Dublin Airport Database Management System

Michael Agbenla (C22495816)

Advanced Databases

TU914/2

System Descriptions

Dublin Airport Database Management System is designed to efficiently handle the diverse operations of the airport, ensuring smooth passenger experiences and optimal resource utilization. Apart from core functionalities like managing passengers, flights, and staff, it integrates advanced features such as baggage tracking, security incident reporting, retail management, and parking space allocation. Dublin Airport, like many modern airports, handles a vast amount of data and diverse operations on a daily basis.

Entity Relationship Model (ER Model): Expanding beyond the basics, our system includes entities such as passengers, flights, airlines, staff, gates, baggage, security incidents, retail stores, products, and parking spaces. Each entity is carefully designed with its unique attributes to capture the intricacies of airport operations.

Choosing Dublin Airport as the subject for the database development project is a personal decision rooted in real-world experience. As a Search Unit Officer at Dublin Airport, I have firsthand insight into the various operational challenges and intricacies involved in managing such a complex and dynamic environment.

Conclusion: In conclusion, the Dublin Airport Database Management System represents a holistic approach to managing airport operations, incorporating both core functionalities and advanced features to meet the diverse needs of modern airports. Through careful analysis, design, and implementation, we aim to create a solution that not only addresses current challenges but also adapts to future requirements, ensuring the continued success of Dublin Airport as a key transportation hub. Dublin Airport requires a robust DBMS to effectively manage its complex operations, ensure regulatory compliance, enhance security measures, optimize resource utilization, and deliver a seamless experience for passengers and stakeholders. By centralizing data and providing powerful tools for analysis and decision-making, a DBMS plays a vital role in the efficient functioning of modern airports like Dublin Airport

The ER model consists of the following entities:

* Passengers
* Flights
* Airlines
* Staff
* Gates
* Baggage
* SecurityIncidents
* RetailStores
* Products

Passengers:

* PassengerID (Primary Key)
* Name
* PassportNumber
* Nationality
* ContactDetails
* TicketNumber (Foreign Key referencing Flights)

Flights:

* FlightID (Primary Key)
* AirlineID (Foreign Key referencing Airlines)
* DepartureDateTime
* ArrivalDateTime
* Origin
* Destination

Airlines:

* AirlineID (Primary Key)
* Name
* Country
* ContactDetails

Staff:

* StaffID (Primary Key)
* Name
* Role
* ContactDetails

Gates:

* GateID (Primary Key)
* Terminal
* Location
* Availability

Baggage:

* BaggageID (Primary Key)
* PassengerID (Foreign Key referencing Passengers)
* FlightID (Foreign Key referencing Flights)
* Weight
* Destination

SecurityIncidents:

* IncidentID (Primary Key)
* Description
* DateOccurred
* StaffID (Foreign Key referencing Staff)

RetailStores:

* StoreID (Primary Key)
* Name
* Location
* ContactDetails

Products:

* ProductID (Primary Key)
* Name
* Description
* Price
* StoreID (Foreign Key referencing RetailStores)

Relational Model:

Passengers (PassengerID, Name, PassportNumber, Nationality, ContactDetails, TicketNumber)

Flights (FlightID, AirlineID, DepartureDateTime, ArrivalDateTime, Origin, Destination)

Airlines (AirlineID, Name, Country, ContactDetails)

Staff (StaffID, Name, Role, ContactDetails)

Gates (GateID, Terminal, Location, Availability)

Baggage (BaggageID, PassengerID, FlightID, Weight, Destination)

SecurityIncidents (IncidentID, Description, DateOccurred, StaffID)

RetailStores (StoreID, Name, Location, ContactDetails)

Products (ProductID, Name, Description, Price, StoreID)

CREATE TABLE Passengers (

PassengerID AUTOINCREMENT PRIMARY KEY,

Name TEXT(50),

PassportNumber TEXT(20),

Nationality TEXT(50),

ContactDetails MEMO,

TicketNumber LONG,

FOREIGN KEY (TicketNumber) REFERENCES Flights(FlightID)

);

CREATE TABLE Flights (

FlightID AUTOINCREMENT PRIMARY KEY,

AirlineID LONG,

DepartureDateTime DATETIME,

ArrivalDateTime DATETIME,

Origin TEXT(50),

Destination TEXT(50),

FOREIGN KEY (AirlineID) REFERENCES Airlines(AirlineID)

);

CREATE TABLE Airlines (

AirlineID AUTOINCREMENT PRIMARY KEY,

Name TEXT(50),

Country TEXT(50),

ContactDetails MEMO

);

CREATE TABLE Staff (

StaffID AUTOINCREMENT PRIMARY KEY,

Name TEXT(50),

Role TEXT(50),

ContactDetails MEMO

);

CREATE TABLE Gates (

GateID AUTOINCREMENT PRIMARY KEY,

Terminal TEXT(10),

Location TEXT(50),

Availability YESNO

);

CREATE TABLE Baggage (

BaggageID AUTOINCREMENT PRIMARY KEY,

PassengerID LONG,

FlightID LONG,

Weight FLOAT,

Destination TEXT(50),

FOREIGN KEY (PassengerID) REFERENCES Passengers(PassengerID),

FOREIGN KEY (FlightID) REFERENCES Flights(FlightID)

);

CREATE TABLE SecurityIncidents (

IncidentID AUTOINCREMENT PRIMARY KEY,

Description MEMO,

DateOccurred DATETIME,

StaffID LONG,

FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)

);

CREATE TABLE RetailStores (

StoreID AUTOINCREMENT PRIMARY KEY,

Name TEXT(50),

Location TEXT(50),

ContactDetails MEMO

);

CREATE TABLE Products (

ProductID AUTOINCREMENT PRIMARY KEY,

Name TEXT(50),

Description MEMO,

Price CURRENCY,

StoreID LONG,

FOREIGN KEY (StoreID) REFERENCES RetailStores(StoreID)

);