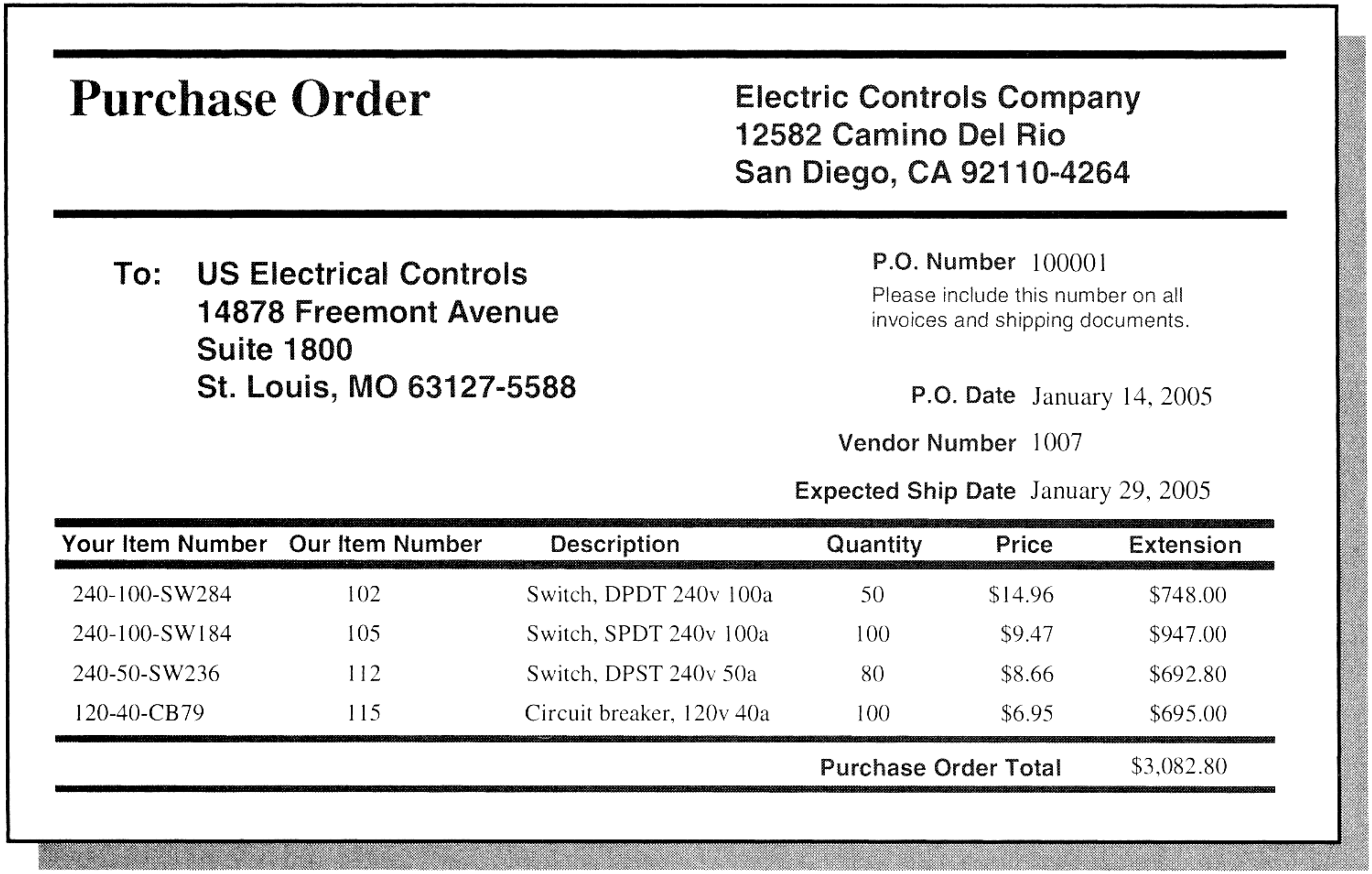


Purchase Order

Electric Controls Company
12582 Camino Del Rio
San Diego, CA 92110-4264

To: US Electrical Controls
14878 Freemont Avenue
Suite 1800
St. Louis, MO 63127-5588

P.O. Number 100001
Please include this number on all invoices and shipping documents.

P.O. Date January 14, 2005

Vendor Number 1007

Expected Ship Date January 29, 2005

Your Item Number	Our Item Number	Description	Quantity	Price	Extension
240-100-SW284	102	Switch, DPDT 240v 100a	50	$14.96	$748.00
240-100-SW184	105	Switch, SPDT 240v 100a	100	$9.47	$947.00
240-50-SW236	112	Switch, DPST 240v 50a	80	$8.66	$692.80
120-40-CB79	115	Circuit breaker, 120v 40a	100	$6.95	$695.00
				Purchase Order Total	$3,082.80

Purchase Order - Attribute Analysis

ATTRIBUTE	TYPE	LENGTH	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

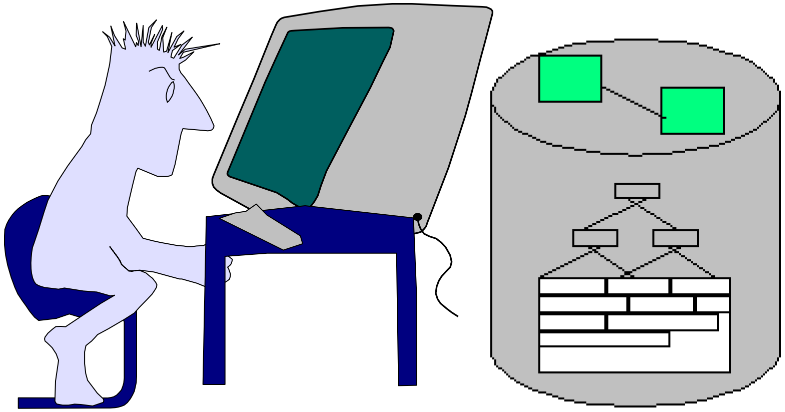
Key PO-NO

Purchase Order Relation in 0NF

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

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



Objectives

- Explain the reasons for normalising data
- Describe the process of normalisation
- Describe the benefits of 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.

Why Normalisation?

- Understand the meaning of data
- Understand inter-relationship and inter-dependence of data
- Minimise data redundancy and inconsistency
- Prevent data update problems

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

Normal Forms cont’d

BCNF any remaining anomalies that result from functional dependencies have been removed

4NF any multi-valued dependencies have been removed

5NF any remaining anomalies 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

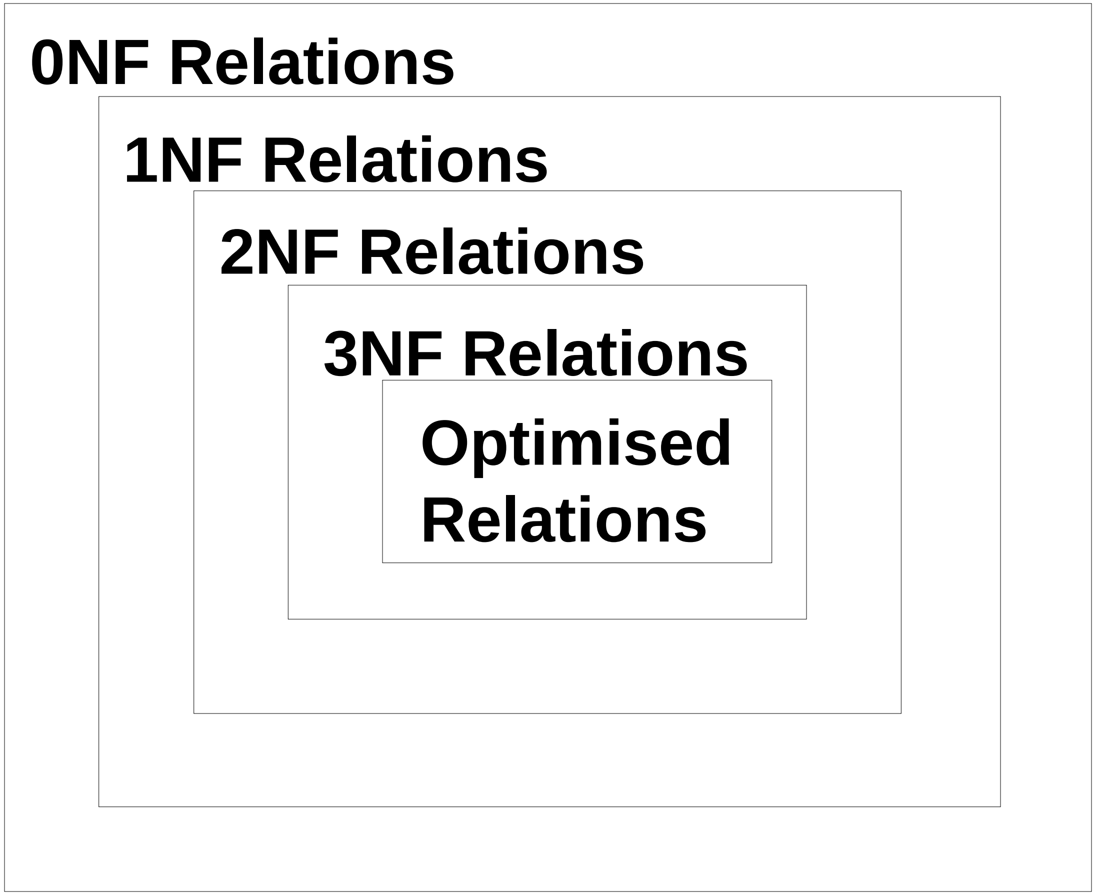
entity types *(e.g. Supplier, Order)*

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)*

Normalisation Process

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



0NF Relations

1NF Relations

2NF Relations

3NF Relations

Optimised Relations

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**.

i.e. 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

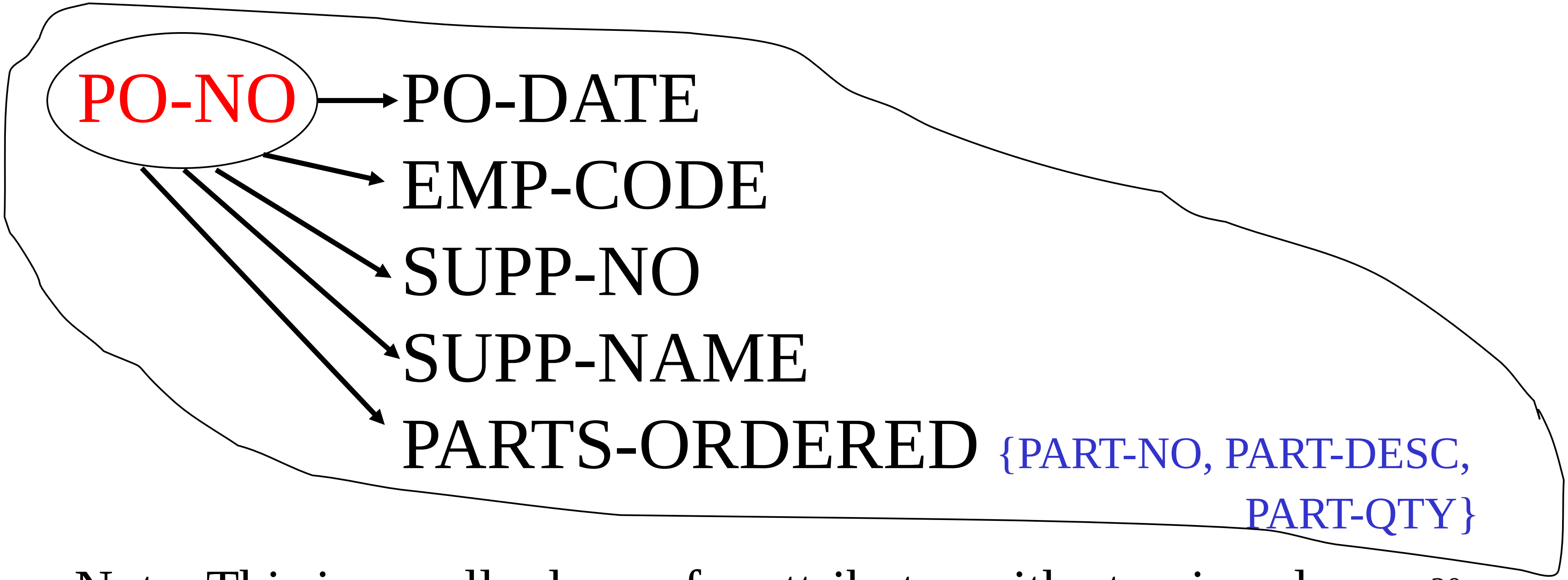
Purchase Order Relation in 0NF

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 Functional Dependency diagram

PO-NO → PO-DATE, EMP-CODE, SUPP-NO, SUPP-NAME, PARTS-ORDERED



PO-NO

PO-DATE

EMP-CODE

SUPP-NO

SUPP-NAME

PARTS-ORDERED {PART-NO, PART-DESC, PART-QTY}

Note: This is usually drawn for attributes with atomic values

Purchase Order Relation in 0NF

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

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 Relation in 0NF

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

PO

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
…	…	…	…	…	…	…	…

Purchase Order Relation in 1NF

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

PO

PO-NO	PO-DATE	EMP-CODE	SUPP-NO	SUPP-NAME
111	01012001	M2	222	AC Stores
112	01012001	S3	105	I Hardware
…	…	…	…	…

Include key PO-NO

New key is PO-NO, PART-NO

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
…	…	…	…

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

Purchase Order Relations in 1NF

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

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

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

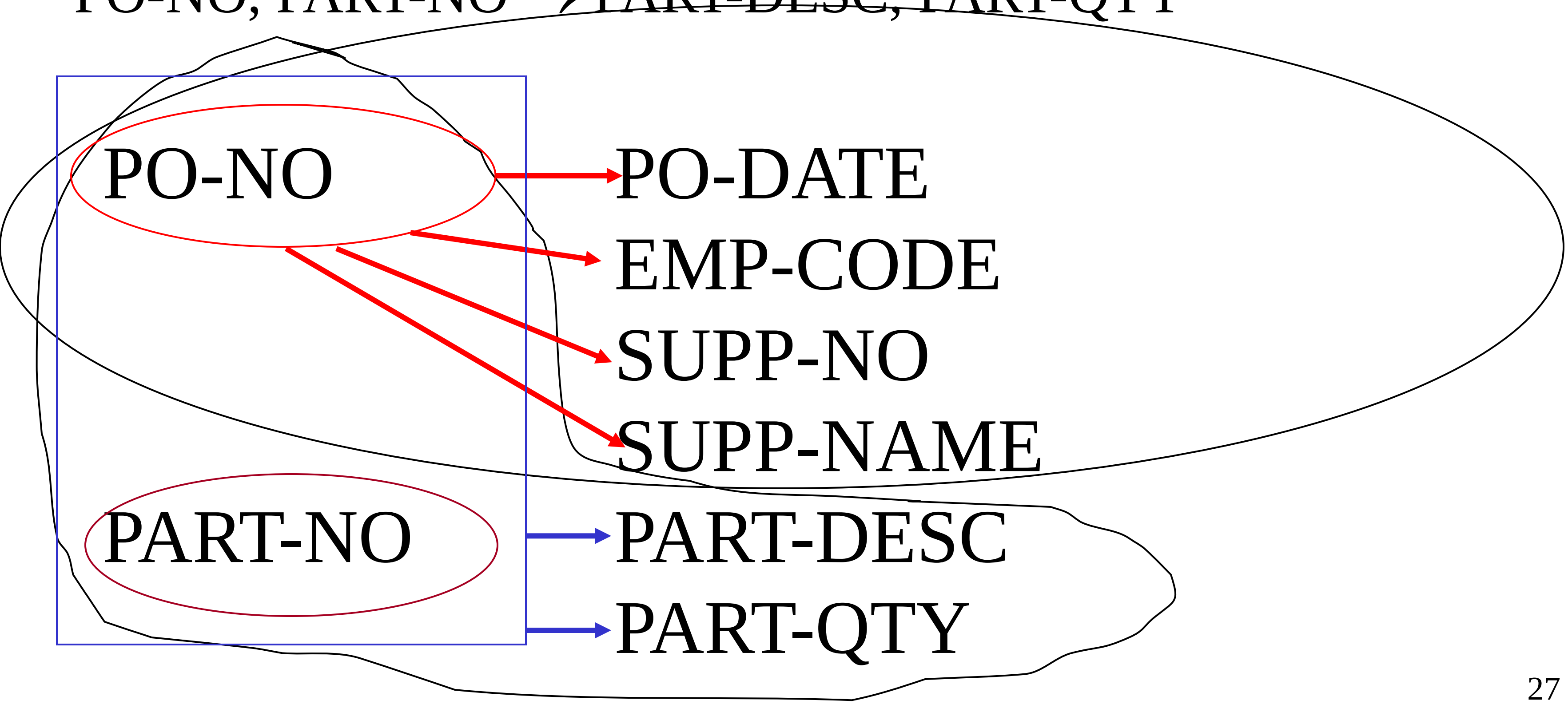
PO-PART

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

Purchase Order Relation in 1NF Functional Dependency diagram

PO-NO → PO-DATE, EMP-CODE, SUPP-NO, SUPP-NAME

PO-NO, PART-NO → PART-DESC, PART-QTY



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

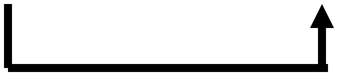
Second Normal Form - 2NF

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

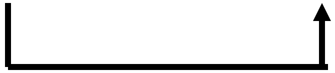
PO-PART(PO-NO, **PART-NO**, PART-DESC,
PART-QTY)



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

Parts Ordered Relation in 1NF

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



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 - consider combining tables that have identical primary keys

Parts Ordered Relation in 1NF

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

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

PART

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

Key is PART-NO

Parts Ordered Relations in 2NF

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

PART(PART-NO, PART-DESC)

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

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

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

PART(PART-NO, PART-DESC)

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

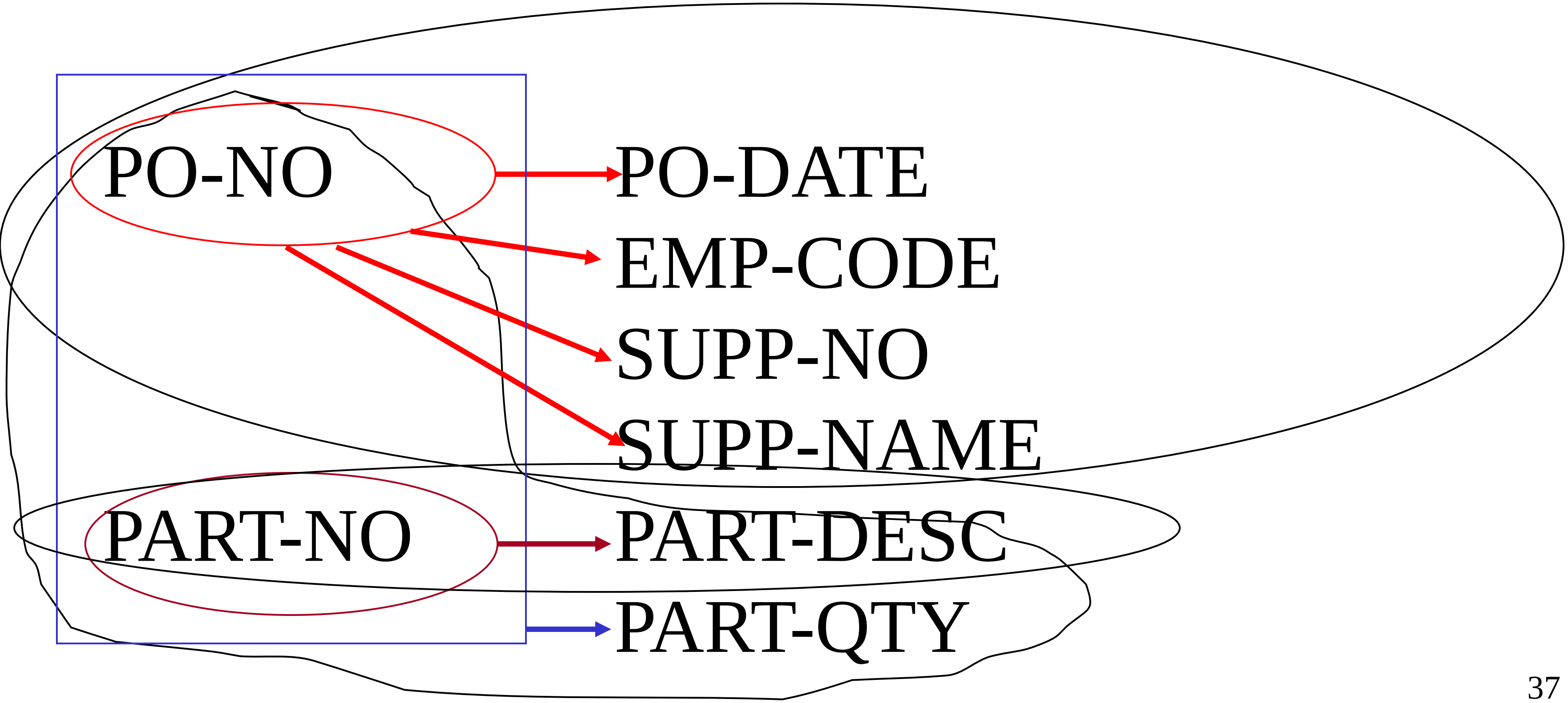
PO-NO	PO-DATE	EMP-CODE	SUPP-NO	SUPP-NAME
111	01012001	M2	222	AC Stores
112	01012001	S3	105	I Hardware
113	02012001	S1	111	BC Trading
114	02012001	M2	150	DO Service
115	03012001	S1	222	AC Stores
116	04012001	S1	100	LM Centre

Purchase Order Relation in 2NF Functional Dependency diagram

PO-NO → PO-DATE, EMP-CODE, SUPP-NO, SUPP-NAME

PO-NO, PART-NO → PART-QTY

PART-NO → PART-DESC



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

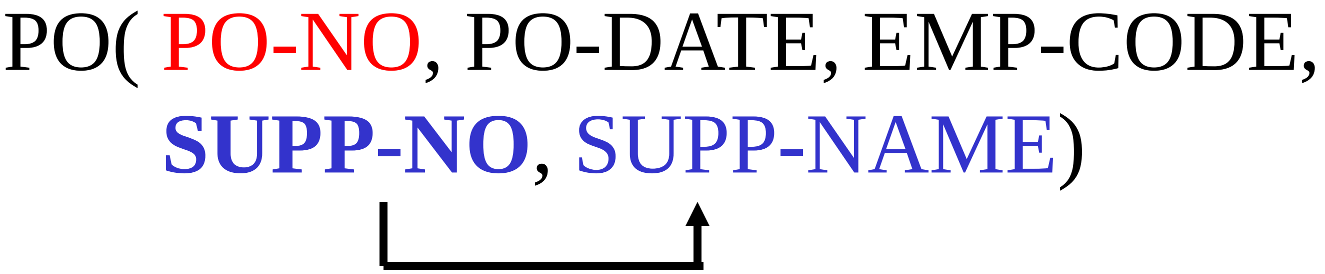
   to change name of Supplier 222 we need to change every tuple in PO containing Supplier No 222

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

Purchase Order Relation in 2NF

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

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

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 - consider combining tables that have identical primary keys

Purchase Order Relation in 2NF

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

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

SUPPLIER

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

Key is SUPP-NO

PO and SUPPLIER Relations in 3NF

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

SUPPLIER(SUPP-NO, SUPP-NAME)

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

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

PART(PART-NO, PART-DESC)

SUPPLIER(SUPP-NO, SUPP-NAME)

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

SUPPLIER

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

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

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

Normalised Relations

It is sufficient to show only the relations using only the following notations, i.e. not necessary to show the tables with data elements.

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

PART(PART-NO, PART-DESC)

SUPPLIER(SUPP-NO, SUPP-NAME)

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

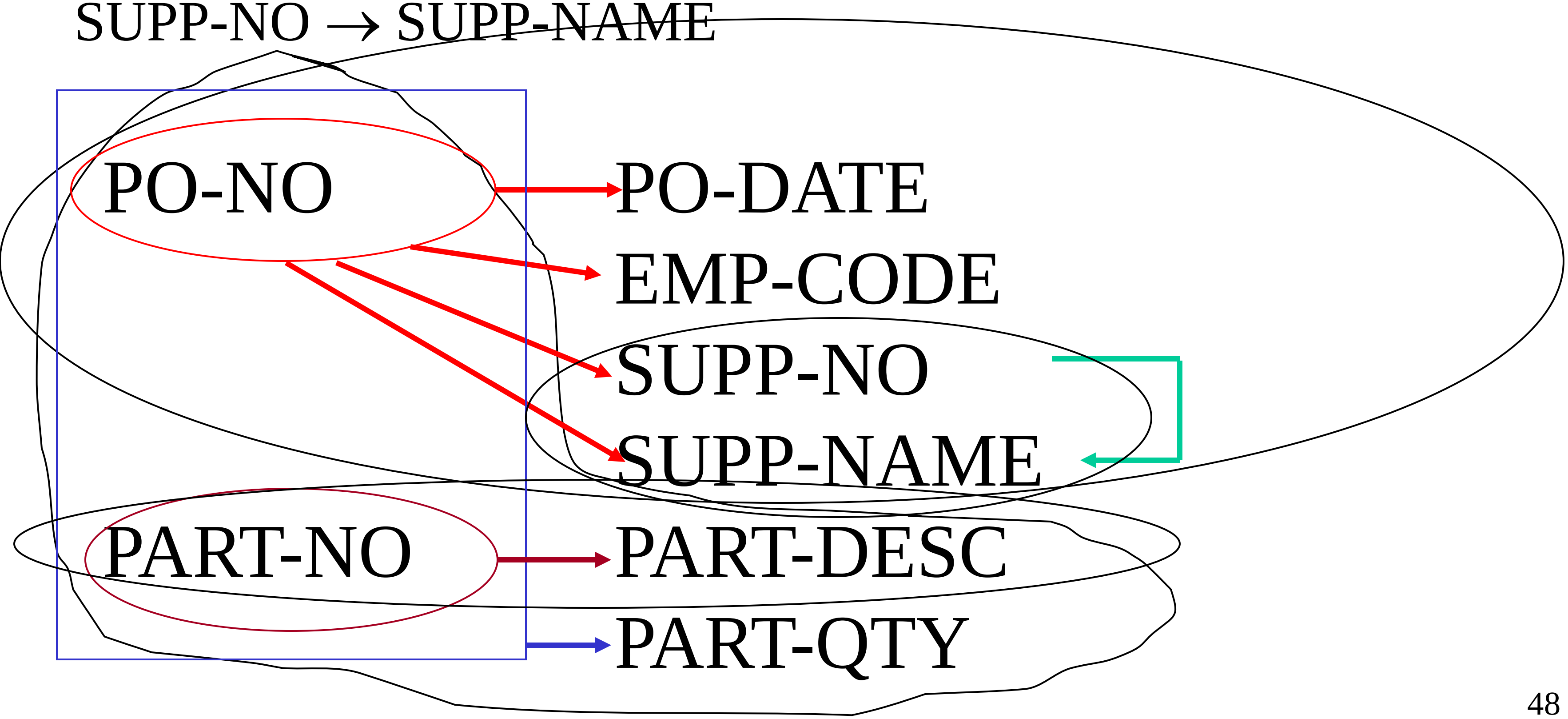
Purchase Order Relation in 3NF Functional Dependency diagram

PO-NO → PO-DATE, EMP-CODE, SUPP-NO

PO-NO, PART-NO → PART-QTY

PART-NO → PART-DESC

SUPP-NO → SUPP-NAME



PO-NO

PO-DATE

EMP-CODE

SUPP-NO

SUPP-NAME

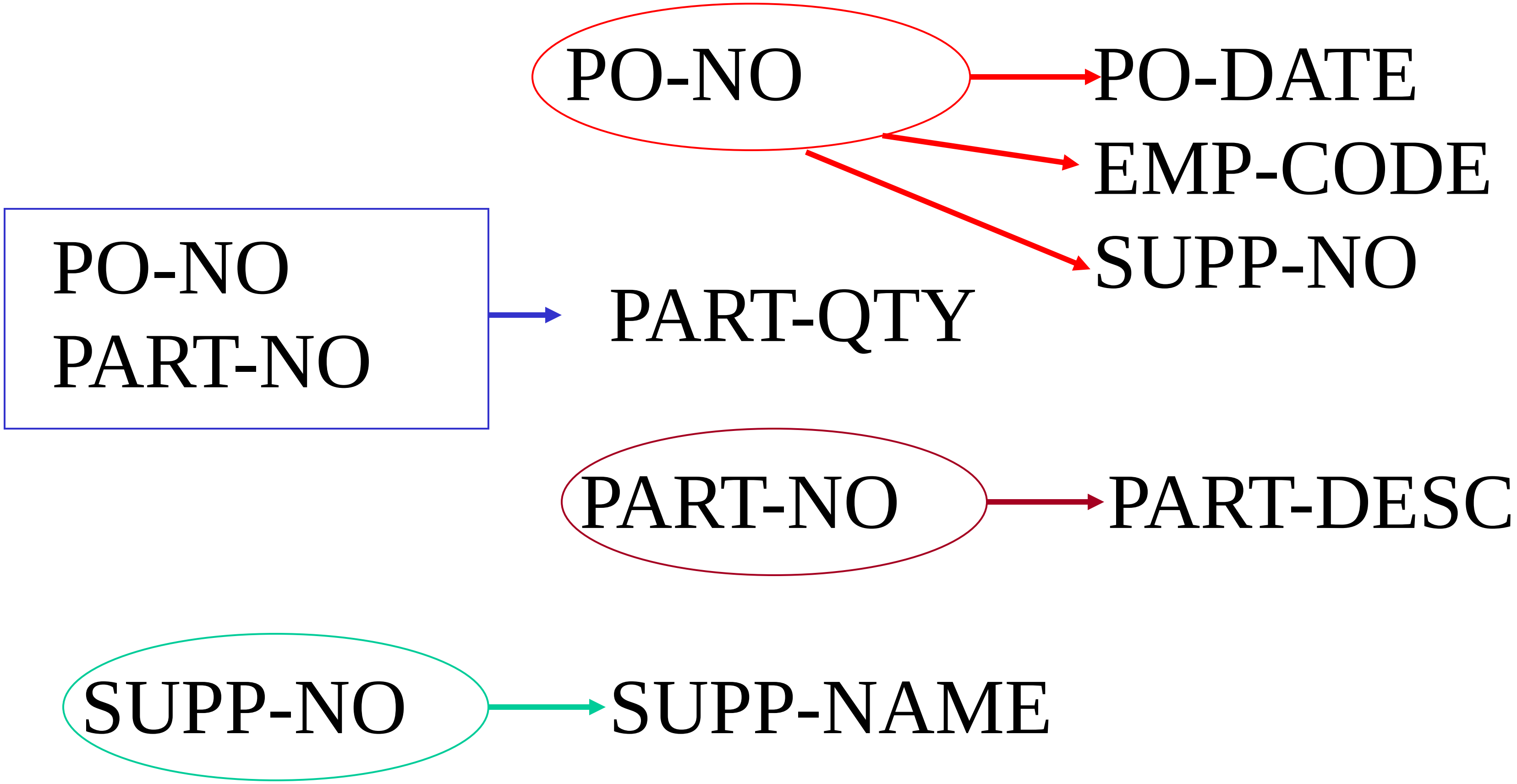
PART-NO

PART-DESC

PART-QTY

Functional Dependency diagram ...

Draw one diagram showing **all** dependencies as in previous slide or a collection of diagrams as shown below



Optimisation

combining tables that have identical primary keys

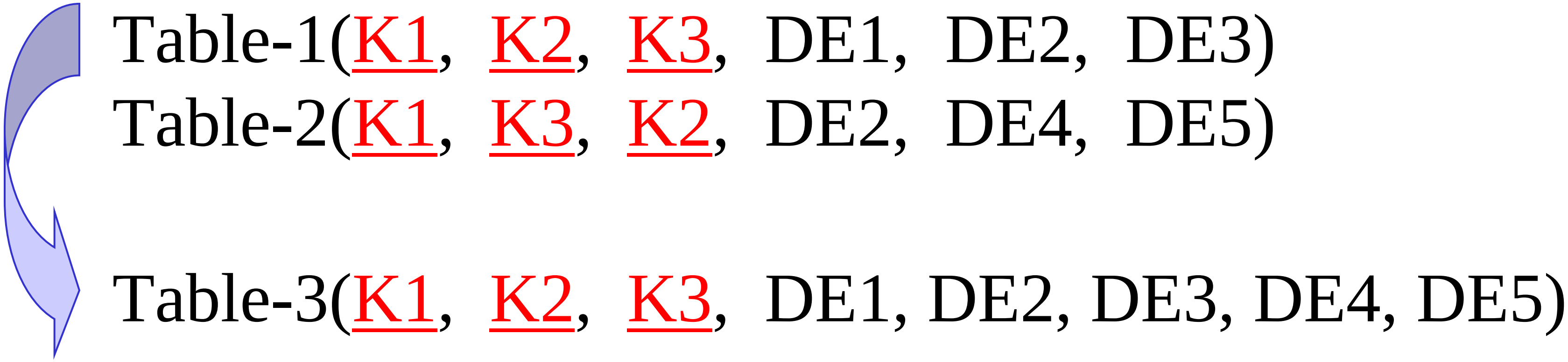


Table-1(K1, K2, K3, DE1, DE2, DE3)

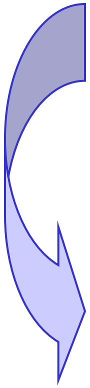
Table-2(K1, K3, K2, DE2, DE4, DE5)

Table-3(K1, K2, K3, DE1, DE2, DE3, DE4, DE5)

- K is a Key-element and DE is a Data-element
- Key sequence is not important
- Don’t lose data elements

Optimisation ...

combining tables that have identical primary keys



STUDENT(NID, Name, Course)
EMP(NID, Name, Designation)

STU-EMP(NID, Name, Course, Designation)

If a student can become an employee, and vice-versa, and also be both at a given time, we can reduce duplication by COMBINING the tables

Optimisation ...

Two Approaches

1. Optimise at the end of each normalization step
2. Optimise after 3NF

- Recommend a combination of both
- Combine simple key relations with few data elements after each step
- Leave other more complex key relations until after 3NF
- Reason for this being that more re-working is necessary if optimise everything, then subsequent revisions or errors correction required.

Generalisation

- This process reduces redundancy and is done in addition to Normalisation
- Examine root tables, i.e. those that represent entity types directly, and their repeating groups (after 1NF).

  e.g. STUDENT, EMPLOYEE, etc.
- Reduce data elements to their Generic Names

  e.g. INSTRUCTOR-NAME becomes NAME,
  STUDENT-NAME becomes NAME,
  ISTRUCTOR-NUMBER becomes NUMBER,
  STUDENT-NUMBER becomes NUMBER,

Generalisation ...

reduces redundancy

STUDENT(NID, Stu-Name, Course)
EMP(NID, Emp-Name, Designation)

STU-EMP(NID, Stu-Name, Emp-Name, Course, Designation)

STU-EMP(NID, Name, Course, Designation)

Generalisation ...

- Hence if any tables have the same generic key and are occurrences of the same entity type (as opposed to different entity-types) we:
    - COMBINE to form a “higher” entity type.
    - Determine primary key of new table
    - Amend old tables associated with the old key(s)

Multi-Typing

- Generalisation leads to the creation of “higher” entity type(s).
- However these super-types contain many null data elements
- Multi-typing addresses this problem by creating super-types and sub-types.
- A sub-type can be thought of as an “instance” of the super-type, i.e. a way of categorising occurrences of the super_type.
- A single occurrence of the super-type may be made up of one or more sub-types
- A sub-type cannot exist without the super-type.

Multi-Typing ...

- Super-Type
  - PERSON(Person-ID, Person-Name, Person-Address, Person-Tel-No)

- Sub-Type
  - EMPLOYEE(Person-ID [EMPLOYEE], Employee-Appointed-Date, Employee-No)

- Sub-Type
  - STUDENT (Person-ID [STUDENT], Student-Enrolment-Date, Student-No)

Benefits of Normalisation

- Data requirements are thoroughly analysed:
  - Data interdependence are recognised
  - Meaning and definition of each data element are established
  - Update anomalies are removed
  - Data inconsistencies are eliminated
- Simplified way of compiling data into a set of non-ambiguous entity types
- Practical approach built on theoretical basis
- Foundation for an optimum physical data base design using any DBMS