# Normalisation

Name limit

The development of the logical data model (i.e. the tables) involves continuous *validation* of the model against the user requirements

Normalisation is the process of testing the correctness of a logical data model.

Database Concepts Normalisation 2

What does normalisation ensure?

- · No data redundancy
- thus removing the possibility of *update anomalies*

Normalisation helps to identify a suitable set of relations to represent data in the database

Database Concepts Normalisation 3

### Definition

Normalisation is a technique for producing a set of relations with desirable properties, given the data requirements of an organisation

The normalisation process was developed by E.F. Codd (1972)

Margalistica

- Informally defined as a series of tests
- Applied to one (or more) relations
- Determines if the relations satisfy or violate the requirements of a given normal form

Three normal forms were initially proposed:

- -1NF
- 2NF
- -3NF

Database Concepts Normalisation 5

A stronger definition of 3NF was subsequently defined:

Boyce-Codd Normal Form (BCNF)

This normal form examines candidate keys and determinants.

Higher normal forms were later defined (4NF, 5NF). These deal with situations which are very rare

atabase Concepts Normalisation 6

A database *schema* is said to consist of a group of relations.

A *relation* is said to consist of a set of attributes.

When the **data requirements** of an organisation are identified, how are these attributes grouped into suitable relations?

Database Concents

. . . .

This decision is made based on the following:

- the common sense of the database designer
- -by mapping ER diagrams onto relations

Some formal method is required to help the database designer to identify the optimal groupings of attributes for each relation in the schema

Database Concept

Mormalication

All normal forms are based on *functional dependencies* among the attributes of a relation

Attributes are classed as either:

- Key attributes
- Non-key attributes

Database Conce

Normalisation

Normalisation is a formal method

It identifies relations based on:

- Primary key (candidate key in BCNF)
- and the functional dependencies between their attributes

Database Conc

Normalisation

Normalisation presents a series of tests applied to *individual* relations so that a schema can be normalised to a specific form to prevent possible *update anomalies* 

Outabase Concep

Normalisation

Data Redundancy & Update Anomalies

The primary aim of normalisation is to *minimise* data redundancy thus reducing file storage space.

Remember, storage space is an expensive commodity! The less we need the better.

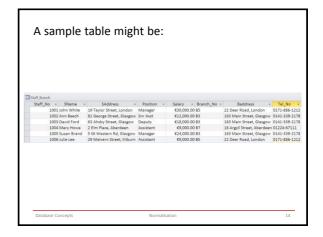
tabase Concepts

Normalisation

Consider the data requirements of an organisation are such that it needs to store details of all its branches and all staff working in those branches.

An initial representation of this data might be:

StaffBranch(StaffNo,Sname,Saddr,Position,Salary, Bno,Baddr,TelNo)



Look at the sample data in the table.

In particular we notice that some branches
(B3, B5) have more than one staff member.

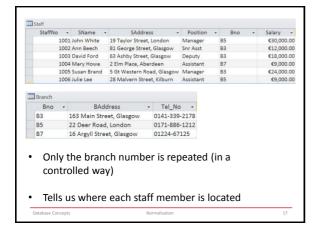
The details of each branch are *repeated* for each staff member at that branch

=> Data Redundancy

Normalisation

A better representation would be:

STAFF(Staffno,Sname,Saddr,Position,Salary,Bno)
BRANCH(Bno,Baddr,TelNo)



Redundant data leads to Update Anomalies
This has serious implications for data consistency and data integrity.

There are three categories of update anomalies:

- Insertion anomalies

- Deletion anomalies

- Modification anomalies

Normalisation 3

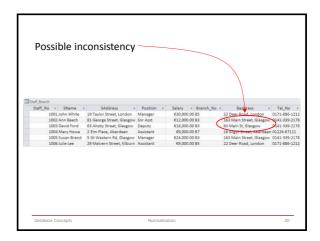
15

#### **Insertion Anomalies**

Insert a new staff member into the relation **StaffBranch**:

- Adding a record for a staff member working at branch B3 <u>must also include</u> details of branch B3
- Must be consistent with all other entries for branch B3

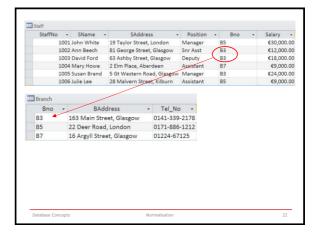
Database Concepts Normalisation 19



The individual staff and branch relations ensure that

- · branch details are recorded only once and
- thus prevents potential inconsistency

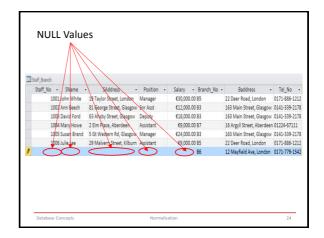
Database Concepts Normalisation 21



Insert a branch that currently has no members of staff into the StaffBranch relation

To do this you must enter NULL in the attributes of staff (since no staff details are available).

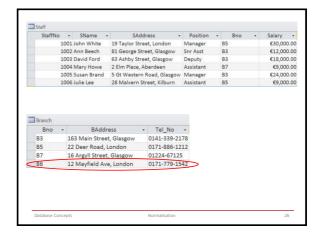
Database Concepts Normalisation 23



Since Staff\_No is the primary key, this is not allowed (Primary key may not be NULL).

The individual *Staff* and *Branch* relations avoid this problem.

Database Concepts Normalisation 25

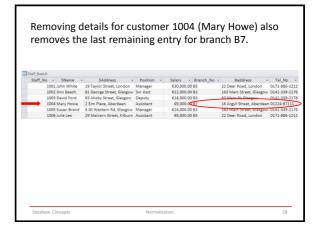


#### **Delete Anomalies**

If we remove a staff member from the StaffBranch relation we also remove information about the branch at which they work.

If the staff member happened to be the *last* member at this branch, we lose all details of that branch from the database

Database Concepts Normalisation 27



# Modification Anomalies

Change the telephone number for branch B3 in the StaffBranch relation

- We must update the rows of all staff located at branch B3
- If some of the rows are not updated, this results in inconsistent data

Database Concepts Normalisation 29

# Conclusions

The relations Staff and Branch have more desirable properties than the relation StaffBranch

We will see how normalisation can be used to derive such relations.

Refer to the PDF document *Normalisation\_Chapter* on X:\ drive.

Database Concepts Normalisation 30

# **Functional Dependencies**

Describes the relationship between attributes in a relation

If A and B are attributes of a relation R B is *functionally dependent* on A (A->B)

if each value of A in R is associated with exactly one value of B in R

Database Concepts

Normalisation

 $A \rightarrow B$ 

A is said to be the *determinant* 

B is said to be the dependent

Database Concepts

Look at the dependencies of the StaffBranch relation:

StaffNo -> Sname, Saddr, Position, Salary, Bno, Baddr, TelNo

Baddr, TelNo Bno -> Baddr -> Bno, TelNo

TelNo -> Bno, Baddr

Database Concepts

Normalisation

Un-normalised Form

A relation which contains one or more repeating groups

i.e. attribute values are non-atomic

This also means that the record length varies.

Database Concepts

33

Normalisation

### 1NF

A relation is in 1NF if

- it contains no repeating groups
- all non-key attributes are functionally dependent on the primary key

Database Concepts

Normalisation

2NF

A relation is in 2NF if

- it is in 1NF
- All non-key attributes are *fully* functionally dependent on the primary key

Database Concepts

Normalisation

B is *fully* functionally dependent on A if:

B is functionally dependent on A and not any subset of A

B is *partially* dependent on A if some attribute can be removed from A and the dependency still holds

Applies only to relations with composite keys

Database Concepts

Normalisation

3NF

A relation is in 3NF if

- it is in 2NF
- Non-key attributes are not transitively dependent on the primary key

Database Concepts

Normalisation

If A, B and C are attributes of a relation

and

A->B and B->C

then C is transitively dependent on A via B

Database Concepts

Normalisation

The StaffBranch relation contains the following dependencies:

StaffNo -> Bno

Bno -> Baddr

Clearly there is a transitive dependency

Staffno -> Baddr (via Bno)

Database Concepts

Normalisation

40

# Example

Consider a Real Estate system which maintains property rental details.

A property has an owner and is rented to a customer.

A customer can rent more than one property but not simultaneously

A customer never rents the same property twice

Database Concepts

Normalisation

Such a relation might contain the following attributes:

> CustNo, Cname PropNo, Paddr

RentStart, RentFinish, Rent

OwnerNo, Oname

Show the data as a set of normalised (3NF) relations

Database Concepts

Normalisation

42