Basics of database systems

Project – Database design

Daniel Tuukkanen

Sami Anttalainen

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2024

TABLE OF CONTENTS

1 DEFINITION

In project ‘Jet database’ database is developed to analyze and make comparisons between different jet aircrafts that have been or are still in service. The database mainly contains information about jet models and their manufacturers, but there is also information about every major air force country and their air forces and pilots. It is important to be able to find the maximum speed of each aircraft and the manufacturer of each aircraft. It is also important to be able to find newer aircraft by sorting them by year of introduction / first flight.

The following database queries need to be implemented: (1) List the information about a specific aircrafts model, introduction year, max speed, and armament. (2) List all jets that can fly above the speed of sound. (3) List jet manufacturer, model name and headquarter location of the manufacturer. (4) List pilots, their air force and the country of the air force. (5) List aircrafts that are faster than double the speed of sound or were introduced after the 1980s. (6) List all aircrafts and the air force they belong to.

2 MODELING

2.1 Conceptual model

Figure 1. shows the ER model of the Jet Model Database. There are six entities and five relationships in the ER model. The entities are Country, AirForce, Pilot, JetModel, MissionType and Manufacturer. The relationships are ProducedBy, Missions, OperatedBy, AssignedTo and BelongsTo. OperatedBy and Missions are N:M relationships and BelongsTo is a 1:1 relationship and the others are 1: N relationships. JetModel is related to Manufacturer entity by ProducedBy relationship and to MissionType by Missions relationship. AirForce is related to Country through a BelongsTo relationship and to Pilot through an AssignedTo relationship. Both JetModel and AirForce are linked to each other by an OperatedBy relationship. All primary keys are underlined.

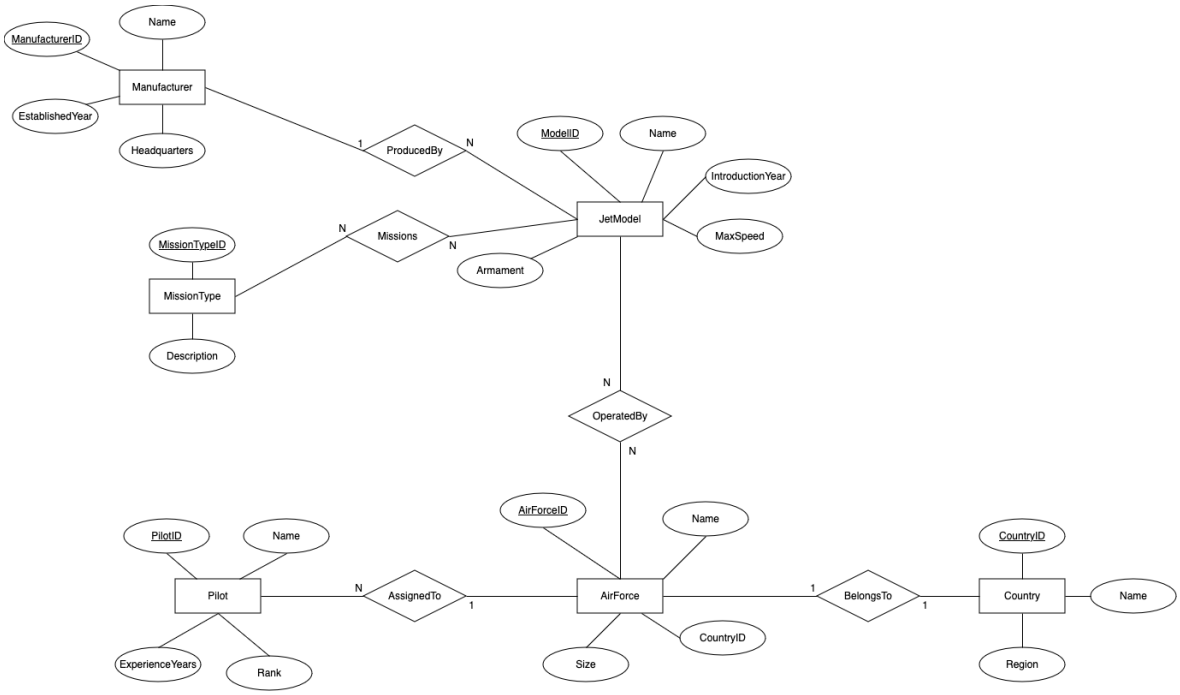

Figure 1: ER model

2.2 Logical model

Figure 2. shows the logical model of the Jet Model Database based on the ER model. The left column shows the keys and data types, and the right column shows the names of the attributes. The datatype INT means integer and VARCHAR (100) means string, which has 100 characters reserved for it. Integrity constraints are not shown on the logical model, but basically every primary key has a NOT NULL constraint and Country also has a UNIQUE constraint, so that there cannot be multiple countries with the same name and Manufacturer

has a CHECK constraint which checks that aircrafts have been introduced after 1890, because the first aircraft was invented in the 20th century. MissionType also has a DEFAULT constraint. Missions and OperatedBy relationships have standard integrity constraints. The logical model should represent the ER model as closely as possible, so no modifications have been made to make it work.

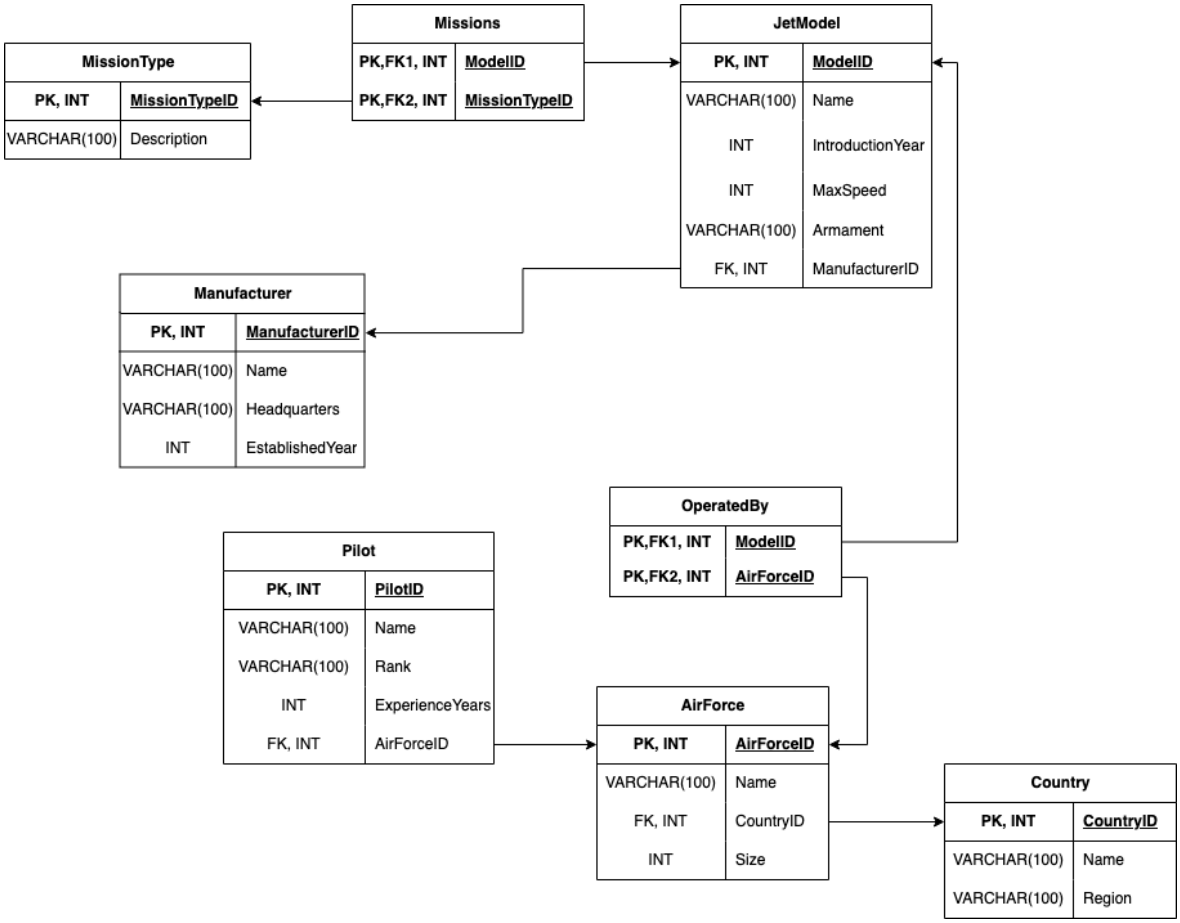

Figure 2: Logical model from the conceptual model

3 DATABASE IMPLEMENTATION

The database contains the following integrity constraints:

- **Country**
  - CountryID, Name and Region cannot be null (NOT NULL)
  - Name must be unique (UNIQUE)
- **Manufacturer**
  - ManufacturerID, Name, Headquarters cannot be null (NOT NULL)
  - Established year hast to be at least 1890 (CHECK)
- **Airforce**
  - AirForceID and Name cannot be null (NOT NULL)
  - CountryID is a foreign key and must be updated or deleted whenever changes occur (ON DELETE, ON UPDATE CASCADE)
- **Pilot**
  - PilotID, Name, Rank cannot be null (NOT NULL)
  - AirforceID is a foreign key and must be updated or deleted whenever changes occur (ON DELETE, ON UPDATE CASCADE)
- **JetModel**
  - ModelID and Name cannot be null (NOT NULL)
  - AirForceID is a foreign key, and must be updated or deleted whenever changes occur (ON DELETE, ON UPDATE CASCADE)
- **MissionType**
  - MissionTypeID cannot be null (NOT NULL)
  - Description of the mission has default string (DEFAULT)
- **Missions**
  - ModelID, MissionTypeID cannot be null (NOT NULL)
  - ModelID is a foreign key and must be updated or deleted whenever changes occur (ON DELETE/UPDATE CASCADE)
  - MissionTypeID is a foreign key and must be updated or deleted whenever changes occur (ON DELETE/UPDATE CASCADE)
- **OperatedBy**
  - ModelID and AirForceID (NOT NULL)

- ModelID and AirForceID combines into primary key.
- ModelID and AirForceIDs are both foreign keys (ON DELETE CASCADE, ON UPDATE CASCADE)

The database implements an index and a Python interface in addition to the integrity constraints.

The JetModelName index, which allows for easier jet name searching and improves the performance of the database.

The Python interface consists of a menu where there are options to either print, insert, delete or update data. There is also an option for running queries that are predetermined. Each of these options has a sub-menu where the user is asked to insert their option of what data they want to be displayed.

4 DISCUSSION

Points

Submitted files	**3p**
Report is readable and sensible	**2p**
Report contains all necessary parts (some parts are separately checked)	**2p**
Report has an ER model (conceptual model)	**2p**
ER model has at least one N:M relationship	**2p**
ER model has max & min cardinalities	**2p**
Number of entities in the ER model (6)	**6p**
Report has a relational model / database diagram / schema (logical model)	**3p**
The previous model defines data types and is as large as the ER model	**2p**
Number of select queries / views (6)	**5p**
Queries are somewhat different, and they work	**4p**
Most queries/views provide useful information	**2p**
At least one query uses two JOIN clauses	**2p**
Each table has at least five rows of data	**1p**
Database can be created with the given SQL commands given in the separate files (copy paste or read)	**2p**
There is at least one of each type of integrity constraints (NOT NULL, UNIQUE, CHECK, DEFAULT)	**4p**
Each FK has ON UPDATE and ON DELETE constraints	**4p**

Reasonable indices used (not just using primary keys)	**1p**
DONE IN PAIR	**-14p**
Python file included	**1p**
Python program runs without large issues	**2p**
Python program uses the database	**1,5p**
Python program runs in a loop	**1,5p**
Python program showcases the queries/views given in the report	**5p**
Python frontend allows insert, update, and delete	**3p**
TOTAL POINTS	**50p**