Introduction to normalisation

Normalisation

- Normalisation = a formal process for deciding which attributes should be grouped together in a relation
- Normalisation is the process of decomposing relations with anomalies to produce smaller, well-structured relations

Steps in normalisation

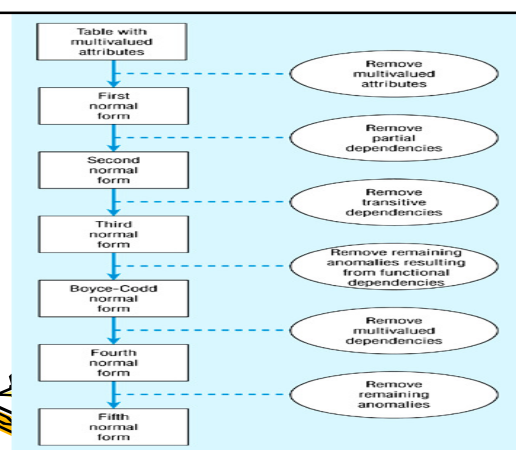
- First normal form: Any multivalued attributes (repeating groups) have been removed, so there is a single value (possibly null) at the intersection of each row and column of the table

Steps in normalisation

- Second normal form: Any partial functional dependencies have been removed
- Third normal form: Any transitive dependencies have been removed
- Boyce/Codd normal form: Any remaining anomalies that result from functional dependencies have been removed

Steps in normalisation

- Fourth normal form: Any multivalued dependencies have been removed
- Fifth normal form: Any remaining anomalies have been removed
- Usually only bother with First to third
- Following Fig shows process:



Functional Dependency

- Functional dependency describes the relationship between attributes in a relation.
- Eg. if A and B are attributes of relation R, B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R.

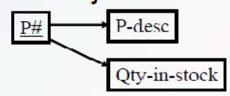
FD Diagram

Stock

P#	P-desc.	Qty-in-stock
P2	nut	5000
P1	bolt	8300
P3	washer	9750
P4	nut	2326

(P#, P-desc, Qty-in-stock)

P# ➔ {P-desc, Qty-in-stock}



Partial Dependencies

- Attribute Y of relation R is Partially functionally dependent on attribute X of relation R if it is functionally dependent on X and functionally dependent on any proper subset of X.

Full Dependencies

- Attribute Y of relation R is Fully functionally dependent on attribute X of relation R if it is functionally dependent on X and not functionally dependent on any proper subset of X

Full & Partial Dependency

Supp-Part (S#, Sname, P#, PackSize)

Full Key Dependency:

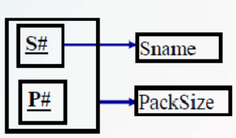
{S#,P#}➔ {Packsize}

Partial Key Dependency:

S# ➔ Sname

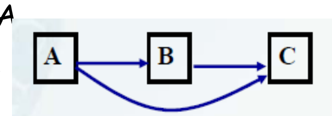
Supp-Part

S#	SNAME	P#	PACKSIZE
S2	Jones	P1	10
S7	Smith	P6	25
S2	Jones	P4	40
S5	Jones	P1	20



Transitive Dependencies

- If $A \rightarrow B$ and $B \rightarrow C$, then attribute A must be the determinant of C, Attribute A transitively determines attribute C or C is transitively dependent on A



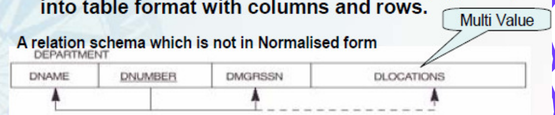
Relationship between determinants and candidate keys

- A candidate key is always a determinant, whilst a determinant may or may not be a candidate key
- A candidate key is a determinant that uniquely identifies the remaining (nonkey) attributes in a relation
- A determinant may be a candidate key, part of a composite candidate key or a nonkey attribute

Unnormalised Form (UNF)

- A table that contains one or more repeating groups.
- To create an unnormalized table:
- transform data from information source (e.g. form) into table format with columns and rows.

A relation schema which is not in Normalised form



Example of a relation instance

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

First normal form (1NF)

- Contains no multivalued attributes
- Usually, when you map ER diagrams into relations, you remove multivalued attributes from entity types on the ER diagram, so there should not be any multivalued attributes remaining
- However, many old legacy systems supported multivalued attributes, so we must understand how to get rid of them

First Normal Form (1NF)

- Each single cell in a relation must only hold a single atomic value.
- This implies that we should disallow composite attributes, multi-valued attributes, and nested relations
- In other words, forbid all attributes whose values for an individual tuple are non-atomic

The steps of transformation from UNF into 1NF

1. Nominate an attribute or group of attributes to act as the key for the unnormalized table.
2. Identify the repeating group(s) in the unnormalized table, which repeats for the key attribute(s).

The steps of transformation from UNF into 1NF

3 a. Remove the repeating group by entering appropriate data into the empty columns of tuples containing the repeating data ('flattening' the table),

1stNF relation with redundancy

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

from UNF into 1NF

3b. or by placing the repeating data along with a copy of the original key attribute (s) into a separate relation.

EMP_PROJ

SSN	ENAME	PROJS	
		PNUMBER	HOURS

EMP_PROJ1

SSN	ENAME

EMP_PROJ2

SSN	PNUMBER	HOURS

EMP_PROJ

SSN	ENAME	PNUMBER	HOURS
123456789	Smith,John B.	1	32.5
		2	7.5
666884444	Narayan,Ramesh K.	3	40.0
453453453	English,Joyce A.	1	20.0
		2	20.0
333445555	Wong,Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya,Alicia J.	30	30.0
		10	10.0
987987987	Jabbar,Ahmad V.	10	35.0
		30	5.0
987654321	Wallace,Jennifer S.	30	20.0
		20	15.0
888665555	Borg,James E.	20	null

Second Normal Form (2NF):

- a relation is in 2NF if it is in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key
- If in 1NF will be in 2NF if <u>any one</u> of the following conditions applies:

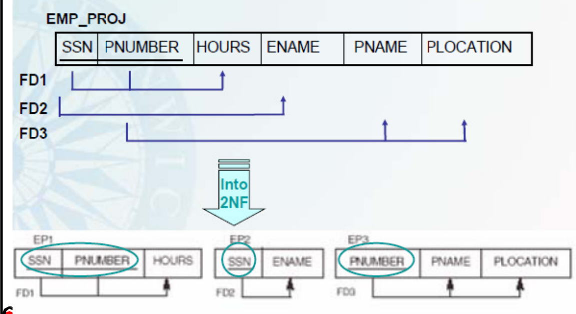
2NF

- The primary key consists of only one attribute.
- No nonkey attributes exist in the relation (thus all of the attributes in the relation are components of the primary key)
- Every nonkey attribute is functionally dependent on the full set of primary key attributes

steps of transformation from 1NF into 2NF

1. Identify the primary key for the 1NF relation.
2. Identify the functional dependencies in the relation.
3. If partial dependencies exist on the primary key remove them by placing them in a new relation along with a copy of their determinant.

Examples of transformation into 2NF



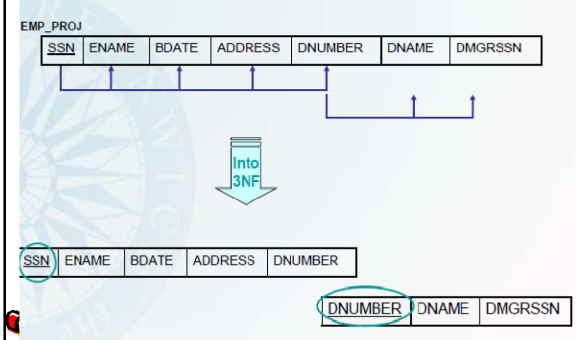
Third Normal Form (3NF)

- A relation schema R is in third normal form (3NF) if it is in 2NF and no non-prime attribute A in R is transitively dependent on the primary key.
- This means that all non-prime attributes should be fully and directly dependent on the primary key.

steps of transformation from 2NF into 3NF

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

Examples of transformation into 3NF



Boyce-Codd Normal Form (BCNF)

- A relation is in BCNF, if and only if every determinant is a candidate key.
- BCNF is a very strict 3NF. All 3NF relations that are based on a single key are in BCNF.
- Violation of BCNF may occur in a relation that contains two (or more) composite keys, which overlap and share at least one attribute in common.

steps of transformation from 3NF into BCNF

1. Identify all candidate keys in the relation.
2. Identify all functional dependencies in the relation.
3. If functional dependencies exist in the relation where their determinants are not candidate keys for the relation, remove the functional dependencies by placing them in a new relation along with a copy of their determinant.

BCNF / Example /1

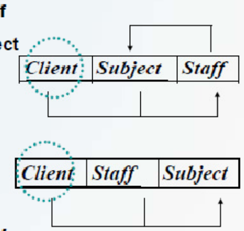
- Consider this scenario:
  - The DSD company provides end user software training in Database, Network & Spreadsheets
  - DSD employs several trainers in each of the three subject.
  - Each trainer teaches only one subject, that is a Database trainer teaches Database only.
  - Corporate customers may elect to purchase training contracts for one or more subjects.

Client	Subject	Staff
1001	Database	Ala
1001	Network	Sati
1002	Database	Ala
1003	Spreadsheet	Phil
1004	Database	Alun

BCNF / Example /2

- 2 composite candidate keys:
  - FD1:{Client, Subject} → Staff
  - FD2:{Client & Staff} → Subject
- These candidate keys are overlapping on Client.

Staff is a determinate but not a candidate key

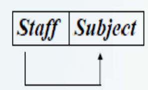


- What is the Problem?
  - Anomalies
- Delete client 1004 will also delete Tony teaches Database. So is for client 1001 on Network.
- Hence, we need to decompose table into two to get rid of redundancies.

Client	Subject	Staff
1001	Database	Ala
1001	Network	Sati
1002	Database	Ala
1003	Spreadsheet	Phil
1004	Database	Alun

Client | *Staff*

Client	Staff
1001	Ala
1001	Sati
1003	Phil
1004	Alun
1002	Ala



Staff	Subject
Ala	Database
Sati	Network
Phil	Spreadsheet
Alun	Database

Other Normal Forms

- Relations in 3NF are sufficient for most practical database applications, however 3NF does not guarantee that all anomalies have been removed

Other Normal Forms

- 4NF - No multivalued dependencies
- 5NF - very rare
- Domain-key NF - The "ultimate" NF…perfect elimination of all possible anomalies – practical utility quite limited

Questions