BITR ANALYTICS

SQL DATABASE DESIGN FOR FLIGHT BOOKING SYSTEM

USING MYSQL and MICROSOFT SQL SERVER

BY

OLAKUNLE ADEGEYE

OF CONTENT

TABLE OF FIGURES

1.0.INTRODUCTION

This report details the process of the first phase of the database design for a flight booking using MySQL and Microsoft SQL Server. The design was done according to the requirement specification from the client which can be accessed in the appendix HERE

The procedures and principles guiding the design of an efficient and optimized database are highlighted.

MySQL Workbench 8.0 was used for MySQL and Microsoft SQL Server Management Studio 18 was used for the Microsoft database.

The database was modelled using the Entity Relationship (ER) diagram to present the conceptual and logical model of the database and showed the relationships between the tables, including the cardinalities.

It shows the process of designing a normalized database and putting it in the 1st normal form, 2nd normal form and 3rd normal form (1NF, 2NF and 3NF). The query statements used in creating the tables, rules and constraints are also presented.

Screenshots of the implementation on both MySQL and Microsoft SQL Server are presented in the report, while links to the full scripts and high-resolution images are available in the appendix.

The report further presents the justifications for the selection of MySQL and MSSQL Server in addition to presenting their limitations.

2.0.DATABASE MODELLING PROCESS

This stage comprises the:

- Conceptual Model
- Logical Model
- Physical Model

2.1.Conceptual model

2.1.1. ER data model

The Entity Relationship data model would be used to present the relationships between the tables.

The ER data model was used because it reduces redundancy among its entities and clearly defines their relationships and cardinalities.

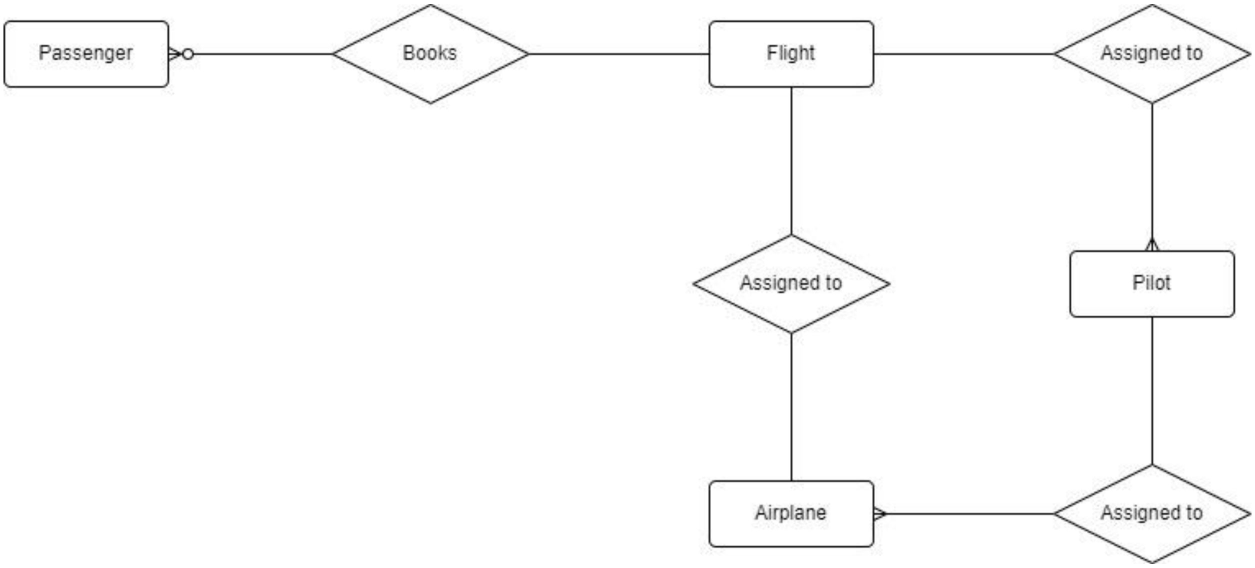


Fig 1: ER data model of Conceptual Model

The ER data model show the relationship between the various entities.

The Passenger has a many-to-one relationship with flights, meaning that many passengers can book one Flight, Flight and Airplane share a one-to-one relationship because only one airplane can be to attached to a specific flight.

Pilot has a two-to-one relationship with flight because 2 pilots are usually attached to a flight – the Captain and the First Officer.

Pilot has a one-to-many relationship with airplanes because one pilot can have the license to fly more than one airplane. Nevertheless, at any specific point in time, one pilot can only be assigned to one airplane at a time.

2.1.2. ER diagram

The ER diagram is used to present the attributes of the conceptual data model because it shows the information in a more organized format.

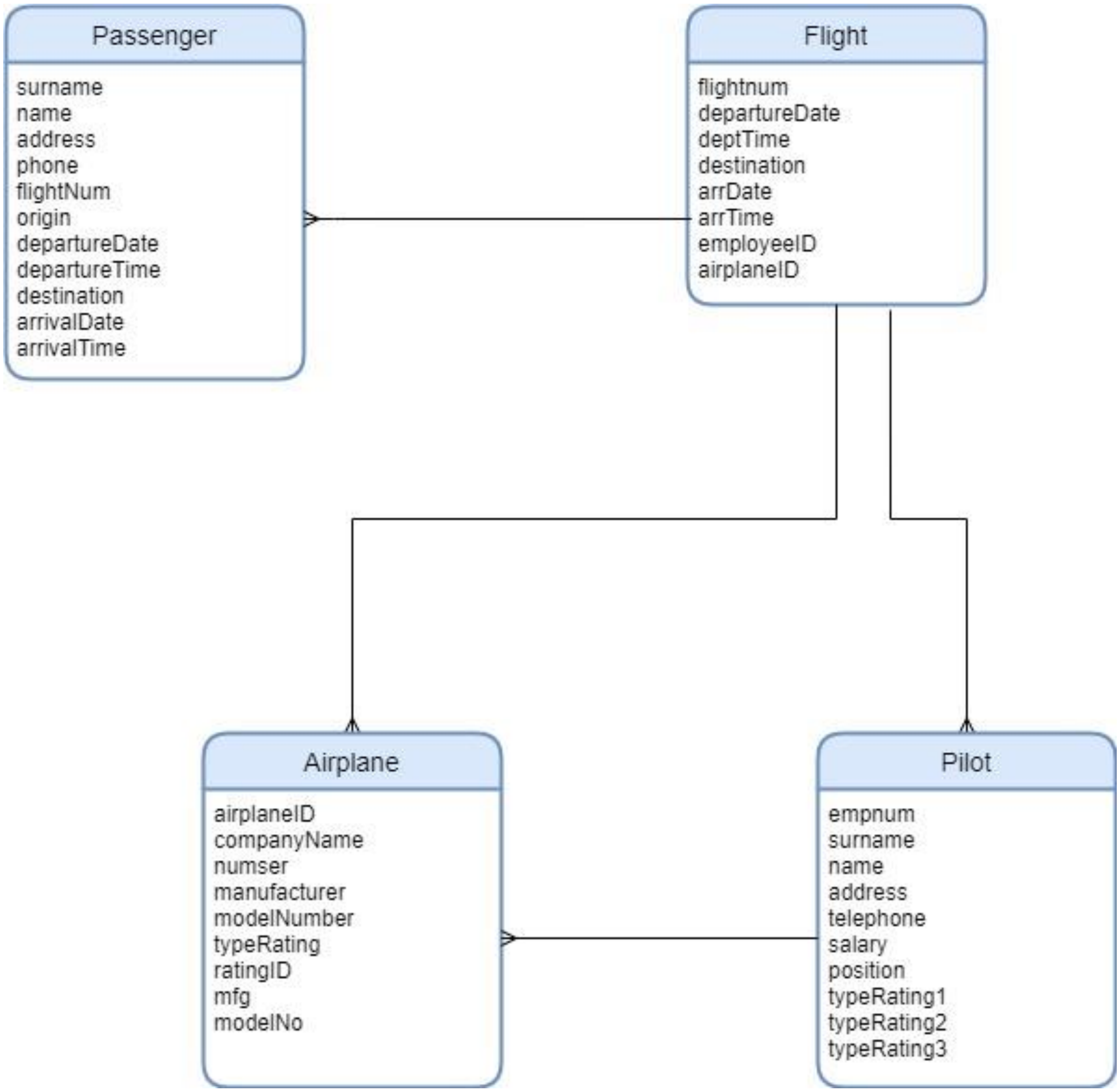


Fig 2: ER diagram of Conceptual Model

The above ER diagram is un-normalized conceptual model which shows the entities and their attributes.

2.2.Logical Model

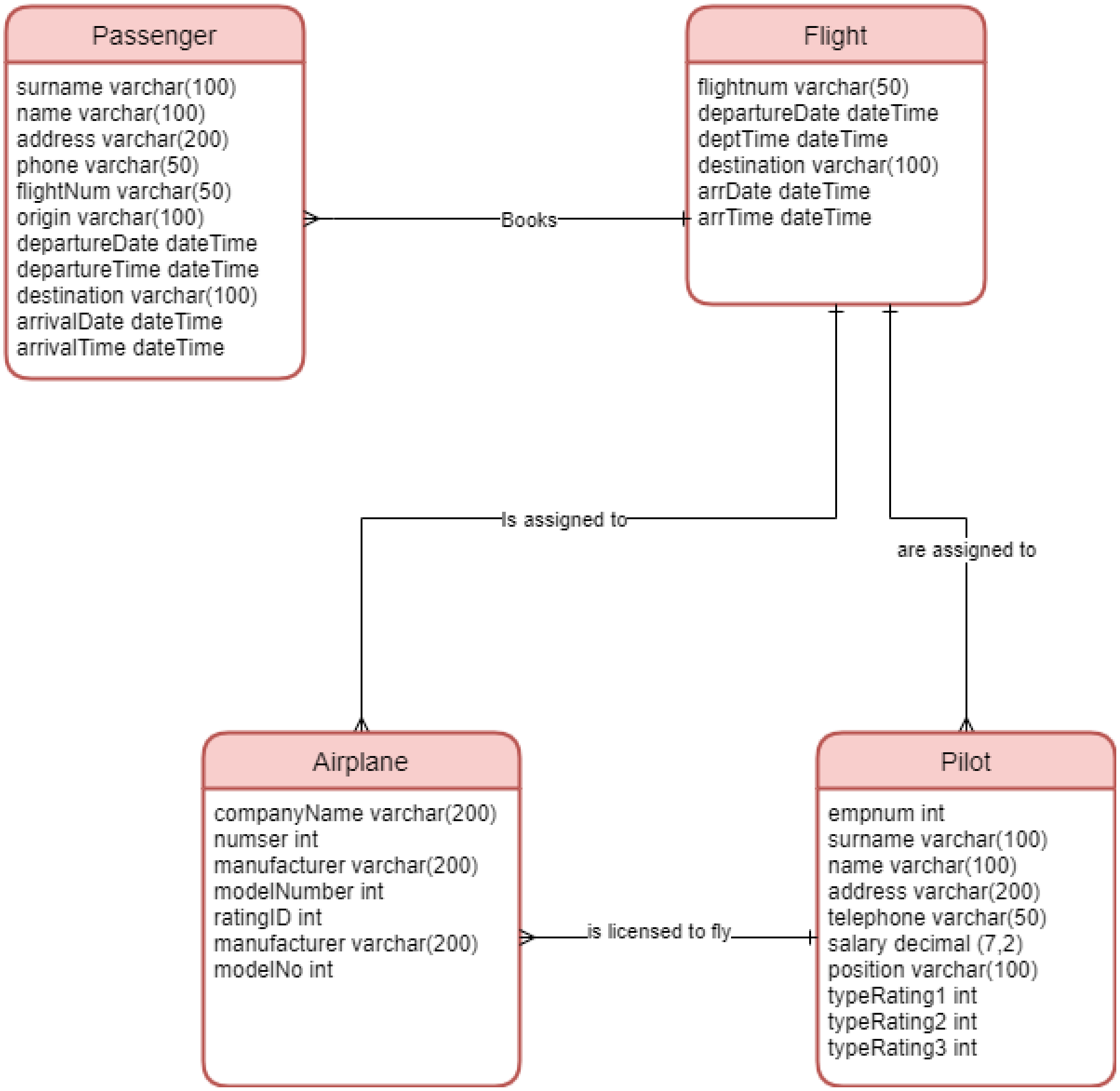


Fig 3: ER diagram of Un-normalized logical model

The above image is an un-normalized logical model containing the entities, attributes, data types and cardinalities. In the following section, the table would be normalized to the first, second and third normal form.

2.2.1. Normalization

In this section below, the tables are normalized in order to remove redundancy, optimize performance and improve data integrity.

1st Normal Form

First normal form: holds the following:

- There should be a primary key for each column
- Each column must have a unique name
- No rows duplicated must be duplicated

According to the rules above, the table is not currently in the 1NF form and would be modified to satisfy the rules.

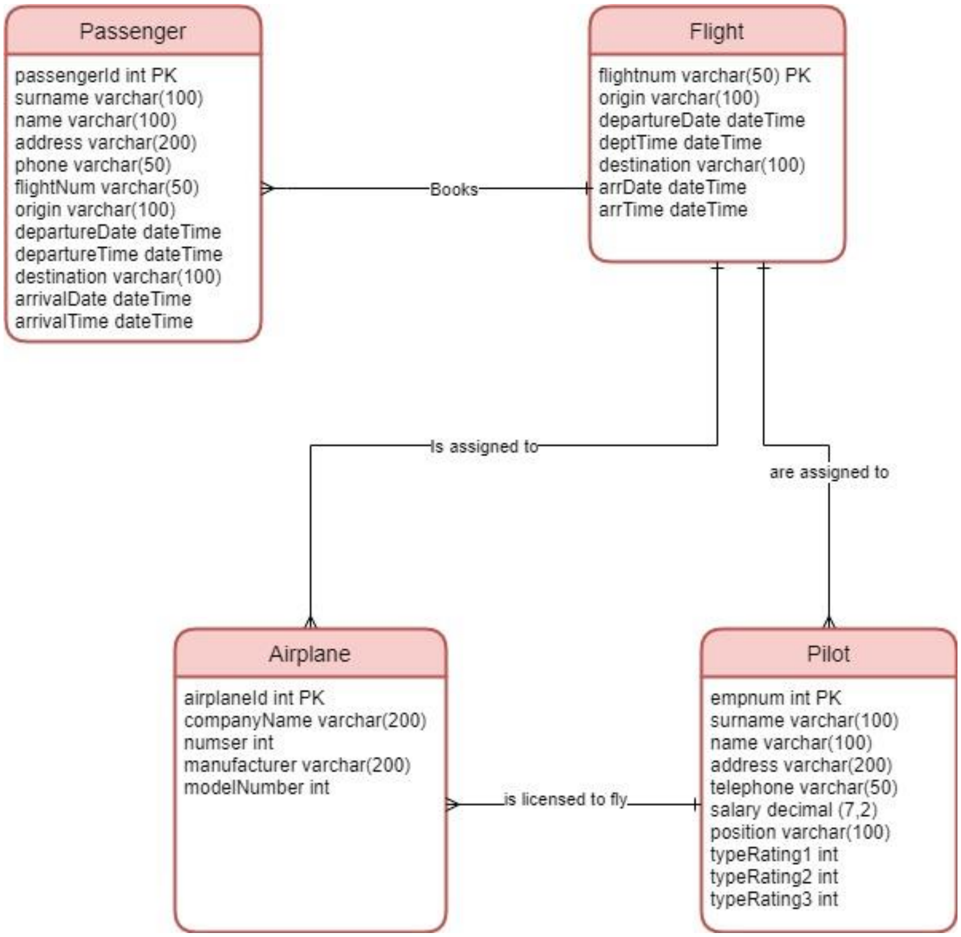


Fig 4: ER diagram of Logical Model in 1NF

Unique Ids were added to the passenger and airplane table respectively. The duplicate manufacturer and model number rows were removed since there was already a previous one which could be referred to for the type rating. With these adjustments, the table is now in the first normal form.

Second Normal Form:

It holds that every column in a table should describe just one entity. According to this rule, the table in fig 4 is not in the second normal form.

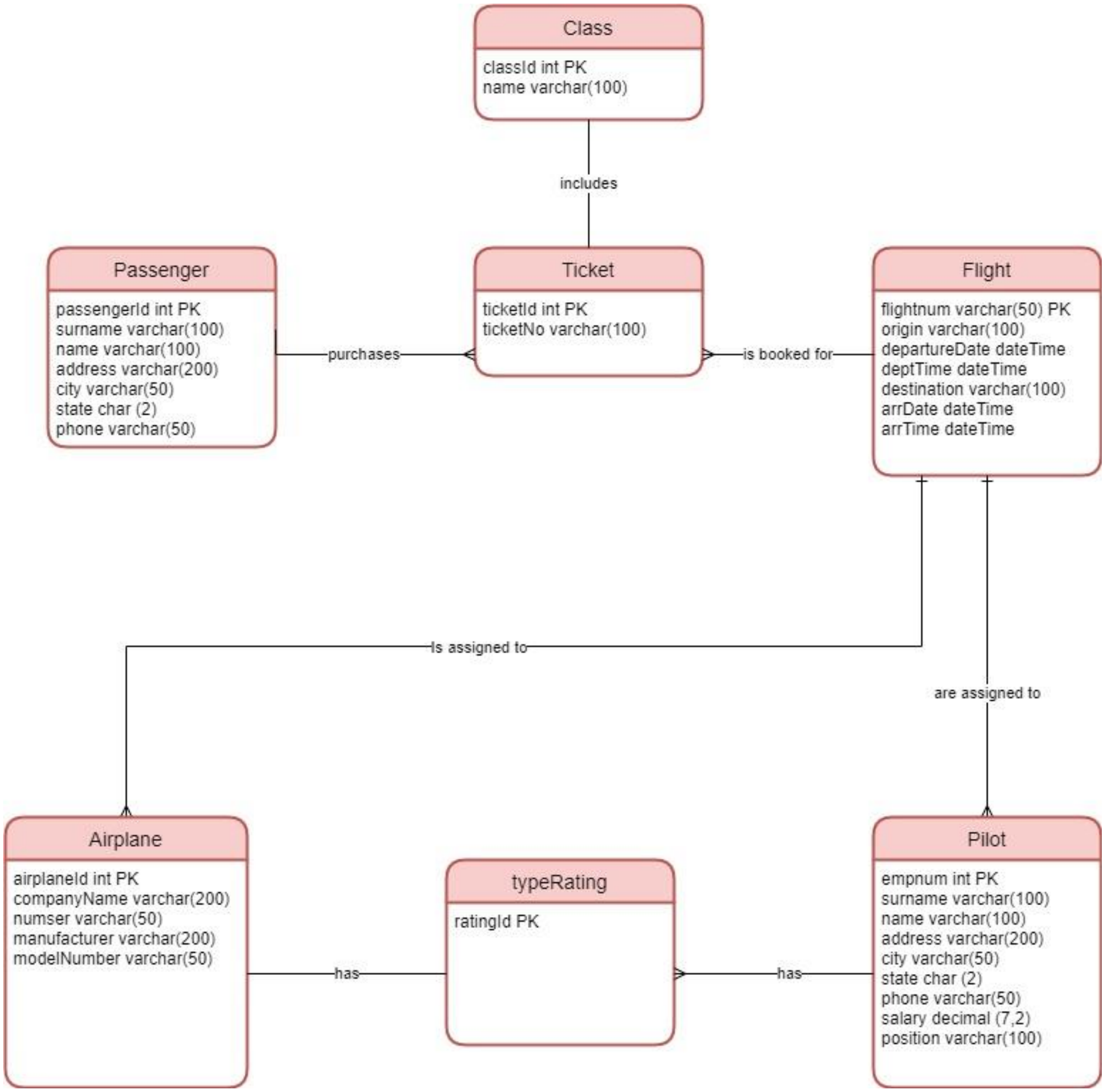


Fig 5: ER diagram of Logical Model in 2NF

In the passenger entity, the attributes of the flights were removed since it does not relate to the passenger. A ticket booking table was added which would contain information about the flight and serve as a sort of Link Table between the Passenger and Flight table. An entity called class was also created, as well as a typerating entity to hold the different flight type ratings

With these adjustments, the model is now in the second normal form.

Third normal form

Third normal form holds that fields that do not depend on the key should be eliminated. It also holds that an attribute should not be such that it can be easily derived from another attribute. For example, the combination of the plane manufacturer and model number gives the aircraft number. Including an attribute for aircraft number would have been violating the third normal form, so it wasn't included. The data model has been optimized to the third normal form.

2.3.Physical Model

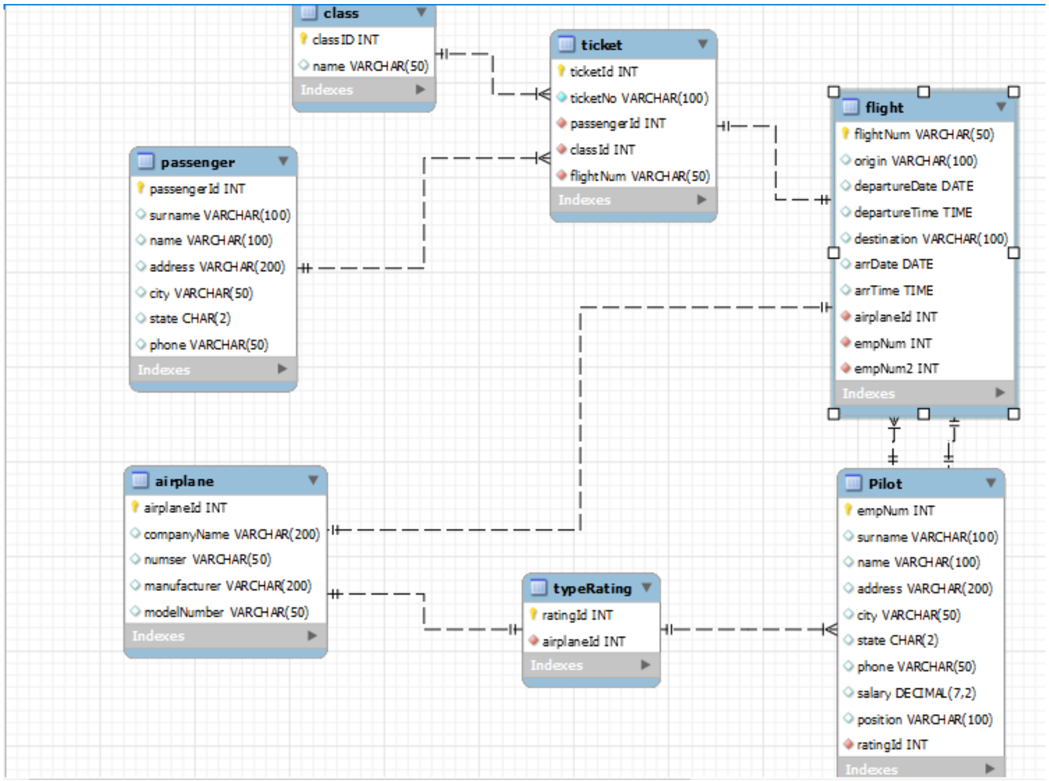


Fig 6: Enhanced Entity Relationship (EER) physical model MySQL

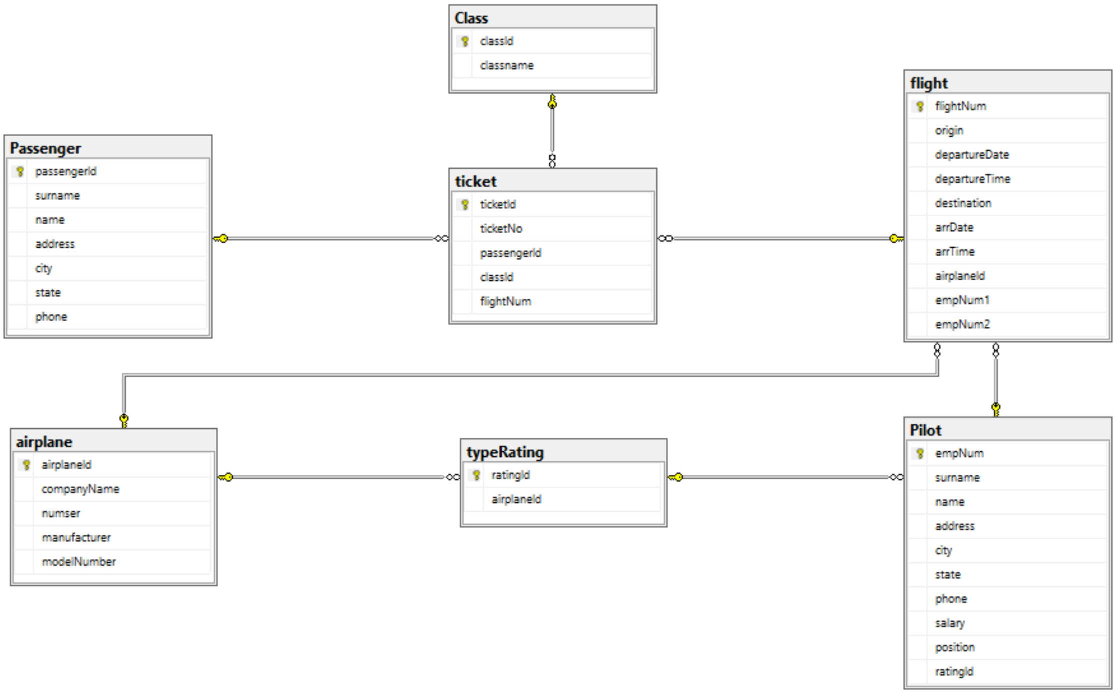
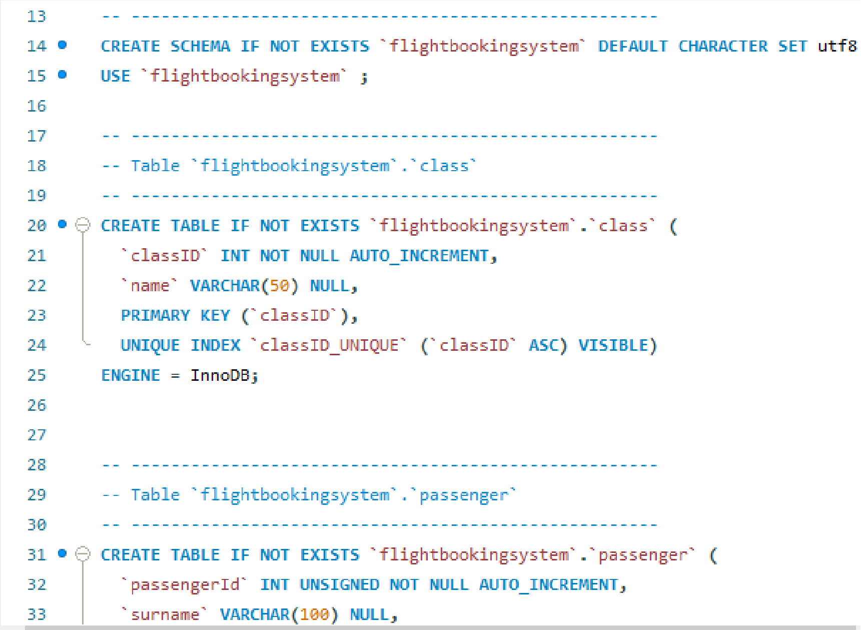


Fig 7: Enhanced Entity Relationship (EER) physical model Microsoft Server

3.0.IMPLEMENTATION QUERIES

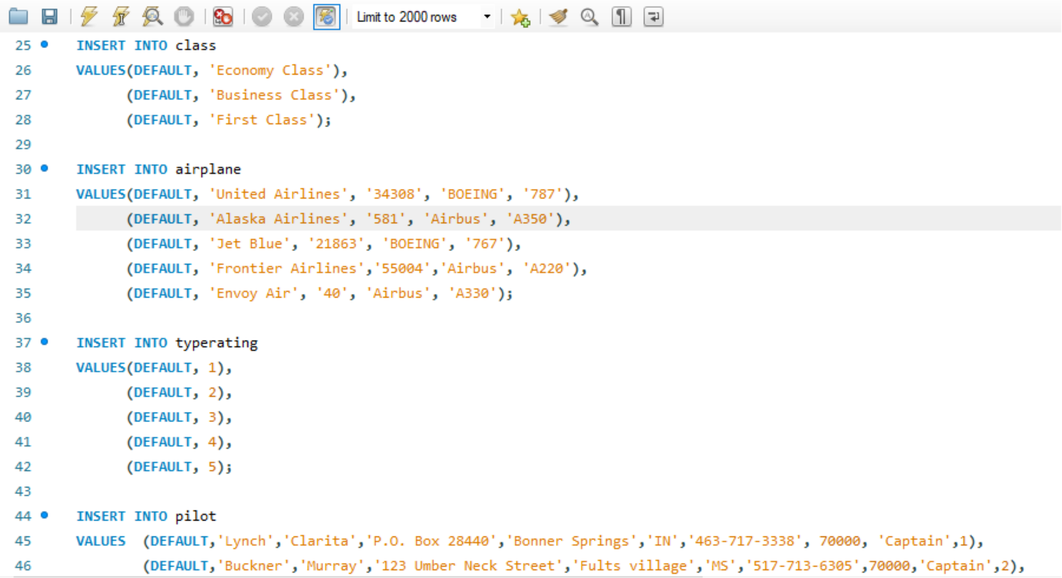
This section presents screenshots from the implementation on both MySQL and Microsoft Server



```sql
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `flightbookingsystem` DEFAULT CHARACTER SET utf8
USE `flightbookingsystem` ;

-- -----------------------------------------------------
-- Table `flightbookingsystem`.`class`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `flightbookingsystem`.`class` (
  `classID` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NULL,
  PRIMARY KEY (`classID`),
  UNIQUE INDEX `classID_UNIQUE` (`classID` ASC) VISIBLE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `flightbookingsystem`.`passenger`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `flightbookingsystem`.`passenger` (
  `passengerId` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `surname` VARCHAR(100) NULL,
```

Fig 8: Database Creation Query in MySQL

Fig 8 is a code snippet of the database creation in MySQL

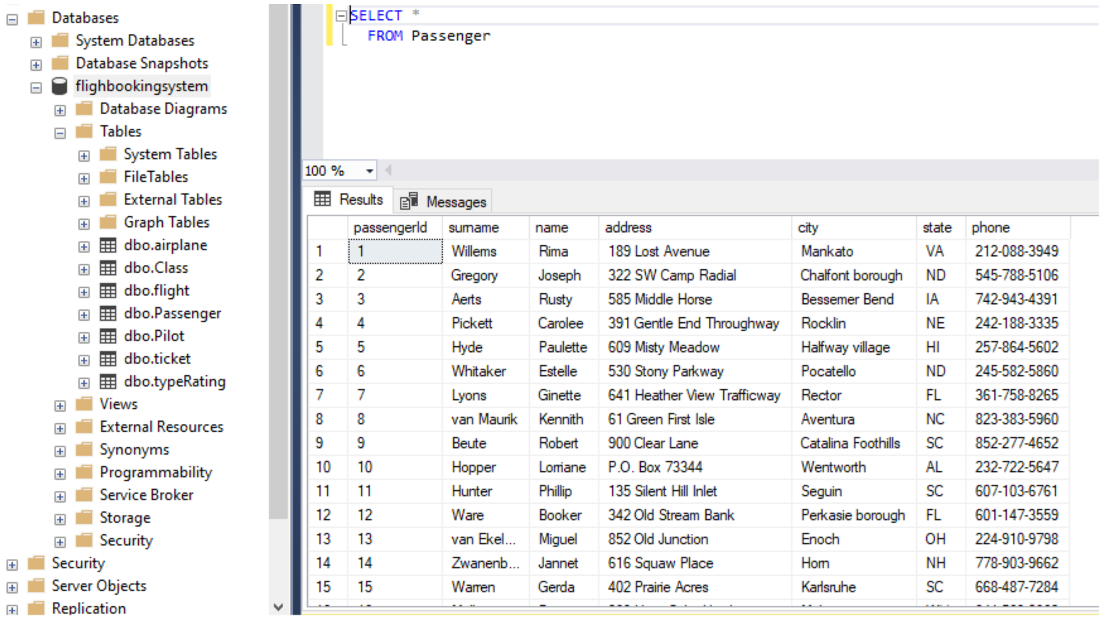


Limit to 2000 rows

```sql
INSERT INTO class
VALUES(DEFAULT, 'Economy Class'),
      (DEFAULT, 'Business Class'),
      (DEFAULT, 'First Class');

INSERT INTO airplane
VALUES(DEFAULT, 'United Airlines', '34308', 'BOEING', '787'),
      (DEFAULT, 'Alaska Airlines', '581', 'Airbus', 'A350'),
      (DEFAULT, 'Jet Blue', '21863', 'BOEING', '767'),
      (DEFAULT, 'Frontier Airlines','55004','Airbus', 'A220'),
      (DEFAULT, 'Envoy Air', '40', 'Airbus', 'A330');

INSERT INTO typerating
VALUES(DEFAULT, 1),
      (DEFAULT, 2),
      (DEFAULT, 3),
      (DEFAULT, 4),
      (DEFAULT, 5);

INSERT INTO pilot
VALUES  (DEFAULT,'Lynch','Clarita','P.O. Box 28440','Bonner Springs','IN','463-717-3338', 70000, 'Captain',1),
        (DEFAULT,'Buckner','Murray','123 Umber Neck Street','Fults village','MS','517-713-6305',70000,'Captain',2),
```

Fig 9: Value Insertion in MySQL

Fig 9 shows snippet of the value insertion used to populate the table in MySQL. Click Here for the full MySQL and Microsoft SQL Server script

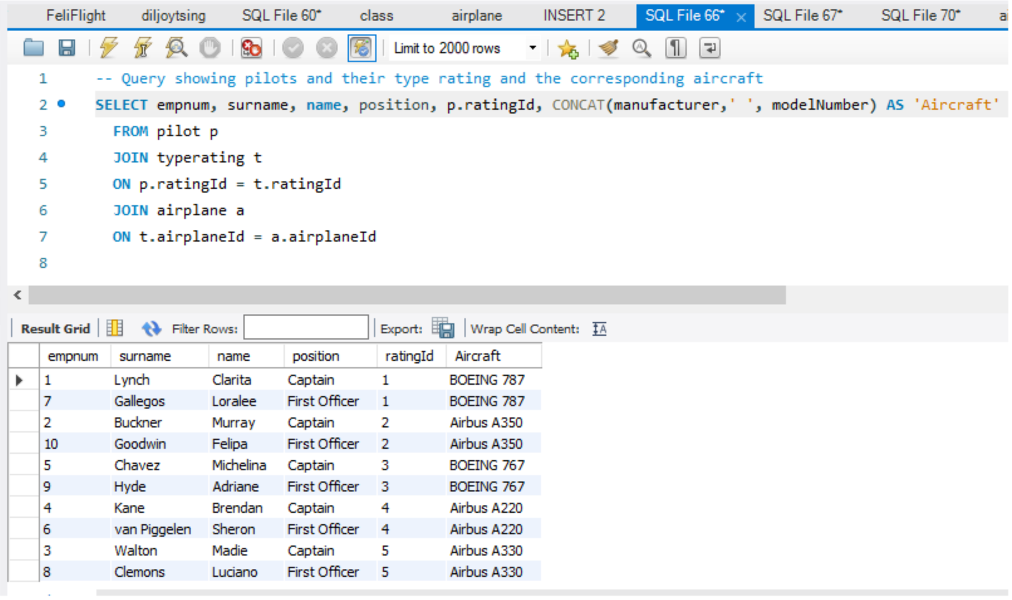


```
SELECT *
  FROM Passenger
```

	passengerId	surname	name	address	city	state	phone
1	1	Willems	Rima	189 Lost Avenue	Mankato	VA	212-088-3949
2	2	Gregory	Joseph	322 SW Camp Radial	Chalfont borough	ND	545-788-5106
3	3	Aerts	Rusty	585 Middle Horse	Bessemer Bend	IA	742-943-4391
4	4	Pickett	Carolee	391 Gentle End Throughway	Rocklin	NE	242-188-3335
5	5	Hyde	Paulette	609 Misty Meadow	Halfway village	HI	257-864-5602
6	6	Whitaker	Estelle	530 Stony Parkway	Pocatello	ND	245-582-5860
7	7	Lyons	Ginette	641 Heather View Trafficway	Rector	FL	361-758-8265
8	8	van Maurik	Kennith	61 Green First Isle	Aventura	NC	823-383-5960
9	9	Beute	Robert	900 Clear Lane	Catalina Foothills	SC	852-277-4652
10	10	Hopper	Lorriane	P.O. Box 73344	Wentworth	AL	232-722-5647
11	11	Hunter	Phillip	135 Silent Hill Inlet	Seguin	SC	607-103-6761
12	12	Ware	Booker	342 Old Stream Bank	Perkasie borough	FL	601-147-3559
13	13	van Ekel...	Miguel	852 Old Junction	Enoch	OH	224-910-9798
14	14	Zwanenb...	Jannet	616 Squaw Place	Horn	NH	778-903-9662
15	15	Warren	Gerda	402 Prairie Acres	Karlsruhe	SC	668-487-7284

Fig 10: Query of Passenger table in Microsoft SQL Server

Fig 10 shows a query statement of the passenger table and the result. The results were limited to 10 records.

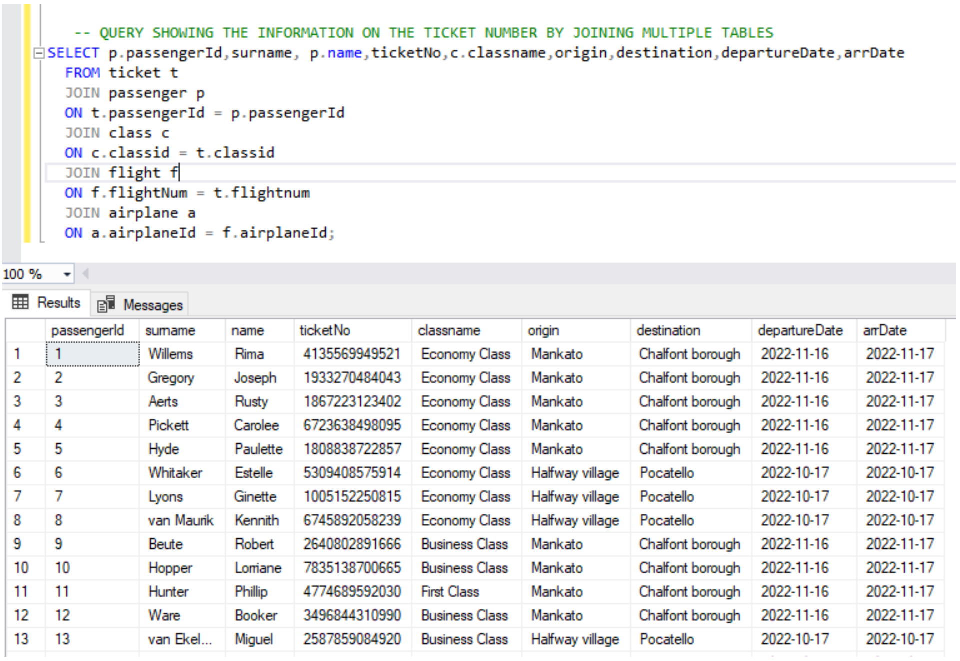


```
-- Query showing pilots and their type rating and the corresponding aircraft
SELECT empnum, surname, name, position, p.ratingId, CONCAT(manufacturer,' ', modelNumber) AS 'Aircraft'
  FROM pilot p
  JOIN typerating t
  ON p.ratingId = t.ratingId
  JOIN airplane a
  ON t.airplaneId = a.airplaneId
```

empnum	surname	name	position	ratingId	Aircraft
1	Lynch	Clarita	Captain	1	BOEING 787
7	Gallegos	Loralee	First Officer	1	BOEING 787
2	Buckner	Murray	Captain	2	Airbus A350
10	Goodwin	Felipa	First Officer	2	Airbus A350
5	Chavez	Michelina	Captain	3	BOEING 767
9	Hyde	Adriane	First Officer	3	BOEING 767
4	Kane	Brendan	Captain	4	Airbus A220
6	van Piggelen	Sheron	First Officer	4	Airbus A220
3	Walton	Madie	Captain	5	Airbus A330
8	Clemons	Luciano	First Officer	5	Airbus A330

Fig 11: Pilot Type Rating in MySQL

Fig 11 shows the rating of the pilots and the aircrafts they are licensed to fly.

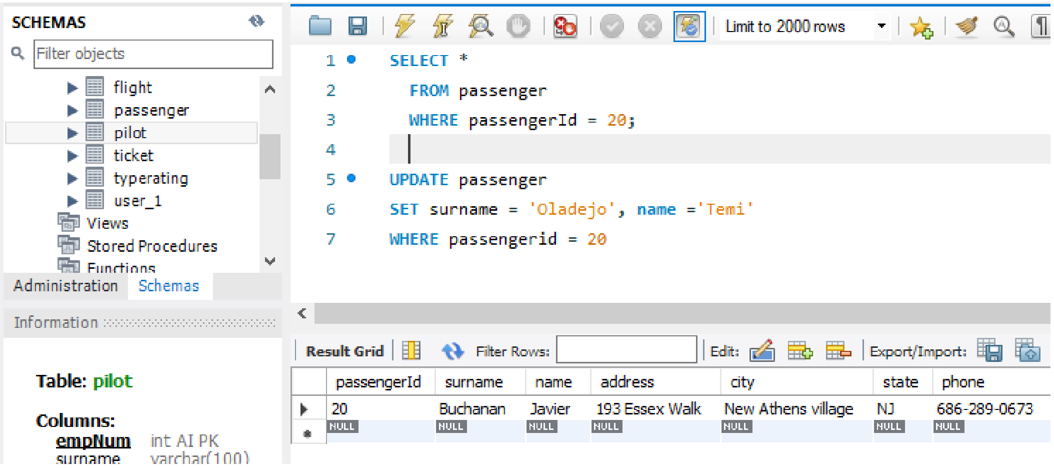


```
 -- QUERY SHOWING THE INFORMATION ON THE TICKET NUMBER BY JOINING MULTIPLE TABLES
SELECT p.passengerId,surname, p.name,ticketNo,c.classname,origin,destination,departureDate,arrDate
  FROM ticket t
  JOIN passenger p
  ON t.passengerId = p.passengerId
  JOIN class c
  ON c.classid = t.classid
  JOIN flight f
  ON f.flightNum = t.flightnum
  JOIN airplane a
  ON a.airplaneId = f.airplaneId;
```

	passengerId	surname	name	ticketNo	classname	origin	destination	departureDate	arrDate
1	1	Willems	Rima	4135569949521	Economy Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
2	2	Gregory	Joseph	1933270484043	Economy Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
3	3	Aerts	Rusty	1867223123402	Economy Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
4	4	Pickett	Carolee	6723638498095	Economy Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
5	5	Hyde	Paulette	1808838722857	Economy Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
6	6	Whitaker	Estelle	5309408575914	Economy Class	Halfway village	Pocatello	2022-10-17	2022-10-17
7	7	Lyons	Ginette	1005152250815	Economy Class	Halfway village	Pocatello	2022-10-17	2022-10-17
8	8	van Maurik	Kennith	6745892058239	Economy Class	Halfway village	Pocatello	2022-10-17	2022-10-17
9	9	Beute	Robert	2640802891666	Business Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
10	10	Hopper	Lorriane	7835138700665	Business Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
11	11	Hunter	Phillip	4774689592030	First Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
12	12	Ware	Booker	3496844310990	Business Class	Mankato	Chalfont borough	2022-11-16	2022-11-17
13	13	van Ekel...	Miguel	2587859084920	Business Class	Halfway village	Pocatello	2022-10-17	2022-10-17

Fig 12: Ticket information in Microsoft SQL Server

Fig 12 shows the information on the ticket. The ticket table was joined with the passenger, flight class, flights and airplane to provide a detailed view of the ticket information.



```
SELECT *
  FROM passenger
  WHERE passengerId = 20;

UPDATE passenger
SET surname = 'Oladejo', name ='Temi'
WHERE passengerid = 20
```

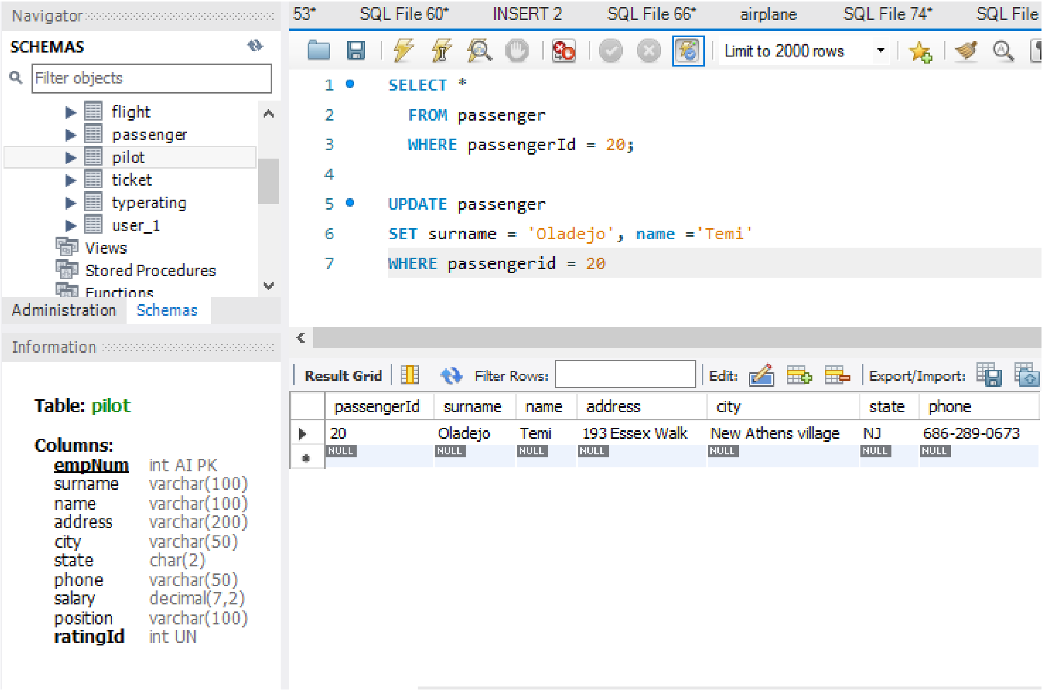
Table: pilot

Columns:
empNum int AI PK
surname varchar(100)

	passengerId	surname	name	address	city	state	phone
▶	20	Buchanan	Javier	193 Essex Walk	New Athens village	NJ	686-289-0673
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Fig 13: Pre-updated table in MySQL

Fig 13 shows the result of passengerId 20 before the table was updated



```sql
SELECT *
  FROM passenger
  WHERE passengerId = 20;

UPDATE passenger
SET surname = 'Oladejo', name ='Temi'
WHERE passengerid = 20
```

	passengerId	surname	name	address	city	state	phone
▶	20	Oladejo	Temi	193 Essex Walk	New Athens village	NJ	686-289-0673
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

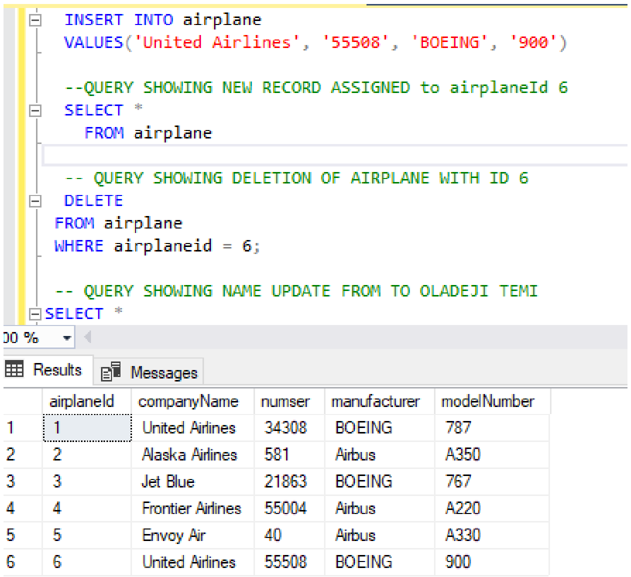
Table: pilot

Columns:

empNum	int AI PK
surname	varchar(100)
name	varchar(100)
address	varchar(200)
city	varchar(50)
state	char(2)
phone	varchar(50)
salary	decimal(7,2)
position	varchar(100)
ratingId	int UN

Fig 14: Post-updated table in MySQL

Fig 14 shows the records of passengerid 20 after it was updated to Oladejo Temi.

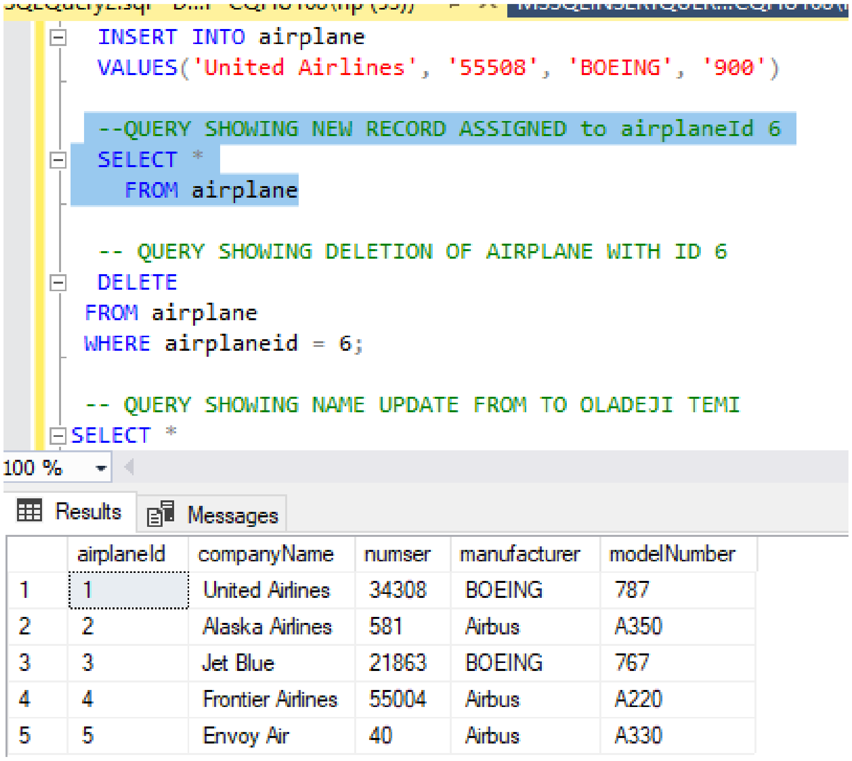


```sql
INSERT INTO airplane
VALUES('United Airlines', '55508', 'BOEING', '900')

--QUERY SHOWING NEW RECORD ASSIGNED to airplaneId 6
SELECT *
  FROM airplane

-- QUERY SHOWING DELETION OF AIRPLANE WITH ID 6
DELETE
FROM airplane
WHERE airplaneid = 6;

-- QUERY SHOWING NAME UPDATE FROM TO OLADEJI TEMI
SELECT *
```

	airplaneId	companyName	numser	manufacturer	modelNumber
1	1	United Airlines	34308	BOEING	787
2	2	Alaska Airlines	581	Airbus	A350
3	3	Jet Blue	21863	BOEING	767
4	4	Frontier Airlines	55004	Airbus	A220
5	5	Envoy Air	40	Airbus	A330
6	6	United Airlines	55508	BOEING	900

Fig 15: Pre-deletion in Microsoft SQL Server

Fig 15 shows 6 records on the airplane table before it was deleted



```
  INSERT INTO airplane
  VALUES('United Airlines', '55508', 'BOEING', '900')

  --QUERY SHOWING NEW RECORD ASSIGNED to airplaneId 6
  SELECT *
    FROM airplane

  -- QUERY SHOWING DELETION OF AIRPLANE WITH ID 6
  DELETE
 FROM airplane
 WHERE airplaneid = 6;

 -- QUERY SHOWING NAME UPDATE FROM TO OLADEJI TEMI
SELECT *
```

	airplaneId	companyName	numser	manufacturer	modelNumber
1	1	United Airlines	34308	BOEING	787
2	2	Alaska Airlines	581	Airbus	A350
3	3	Jet Blue	21863	BOEING	767
4	4	Frontier Airlines	55004	Airbus	A220
5	5	Envoy Air	40	Airbus	A330

Fig 16: Post-deletion in Microsoft SQL Server

4.0.JUSTIFICATION FOR SELECTION OF MYSQL AND MICROSOFT SERVER

Both RDBMS provide high quality performance and are two of the most widely used dbms

Below are some of their individual advantages and disadvantages.

4.1.MySQL

Advantages of MySQL

- Atop of its advantages is that MySQL is free to use. It is an open-source system with a vibrant community of developers and users, which ensures that there are always constant updates to the system and support from other users.
- It is also one of the most secure databases and is used by top companies such as WordPress, Facebook and Twitter.
- It has an intuitive GUI with self-managing features that automate a lot of features such as database design, administration and configuration.
- It is designed to meet the requirements of demanding applications while offering optimized speed performance and unique memory cache.
- It provides extensive support for transactional features such as isolated, complete atomic, consistent, and multi-version transaction support

Disadvantages

- In comparison to a database system like Microsoft Server, MySQL does not handle transactions as efficiently.
- It is not the most efficient solution for handling extremely large databases, especially compared to Microsoft Server.
- Despite the support for its development, it still has a few stability issues.
- It is not the most scalable solution.

4.2.Microsoft Server

Microsoft Server Advantages

Advantages

- Compared to other databases like MySQL, it is very efficient with managing transactions.
- It has a complex encryption algorithm to protect data.
- It runs queries much faster than MySQL
- It has strong data recovery mechanism.
- It comes in various editions which gives users the flexibility to choose based on their needs and budget.
- It has a very comprehensive documentation and reliable support filled with highly skilled professionals, including a very vibrant and constantly growing community.
- It provides various options with the possibility for offloading the administration of the database to the cloud with Azure SQL database or to a virtual machine with Microsoft Server.

Disadvantages

- As opposed to MySQL which is open source and free, the major drawback is the cost of licensing. The Enterprise version is very expensive and has a subscription- based fee.
- It is quite difficult to migrate to another database

5.0.CONCLUSION

The database development employed both MySQL and Microsoft SQL Server to design a database system for an airline booking system. The database allows user to query information regarding the passengers, pilots, flights and booking. It is designed in an efficient way that avoids duplication of attributes and wastage of storage. More detailed information and relationships can be easily accessed by writing join statements.

The report highlights the importance of adopting a systematic process to developing a database by first modelling the database before developing it. By using an ER data model to model the solution; adopting a procedural process by going through the conceptual, logical and physical model; and using normalization principles, it was possible to highlight the initial flaws and inefficiencies in the design that would have made for a complex and expensive database.

This database is therefore presented in the third normal form.

Further optimizations to the database would be implemented in the second phase of the project.

REFERENCES

Branson, T. 2016, 8 major advantages of using MySQL [Online] Available from: https://www.datamation.com/storage/8-major-advantages-of-using-mysql/ [Accessed 28th October, 2022]

Chapple, M. 2022, The Basics of Database Normalization. [Online] Available from: https://www.lifewire.com/database-normalization-basics-1019735 [Accessed 4th November, 2022]

Draw.io 2022, Entity Relationship Diagrams with Draw.io [Online] Available from:

https://drawio-app.com/entity-relationship-diagrams-with-draw-io/#:~:text=Create%20an%20entity%20relationship%20diagram,from%20the%20General%20shape%20library [Accessed 26th October, 2022]

JavatPoint 2022, Normalization [Online] Available from: https://www.javatpoint.com/dbms-normalization [Accessed 3rd November, 2022]

Lucidchart 2022, How to draw an ER Diagram. [Online] Available from: view-source: https://www.lucidchart.com/pages/how-to-draw-ERD [Accessed on 25th October, 2022]

Microsoft 2021, Create entity relationship diagrams in Visio. [Online] Available from: https://support.microsoft.com/en-us/office/create-entity-relationship-diagrams-in-visio-7e44448c-9415-490b-8af1-f548f46ae90c [Accessed 26th October, 2022]

Microsoft 2021, Database Design Basics [Online] Available from https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5 [Accessed 20th October 2022]

Microsoft 2022, Description of the database normalization basics. [Online] Available from https://learn.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description [Accessed 3rd November, 2022]

Microsoft 2022, What is SQL Server Management Studio (SSMS)? [Online] Available from: https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16 [Accessed 2nd November, 2022]

Microsoft 2022, Windows Server Documentation. [Online] Available from: https://learn.microsoft.com/en-us/windows-server/ [Accessed 2nd November, 2022]

MySQL 2022, 13.1.17.5 FOREIGN KEY Constraints. [Online] Available from: https://dev.mysql.com/doc/refman/5.6/en/create-table-foreign-keys.html [Accessed 1st November, 2022]

MySQL 2022, Chapter 13 SQL Statements. Online] Available from: https://dev.mysql.com/doc/refman/5.6/en/sql-statements.html [Accessed 1st November 2022]

Oracle 2010, Guide for Developing High-Performance database applications. *An Oracle White Paper*. [Online] Available from: https://www.oracle.com/technetwork/database/performance/perf-guide-wp-final-133229.pdf [Accessed 18th October, 2022]

Pijacek, R. 2019, Microsoft SQL Server Pros and Cons. [Online] Available from: https://learnsql.com/blog/microsoft-sql-server-pros-and-cons/ [Accessed 27th October 2022]

APPENDIX

APPENDIX A: SQL SOURCE CODE

Source Code for MySQL Implementation

MySQL Database Creation HERE

MySQL Data Insertion HERE

MySQL Database Query HERE

Source code for Microsoft Server Implementation

Microsoft SQL Server Database Creation HERE

Microsoft SQL Server Data Insertion HERE

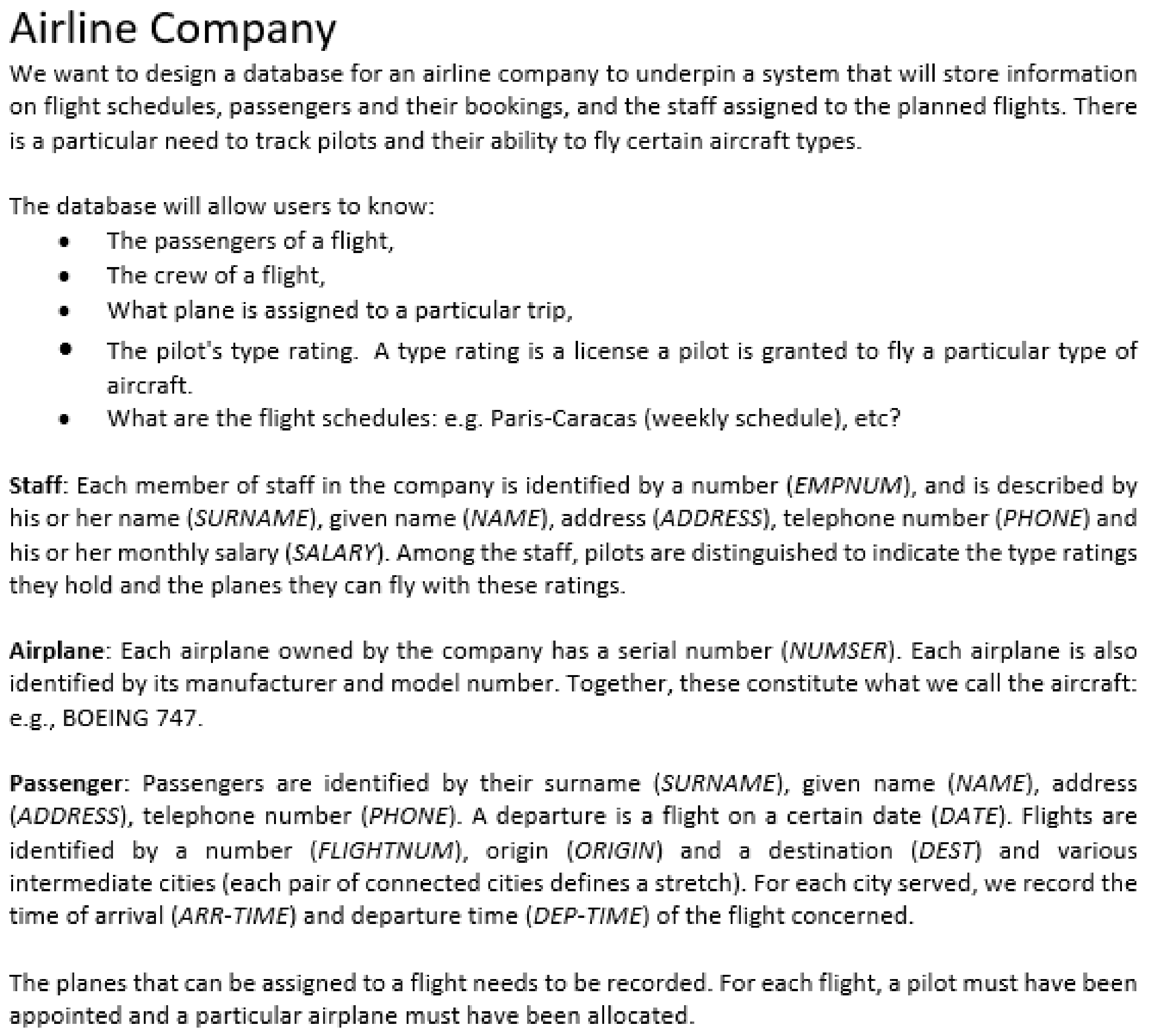
Microsoft SQL Server Database Query HERE

Bundles

MySQL and Microsoft Server Scripts Bundle HERE

Higher Resolution Images Bundle HERE

APPENDIX B: REQUIREMENTS



Airline Company

We want to design a database for an airline company to underpin a system that will store information on flight schedules, passengers and their bookings, and the staff assigned to the planned flights. There is a particular need to track pilots and their ability to fly certain aircraft types.

The database will allow users to know:

- The passengers of a flight,
- The crew of a flight,
- What plane is assigned to a particular trip,
- The pilot's type rating. A type rating is a license a pilot is granted to fly a particular type of aircraft.
- What are the flight schedules: e.g. Paris-Caracas (weekly schedule), etc?

Staff: Each member of staff in the company is identified by a number (*EMPNUM*), and is described by his or her name (*SURNAME*), given name (*NAME*), address (*ADDRESS*), telephone number (*PHONE*) and his or her monthly salary (*SALARY*). Among the staff, pilots are distinguished to indicate the type ratings they hold and the planes they can fly with these ratings.

Airplane: Each airplane owned by the company has a serial number (*NUMSER*). Each airplane is also identified by its manufacturer and model number. Together, these constitute what we call the aircraft: e.g., BOEING 747.

Passenger: Passengers are identified by their surname (*SURNAME*), given name (*NAME*), address (*ADDRESS*), telephone number (*PHONE*). A departure is a flight on a certain date (*DATE*). Flights are identified by a number (*FLIGHTNUM*), origin (*ORIGIN*) and a destination (*DEST*) and various intermediate cities (each pair of connected cities defines a stretch). For each city served, we record the time of arrival (*ARR-TIME*) and departure time (*DEP-TIME*) of the flight concerned.

The planes that can be assigned to a flight needs to be recorded. For each flight, a pilot must have been appointed and a particular airplane must have been allocated.