Chapter – 5: Relational Database Design:
Normalisation

Normalisation

Is derivation of data as a set of

Non-Redundant,

Consistent and

Inter-Dependent Relations

Normalisation

- Normalisation is a set of data design standards.
- It is a process of decomposing unsatisfactory relations into smaller relations.
- Like entity–relationship modelling were developed as part of database theory.

Normalisation - Advantages

Reduction of data redundancy within tables:

- Reduce data storage space
- Reduce inconsistency of data
- Reduce update cost
- Remove many-to-many relationship
- Improve flexibility of the system

Normalisation - Disadvantages

Reduction in efficiency of certain data retrieval as relations may be joined during retrieval.

- Increase join
- Increase use of indexes: storage (keys)
- Increase complexity of the system

Normal Forms

A state of a relation that results from applying simple rules regarding functional dependencies (or relationships between attributes) to that relation.

0NF	multi-valued attributes exists
1NF	any multi-valued attributes have been removed
2NF	any partial functional dependencies have been removed
3NF	any transitive dependencies have been removed

Functional Dependencies and Keys

Functional dependency:A constraint between two attributes or two sets of attributes

The functional dependency of B on A is represented by an arrow: **A → B**

e.g.

NID (SSN) → Name, Address, Birth date

VID → Make, Model, Colour

ISBN → Title, First Author

Functional Dependencies and Keys

Functional dependency (*definition*)

For any relation R (*e.g. book*), attribute B (*e.g. title*) is functionally dependent on attributes A (*e.g. ISBN*), if for every valid instance of A (*e.g. 981-235-996-6*), that value of A uniquely determines the value of B (*e.g. Modern Database Management*)

Input for the Normalisation Process

Database Design process (phase 1)

data requirements and data analysis

entity types *(e.g. Supplier, Order)*

attributes describing each entity type with its meaning *(e.g. supplier name and part name)*

attributes relationships to other attributes. *(e.g.supplier no of Supplier to supplier no of purchase Order)*

Purchase Order - Attribute Analysis

ATTRIBUTE	TYPE	LEN-GTH	DESCRIPTION
PO-NO	N	3	Unique purchase order (PO) number. Many parts can be ordered in one PO
PO-DATE	D	8	DDMMYYYY date when PO written
EMP-CODE	C	2	Unique code of employee who wrote the PO
SUPP-NO	N	3	Unique number assigned to supplier
SUPP-NAME	C	20	Supplier name
PART-NO	N	2	Unique number assigned to each part
PART-DESC	C	10	Part description
PART-QTY	N	2	Quantity of parts ordered in given PO

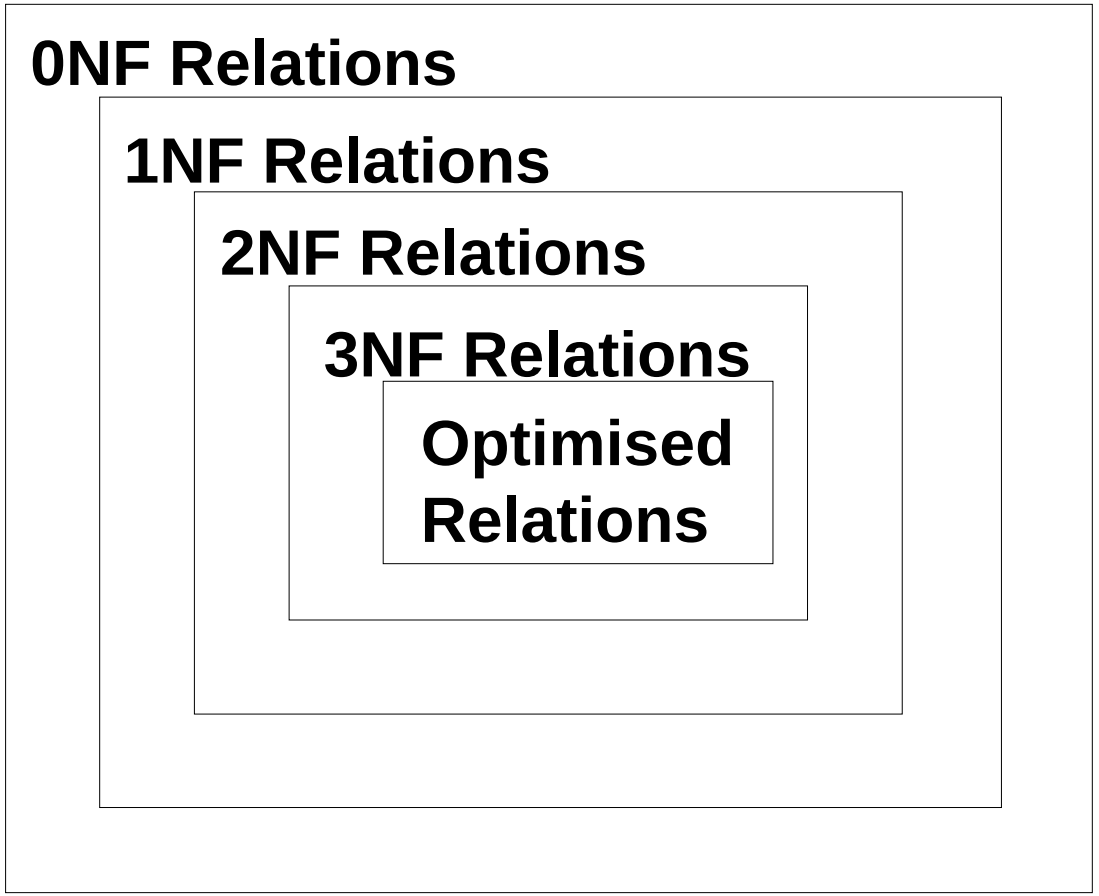
Key PO-NO

Purchase Order Relation in 0NF

PO-NO	PO-DATE	EMP-CODE	SUPP-NO	SUPP-NAME	PART-NO	PART-DESC	PART-QTY
111	01012001	M2	222	AC Stores	P1	Nut	10
					P2	Bolt	5
					P3	Nail	3
					P5	Screw	6
112	01012001	S3	105	I Hardware	P2	Bolt	2
					P5	Screw	1
113	02012001	S1	111	BC Trading	P1	Nut	3
					P3	Nail	4
114	02012001	M2	150	DO Service	P6	Plug	5
115	03012001	S1	222	AC Stores	P7	Pin	8
116	04012001	S1	100	LM Centre	P8	Fuse	2

Normalisation Process

Apply a set of normalisation rules to all the attributes of the entity types identified in the data requirement step.



Output of the Normalisation Process

- A list of normalised entity types in at least third normal form (3NF), such that all non-key attributes of each entity type fully depend on the whole key and nothing but the key

First Normal Form - 1NF

A relation is in First Normal Form (1NF) if **ALL** its attributes are **ATOMIC**.

ie. If there are no repeating groups.

If each attribute is a primitive.

e.g. integer, real number, character string, but not lists or sets

non-decomposable data item

single-value

Purchase Order Relation in 0NF

PO(PO-NO, PO-DATE, EMP-CODE, SUPP-NO, SUPP-NAME, PARTS-ORDERED{PART-NO, PART-DESC, PART-QTY})

Within a single purchase order we could find several part numbers, part descriptions and part quantities. Hence, parts ordered can be decomposed.

Purchase Order Relation in 0NF

PO-No	PO-Date	Emp-Code	Supp-No	Supp-Name	Part-No	Part-Desc	Part-Qty
111	01012001	M2	222	AC Stores	P1	Nut	10
					P2	Bolt	5
					P3	Nail	3
					P5	Screw	6
112	01012001	S3	105	I Hardware	P2	Bolt	2
					P5	Screw	1
113	02012001	S1	111	BC Trading	P1	Nut	3
					P3	Nail	4
114	02012001	M2	150	DO Service	P6	Plug	5
115	03012001	S1	222	AC Stores	P7	Pin	8
116	04012001	S1	100	LM Centre	P8	Fuse	2

First Normal Form - 1NF

- 1NF deals with the *shape* of a record type
- All occurrences of a record type must contain the same number of fields
- A relational schema is at least in 1NF

1NF - Actions Required

1) Examine for repeat groups of data
2) Remove repeat groups from relation
3) Create new relation(s) to include repeated data
4) Include key of the 0NF to the new relation(s)
5) Determine key of the new relation(s)

Purchase Order Relations in 1NF

PO

PO-NO	PO-DATE	EMP-CODE	SUP P-NO	SUPP-NAME
111	01012001	M2	222	AC Stores
112	01012001	S3	105	I Hardware
113	02012001	S1	111	BC Trading
114	02012001	M2	150	DO Service
115	03012001	S1	222	AC Stores
116	04012001	S1	100	LM Centre

PO-PART

PO-NO	PART-NO	PART-DESC	PART-QTY
111	P1	Nut	10
111	P2	Bolt	5
111	P3	Nail	3
111	P5	Screw	6
112	P2	Bolt	2
112	P5	Screw	1
113	P1	Nut	3
113	P3	Nail	4
114	P6	Plug	5
115	P7	Pin	8
116	P8	Fuse	2

Problems - 1NF

1. INSERT PROBLEM

   cannot know available parts until an order is placed (e.g. P4 is bush)

2. DELETE PROBLEM

   loose information of part P7 if we cancel purchase order 115 (i.e. Delete PO-PART for Part No P7)

3. UPDATE PROBLEM:

   to change description of Part P3 we need to change every tuple in PO-PART containing Part No P3

Second Normal Form - 2NF

A relation is in 2NF if it is in 1NF and every non-key attribute is dependent on the whole key

i.e. Is not dependent on part of the key only.

PO-PART Relation (Parts Ordered) in 1NF

PO-PART(PO-NO, PART-NO, PART-DESC, PART-QTY)

Part Description is depended only on Part No, which is part of the key of PO-PART.

Parts Ordered Relation in 1NF

PO-No	Part-No	Part-Desc	Part-Qty
111	P1	Nut	10
111	P2	Bolt	5
111	P3	Nail	3
111	P5	Screw	6
112	P2	Bolt	2
112	P5	Screw	1
113	P1	Nut	3
113	P3	Nail	4
114	P6	Plug	5
115	P7	Pin	8
116	P8	Fuse	2

Second Normal Form - 2NF

Deals with the relationship between non-key and key fields

A non-key field cannot be a fact about a subset of a key

It is relevant when the key is composite, i.e. consists of several fields

2NF - Actions Required

If entity has a concatenated key

1) Check each attribute against the whole key
2) Remove attribute and partial key to new relation
3) Optimise relations

Parts Ordered Relations in 2NF

PO-PART

PO-NO	PART-NO	PART-QTY
111	P1	10
111	P2	5
111	P3	3
111	P5	6
112	P2	2
112	P5	1
113	P1	3
113	P3	4
114	P6	5
115	P7	8
116	P8	2

PART

PART-NO	PART-DESC
P1	Nut
P2	Bolt
P3	Nail
P5	Screw
P6	Plug
P7	Pin
P8	Fuse

Purchase Order Relations in 2NF

PART

PART-NO	PART-DESC
P1	Nut
P2	Bolt
P3	Nail
P5	Screw
P6	Plug
P7	Pin
P8	Fuse

PO-PART

PO-NO	PART-NO	PART-QTY
111	P1	10
111	P2	5
111	P3	3
111	P5	6
112	P2	2
112	P5	1
113	P1	3
113	P3	4
114	P6	5
115	P7	8
116	P8	2

PO

PO-NO	PO-DATE	EMP-CODE	SUPP-NO	SUPP-NAME
111	01012001	M2	222	AC Stores
112	01012001	S3	105	I Hardware
113	02012001	S1	111	BC Trading
114	02012001	M2	150	DO Service
115	03012001	S1	222	AC Stores
116	04012001	S1	100	LM Centre

Problems - 2NF

1. INSERT PROBLEM

   cannot know available suppliers until an order is placed (e.g. 200 is hardware stores)

2. DELETE PROBLEM

   loose information of supplier 100 if we cancel purchase order 116 (i.e. Delete PO for Supplier No 100)

3. UPDATE PROBLEM

   to change name of Supplier 222 we need to change every tuple in PO containing Supplier No 222

Third Normal Form - 3NF

A relation is in 3NF if it is in 2NF and each non-key attribute is only dependent on the whole key, and not dependent on any non-key attribute.

i.e. no transitive dependencies

PO Relation in 2NF

PO(PO-NO, PO-DATE, EMP-CODE, SUPP-NO,
SUPP-NAME)

Supplier name is a non-key field depended on another non-key field (i.e. the supplier no) in addition to be depended on the key purchase order no

Third Normal Form - 3NF

Deals with the relationship between non-key fields

A non-key field cannot be a fact about another non-key field

3NF - Actions Required

1) Check each non-key attribute for dependency against other non-key fields
2) Remove attribute depended on another non-key attribute from relation
3) Create new relation comprising the attribute and non-key attribute which it depends on
4) Determine key of new relation
5) Optimise

PO and SUPPLIER Relations in 3NF

PO

PO-NO	PO-DATE	EMP-CODE	SUPP-NO
111	01012001	M2	222
112	01012001	S3	105
113	02012001	S1	111
114	02012001	M2	150
115	03012001	S1	222
116	04012001	S1	100

SUPPLIER

SUPP-NO	SUPP-NAME
100	LM Centre
105	I Hardware
111	BC Trading
150	DO Service
222	AC Stores

Purchase Order Relations in 3NF

SUPPLIER

SUPP-NO	SUPP-NAME
222	AC Stores
105	I Hardware
111	BC Trading
150	DO Service
100	LM Centre

PART

PART-NO	PART-DESC
P1	Nut
P2	Bolt
P3	Nail
P5	Screw
P6	Plug
P7	Pin
P8	Fuse

PO-PART

PO-NO	PART-NO	PART-QTY
111	P1	10
111	P2	5
111	P3	3
111	P5	6
112	P2	2
112	P5	1
113	P1	3
113	P3	4
114	P6	5
115	P7	8
116	P8	2

PO

PO-NO	PO-DATE	EMP-CODE	SUPP-NO
111	01012001	M2	222
112	01012001	S3	105
113	02012001	S1	111
114	02012001	M2	150
115	03012001	S1	222
116	04012001	S1	100

Further Normalization

- BCNF or Boyce–Codd Normal form
- 4th Normal form
- 5th Normal form

  In a normal situation normalization up-to 3NF is quite sufficient.Certain relations may even be de-normalized on account of efficiency. The Normalizations which are discussed next are not practically enforced most of the time.
- But a relation in 3NF does not guarantee that all anomalies have been removed, hence the additional normalizations.

Definition of BCNF

- In every functional dependency A→B in a relation, *A must be a candidate key*.
- Eliminates any remaining update anomalies in 3NF relations.

- **Students**
  - Each student may have many Subjects.
  - For each major they do, a student has a supervisor.
- **Subjects**
  - Each major has several possible supervisors
- **Supervisors**
  - Each supervisor only supervises one major.
  - Within their one major, supervisors supervise many students.
- **Student_major relation**
- The relation will have the following attributes for each tuple:

  student-id

  major

  Supervisor

- Let’s choose (student-id, major) as the primary key.
  (student-id, supervisor) is an alternate key
  student_major(student-id, major, supervisor)

student-id	major	supervisor
999	Physics	Rohan
789	Physics	Mira
456	Biology	Sobhit
123	Music	Sohan
123	Physics	Rohan

- **Functional dependencies 1**

  (student-id, major) → supervisor

Each major has **several** possible supervisors. (We can't deduce *any* functional dependency from this statement.)

- **Functional dependencies 2**

  Supervisor→. Major

- There are only 2 functional dependencies that we can write from the information we know:

  (student-id,major)→supervisor

  supervisor→major

Is student_major in 1NF?

(Check Repetition Group)

student-id	major	supervisor
999	Physics	Rohan
789	Physics	Mira
456	Biology	Sobhit
123	Music	Sohan
123	Physics	Rohan

- Yes.
- There are no repeating groups within tuples.

Is student_major in 2NF?

(Check Partial Key Dependency)

student-id	major	supervisor
999	Physics	Rohan
789	Physics	Mira
456	Biology	Sobhit
123	Music	Sohan
123	Physics	Rohan

- Yes.
- There are no partial key dependencies.
- A partial key dependency means there exists a non-key attribute dependent on only part of a candidate key.

Look at the primary key…

- student-id does not determine supervisor on its own.
- major does not determine supervisor on its own.
- i.e. The whole PK is required to identify supervisor.

Is student_major in 3NF?

(Check Transitive Key Dependency)

student-id	major	supervisor
999	Physics	Rohan
789	Physics	Mira
456	Biology	Sobhit
123	Music	Sohan
123	Physics	Rohan

- Yes.
- There are no transitive key dependencies.
- But isn't supervisor→major a transitive dependency??
- NO!
- Supervisor and major are *key attributes (each* forms part of a candidate key). Transitive dependencies only exist between non-key attributes.

But there are still anomalies!

student-id	Subject	supervisor
999	Physics	Rohan
789	Physics	Mira
456	Biology	Sobhit
123	Music	Sohan
123	Physics	Rohan

- **Modification anomaly**:
- e.g. change the name of Rohan to Rima.

  Must change this information in many places.
- **Deletion anomaly:**
- e.g. delete student 456, Lost information that Sobhit supervises Biology!
- **Insertion anomaly:**
- e.g. record the fact that Mina supervises programming. Not possible unless a student Subjects in programming.

BCNF solves these problems but how and why??

- These anomalies arise because of the functional dependency:

  supervisor→major

- It turns out that if we stipulate that supervisor is a candidate key, we will not have these anomalies.
- Split the 3NF relation into two:

  Student_supervisor (student-id,supervisor)

  Supervisor_major (supervisor, major)

- Attributes dependent on the non-candidate key attribute moved to a new relation, with the non-candidate key as the primary key. Now the anomalous FD has a candidate (primary) key on the left hand side.
- The original relation can be re-created by joining the two individual relations.

Definition of BCNF

- In every functional dependency A→B in a relation, *A must be a candidate key*.
- Eliminates any remaining update anomalies in 3NF relations.

Multivalued Dependencies (MVDs)

	α	β	$R - \alpha - \beta$
t_1	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

- Let R be a relation schema and let $\alpha \subseteq R$ and $\beta \subseteq R$. The *multivalued dependency*

  $$\alpha \twoheadrightarrow \beta$$

  holds on R if in any legal relation $r(R)$, for all pairs for tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r such that:

  $t_1[\alpha] = t_2[\alpha] = t_3[\alpha]$ $t_4[\alpha]$
  $t_3[\beta] = t_1[\beta]$
  $t_3[R - \beta] = t_2[R - \beta]$
  $t_4[\beta] = t_2[\beta]$
  $t_4[R - \beta] = t_1[R - \beta]$

Multivalued Dependencies

- There are database schemas in BCNF that do not seem to be sufficiently normalized
- Consider a database

  classes(course, teacher, book)

  such that $(c,t,b) \in classes$ means that *t* is qualified to teach *c*, and *b* is a required textbook for *c*
- The database is supposed to list for each course the set of teachers any one of which can be the course's instructor, and the set of books, all of which are required for the course (no matter who teaches it).

Multivalued Dependencies (Cont.)

course	*teacher*	*book*
database	Avi	DB Concepts
database	Avi	Ullman
database	Hank	DB Concepts
database	Hank	Ullman
database	Sudarshan	DB Concepts
database	Sudarshan	Ullman
operating systems	Avi	OS Concepts
operating systems	Avi	Shaw
operating systems	Jim	OS Concepts
operating systems	Jim	Shaw

classes

- There are no non-trivial functional dependencies and therefore the relation is in BCNF
- Insertion anomalies – i.e., if Sara is a new teacher that can teach database, two tuples need to be inserted

(database, Sara, DB Concepts)
(database, Sara, Ullman)

Multivalued Dependencies (Cont.)

- Therefore, it is better to decompose *classes* into:

course	*teacher*
database	Avi
database	Hank
database	Sudarshan
operating systems	Avi
operating systems	Jim

teaches

course	*book*
database	DB Concepts
database	Ullman
operating systems	OS Concepts
operating systems	Shaw

text

We shall see that these two relations are in Fourth Normal Form (4NF)

Example (Cont.)

- In our example:

  $course \rightarrow\rightarrow$ teacher
  $course \rightarrow\rightarrow book$

- The above formal definition is supposed to formalize the notion that given a particular value of *Y* (*course*) it has associated with it a set of values of *Z (teacher)* and a set of values of *W (book)*, and these two sets are in some sense independent of each other.
- Note:
  - If $Y \rightarrow Z$ then $Y \rightarrow\rightarrow Z$
  - Indeed we have (in above notation) $Z_1 = Z_2$
    The claim follows.