NORMALIZATION

EGCO321 DATABASE SYSTEMS

KANAT POOLSAWASD
DEPARTMENT OF COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

AVOIDANCE OF MODIFICATION ANOMALY

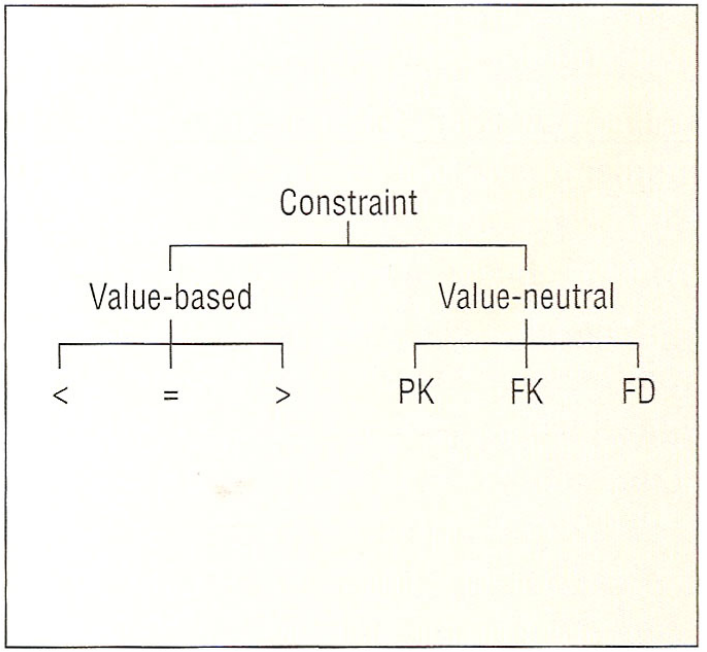
- Modification anomaly is an unexpected side effect that occurs when changing the data in a table with excessive redundancies.
- To understand more precisely the impact of modification anomalies, let us consider a poorly design database. Imagine that a university database consist of the single table show in this table.

TABLE 7.1 Sample Data for the Big University Database Table

StdSSN	StdCity	StdClass	OfferNo	OffTerm	OffYear	EnrGrade	CourseNo	CrsDesc
S1	SEATTLE	JUN	O1	FALL	2006	3.5	C1	DB
S1	SEATTLE	JUN	O2	FALL	2006	3.3	C2	VB
S2	BOTHELL	JUN	O3	SPRING	2007	3.1	C3	OO
S2	BOTHELL	JUN	O2	FALL	2006	3.4	C2	VB

DATABASE CONSTRAINTS

- Constrains can be characterized as value-based versus value-neutral (See below figure)
  - A value-based constrain involves a comparison of a column to a constrain using a comparison operator.
  - A value-neutral constrain involves a comparison of column.



FUNCTIONAL DEPENDENCY (1)

- A functional dependency is another important kind of value-neutral constrain.
- A Function Dependency (FD) is a constraint about two or more columns of a table X determines Y (X → Y) if there exists at most one value of Y for every value of X.
- For example, Social Security number determines City (StdSSN → StdCity) in the university database table if there is at most one city value for every Social Security number.
- A column appearing on the left-hand side of an FD is called a determinant or alternatively, an LHS for left-hand side. In this example, StdSSN is a determinant.

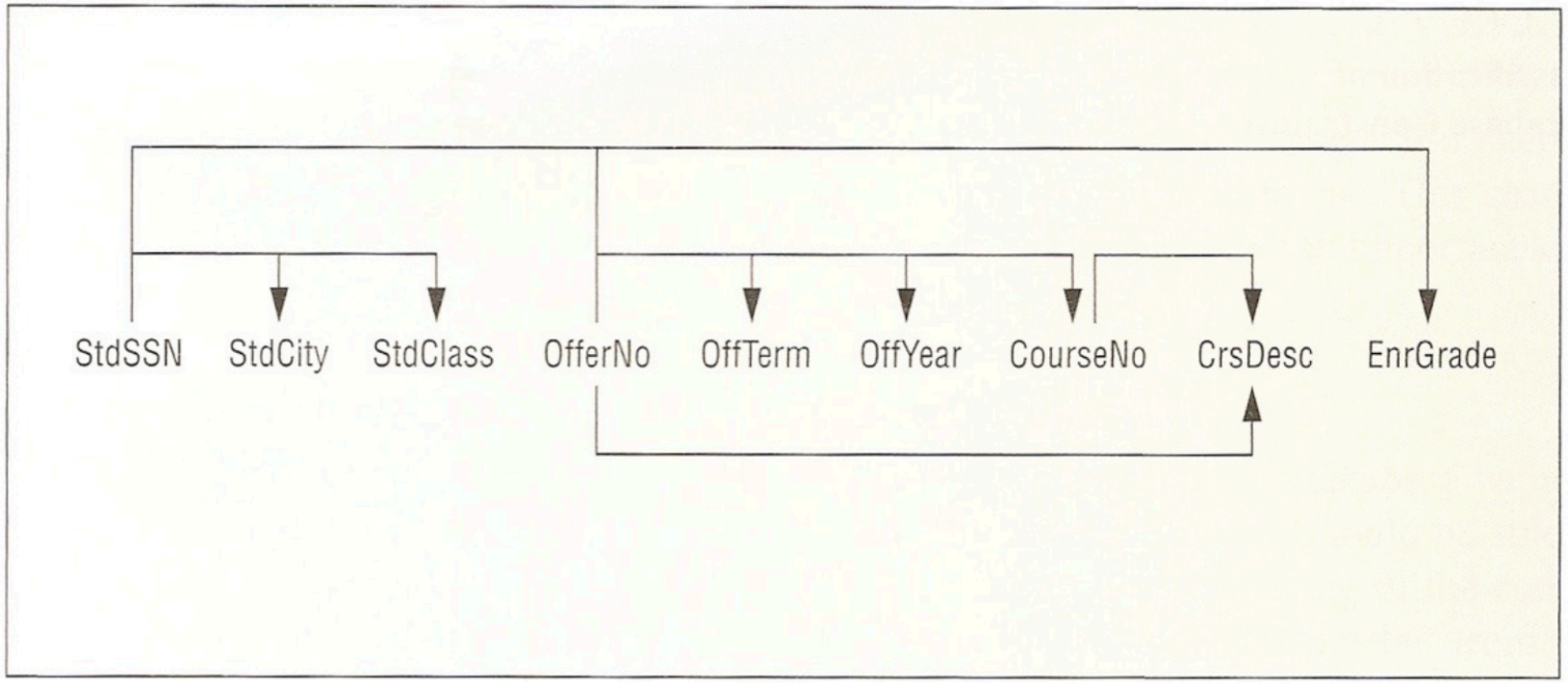
FUNCTIONAL DEPENDENCY (2)

- You can also think about functional dependencies as identifying potential candidate keys. By stating that $X \rightarrow Y$, if X and Y are placed together in a table without other columns, X is a candidate key.

FUNCTIONAL DEPENDENCY (3)

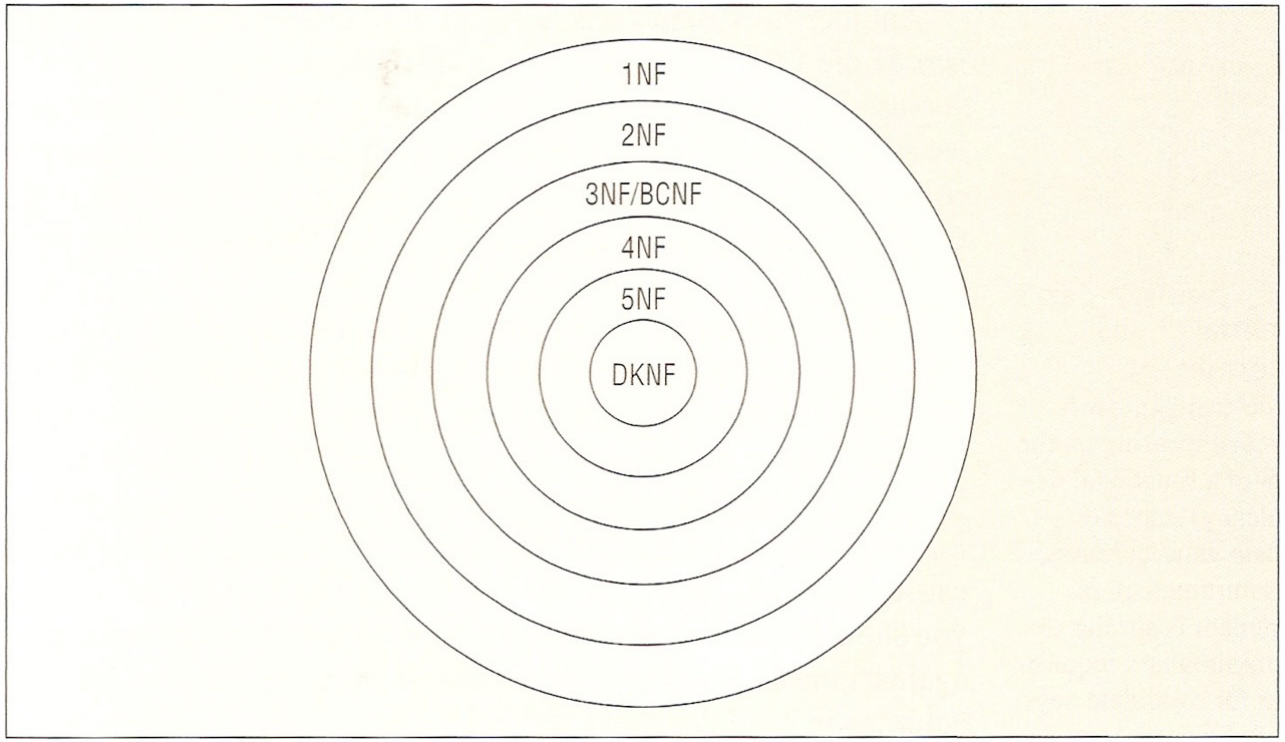
TABLE 7.1 Sample Data for the Big University Database Table

StdSSN	StdCity	StdClass	OfferNo	OffTerm	OffYear	EnrGrade	CourseNo	CrsDesc
S1	SEATTLE	JUN	O1	FALL	2006	3.5	C1	DB
S1	SEATTLE	JUN	O2	FALL	2006	3.3	C2	VB
S2	BOTHELL	JUN	O3	SPRING	2007	3.1	C3	OO
S2	BOTHELL	JUN	O2	FALL	2006	3.4	C2	VB



NORMAL FORMS

- Normal form is a rule about allowable dependencies. Each normal form removes certain kinds of redundancies.

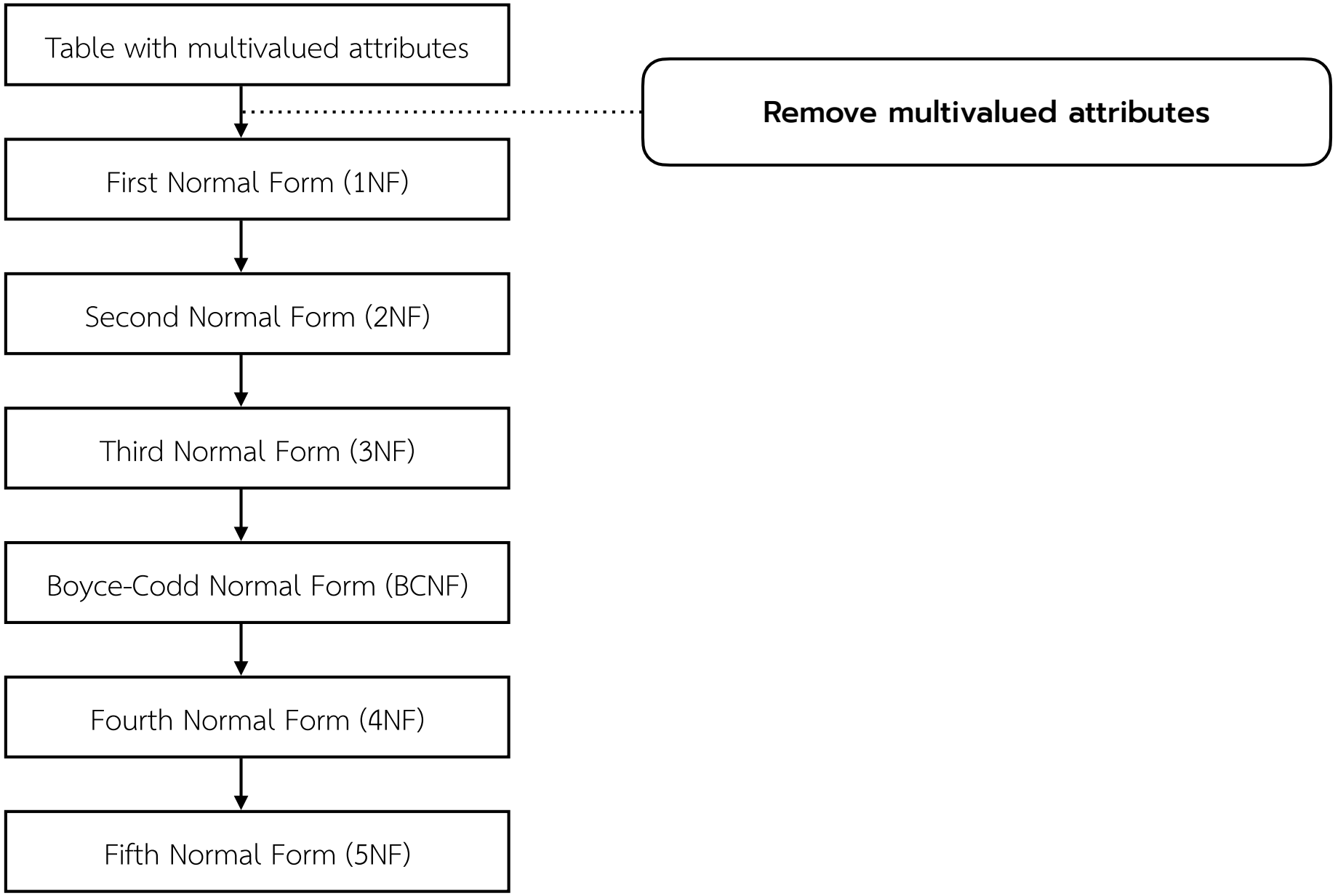


UNNORMALIZED FORM (UNF)

Table name: RPT_FORMAT **Database name: Ch05_ConstructCo**

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

STEPS IN NORMALIZATION



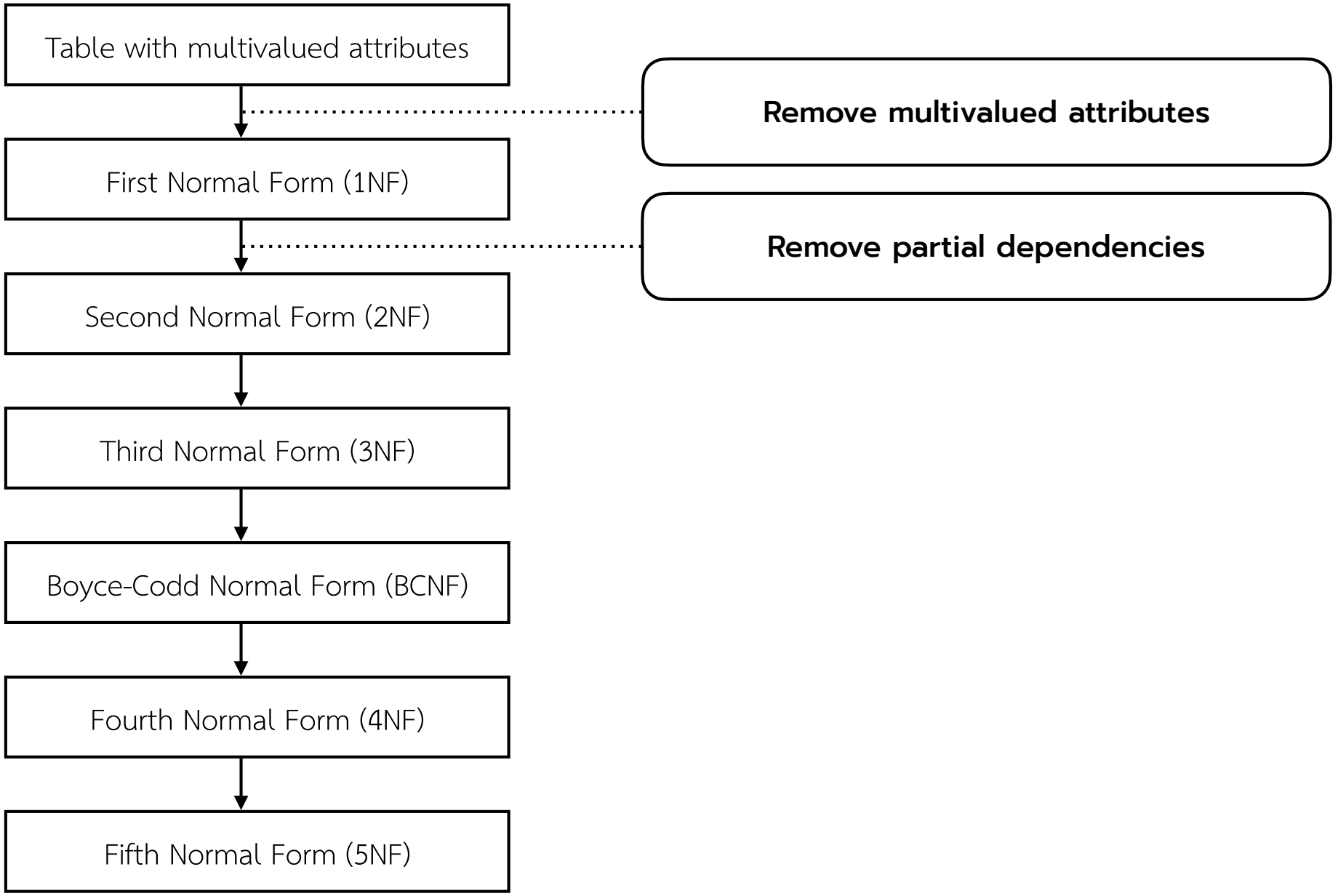
FIRST NORMAL FORM (1NF)

- 1NF prohibits nesting or repeating groups in tables. A table not in 1NF is unnormalized or non-normalized.

Table name: DATA_ORG_1NF **Database name: Ch05_ConstructCo**

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
15	Evergreen	101	John G. News	Database Designer	105.00	19.4
15	Evergreen	105	Alice K. Johnson *	Database Designer	105.00	35.7
15	Evergreen	106	William Smithfield	Programmer	35.75	12.6
15	Evergreen	102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
18	Amber Wave	118	James J. Frommer	General Support	18.36	45.3
18	Amber Wave	104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	96.75	48.4
22	Rolling Tide	113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	26.87	22.0
22	Rolling Tide	106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
25	Starflight	115	Travis B. Bawangi	Systems Analyst	96.75	45.8
25	Starflight	101	John G. News *	Database Designer	105.00	56.3
25	Starflight	114	Annelise Jones	Applications Designer	48.10	33.1
25	Starflight	108	Ralph B. Washington	Systems Analyst	96.75	23.6
25	Starflight	118	James J. Frommer	General Support	18.36	30.5
25	Starflight	112	Darlene M. Smithson	DSS Analyst	45.95	41.4

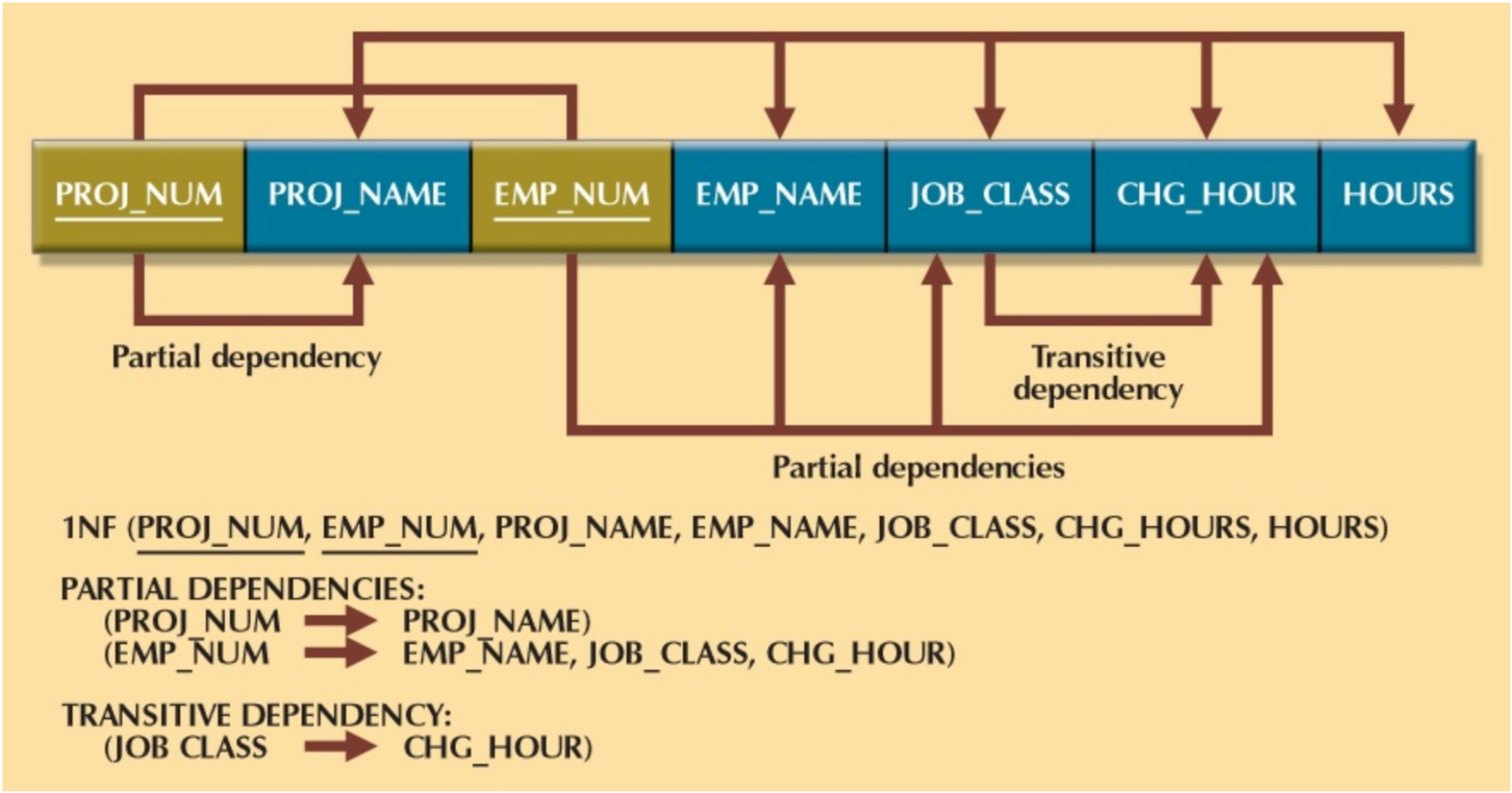
STEPS IN NORMALIZATION



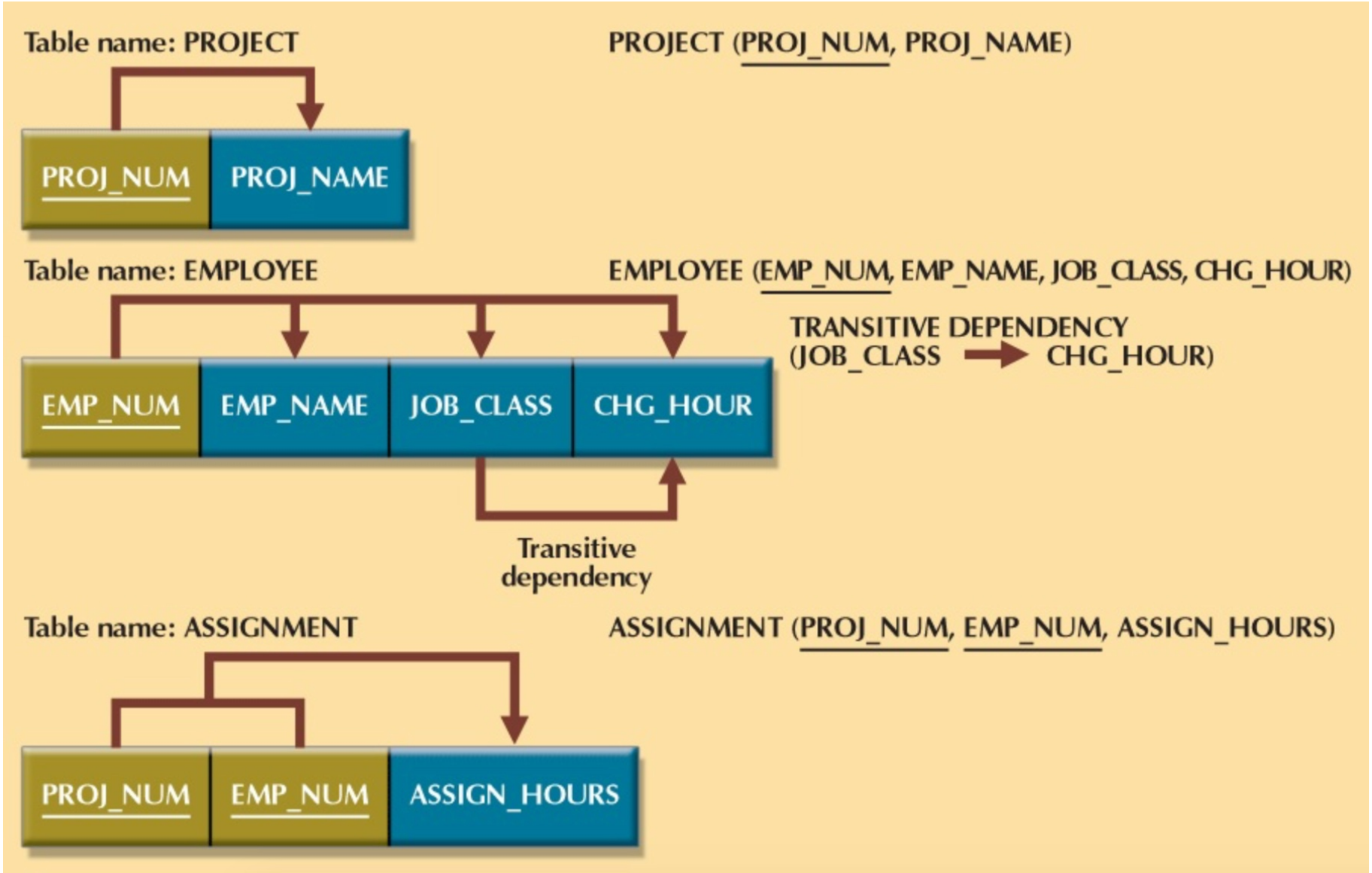
SECOND NORMAL FORM (2NF)

- Convert to 2NF by eliminating partial functional dependencies.
  - Identify partial functional dependencies.
  - For each primary key attribute (or composition of attributes) that is a determinant in a partial dependency create a new relation/table with that attribute(s) as the primary key.
  - Move the non-key attributes that were dependents of that attribute(s) from the old table to the new one.
  - Any remaining non-key attributes stay in the original table with the original primary key.

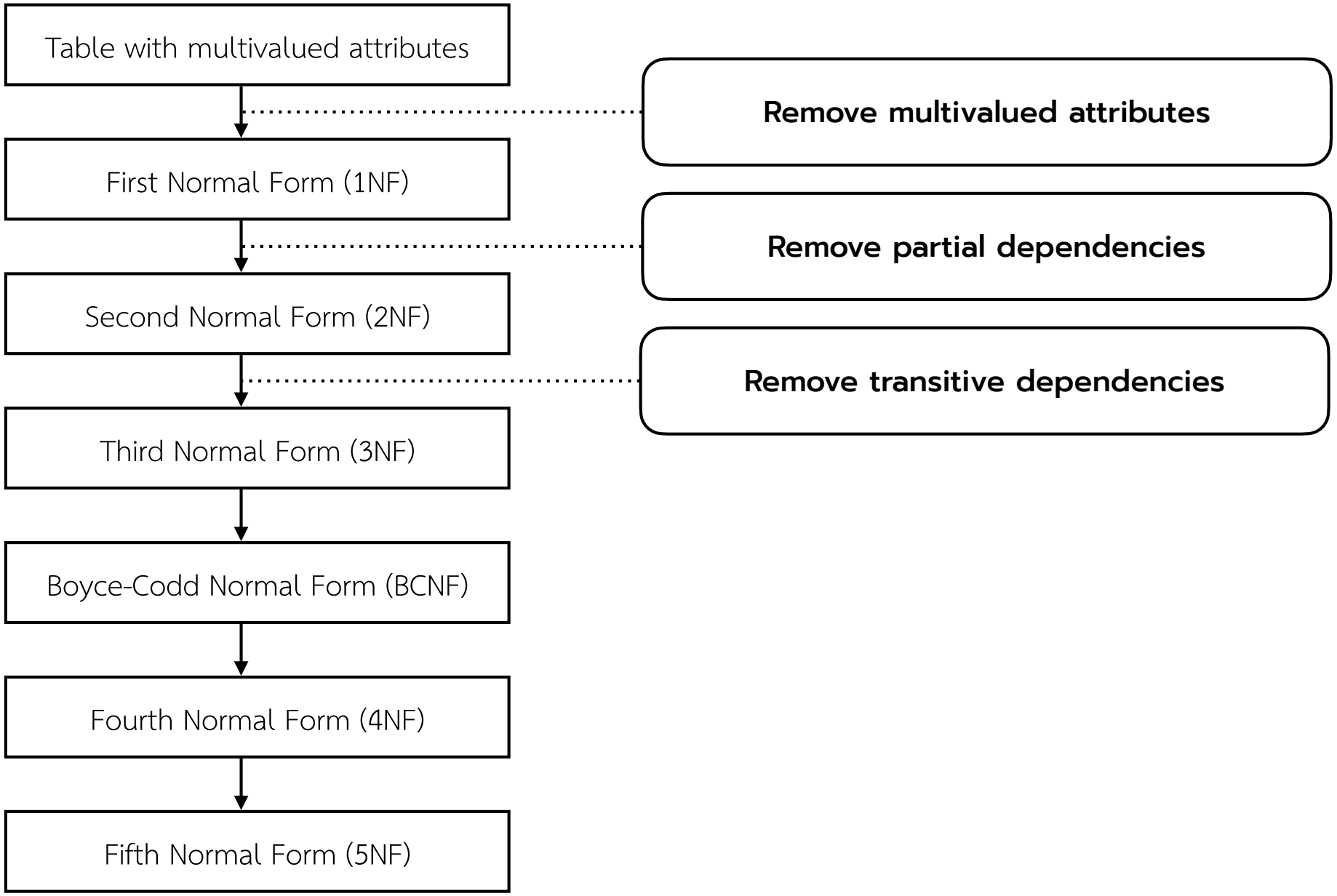
1NF DEPENDENCY DIAGRAM



2NF CONVERSION RESULT



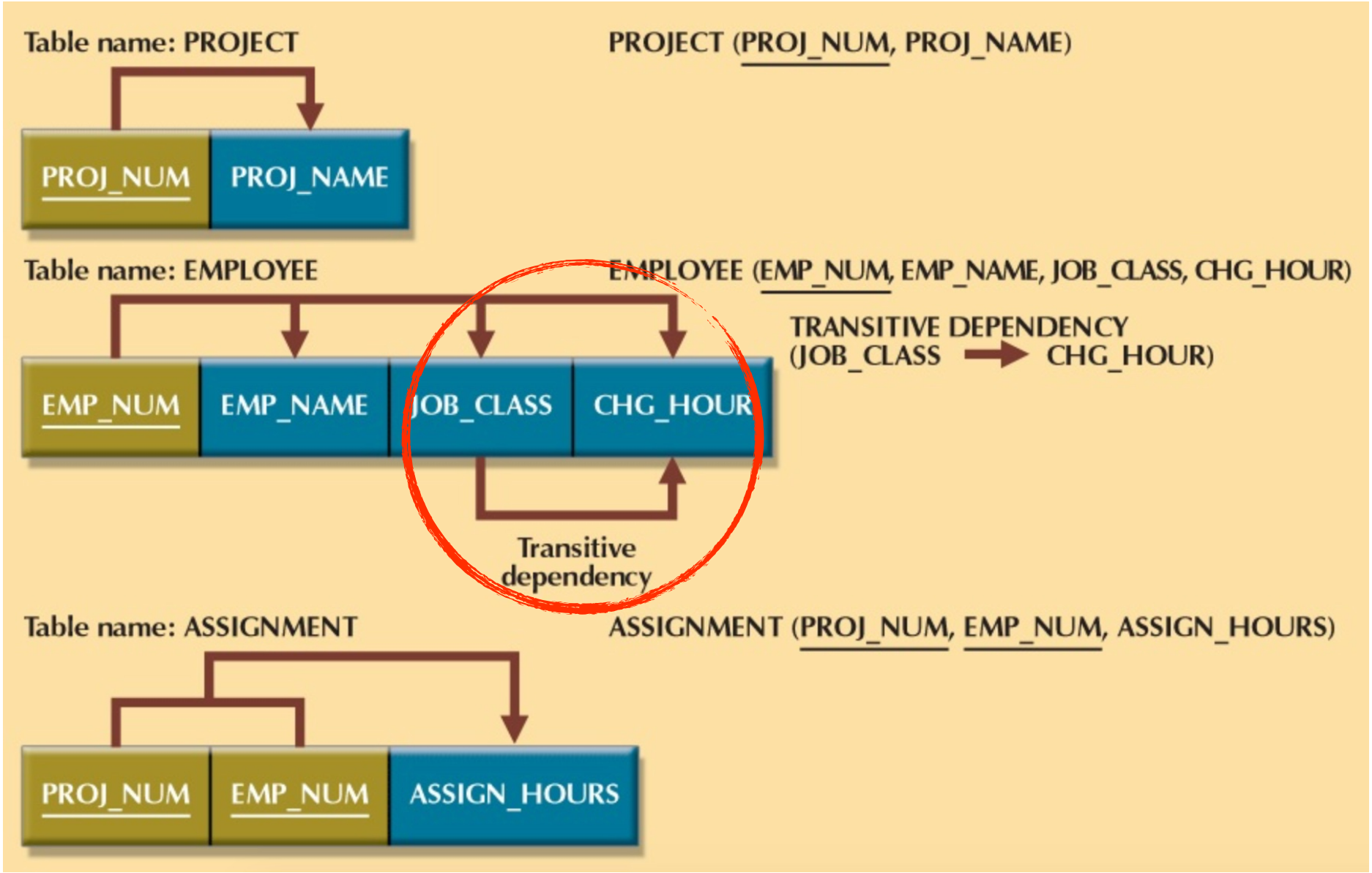
STEPS IN NORMALIZATION



THIRD NORMAL FORM (3NF)

- Convert to 3NF by eliminating transitive dependencies.
  - Identify transitive dependencies in which a non-key attribute (or composition of attributes) determines some other non-key attribute(s).
  - For each non-key attribute(s) that is a determinant of other non-key attribute(s), create a new table in which that determinant attribute(s) is the primary key.
  - Move the non-key attribute(s) that were dependents of that attribute(s) from the old table to the new one.
  - Leave the determinant attribute(s) in the old table as a foreign key linking to the new table.

2NF CONVERSION RESULT



3NF CONVERSION RESULT

Table name: PROJECT

PROJECT (PROJ_NUM, PROJ_NAME)

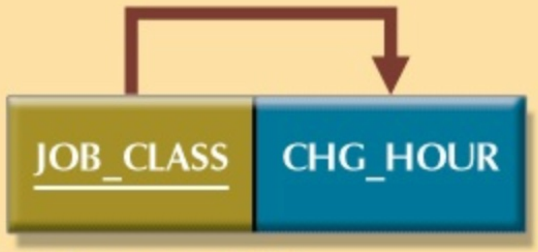


Table name: JOB

JOB (JOB_CLASS, CHG_HOUR)

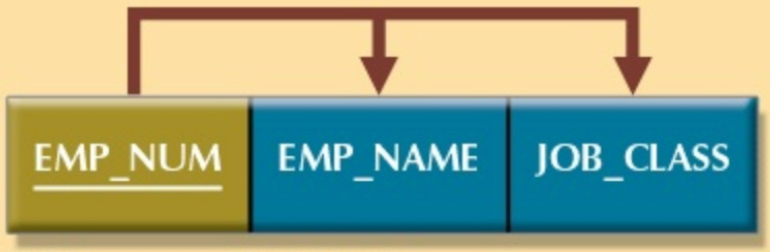


Table name: EMPLOYEE

EMPLOYEE (EMP_NUM, EMP_NAME, JOB_CLASS)

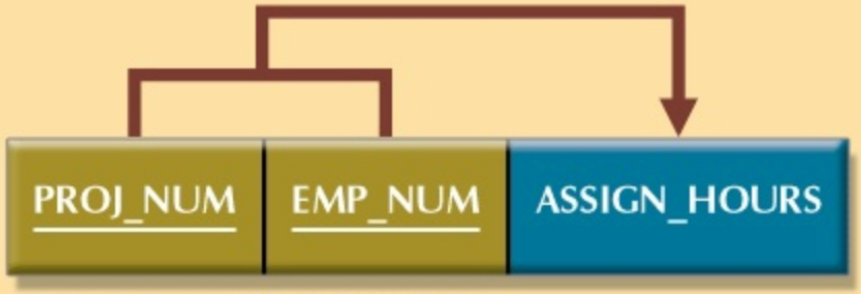


Table name: ASSIGNMENT

ASSIGNMENT (PROJ_NUM, EMP_NUM, ASSIGN_HOURS)

COMPLETED DATABASE (1)

Database name: Ch05_ConstructCo

Table name: PROJECT

PROJ_NUM	PROJ_NAME	EMP_NUM
15	Evergreen	105
18	Amber Wave	104
22	Rolling Tide	113
25	Starflight	101

Table name: JOB

JOB_CODE	JOB_DESCRIPTION	JOB_CHG_HOUR
500	Programmer	35.75
501	Systems Analyst	96.75
502	Database Designer	105.00
503	Electrical Engineer	84.50
504	Mechanical Engineer	67.90
505	Civil Engineer	55.78
506	Clerical Support	26.87
507	DSS Analyst	45.95
508	Applications Designer	48.10
509	Bio Technician	34.55
510	General Support	18.36

COMPLETED DATABASE (2)

Table name: EMPLOYEE

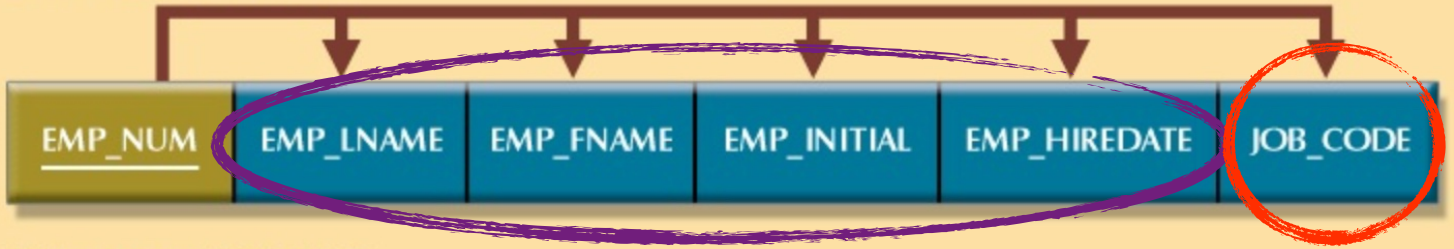


Table name: EMPLOYEE

EMP_NUM	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_HIREDATE	JOB_CODE
101	News	John	G	08-Nov-00	502
102	Senior	David	H	12-Jul-89	501
103	Arbough	June	E	01-Dec-97	503
104	Ramoras	Anne	K	15-Nov-88	501
105	Johnson	Alice	K	01-Feb-94	502
106	Smithfield	William		22-Jun-05	500
107	Alonzo	Maria	D	10-Oct-94	500
108	Washington	Ralph	B	22-Aug-89	501
109	Smith	Larry	W	18-Jul-99	501
110	Olenko	Gerald	A	11-Dec-96	505
111	Wabash	Geoff	B	04-Apr-89	506
112	Smithson	Darlene	M	23-Oct-95	507
113	Joenbrood	Delbert	K	15-Nov-94	508
114	Jones	Annelise		20-Aug-91	508
115	Bawangi	Travis	B	25-Jan-90	501
116	Pratt	Gerald	L	05-Mar-95	510
117	Williamson	Angie	H	19-Jun-94	509
118	Frommer	James	J	04-Jan-06	510

COMPLETED DATABASE (3)

Table name: ASSIGNMENT

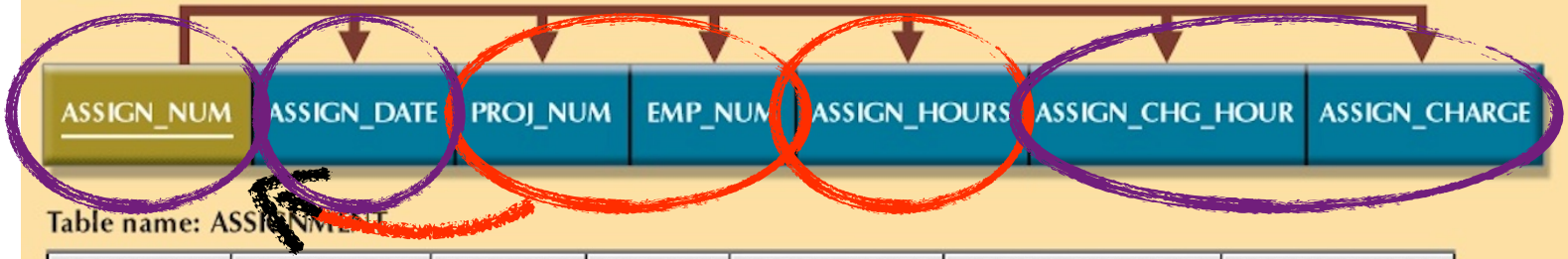
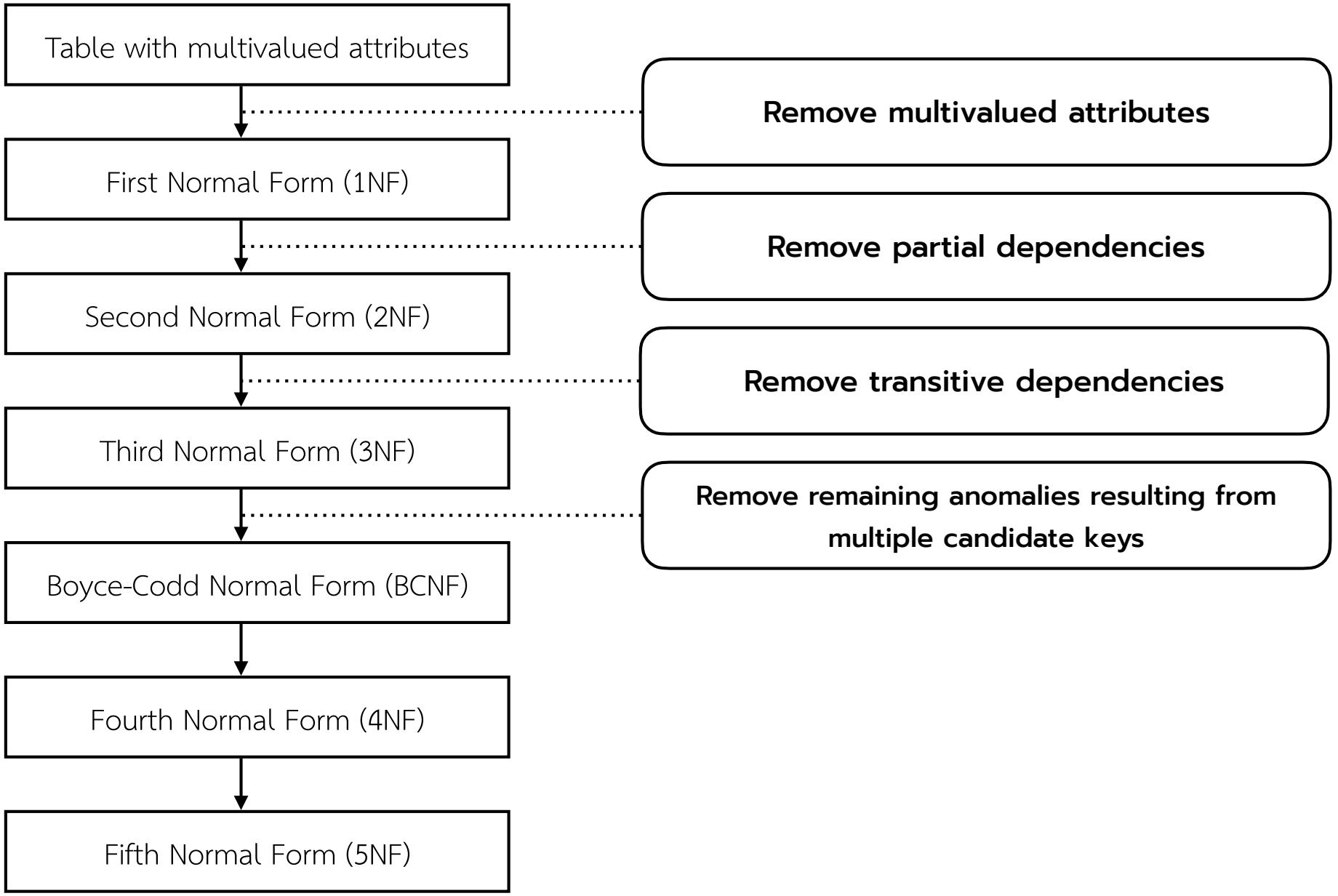


Table name: ASSIGNMENT

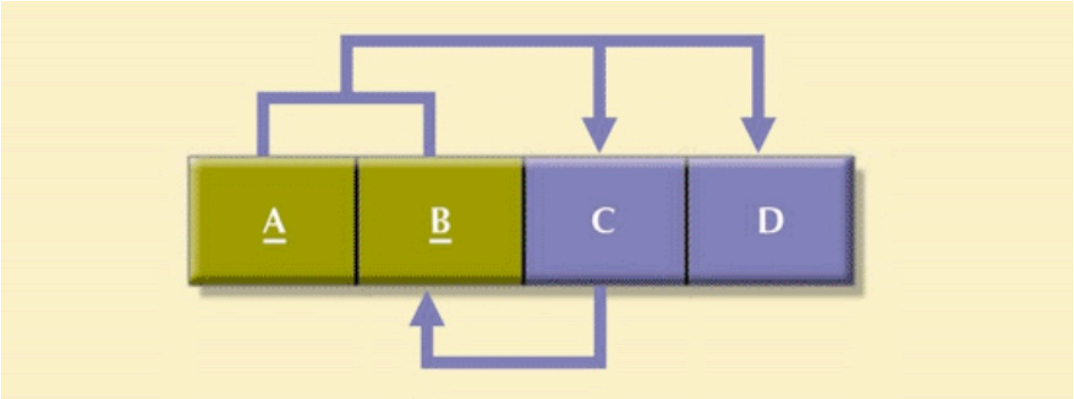
ASSIGN_NUM	ASSIGN_DATE	PROJ_NUM	EMP_NUM	ASSIGN_HOURS	ASSIGN_CHG_HOUR	ASSIGN_CHARGE
1001	04-Mar-08	15	103	2.6	84.50	219.70
1002	04-Mar-08	18	118	1.4	18.36	25.70
1003	05-Mar-08	15	101	3.6	105.00	378.00
1004	05-Mar-08	22	113	2.5	48.10	120.25
1005	05-Mar-08	15	103	1.9	84.50	160.55
1006	05-Mar-08	25	115	4.2	96.75	406.35
1007	05-Mar-08	22	105	5.2	105.00	546.00
1008	05-Mar-08	25	101	1.7	105.00	178.50
1009	05-Mar-08	15	105	2.0	105.00	210.00
1010	06-Mar-08	15	102	3.8	96.75	367.65
1011	06-Mar-08	22	104	2.6	96.75	251.55
1012	06-Mar-08	15	101	2.3	105.00	241.50
1013	06-Mar-08	25	114	1.8	48.10	86.58
1014	06-Mar-08	22	111	4.0	26.87	107.48
1015	06-Mar-08	25	114	3.4	48.10	163.54
1016	06-Mar-08	18	112	1.2	45.95	55.14
1017	06-Mar-08	18	118	2.0	18.36	36.72
1018	06-Mar-08	18	104	2.6	96.75	251.55
1019	06-Mar-08	15	103	3.0	84.50	253.50
1020	07-Mar-08	22	105	2.7	105.00	283.50
1021	08-Mar-08	25	108	4.2	96.75	406.35
1022	07-Mar-08	25	114	5.8	48.10	278.98
1023	07-Mar-08	22	106	2.4	35.75	85.80

STEPS IN NORMALIZATION

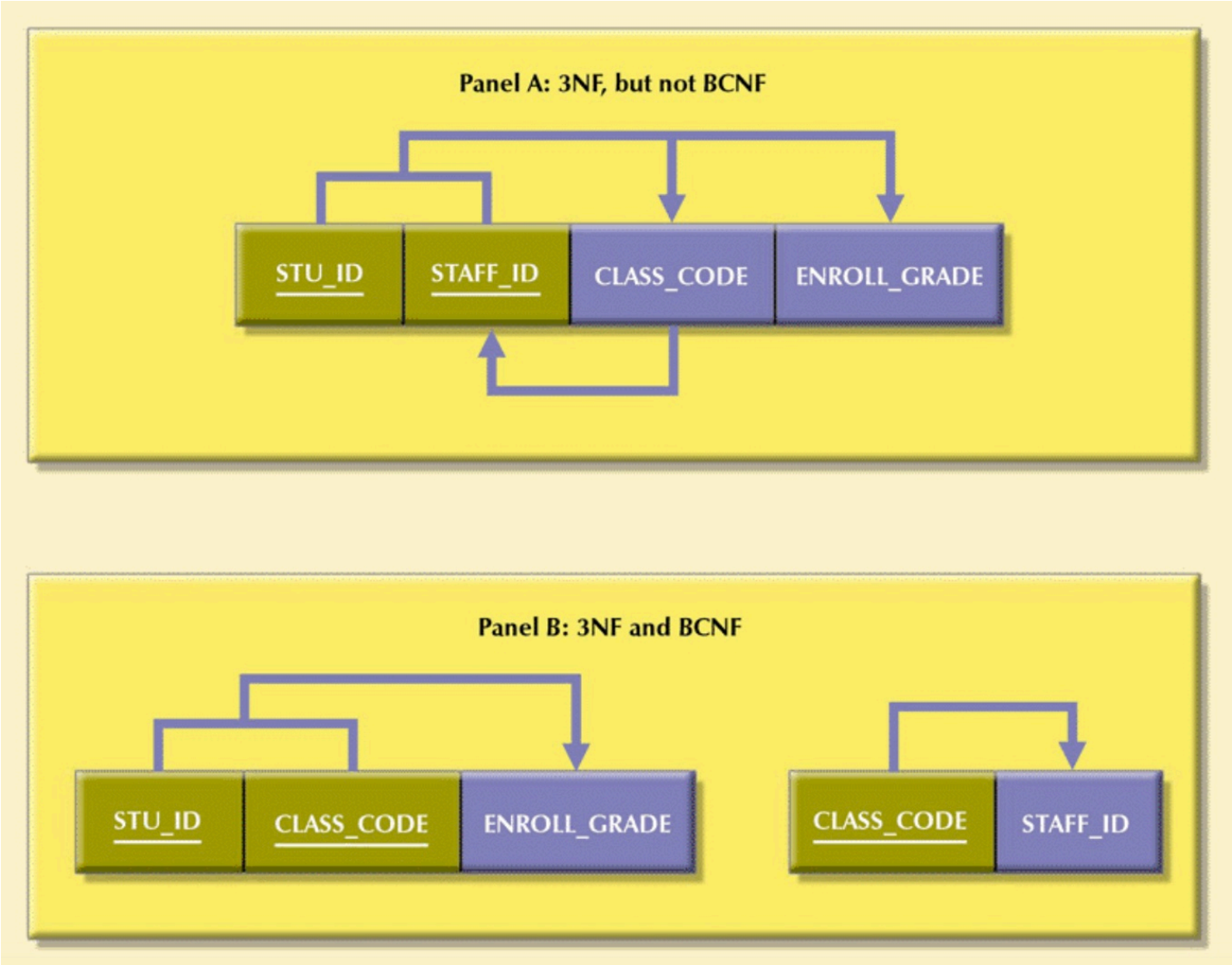


BOYCE-CODD NORMAL FORM (BCNF)

- A table is in Boyce-Codd normal form (BCNF) if every determinant in the table is a candidate key. (A determinant is any attribute whose value determines other values with a row.)
- If a table contains only one candidate key, the 3NF and the BCNF are equivalent.
- BCNF is a special case of 3NF.



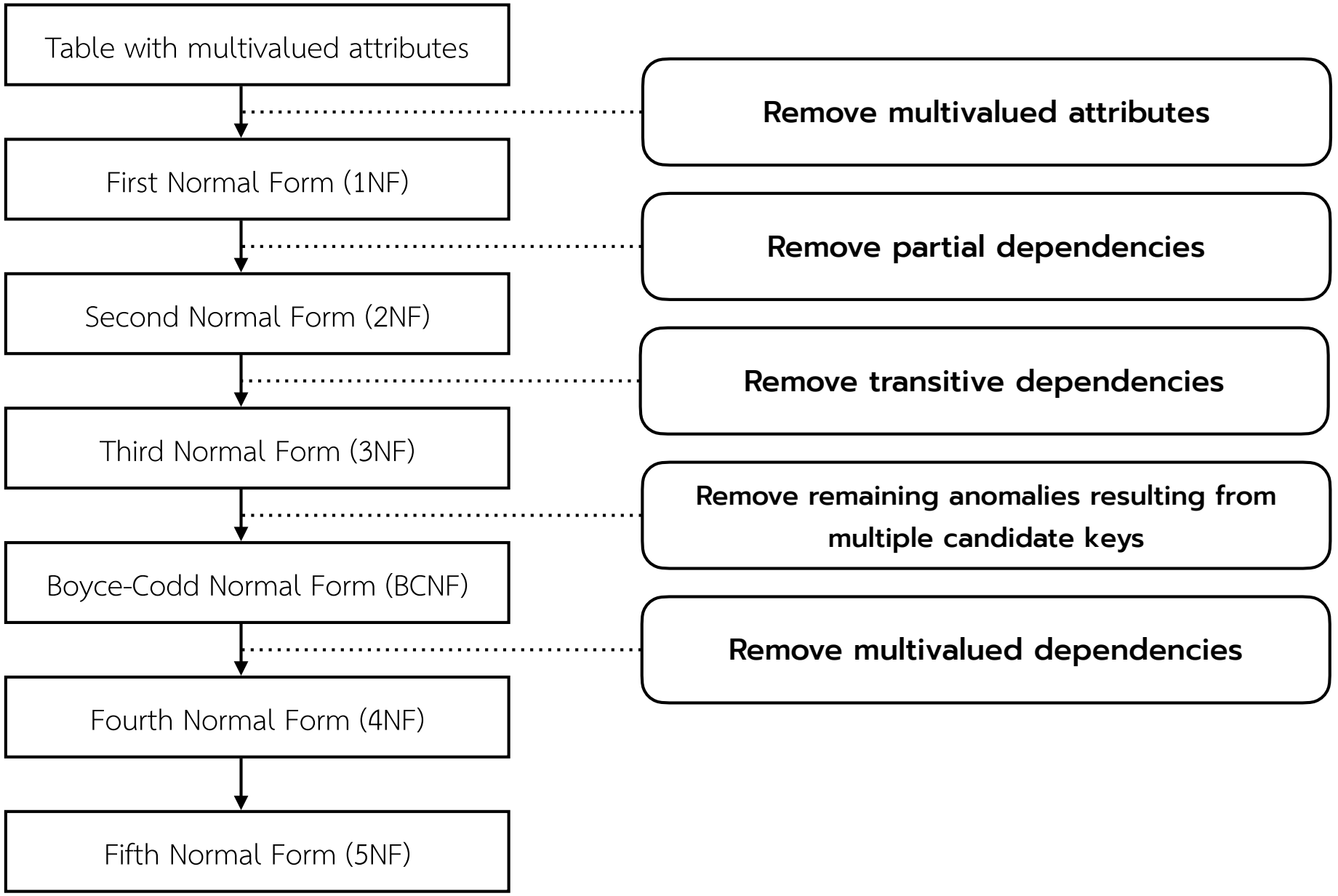
BOYCE-CODD NORMAL FORM (BCNF)



SIMPLE SYNTHESIS PROCEDURE

- Steps of the simple synthesis procedure
  - Eliminate extraneous columns from the LHS of FDs
  - Remove derived FDs from the FD list.
  - Arrange the FDs into groups having the same determinant.
  - For each FD group, make a table with the determinant as the primary key.
  - Merge tables in which on table contains all columns of the order table.
    - Choose the primary key of one of the separate tables as the primary of the new, merged table
    - Define unique constraints for the other primary key that were not designed as the primary key of the new table.

STEPS IN NORMALIZATION



MULTIVALUED DEPENDENCIES

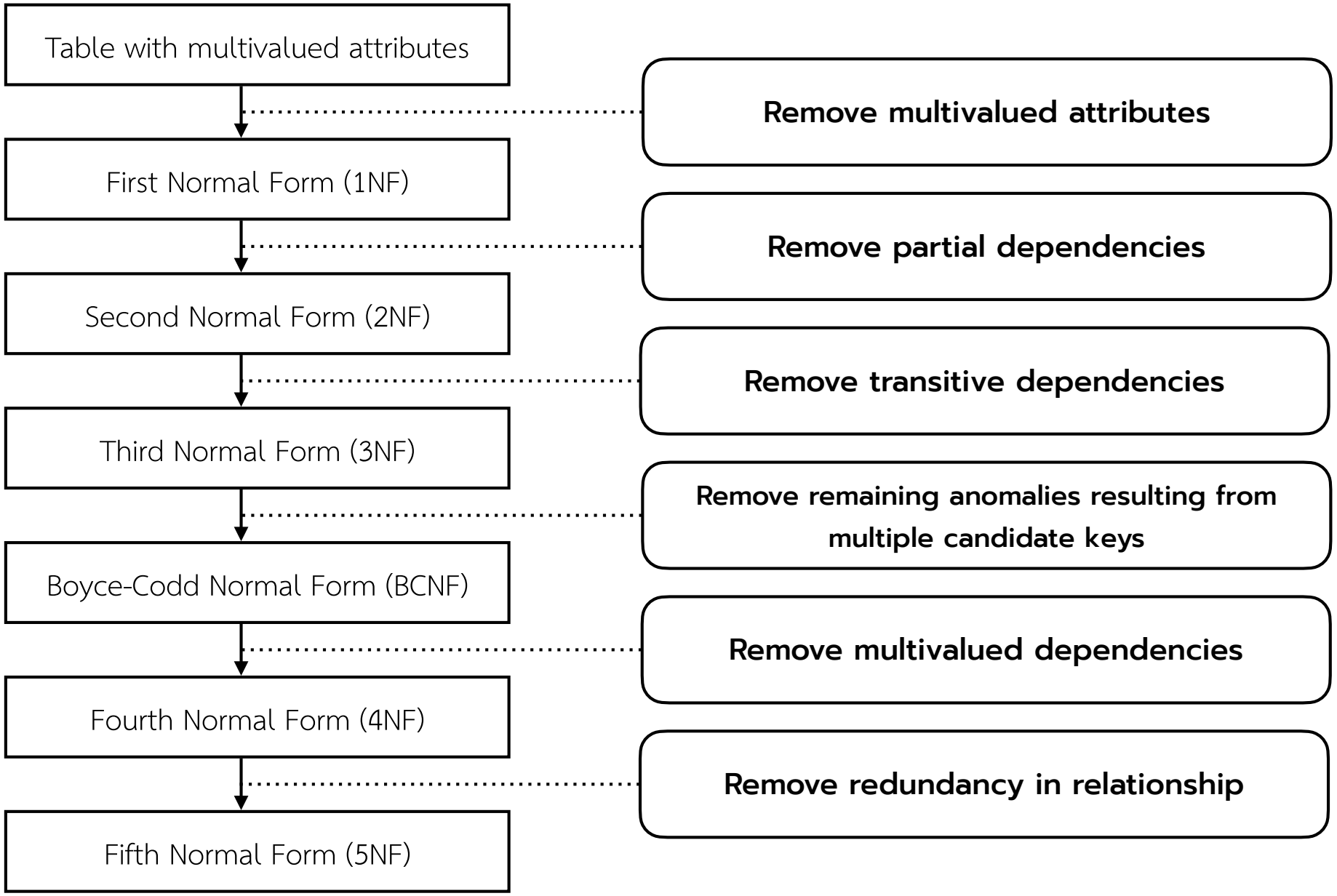
- MVD (Multivalued Dependencies) A →→ B | C (read a multi-determines B or C) means that
  - A given value is associated with a collection of B and C values, and
  - B and C are independent given the relationships between A and B, and A and C.
- Forth Normal Form (4NF) prohibits redundancies caused by multivalued dependencies.

4NF EXAMPLE

- The below is not in 4NF, since
  - More than one movie can have the same listing
  - Many shooting locations can have the same movie

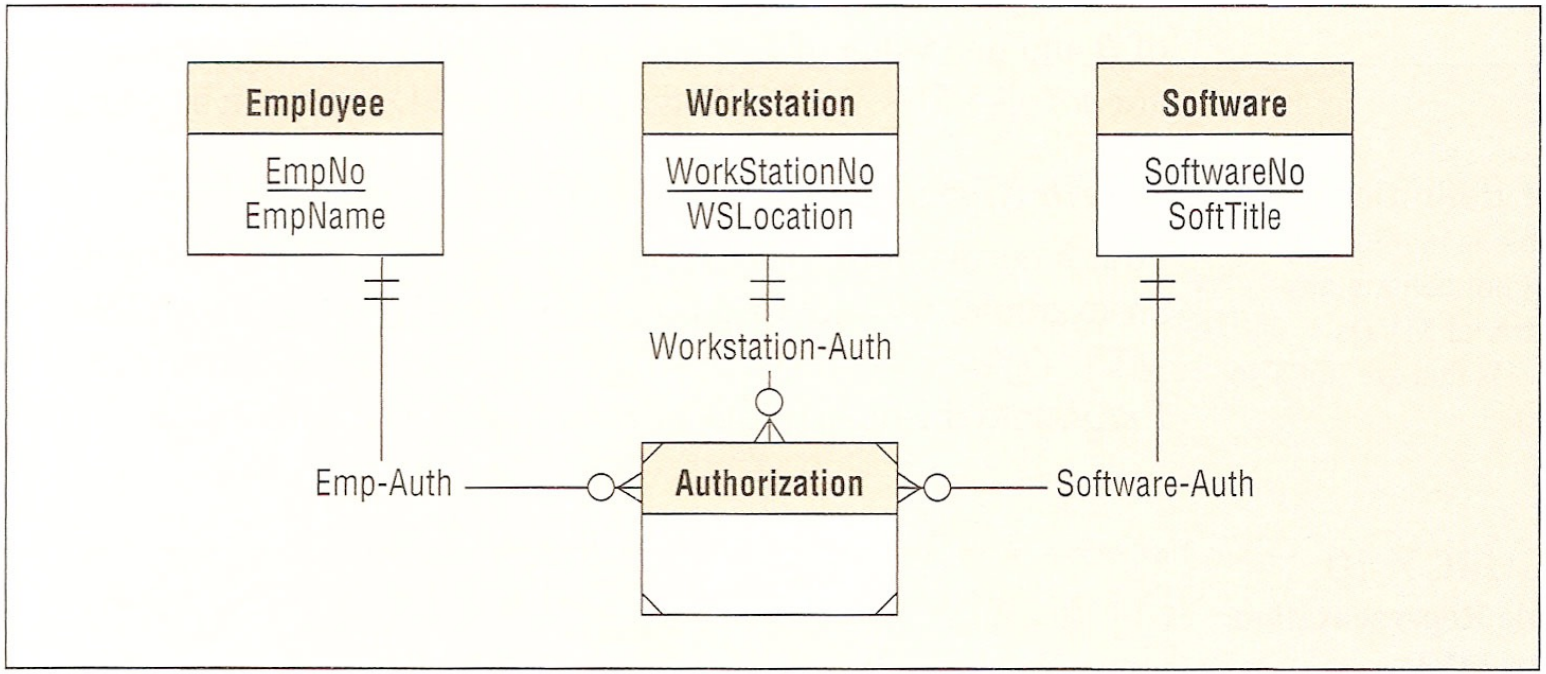
Movie_Name	Shooting_Location	Listing
MovieOne	UK	Comedy
MovieOne	UK	Thriller
MovieTwo	Australia	Action
MovieTwo	Australia	Crime
MovieThree	India	Drama

STEPS IN NORMALIZATION



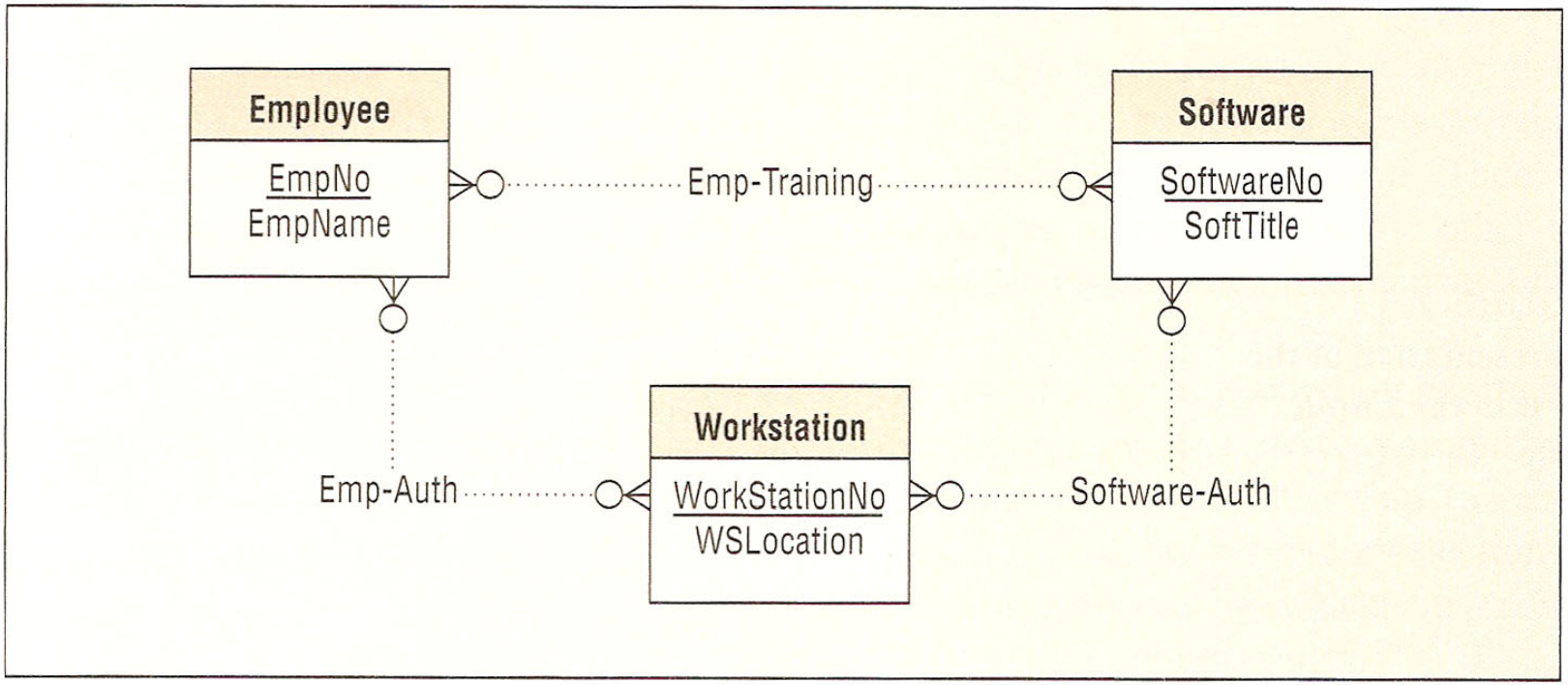
HIGHER LEVEL NORMAL FORMS (1)

- Fifth normal form (5NF) applies to M-way relationships like 4NF. Unlike 4NF, 5NF involves situation when a three-way relationship should be replaced with three binary relationships, not two binary relationships as for 4NF.



HIGHER LEVEL NORMAL FORMS (2)

- Replacement of associative entity type with three binary relationships



SIXTH NORMAL FORM (6NF)

- 6NF, the relation variable is decomposed into irreducible components.
- A relation is in 6NF, only if, It is in 5NF, and every join dependency on the relation is trivial

DOMAIN KEY NORMAL FORM

- After learning about normal form, you may be asking questions such as “Where does it stop?” and “Is there an ultimate normal form?”
- Fortunately, the answer to the last question is YES. In 1981 paper, Dr. Ronald Fagin proposed domain key normal form (DKNF) as the ultimate normal form.

OVER NORMALIZATION

- In practice, normalization to fifth Normal Form is unusual. Normally, the Database designer reaches for his hat and coat after reaching Boyce/Codd Normal Form (BCNF), which is 4NF, and few databases you have ever seen are even reliably BCNF.
- When you normalize, you are adding more tables to your database, and you are doing so many JOIN to retrieve data that it is causing notable performance penalties and deadlocks on your database.

DENORMALIZATION

- Denormalization is a strategy that database managers use to increase the performance of a database infrastructure. It involves adding redundant data to a normalized database to reduce certain types of problems with database queries that combine data from various tables into a single table.
- The definition of denormalization is dependent on the definition of normalization, which is defined as the process of organizing a database into tables correctly to promote a given use.

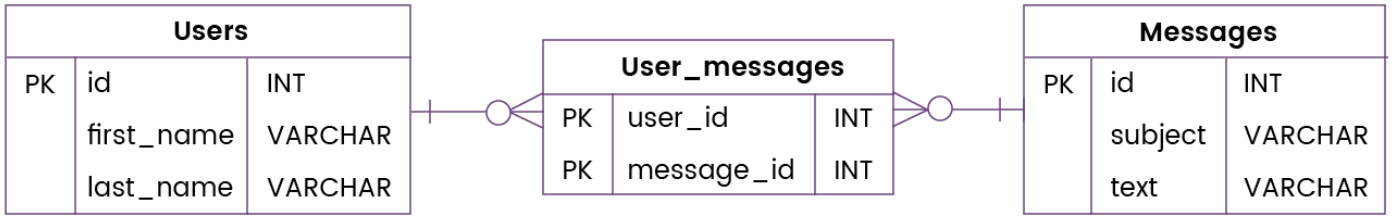
DATABASE DENORMALIZATION TECHNIQUES

- Storing derivable data
- Using pre-joined tables
- Using hardcoded values
- Keeping details with the master
- Repeating detail with master
- Adding short-circuit keys

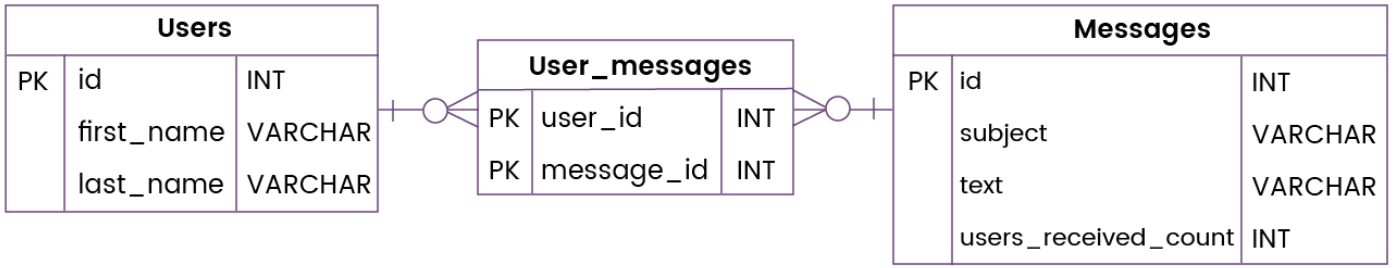
STORING DERIVABLE DATA

- Storing frequently executed calculations is worthwhile in situations where:
  - The use of the derived value is frequent.
  - The source values do not change.

Normalized database



Denormalized database



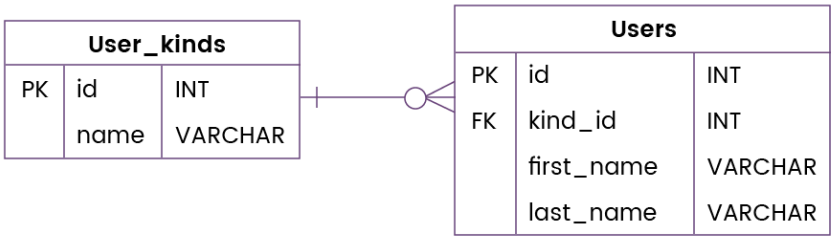
PRE-JOINED TABLES

- Pre-joined tables store the frequently used pieces of information together into one table. The process comes in handy when:
  - Queries frequently execute on the tables together.
  - The join operation is costly.

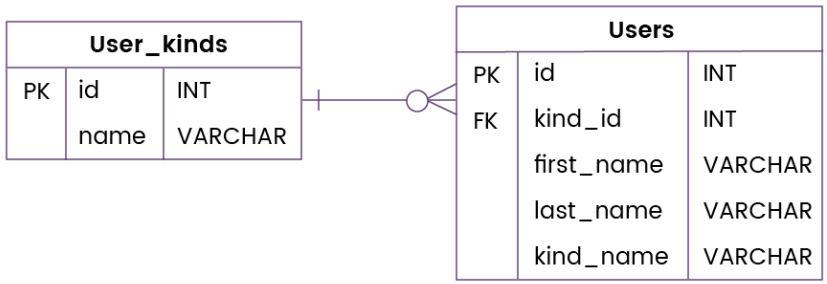
USING HARDCODED VALUES

- Hard-coded values remove a reference to a commonly used entity. Use this method in situations where:
  - The values are considered static.
  - The number of values is small.

Normalized database

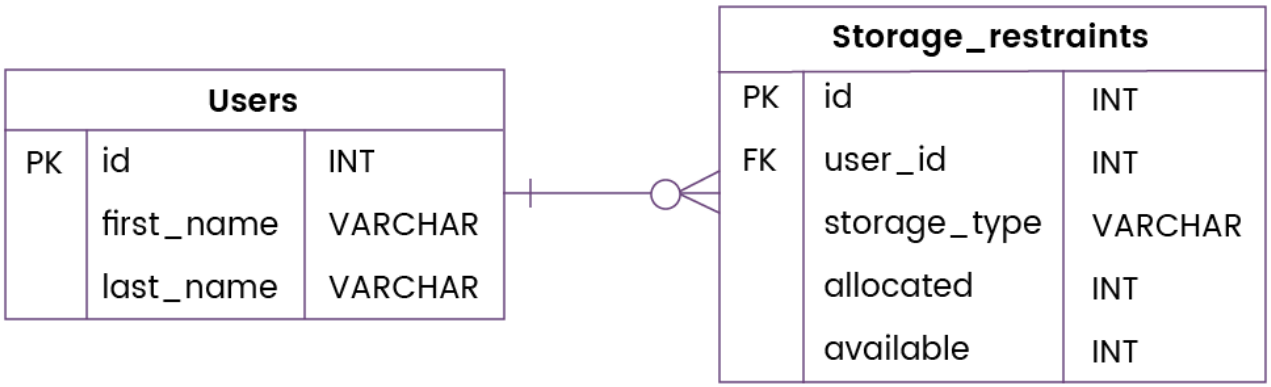


Denormalized database

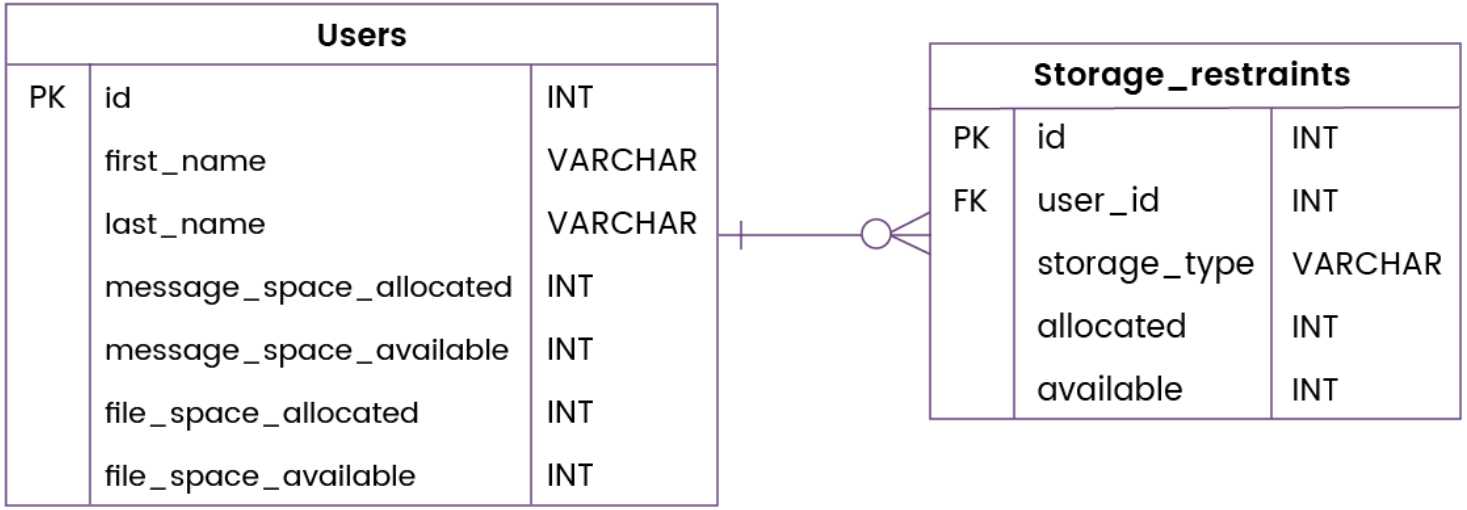


KEEPING DETAILS WITH THE MASTER

Normalized database



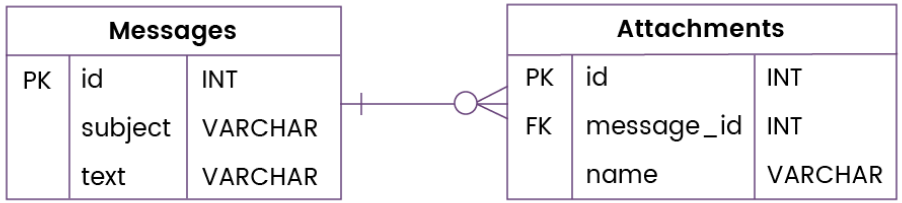
Denormalized database



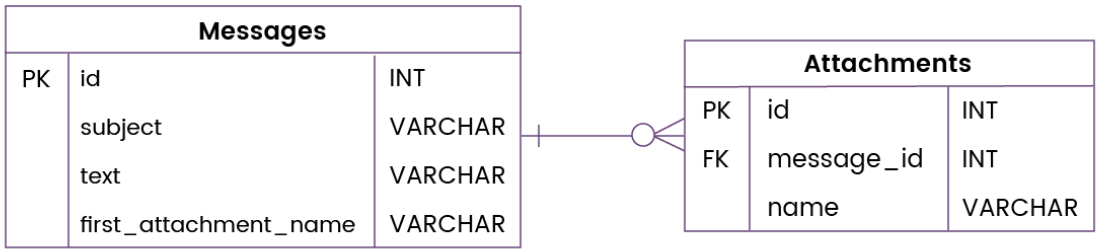
REPEATING DETAIL WITH MASTER

- Queries often need just a single detail added to the master table instead of pre-joining multiple values. Use this method when:
  - JOINs are costly for a single detail.
  - The master table requires the information often.

Normalized database



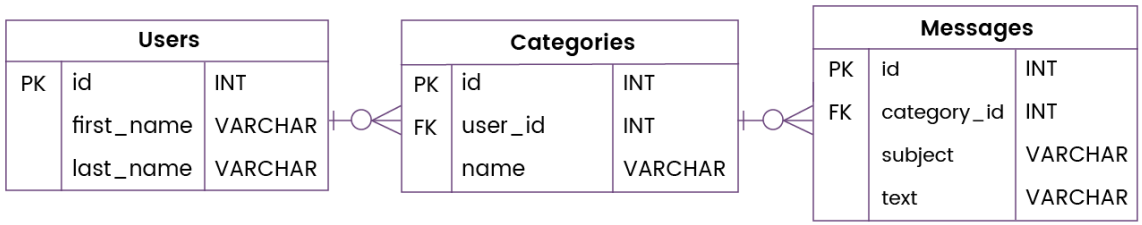
Denormalized database



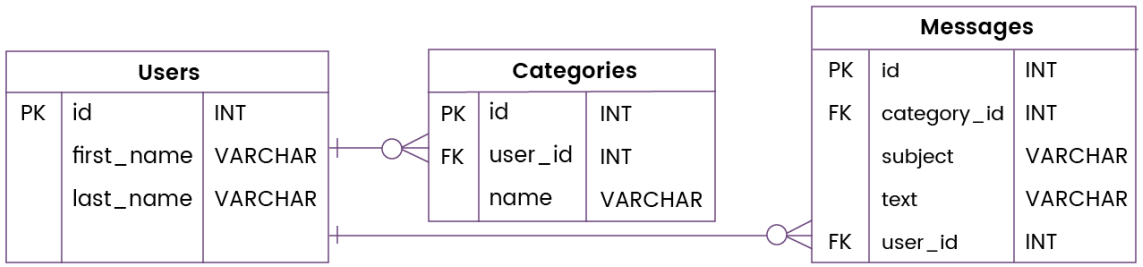
ADDING SHORT-CIRCUIT KEYS

- In a database with three or more tables of related information, the short-circuit keys method skips the middle table(s) and "short-circuits" the grandparent and grandchild tables.
  - A database has over three levels of master-detail.
  - The values from the grandparent and grandchild are often needed and the parent information is not as valuable.

Normalized database



Denormalized database



DENORMALIZE ADVANTAGES

- **Speed.** Since JOIN queries are costly on a normalized database, retrieving data is faster.
- **Simplicity.** Fetching data is more straightforward because of the smaller number of tables.
- **Fewer errors.** Working with a smaller number of tables means fewer bugs when retrieving information from a database.

DENORMALIZE DISADVANTAGES

- **Complexity.** Updating and inserting into a database is more complex and costly.
- **Inconsistency.** Finding the correct value for a piece of information is more challenging because the data is hard to update.
- **Storage.** More significant storage space is necessary due to introduced redundancies.