

Information Engineering

NORMALISATION

B HOLLAND



What is it?

- Normalisation is a database design technique, which begins to examine the relationships (called functional dependencies) between attributes)
- Normalisation is a formal process based on mathematics which defines a set of rules for creating structures that are arranged in the best way to enable the storage and retrieval of data.
- The normal forms are based on the requirement to model data in two dimensional structures consisting of a variable number of rows and a variable number of columns
- The process was defined by DR E.F. Codd in the early 70's
- Attributes describe some aspect of the data or of the relationships between the data that is important to the organisation.
- Normalisation uses a series of tests (described as normal forms to help identify a set of suitable relationships that best supports the data requirements of the organisation

The Process and the Purpose of :

- ⦿ To put data into a form that conforms to relational principles. E.g single valued columns each relation representing one entity.
- ⦿ To avoid redundancy by storing each fact within the database only once.
- ⦿ To put data into a form that is more able to accommodate change.
- ⦿ To avoid certain difficulties in updating
- ⦿ To facilitate the enforcement of constraints on the data.

Normalisation and E-R modelling

- The two techniques can be used together to produce a more correct model of the data.
- You need to have a fairly complete list of all the data items that need to be held in the database before you can begin to normalise.
- In a large system this could prove quite complicated and it would be better to organise the data into preliminary groups - i.e. entities.
- At the end of normalisation you may need to revise your E-R model to show the normalised entities.

Data redundancy

Transforming data from a problem into relations while ensuring data integrity and eliminating data redundancy.

- Data integrity : consistent and satisfies data constraint rules
- Data redundancy: if data can be found in two places in a single database (direct redundancy) or calculated using data from different parts of the database (indirect redundancy) then redundancy exists.

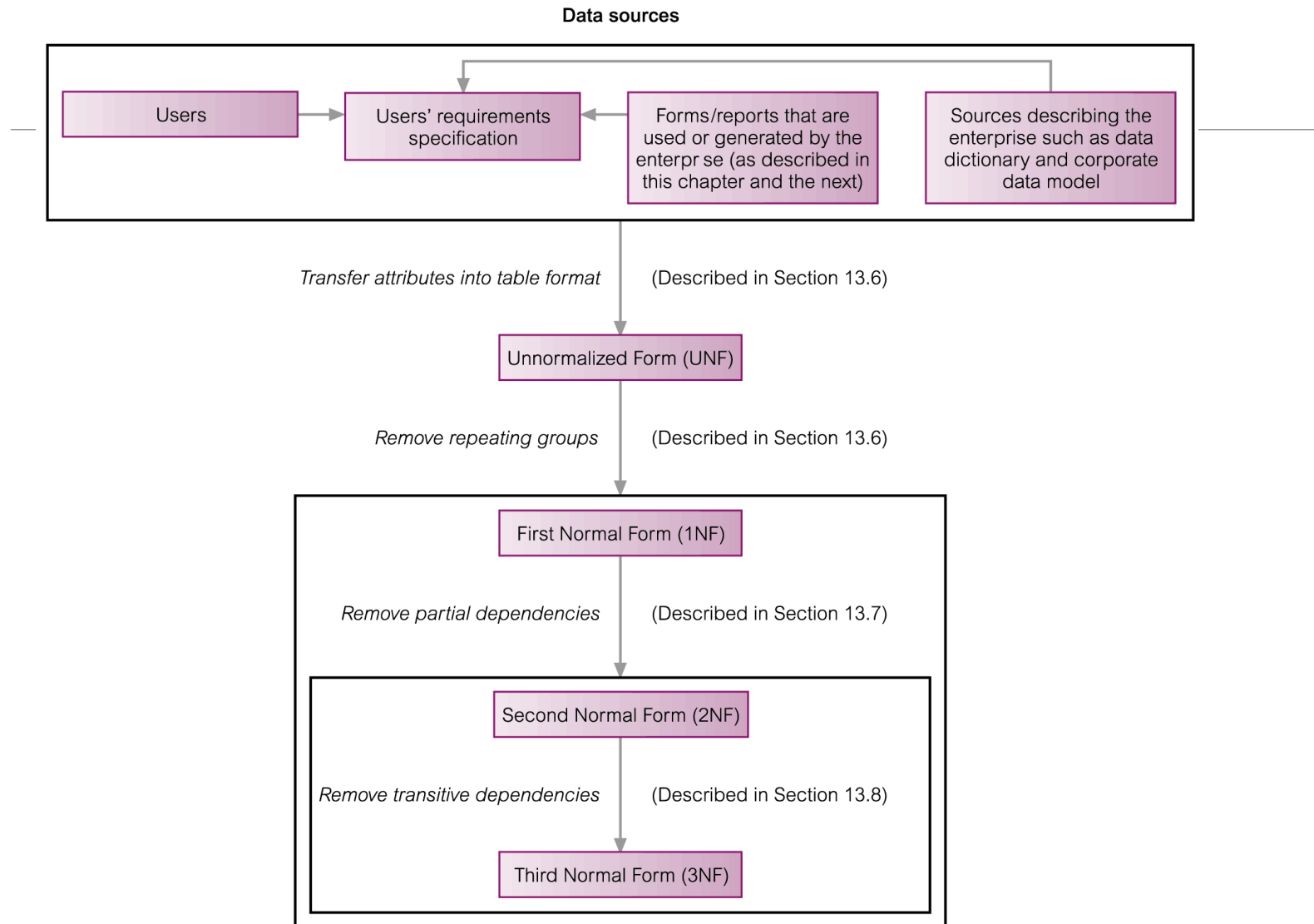
Normalisation should remove redundancy, but not at the expense of data integrity.

Problems of redundancy

If redundancy exists then this can cause problems during normal database operations:

- When data is inserted the database the data must be duplicated where ever redundant versions of that data exists.
- When data is updated, all redundant data must be simultaneously updated to reflect that change.

The Process of Normalisation



First Normal Form

- Un-normalised form (UNF) describes a table that contains one or more repeating groups.
- An entity is said to be in First Normal Form if all the attribute values are atomic. (A relationship in which the intersections of each row and column contains one and only one value)
- Each column of the table should contain only one values not multiples as in a list.
- 1NF – the aim is to remove all of the repeated occurrences of attributes in the logical design, and therefore ensure that the final physical database will be free of redundant data.

UNF to 1NF

Nominate an attribute or group of attributes to act as the key for the un-normalised table.

Identify the repeating group(s) in the un normalised table which repeats for the key attribute(s).

UNF to 1NF

Remove the repeating group by

- Entering appropriate data into the empty columns of rows containing the repeating data ('flattening' the table).
- Or by
- Placing the repeating data along with a copy of the original key attribute(s) into a separate relation.

Convert to 1NF

Student_ number	Student name	Student address	Cours e-id	Course type	Course _title	Course duration	Course mode	Grade
001	A Bell	London	A01	HND	Computi ng	120	Full Time	MERIT
001	A Bell	London	B02	RSA	French	20	Part time	PASS
002	B Careful	Croydon	C01	HNC	Business studies	80	Part time	DISTIC -TION
002	B Careful	Croydon	B01	RSA	Italian	20	Part time	MERIT
003	G upshot	Ealing	A02	HND	Business	120	Full time	PASS

INF

Examination of the data in the above table reveals that attributes Student number, Student name and Students Address are repeated unnecessarily.

To achieve 1NF, these attributes should be placed in a separate table called student table. The remaining attributes will be placed in a course table

1NF

Student Table

Student number	Student name	Student address
001	A Bell	London
002	B Careful	Croydon
003	G Upshot	Ealing

Course Table

Course_id	Course_type	Course_title	Course_duration	Course_mode	Grade
A01	HND	Computing	120	Full Time	MERIT
B02	RSA	French	20	Part time	PASS
C01	HNC	Business studies	80	Part time	DISTINCTION
B01	RSA	Italian	20	Part time	MERIT
A02	HND	Business	120	Full time	PASS

Second Normal Form

- An entity is in Second Normal Form if it is in First Normal Form and if every non-key attribute is functionally dependent on a unique identifier.
- To demonstrate – consider the student table. Each student will have one student number, but it is possible for more than one student to have the same name and, possibly the same address.
- Therefore the attribute student number can be used to identify the name and address they are dependent on student number.

Second Normal Form (2NF)

Based on the concept of full functional dependency.

Full functional dependency indicates that if

- A and B are attributes of a relation,
- B is fully dependent on A if B is functionally dependent on A but not on any proper subset of A.

2NF

In order to convert to 2NF, any attribute which is not dependent on a key attribute must be removed from the table and placed in another table with the key attribute on which it is dependent.

The student table is already in 2NF

In the course table the key attribute is Course_id . Each non-key attribute is then checked to determine if it is dependent upon Course_id. This process reveals the following attributes as dependent

2NF

- ⦿ *Course-Type* because each course can only be of one type, but there may be many courses of that type. For example, there are many different HND courses.
- ⦿ *Course_title* because each course can only have one title, but there may be many courses with the same title. For example there may be many courses with the title Business studies
- ⦿ *Course_duration* because each course has only one duration but there may be courses which last two years, such as HND and A level.
- ⦿ Grade is not dependent on course_id as a grade is awarded to a student.

Normalisation into 2NF

Course Table

Course_id	Course_type	Course_title	Course_duration	Course_mode
A01	HND	Computing	120	Full Time
B02	RSA	French	20	Part time
C01	HNC	Business studies	80	Part time
B01	RSA	Italian	20	Part time
A02	HND	Business	120	Full time

Class Table

Course_id	Student_number	Grade
A01	001	MERIT
B02	001	PASS
C01	002	DISTINCTION
B01	002	MERIT
A02	003	PASS

Third Normal Form

- An entity is said to be in Third Normal Form if it is in 2NF, and if all non-key attributes are mutually independent.
- Any non key attributes which is dependent on any other non key attribute should be removed and put into a new table.
- We only need to consider the course table
- In the course table, the attributes Course_duration and Course_mode are dependent upon non- key attribute Course type. This is because all HND courses are full-time and 120 hours all HNC courses are part time and 80 hours etc.

2NF to 3NF

- Identify the primary key in the 2NF relation.
- Identify functional dependencies in the relation.
- If transitive dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their dominant.

General Definitions of 2NF and 3NF

Second normal form (2NF)

- A relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on *any candidate key*.

Third normal form (3NF)

- A relation that is in first and second normal form and in which no non-primary-key attribute is transitively dependent on *any candidate key*.

3NF

Course Table

Course_id	Course_type
A01	HND
B02	RSA
C01	HNC
B01	RSA
A02	HND

Course Definition Table

Course_type	Course_duration	Course_mode
HND	120	Full Time
RSA	20	Part time
HNC	80	Part time

-
- Once each stage of the normalisation process is completed, the tables are said to be fully normalised.