4005CEM- Database Systems

International Space Station Report

By Ayokunle Toye Ikusemori

Table of Contents

Contents

Table of Figures

Acronyms

ISS – International Space Station

DBMS – Database Management System

RDBMS – Relational Database Management System

UNF – Unnormalized Form

1NF- First Normal Form

2NF – Second Normal Form

3NF- Third Normal Form

ERD – Entity Relationship Diagram

SQL – Structured Query Language

Scenario

"The International Space Station (ISS) is a habitable artificial satellite in low Earth orbit. It is the ninth space station to be inhabited by crews following previous orbital stations that were launched by the US the former Soviet Union and later Russia. The ISS is intended to be a laboratory, observatory and factory in space as well as to provide transportation, maintenance, and act as a staging base for possible future missions to the Moon, Mars and beyond. In order to support the crew and overall operation of ISS the space agencies in charge of running the station conduct regular missions to launch spacecraft carrying payloads of essential or replacement equipment up to ISS. A payload inventory (Table 1) is recorded of each mission, consisting of the space agency leading the mission and the equipment payload to be sent up to ISS. The overall weight of the payload is also determined in order to calculate the fuel needed for orbital insertion of the spacecraft to successfully rendezvous with ISS"

From this scenario we can assume that the table being used will be constantly updated from the fact that the payload inventory is recorded for every mission. We can also assume that the database will eventually hold large amounts of data since there is no mention of missions being deleted from the table.

Table – Payload mission data

Mission No.	Agcy No.	Lead Agency	Country	Mission Date	Equipment	Qty	Item Weight	Total Weight
ISS-2237	178	JAXA	Japan	14/12/2013	Potable water dispenser	2	100kg	211kg
					Flexible air duct	6	0.5kg	
					Small storage rack	4	2kg	
ISS-3664	526	ESA	EU	16/01/2014	Bio filter	6	0.2kg	1.20kg
ISS-2356	167	NASA	USA	12/02/2014	Small storage rack	3	2kg	69kg
					Battery Pack	2	5kg	
					Urine transfer tubing	2	1.5kg	
					02 scrubber	1	50kg	
ISS-1234	32	Roskosmos	Russia	16/04/2014	Small storage rack	1	2kg	2.5kg
					Flexible air duct	2	0.5kg	

Introduction

In order to conclude on which will be suitable for the scenario presented, the benefits and drawbacks of different approaches to storing data need to be identified. The approaches to be looked at are the DBMS approach and the RDBMS approach.

DBMS Approach

Databases are a "complex object" containing "interrelated stored data" which the DBMS approach utilises while using "generalized software system[s]" (Toby J. Teorey, 2005-10-18, p. 2) in order to create databases and manipulate their data as files. This is usually adopted when a large amount of data needs to be managed or is expected to be in a database due to the addition of records. DBMS' allow for data to be searched efficiently using simple queries, some security, and requires very little to run as software and on hardware.

Since DBMS' work best when handling big amounts of simple data, they become unnecessary with embedded systems since they have low amounts of storage and cannot store large databases. Most DBMS' do not allow for multiple user access which could be detrimental when trying to modify a large part of the file at a time and do not support normalisation which could lead to data anomalies.

RDBMS Approach

The RDBMS approach is similar to the DBMS with the multitude of benefits it provides such as queries and the ability to let multiple users access data at one time. However, RDBMS' add more complexity and as a result provides additional benefits. RDBMS' are "the most widely used model for database design" (Elvis C. Foster, 2014-12-18, p. 31) and presents data using a structured format of tables and rows that are related to each other which allows queries to be run across multiple tables at once, making the search for specific data easier. It uses the relational model to show relationships between tables which allows for the data to be normalised ensuring consistency across the entire database. RDBMS' also utilise the ACID concept to ensure integrity of the data in the database. Another major benefit of the RDBMS approach is that it supports multiple user access which makes the modification of a large database particularly easier.

A problem with the RDBMS approach is that it requires a lot from software and hardware reducing its availability and accessibility. Due to its complexity, another issue with the RDBMS approach is that the larger the amount of data in a RDBMS system gets, the more precise and organised the database needs to become due to how the data is interrelated.

Should the data be kept the way it is?

If we look at the data in Table 1 we can see it can be easily implemented into a DBMS as a file due to the data being raw data that has been categorised/structured which the approach is good for. Due to this the data could not be implemented into a RDBMS as it has not been normalised nor the relationships identified. However, for the ISS it would be better to use the RDBMS as the data will need to be accessed by multiple users due to the importance of their missions and will contain a large amount of complex data which the RDBMS is better at managing compared to a DBMS.

A RDBMS should be used for this data.

Main Body

Having decided on using a RDBMS, in order to integrate the data, we need to normalise it, identify the attributes and the relationships between the tables that will be produced.

Normalisation- 1NF

The columns in Table 1 will serve as our attributes with “Mission No” being the primary key as it is the most unique field in the table

We can consider Table 1 to be in UNF. In order to achieve 1NF, all tables need to have a Primary Key, the data must be atomic (no multi value attributes) and there must not be any repeating data.

Table - Payload Mission Data (UNF)

Mission No.	Agcy No.	Lead Agency	Country	Mission Date	Equipment	Qty	Item Weight	Total Weight
ISS-2237	178	JAXA	Japan	14/12/2013	Portable water dispenser	2	100kg	211kg
					Flexible air duct	6	0.5kg	
					Small storage rack	4	2kg	
ISS-3664	526	ESA	EU	16/01/2014	Bio filter	6	0.2kg	1.20kg
ISS-2356	167	NASA	USA	12/02/2014	Small storage rack	3	2kg	69kg
					Battery Pack	2	5kg	
					Urine transfer tubing	2	1.5kg	
					02 scrubber	1	50kg	
ISS-1234	32	Roskosmos	Russia	16/04/2014	Small storage rack	1	2kg	2.5kg
					Flexible air duct	2	0.5kg	

Looking at Table 1 we can see multiple values under one column of a tuple which means the data is not atomic. To fix this we must make a separate Mission Equipment table as shown below. Calculated fields like Total weight also have to be removed.

Table - Payload Mission Data (1NF)

Mission Table

Mission No.(PK)	Agcy No.	Lead Agency	Country	Mission Date
ISS-2237	178	JAXA	Japan	14/12/2013
ISS-3664	526	ESA	EU	16/01/2014
ISS-2356	167	NASA	USA	12/02/2014
ISS-1234	32	Roskosmos	Russia	16/04/2014

Mission Equipment

Composite Key with Equipment			
Mission No. (FK)	Equipment	Qty	Item Weight
ISS-2237	Portable water dispenser	2	100kg
ISS-2237	Flexible air duct	6	0.5kg
ISS-2237	Small storage rack	4	2kg
ISS-3664	Bio filter	6	0.2kg
ISS-2356	Small storage rack	3	2kg
ISS-2356	Battery Pack	2	5kg
ISS-2356	Urine transfer tubing	2	1.5kg
ISS-2356	02 scrubber	1	50kg
ISS-1234	Small storage rack	1	2kg
ISS-1234	Flexible air duct	2	0.5kg

Normalisation – 2NF

For data to be in second normal form it must be in first normal form and no attribute is dependant on part of the primary key.

Table - Payload Mission Data(2NF)

Mission Table

Mission No.(PK)	Agcy No.	Lead Agency	Country	Mission Date
ISS-2237	178	JAXA	Japan	14/12/2013
ISS-3664	526	ESA	EU	16/01/2014
ISS-2356	167	NASA	USA	12/02/2014
ISS-1234	32	Roskosmos	Russia	16/04/2014

Mission Equipment

Composite Key with Equipment			
Mission No. (FK)	Equipment	Qty	Item Weight
ISS-2237	Portable water dispenser	2	100kg
ISS-2237	Flexible air duct	6	0.5kg
ISS-2237	Small storage rack	4	2kg
ISS-3664	Bio filter	6	0.2kg
ISS-2356	Small storage rack	3	2kg
ISS-2356	Battery Pack	2	5kg
ISS-2356	Urine transfer tubing	2	1.5kg
ISS-2356	02 scrubber	1	50kg
ISS-1234	Small storage rack	1	2kg
ISS-1234	Flexible air duct	2	0.5kg

Looking at Table 4, We can see that Equipment dependant on Mission No to form a primary key.

To fix this we need to create one with the Equipment column being the primary key since the equipment data is unique.

Table - Payload Mission Data (2NF)

Mission

Mission No. (PK)	Agcy No.	Lead Agency	Country	Mission Date
ISS-2237	178	JAXA	Japan	14/12/2013
ISS-3664	526	ESA	EU	16/01/2014
ISS-2356	167	NASA	USA	12/02/2014
ISS-1234	32	Roskosmos	Russia	16/04/2014

Composite Key

Mission No. (FK)	Equipment (FK)	Qty
ISS-2237	Portable water dispenser	2
ISS-2237	Flexible air duct	6
ISS-2237	Small storage rack	4
ISS-3664	Bio filter	6
ISS-2356	Small storage rack	3
ISS-2356	Battery Pack	2
ISS-2356	Urine transfer tubing	2
ISS-2356	02 scrubber	1
ISS-1234	Small storage rack	1
ISS-1234	Flexible air duct	2

Equipment (PK)	Item Weight
Portable water dispenser	100kg
Flexible air duct	0.5kg
Small storage rack	2kg
Bio filter	0.2kg
Battery Pack	5kg
Urine transfer tubing	1.5kg
02 scrubber	50kg
Flexible air duct	0.5kg

Equipment Table

Mission Equipment

Normalisation - 3NF

For data to be in third normal form, all attributes are solely dependent on the primary key.

Table - Payload Mission Data (2NF)

Mission

Mission No. (PK)	Agcy No.	Lead Agency	Country	Mission Date
ISS-2237	178	JAXA	Japan	14/12/2013
ISS-3664	526	ESA	EU	16/01/2014
ISS-2356	167	NASA	USA	12/02/2014
ISS-1234	32	Roskosmos	Russia	16/04/2014

Composite Key

Mission No. (FK)	Equipment (FK)	Qty
ISS-2237	Portable water dispenser	2
ISS-2237	Flexible air duct	6
ISS-2237	Small storage rack	4
ISS-3664	Bio filter	6
ISS-2356	Small storage rack	3
ISS-2356	Battery Pack	2
ISS-2356	Urine transfer tubing	2
ISS-2356	02 scrubber	1
ISS-1234	Small storage rack	1
ISS-1234	Flexible air duct	2

Mission Equipment

Equipment (PK)	Item Weight
Portable water dispenser	100kg
Flexible air duct	0.5kg
Small storage rack	2kg
Bio filter	0.2kg
Battery Pack	5kg
Urine transfer tubing	1.5kg
02 scrubber	50kg
Flexible air duct	0.5kg

Equipment Table

Looking at Table 6, we can see that Lead Agency and Country are not solely dependent on the primary key but also Agcy No. To fix this we need to create an Agency table as shown below

Table - Payload Mission Data (3NF)

Mission

Mission No.(PK)	Agcy No. (FK)	Mission Date
ISS-2237	178	14/12/2013
ISS-3664	526	16/01/2014
ISS-2356	167	12/02/2014
ISS-1234	32	16/04/2014

Agency

Agcy No. (PK)	Lead Agency	Country
178	JAXA	Japan
526	ESA	EU
167	NASA	USA
32	Roskosmos	Russia

Composite Key

Mission No. (FK)	Equipment (FK)	Qty
ISS-2237	Portable water dispenser	2
ISS-2237	Flexible air duct	6
ISS-2237	Small storage rack	4
ISS-3664	Bio filter	6
ISS-2356	Small storage rack	3
ISS-2356	Battery Pack	2
ISS-2356	Urine transfer tubing	2
ISS-2356	02 scrubber	1
ISS-1234	Small storage rack	1
ISS-1234	Flexible air duct	2

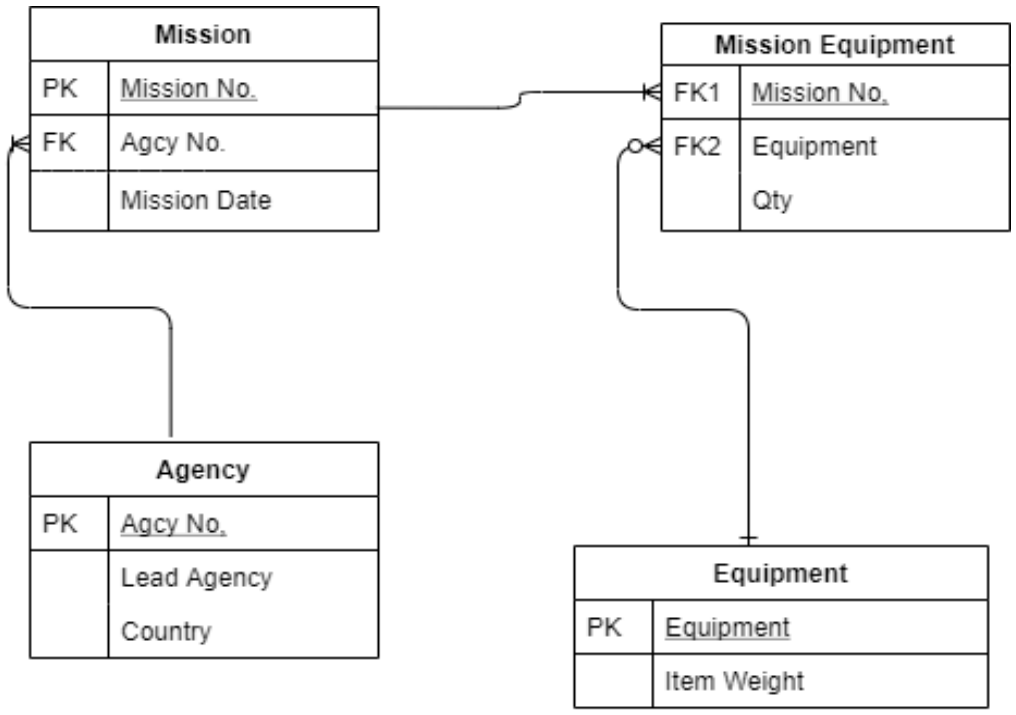
Equipment (PK)	Item Weight
Portable water dispens	100kg
Flexible air duct	0.5kg
Small storage rack	2kg
Bio filter	0.2kg
Battery Pack	5kg
Urine transfer tubing	1.5kg
02 scrubber	50kg
Flexible air duct	0.5kg

Equipment Table

Mission Equipment

The data is now in 3NF which allows us to form an ERD.

ERD


Mission
PK
Mission No.
FK
Agcy No.
Mission Date
Mission Equipment
FK1
Mission No.
FK2
Equipment
Qty
Agency
PK
Agcy No.
Lead Agency
Country
Equipment
PK
Equipment
Item Weight

Creating and Populating the database using SQL

In order to create and manage data on RDBMS' we need to use SQL. Before creating a database however, we need to determine the data type of our attributes and their lengths.

Agency Table Attributes Type and length

Agcy No. (INTEGER (3 – based on length in Table 7), PRIMARY KEY)

Lead Agency (TEXT (9 – based on the longest amount of characters in Table 7))

Country (TEXT (48 – Longest country name))

Creation of Agency Table using SQL

```
CREATE TABLE "Agency" (
        "Agcy No."      INTEGER(3) NOT NULL,
        "Lead Agency"   TEXT(9) NOT NULL,
        "Country"       TEXT(48) NOT NULL,
        PRIMARY KEY ("Agcy No.")
);
```

Equipment attribute type and length

Equipment (TEXT – (Maximum number), PRIMARY KEY)

Item Weight (KG)- (REAL (5,2 – to accommodate for decimals as seen in Table 7)

Creation of Equipment Table using SQL

```
CREATE TABLE "Equipment_Table" (
        "Equipment"     TEXT NOT NULL,
        "Item Weight(KG)"       REAL(5,2) NOT NULL,
        PRIMARY KEY("Equipment")
);
```

Mission Table attribute type and length

Mission No. (TEXT (8), PRIMARY KEY)

Agcy No. (INTEGER (3-based on the length in Table 7), FOREIGN KEY)

Mission Date (DATE)

Creation of Mission Table using SQL

```
CREATE TABLE "Mission" (
        "Mission No."    TEXT(8),
        "Agcy No."       INTEGER(3) NOT NULL,
        "Mission Date"   DATE NOT NULL,
        FOREIGN KEY("Agcy No.") REFERENCES "Agency"("Agcy No."),
        PRIMARY KEY("Mission No.")
);
```

Mission Equipment attribute type and length

Mission No. (TEXT (8), PRIMARY KEY,FOREIGN KEY)

Equipment (TEXT – (Up to Max number), PRIMARY KEY, FOREIGN KEY)

Qty (INTEGER(Up to Max number)

Creation of Mission Equipment Table using SQL

```
CREATE TABLE "Mission Equipment" (
        "Mission No."    INTEGER(3) NOT NULL,
        "Equipment"      TEXT NOT NULL,
        "Qty"    INTEGER NOT NULL,
        PRIMARY KEY("Mission No.","Equipment"),
        FOREIGN KEY("Mission No.") REFERENCES "Mission Equipment"("Mission No."),
        FOREIGN KEY("Equipment") REFERENCES "Equipment_Table"("Equipment")
);
```

Populating the tables using SQL

The way in which you add a record into a table is the across all the table apart from the column names and the values. Utilising the INSERT INTO “” VALUES “” query you can add new data.

An example will be shown with the Agency table below

```
INSERT INTO "Agency"
("Agcy No.", "Lead Agency", "Country")
VALUES (178, 'JAXA', 'JAPAN');
```

Reflection

This report has highlighted the benefits and drawbacks of storing data with a DBMS or a RDBMS. The process in which the data can be normalised and can be optimally integrated into a RDBMS, the formation of the ERD using the data provided and the creation and implementation of data into a RDBMS using SQL.

This process is important as it prevents multiple issues such as a multitude of anomalies like insertion and update anomalies. It makes large amounts of data less cluttered which in turn makes it easier to present and manage. Implementing data into a RDBMS through this process is necessary as you are unable to do certain steps without doing others. For example, you cannot create a efficient table using SQL if you have not identified the relationships by getting the data into 3NF

Conclusion

In conclusion, considering the benefits and the drawbacks of an RDBMS, the ISS would greatly benefit from using an RDBMS to store their data as it would effectively and efficiently manage the potential large amounts of data that they would provide. Keeping the data in the format it was being stored in before would be dangerous and could cause a lot of major security problems as users would be able to access data that they would not be supposed to

References

Elvis C. Foster, S. V. G., 2014-12-18. *Database Systems:The Relational Model.* Shrewsbury,USA: Apress.

Manning, A., 2015. *Databases For Small Business.* Berkeley, CA: Apress.

Toby J. Teorey, ,. S. L. ,. T. N. ,. a. H. V. J., 2005-10-18. *Database Modeling and Design : Logical Design.* s.l.:Elsevier Science & Technology.