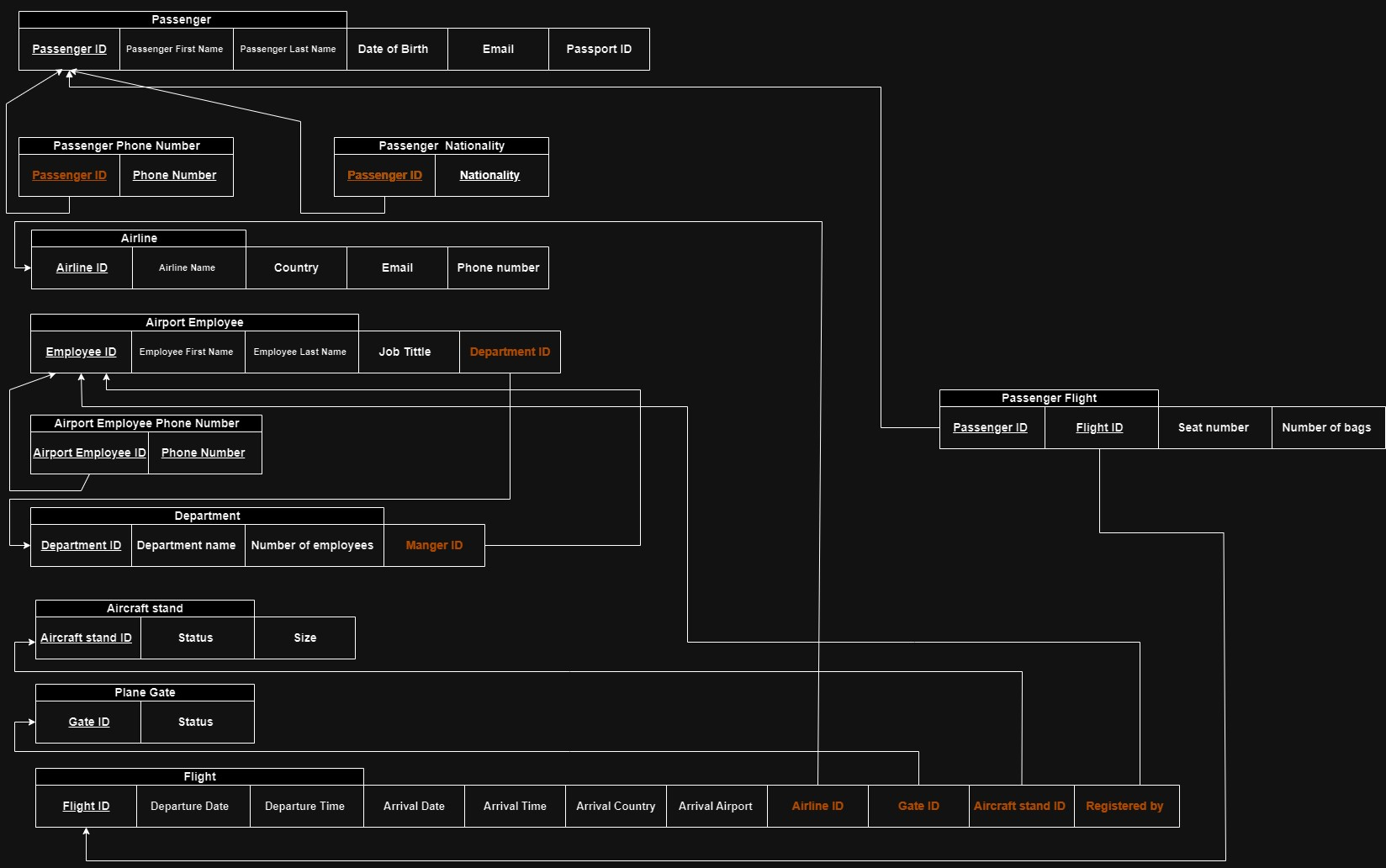


#### 2.1.4.1 Description of the conceptual design

Description of the airport conceptual design: based on the requirements of the airport system as database developer we found that we have 7 entities must be used to do the conceptual database design which they are Airport Employee, Department, Plane Gate, Flight, Airline, Aircraft stand, Passenger. There are relationships between these entities:

* The relationship between the passenger and the flight is many-to-many because many passengers can reserve many flights at the same time and many passengers can reserve the same flight.
* The relationship between flight and aircraft stand is one-to-many because many flights can be parked in the same aircraft stand. (Many flights can park at the same aircraft stand).
* The relationship between the airline and flight is one-to-many because each flight can be owned by one airline company, but the airline companies can own many fights. (Each airline can host many flights).
* The relationship between plane gate and flight is one-to-many because many flights can use the plane gate to let the passengers getting off the airplane to the airport through one plane gate (Many flights can assign to the same plane gate).
* The relationship between department and airport employees is one- to-many because each department has many employees who work in the department.
* The relationship between the airport employee and flight is one-to- many because many flights can be registered with the same airport employee.
* The relationship between departure and airport employees is one-to-one because one manager can manage one department.

## 2.2 Schema after Mapping



Passenger (Passenger ID, Passenger first name, Passenger last name, Birth of Date, Email, Passport ID).

Passenger Phone Number {Passenger ID, Phone Number}.

Passenger Nationality {Passenger ID, nationality}.

Airline (Airline ID, Airline Name, Country, Email, Phone Number).

Airport Employee (Employee ID, Employee first name, Employee last name, Job tittle, Department ID)

Airport Employee Phone Number {Employee ID , Phone Number}.

Department (Department ID, Department Name, Number of Employees, Manger ID).

Aircraft Stand (Aircraft Stand ID, Status, Size)

Plane Gate (Gate ID, Status)

Flight (Flight ID, Departure Date, Departure Time, Arrival Date, Arrival Time, Arrival Country, Arrival Airport, Airline ID, Aircraft Stand, Gate ID, Registered by).

Passenger Flight {Passenger ID, Flight ID, seat Number, Number of Bags}.

## 2.3 Normalization

### 2.3.1: 1st NF

Definition of 1st NF: Modifying multivalued columns to ensure data integrity and minimize redundancy. It is important to have an identifier for each row and atomic values for each attribute in a table. If there are recurring groups, multivalued attributes or composite attributes within a cell this rule is violated. To rectify this issue, we separate these properties into tables. Establish associations, between them using primary keys and foreign keys. This standardization process helps maintain the integrity of the data. So, we can ensure data integrity and minimize redundancy. It is important to have an identifier for each row and atomic values for each attribute in a table.

when it is violated:

* Multi-Valued Attributes.
* Composite Attributes.

|  |  |  |  |
| --- | --- | --- | --- |
| Relations | Attributes | Violation Description | Solution – Relations |
| The relations schema | The attribute name | Describe why it is not in the 1st NF (the violation) | Show the schema for each affected relation. |
| Employee (Employee ID, employee name, birth date, phone number) | Airport Employee (Employee ID, employee first name, birth date, employee first name)  Employee\_Phone number {Employee ID, Phone number} | Employee name is the problem because it is a composite attribute which contain many values. Also, the phone number is problem because it is a multivalued attribute so the employee can have more than one phone number so it’s not 1st NF. | So, firstly we have to sperate the composite attribute to two attributes so we will have an attribute called first name and another attribute called last name.  “Airport Employee (Employee ID, Employee first Name, Employee last Name, job tittle, terminal ID).”  Then we have to create a new table named “Employee Phone number “to fix the multivalued problem by creating a table contain the Employee ID as primary key and phone number as primary key to, and the Employee ID will foreign key too.  Employee Phone number {Employee ID, phone number}. |
| Passenger (Passenger ID, nationality, Time and date for flight) | Passenger\_Nationality {Passenger ID, Nationality}  Passenger (Passenger ID, nationality, flight time, flight date) | Our problem in the nationality and the time and date for the flight. Nationality because some passengers have more than one nationality so we will have multivalued. Also, the time and date to flight is a problem because time and date contain composite value for example 23/march/2023 at 9pm. | To fix the first problem which is the nationality we have to create another table named Passenger Nationality this table contain two attributes passenger ID, and Nationality.  Passenger Nationality {Passenger ID, Nationality}.  Then to fix the date and time to flight problem we have to split the date and time to two attributes.  Passenger (Passenger ID, Date for the flight, Time for the flight, Date of birth, seat number). |

### 2.3.2: 2nd NF

The definition of 2nd NF is a relational database design idea for database normalization. By arranging the data into several linked tables, normalization primarily aims to eliminate data redundancy and improve data integrity. The guidelines of First Normal Form (1NF) are expanded upon in Second Normal Form. uses the idea of ​​a fully functioning dependency as its basis. A composite key relationship, or a primary key relationship consisting of two or more attributes, is in the second normal form. A primary key has only one automatic attribute with a value of at least 2NF. (If a relation is 1NF and all non-key attributes are determined by the full primary key, then and only then is the relation in 2NF) non-primary means attributes that are other than primary key(s) (simple attributes)

When it is violated:

When we have two primary keys, they are not composite and there are separate attributes depending on these primary keys.

|  |  |  |  |
| --- | --- | --- | --- |
| Relations | FDs | Violation Description | Solution – Relations |
| The relations schema | Show the functional dependencies causing the violation | Describe why it is not in the 2nd NF (the violation) | Show the schema for each affected relation. |
| Airport employee Department (employee ID, employee first name, employee last name, birth date, department ID, department name, number of employees in the department, job title) | FD1: Employee ID employee first name, employee last name, Birth Date.  FD2: Department ID department name, number of employees in the department. | Because we got two primary keys, and they are not composite key and there are specific attributes depends on different primary key so each primary keys have a specific attribute depends on it so it’s not 2nd NF.  As we can see in this situation, we have many attributes depends on the **employee ID** only which they are the employee first name, employee last name, and birth date. But department name, and number of employees in the department, are depending on the **Department ID**. | To solve the violation, we got we have to create two table each table has one of the primary keys then all the attributes that depending on that primary key will be add in the same table with the that primary key. Then we will create a third table that contain only the primary keys with what depends on them employee ID, Department ID).  Airport Employee (Employee ID employee first name, employee last name, Birth Date).  Department (Department ID department name, department manager).  Employee\_Department (**employee ID, department ID, job title**) |
| Airline (Airline ID, airline name, airline country, airline email, flight ID, departure country, arrival country, arrival airport, ticket fee). | FD1: Airline ID airline name, airline email, airline country.  FD2: Flight ID flight time, departure country, arrival country, arrival airport. | Because we have two primary keys within the same table it created a problem because they are not composite so there are attributes depends on one of the primary keys and another attribute depends on the other primary key.  As we can see in our example the airline name, airline email, airline country they are only depends on the **airline ID** primary key. On the other hand, the flight time, departure country, arrival country, and arrival airport they are depending on the **flight ID** primary key only. | To solve the violation that we got we have to create two tables each table have a primary key with the attributes that depend on it.  Then we will create a third table that contain only the primary keys with what depends on them with what attributes that depends on both of the primary keys, (Airline ID, Flight ID).  Airline (Airline ID, airline name, airline email, airline country).  Flight (Flight ID, flight time, departure country, arrival country, arrival airport).  Airline\_Flight (**Airline ID, Flight ID,** ticket fee). |

### 2.3.3: 3rd NF

The Definition of 3rd NF is a relation in both in second normal form and third normal form, there is no transitive dependency for non-prime characteristics. Additionally, it is a database normalization technique used in relational databases to reduce duplicate data and enhance data integrity. No transitive functional dependencies should exist in 3NF. As a result, a non-key attribute (column) that depends on another non-key attribute must do so directly rather than transitively.

When it is violated:

When we have attribute (a) then attribute (b) depending on attribute (a) and attribute (c) depends on attribute (b).

|  |  |  |  |
| --- | --- | --- | --- |
| Relations | FDs | Violation Description | Solution – Relations |
| The relations schema | Show the functional dependencies causing the violation | Describe why it is not in the 3rd NF (the violation) | Show the schema for each affected relation. |
| Flight (flight ID, departure date, departure country, arrival postal code, arrival region, arrival country) | FD: Arrival postal code (a) arrival region (b) arrival country (c).  FD: Arrival postal code (a) arrival country (c) | Because as we can see that arrival region (b) as attribute depends on the arrival postal code (a) then arrival country (c) depends on arrival region (b). So, b depends on (a) and (c) depends on (b).  So, as we can see that their transitivity. | We have to create new table contains only the attributes that depending on each other and the attribute (a) which is the arrival postal code will be the primary key. Then we will have a foreign key in the first table (Flight) from the primary key in the new table. (Arrival address).  Flight (flight ID, departure date, departure country, **arrival postal code**).  Arrival address (arrival postal code, arrival region, arrival country). |
| Passenger (passenger ID, passenger first name, passenger last name, flight number, arrival country, arrival time) | FD: Flight number (a) arrival country (b) arrival time (c)  flight number (a) arrival time (c). | As we can see its not 3rd NF because arrival country depending on the flight number. And arrival time (c) depending on the arrival country (b). So, as we can see that their transitivity. | To solve the violation that we got we have to create a new table that contain only the transitive attributes the attributes that depending on each other. As result, the table will contain the flight number (a), arrival country (b), and arrival time. And flight number will be a primary key. Then we will have a foreign key in the first table (Passenger) from the primary key in the new table(Flight information).  Passenger (passenger ID, passenger first name, passenger last name, **flight**).  Flight information (Flight number, arrival country, arrival time). |

## 2.4 Logical Design

2.4.1 Definition of logical design is the process of figuring out how to organize the qualities of the entities in a specific business context into database structures, like the tables of a relational database. All conceptual model objects must be mapped to the precise constructs utilized by the chosen database model in order to comply with logical design requirements. Also, it is the process of creating an enterprise data model based on a certain data model. (It usually follows the requirements analysis phase and is a crucial stage in the system development life cycle). But without going into technical implementation specifics. As result the logical design seeks to identify the system's functional requirements and the relationships between its components. This phase is essential for bridging the gap between the technical design and the commercial requirements.

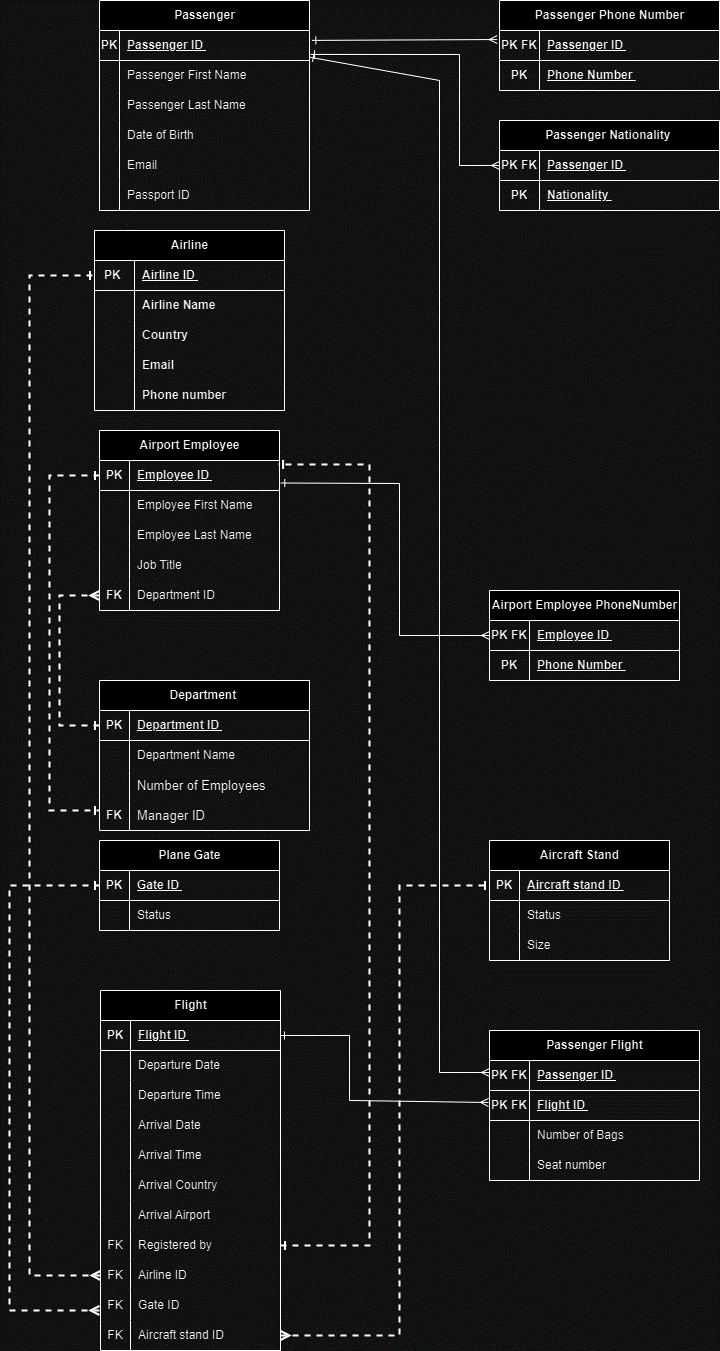
2.4.2 The goal of logical design

* The purpose of logical design is to create a relational schema that effectively and accurately reflects all of the data provided by an Entity-Relationship schema created during the conceptual design phase.
* Functionality: To determine and specify the fundamental tasks, procedures, and actions that the system must carry out in order to satisfy the demands of the user or the business. This guarantees that the logical design effectively achieves its intended aim.
* Data organization involves categorizing the characteristics of entities or data elements into database structures, such as relational database tables, tailored to a specific business context. This systematic arrangement ensures that data is efficiently stored and can be easily retrieved and manipulated when needed.

### 2.4.3 logical design components

* Tables.
* Entities.
* Attributes.
* Relations between the tables.
* Primary keys.
* Foreign keys.

### 2.4.4 Logical design figure



#### 2.4.4.1 Description of the figure

As we can see the logical design is designed after going on the mapping steps and normalization. In logical design we are able to determine what are the attributes of each entity and the relationship between the entities. Also, we are able to determine the primary keys and foreign keys. And from the figure we are able to know the weak and strong relations based on the line (dashed line is a strong relation on the other hand the sold line is the weak relation).

## 2.5 physical design

2.5.1 Definition of physical design is the process of converting the conceptual design into a comprehensive technical specification that describes how the system will be installed and used in a particular computer environment. So, in physical design we decide how the database will be organized for data storage and how users will access that data in order to maintain the database's integrity, security, and performance.

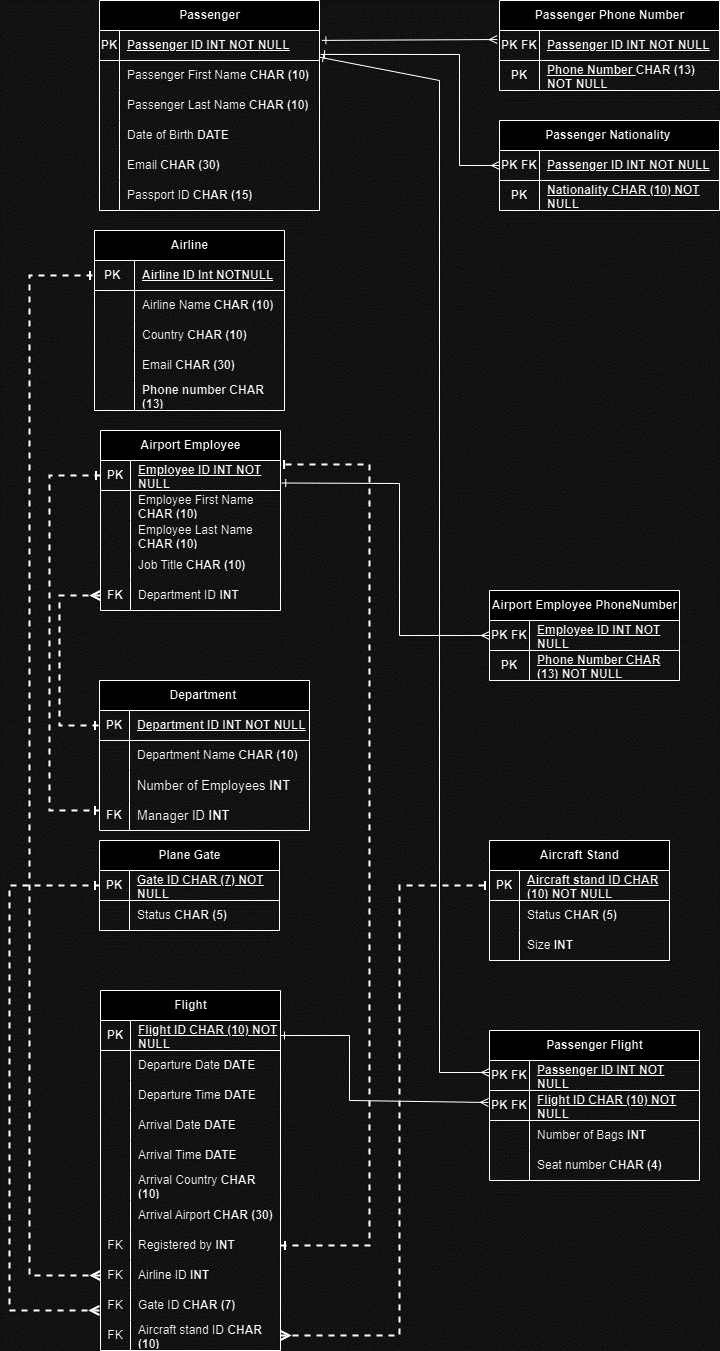
### 2.5.2 the goal of physical design

* To choose the implementation strategy for the logical database design.
* Implement strong security measures by adding a protection layer against illegal access, data breaches, and other potential threats. Because physical design shows us the permissions of everyone.
* Determining the precise storage structures and access techniques to efficiently retrieve data.

### 2.5.3 physical design components

* Tables.
* Entities.
* Attributes.
* Relations between the tables.
* Primary keys.
* Foreign keys.
* Types of each attribute.

### 2.5.4 Physcial desgin figure



#### 2.5.4.1 Description of the figure

As we can see in the figure, we will be able to determine the types of each attribute. Because I added the types of data in the physical design so we can implement our physical design easier.

## 2.6 Effectiveness of the Design

Based on the scenario I got and my understanding of the requirements and user and system requirements. To build the best possible, and efficient database system. Our database system intends to serve a variety of airport stakeholders, including the departments, flights, airline companies, plane gates, aircraft stands, passengers, and employees. The system must meet the specific needs and expectations of each user group, which range from effective administration and monitoring of airport operations to flawless passenger experiences.

The database system requirements, which cover objects like data integrity, performance, security, scalability, and dependability, are equally important. These requirements must be met for the database to function smoothly and efficiently, safeguard sensitive data, and provide an effortless user experience.

Firstly, the data requirements I found them based on the scenario that I got from the client, and I determined all the possible requirements to build the most efficient system. So, my system requirements met the business requirements. Also, I chose the best requirements to build the most efficient interaction system. All the requirements are chosen to create a design that can interact with all possible entities in an efficient way without any issues. As result of that we will be able to build a system that can decrease the problems and can make the users lives easier.

Secondly, I choose all the necessary attributes for each entity to ensure data accuracy, enable sorting and filtering, support decision making of my system, improve data integrity, optimize performance, and ensure security. As result, defining the most essential, important, and possible attributes helped to build a well-designed database. So, my system builds a strong and efficient database design that satisfies the demands of its users and supports numerous airport operations and administration tasks requiring management characteristics.

Thirdly, after determining the requirements, entities. I design a well-defined, conceptual design to meet all the entities that I have. Therefore, without delving into the intricacies of how the data will be physically stored or implemented, it served as a blueprint that defined the general structure, linkages, and organization of the data for me. Therefore, my conceptual design enabled me to accurately grasp the system.

Also, in my design to avoid any space issues, understanding the flow of the database, and integrity issues. I split the names of the passengers and employees into two attributes (first name, and last name). Also, I created a new three tables for passengers phone numbers, employee phone number, and passengers nationality to avoid any composite attributes.

Then I used eight mapping techniques. As a result, my design where abled to define the relationship between the attributes and entities. Which attributes belong to which entity. Also, my design optimized to increase the speed of data management and retrieval. Designers may increase efficiency by taking into account how data will be accessed and stored. So, my design is accurate, easier to maintain, boosts performance, and promotes data security.

After the mapping I used the normalization to avoid any data redundancy. So, my design improves data integrity, space efficiency, simplified data maintenance, and scalability.

Then I built the logical design, my design where able to determine all the relations between the entities and the attributes with their primary keys, and foreign keys. So, I got an indexing strategy. As a result, my design enhances query efficiency, judgments on which characteristics should be indexed are made as part of the logical architecture. Data retrieval and search activities can be sped up using well selected indexes.

Finally, we build the physical design. And my system was able to specify the date type and space of each attribute.

Based on the techniques, ways, strategies, and design. We can find out that I design a very strong, effective, well-defined, and helpful database design. As a result, I built a design that can be used to decrease the cost of submitting the design to servers and software’s.

# References:

Techopedia.com. (2019). *What is First Normal Form (1NF)? - Definition from Techopedia*. [online] Available at: <https://www.techopedia.com/definition/25955/first-normal-form-1nf>.

*Database Design Phase 2: Conceptual Design*. (n.d.). MariaDB KnowledgeBase. <https://mariadb.com/kb/en/database-design-phase-2-conceptual-design/>

‌ *Departments of Airport*. (n.d.). Ashgabatairport.gov.tm. Retrieved August 2, 2023, from <https://ashgabatairport.gov.tm/en/airport-service>

‌Upadhyay, M. (2019, July 31). *Second Normal Form (2NF) - GeeksforGeeks*. GeeksforGeeks. <https://www.geeksforgeeks.org/second-normal-form-2nf/>

Atzeni, P., Ceri, S., Paraboschi, S., & Torlone, R. (n.d.). *These slides are for use with Database Systems Concepts, Languages and Architectures*. <https://users.dimi.uniud.it/~angelo.montanari/logicDesign.pdf>

CHAPTER 7: LOGICAL DATABASE DESIGN. (2019). *Fundamentals of Database Management Systems, Second Edition*. O’Reilly | Safari. <https://www.oreilly.com/library/view/fundamentals-of-database/9781118213575/15_chap07.html> ‌

Thiru. (2017). *Logical Design*. Myreadingroom.co.in. <http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/508-logical-design.html>

‌*Physical Design*. (n.d.). Www.myreadingroom.co.in. <http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/509-physical-design.html>

*Physical Database Design - an overview | ScienceDirect Topics*. (n.d.). Www.sciencedirect.com. <https://www.sciencedirect.com/topics/computer-science/physical-database-design>

Upadhyay, M. (2019, July 31). *Third Normal Form (3NF) - GeeksforGeeks*. GeeksforGeeks. <https://www.geeksforgeeks.org/third-normal-form-3nf/>

Becker, R. (2019). *What is Third Normal Form (3NF)? - Definition from Techopedia*. Techopedia.com. <https://www.techopedia.com/definition/22561/third-normal-form-3nf>