

Normalization II

CS 341 Database Systems

Normalization Forms

Normal Form	Characteristic
First normal form (1NF)	Table format, no repeating groups, and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	3NF and every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	BCNF and no independent multivalued dependencies
Fifth normal form (5NF or PJNF)	4NF and cannot have lossless decomposition into smaller tables

Normalize up to 3NF

DreamHome Lease

DreamHome Lease

DreamHome Lease

DreamHome Lease

Client Number CR76
(Enter if known)

Full Name John Kay
(Please print)

Property Number PG4

Property Address 6 Lawrence St, Glasgow

Monthly Rent 350

Rent Start 01/07/12

Rent Finish 31/08/13

Owner Number CO40
(Enter if known)

Full Name Tina Murphy
(Please print)

Figure 14.9 Collection of (simplified) DreamHome leases.

ClientRental

clientNo	cName	propertyNo	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	John Kay	PG4	6 Lawrence St, Glasgow	1-Jul-12	31-Aug-13	350	CO40	Tina Murphy
		PG16	5 Novar Dr, Glasgow	1-Sep-13	1-Sep-14	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence St, Glasgow	1-Sep-11	10-June-12	350	CO40	Tina Murphy
		PG36	2 Manor Rd, Glasgow	10-Oct-12	1-Dec-13	375	CO93	Tony Shaw
		PG16	5 Novar Dr, Glasgow	1-Nov-14	10-Aug-15	450	CO93	Tony Shaw

1NF

ClientRental

clientNo	propertyNo	cName	pAddress	rentStart	rentFinish	rent	ownerNo	oName
CR76	PG4	John Kay	6 Lawrence St, Glasgow	1-Jul-12	31-Aug-13	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Dr, Glasgow	1-Sep-13	1-Sep-14	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence St, Glasgow	1-Sep-11	10-Jun-12	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Rd, Glasgow	10-Oct-12	1-Dec-13	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Dr, Glasgow	1-Nov-14	10-Aug-15	450	CO93	Tony Shaw

Functional Dependencies

ClientRental

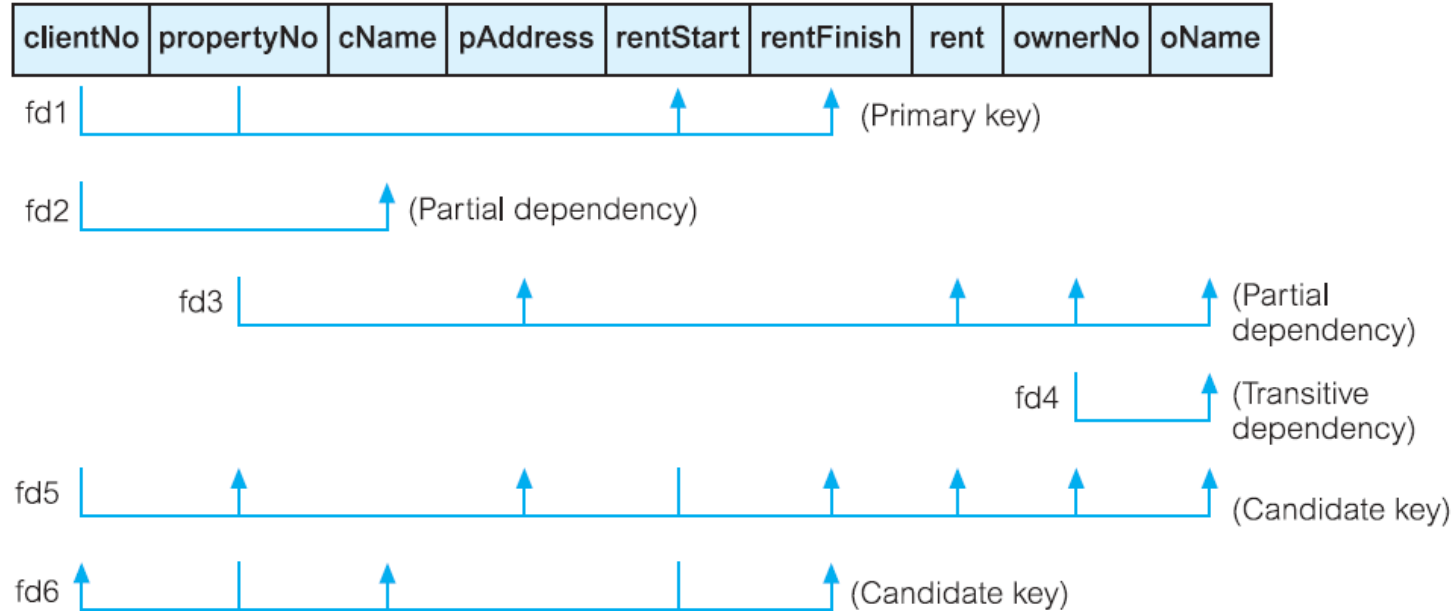


Figure 14.12 Functional dependencies of the ClientRental relation.

The ClientRental relation is defined as follows:

ClientRental (clientNo, propertyNo, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

Identifying Dependencies

- 1NF

ClientRental (clientNo, propertyNo, cName, pAddress, rentStart, rentFinish, rent, ownerNo, oName)

Partial dependencies

clientNo \rightarrow cName

propertyNo \rightarrow pAddress, rent, ownerNo, oName

Transitive dependencies

ownerNo \rightarrow oName

Converting to 2 NF

Partial dependencies

clientNo \rightarrow cName

propertyNo \rightarrow pAddress, rent, ownerNo, oName

Rental (clientNo, propertyNo, rentStart, rentFinish)

Client (clientNo, cName)

PropertyOwner (propertyNo, pAddress, rent, ownerNo, oName)

2NF



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-12	31-Aug-13
CR76	PG16	1-Sep-13	1-Sep-14
CR56	PG4	1-Sep-11	10-Jun-12
CR56	PG36	10-Oct-12	1-Dec-13
CR56	PG16	1-Nov-14	10-Aug-15

PropertyOwner

propertyNo	pAddress	rent	ownerNo	oName
PG4	6 Lawrence St, Glasgow	350	CO40	Tina Murphy
PG16	5 Novar Dr, Glasgow	450	CO93	Tony Shaw
PG36	2 Manor Rd, Glasgow	375	CO93	Tony Shaw

Figure 14.14 Second normal form relations derived from the ClientRental relation.

Converting to 3NF

Transitive dependencies

ownerNo \rightarrow Oname

Owner (ownerNo, oName)

PropertyForRent (propertyNo, pAddress, rent, ownerNo)

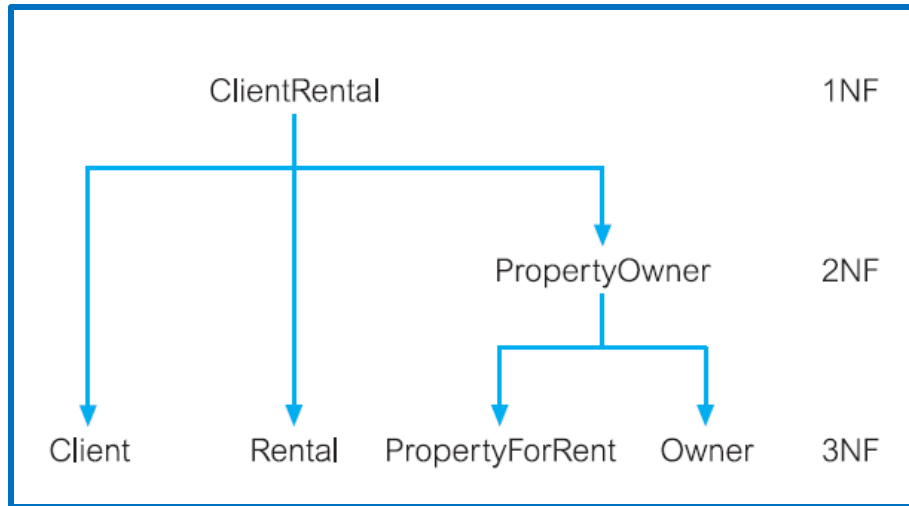
Rental (clientNo, propertyNo, rentStart, rentFinish)

Client (clientNo, cName)

3NF

Figure 14.17

A summary of the 3NF relations derived from the ClientRental relation.



Client

clientNo	cName
CR76	John Kay
CR56	Aline Stewart

Rental

clientNo	propertyNo	rentStart	rentFinish
CR76	PG4	1-Jul-12	31-Aug-13
CR76	PG16	1-Sep-13	1-Sep-14
CR56	PG4	1-Sep-11	10-Jun-12
CR56	PG36	10-Oct-12	1-Dec-13
CR56	PG16	1-Nov-14	10-Aug-15

PropertyForRent

propertyNo	pAddress	rent	ownerNo
PG4	6 Lawrence St, Glasgow	350	CO40
PG16	5 Novar Dr, Glasgow	450	CO93
PG36	2 Manor Rd, Glasgow	375	CO93

Owner

ownerNo	oName
CO40	Tina Murphy
CO93	Tony Shaw

A short, thick, dark red vertical bar is positioned to the left of the title text.

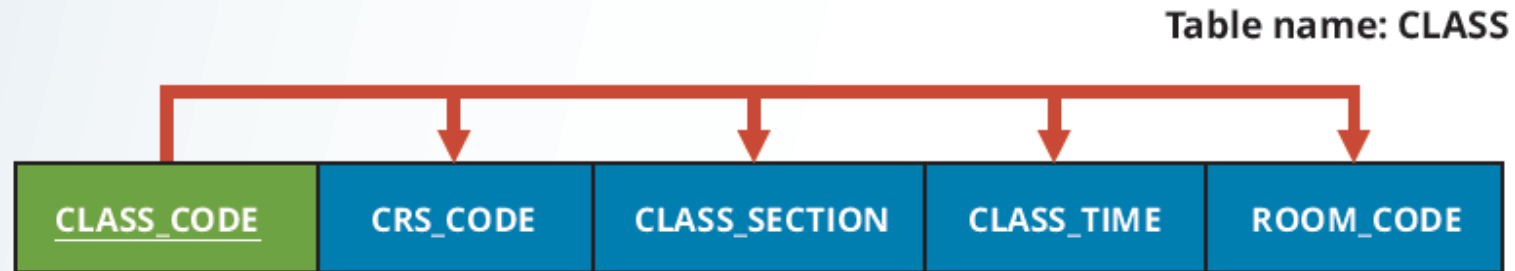
Higher-Level Normal Forms



Boyce-Codd Normal Form (BCNF)

- *Every determinant in the table is a candidate key*
 - Determinant is attribute whose value determines other values in row
 - 3NF table with one candidate key is already in BCNF

Figure 6.7 Tables with Multiple Candidate Keys

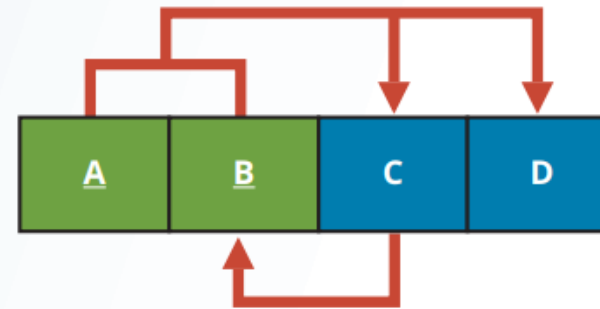


- The CLASS table has two candidate keys:
 - CLASS_CODE
 - CRS_CODE + CLASS_SECTION
- **1NF:** Key attributes defined, all non-key attributes are determined by the key. (True for both candidate keys).
- **2NF:** 1NF and no partial dependencies on either candidate key.
- **3NF:** 2NF and no transitive dependencies.

3NF Table Not in BCNF

- Note these functional dependencies
- $A + B \rightarrow C, D$
- $A + C \rightarrow B, D$
- $C \rightarrow B$
- Candidate Keys: $(A+B)$ and $(A+C)$

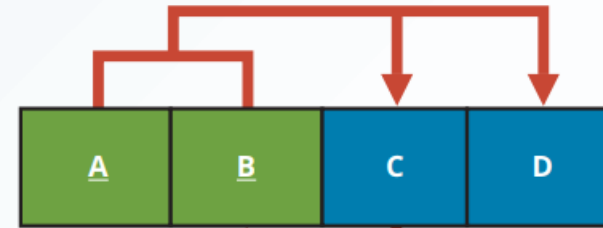
Figure 6.8 A Table That Is in 3NF but not in BCNF



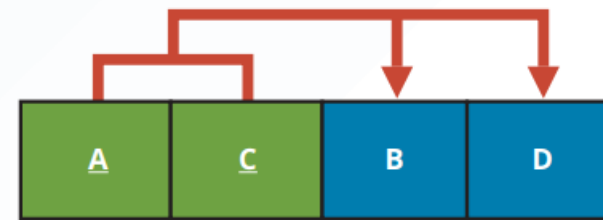
Decomposition of Table Structure to Meet BCNF

Figure 6.9 Decomposition to BCNF

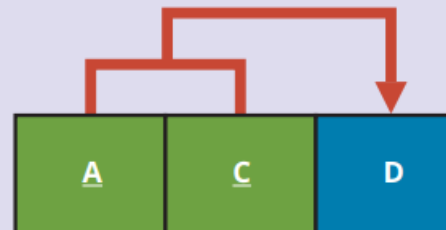
3NF, but not BCNF



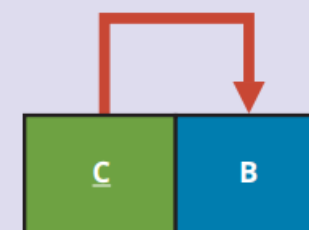
1NF



Partial dependency



3NF and BCNF



3NF and BCNF

Scenario Conditions

- CLASS_CODE identifies a class uniquely. This condition illustrates the case in which a course might generate many classes for example same course has more sections.
- A student can take many classes and earn grades.
- A staff member can teach many classes, but each class is taught by only one staff member.

STU_ID + CLASS_CODE (Candidate Key)

STU_ID + STAFF_ID (Candidate Key)

Table 6.5 Sample Data for a BCNF Conversion

Stu_ID	Staff_ID	Class_code	Enroll_grade
125	25	21334	A
125	20	32456	C
135	20	28458	B
144	25	27563	C
144	20	32456	B

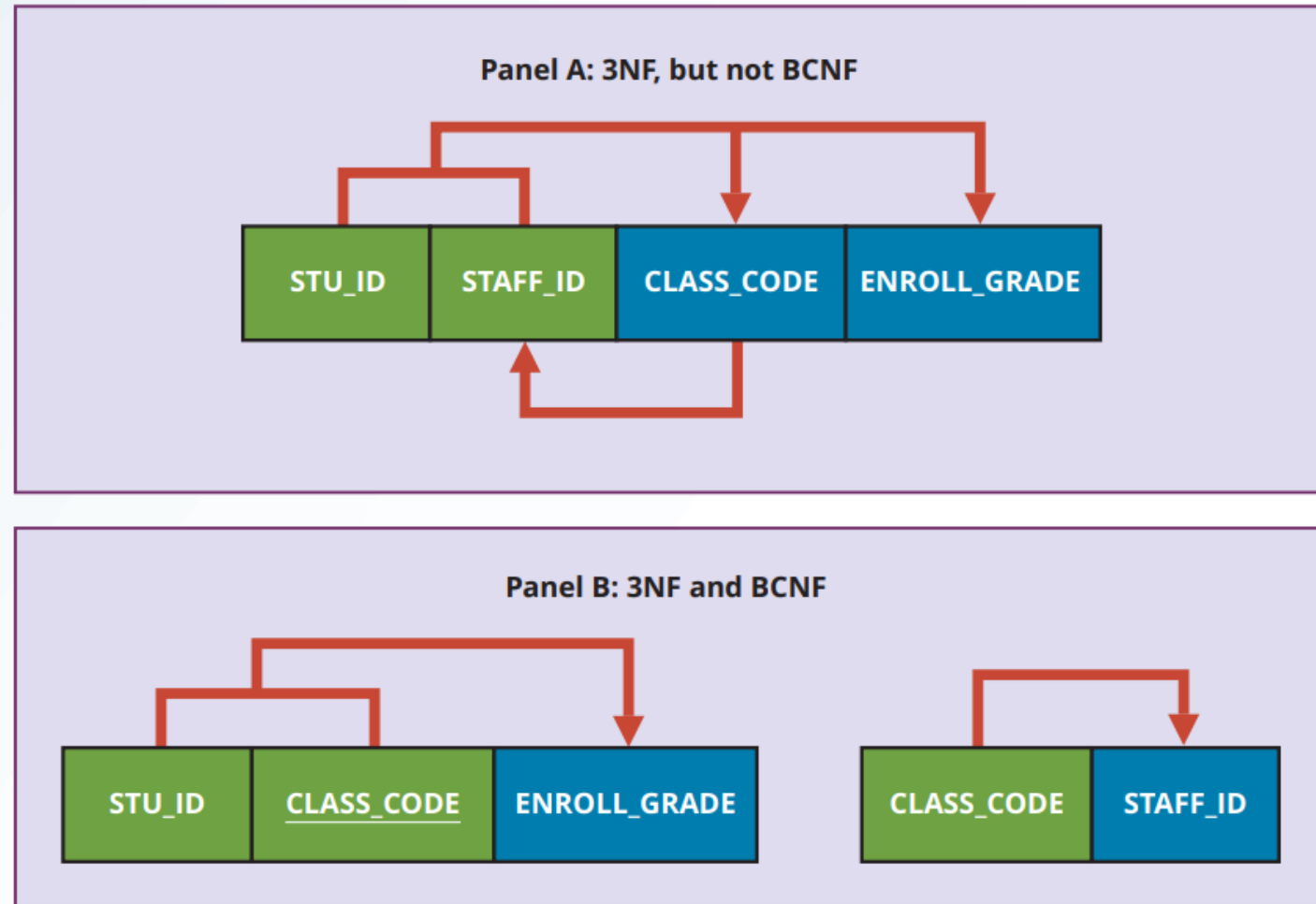
STU_ID + STAFF_ID \rightarrow CLASS_CODE, ENROLL_GRADE

CLASS_CODE \rightarrow STAFF_ID

Anomalies

- The problem: *Trying to describe two things:
staff assignments to classes and student enrollment information.*
- Such a dual-purpose table structure will cause **anomalies**.
 - For example, if a different staff member is assigned to teach class 32456, two rows will require updates, thus producing an update anomaly.
 - If student 135 drops class 28458, information about who taught that class is lost, thus producing a deletion anomaly.
- *The solution to the problem is to decompose the table structure*

Figure 6.10 Another BCNF Decomposition



Fourth Normal Form (4NF)

- *A relation will be in 4NF if it is in Boyce Codd Normal Form (BCNF) and has no multi-valued dependency.*
- For a dependency $A \twoheadrightarrow B$, if for a single value of A, multiple values of B exists, then the relation will be a multi-valued dependency.

Fourth Normal Form (4NF)

- Suppose an employee can have multiple *assignments* and can also be involved in multiple *service* organizations.
- Suppose:
 - Employee 10123 does volunteer work for the Red Cross and United Way.
 - The same employee might be assigned to work on three projects: 1, 3, and 4.

Table name: VOLUNTEER_V1

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	UW	3
10123		4

Table name: VOLUNTEER_V3

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	1
10123	RC	3
10123	UW	4

Table name: VOLUNTEER_V2

EMP_NUM	ORG_CODE	ASSIGN_NUM
10123	RC	
10123	UW	
10123		1
10123		3
10123		4

Rules for 4NF

If you follow proper design rules, you will not have such problems.

Discussion on 4NF is largely academic if your tables conform to the following rules.

1. All attributes must be dependent on the primary key, but they must be independent of each other.
2. No row may contain two or more multivalued facts about an entity.

Conversion to 4 NF

- The solution is to eliminate the problems caused by the *multivalued dependency*.
- You do this by creating **new tables** for the components of the multivalued dependency.

Figure 6.12 A Set of Tables in 4NF



Database name: CH06_Service

Table name: PROJECT

PROJ_CODE	PROJ_NAME	PROJ_BUDGET
1	BeThere	1023245.00
2	BlueMoon	20198608.00
3	GreenThumb	3234456.00
4	GoFast	5674000.00
5	GoSlow	1002500.00

Table name: EMPLOYEE

EMP_NUM	EMP_LNAME
10121	Rogers
10122	O'Leery
10123	Panera
10124	Johnson

Table name: ORGANIZATION

ORG_CODE	ORG_NAME
RC	Red Cross
UW	United Way
WF	Wildlife Fund

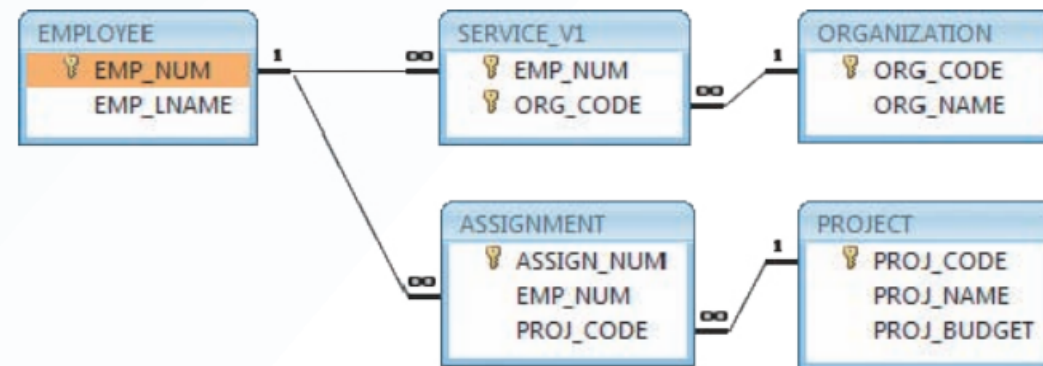
Table name: ASSIGNMENT

ASSIGN_NUM	EMP_NUM	PROJ_CODE
1	10123	1
2	10121	2
3	10123	3
4	10123	4
5	10121	1
6	10124	2
7	10124	3
8	10124	5

Table name: SERVICE_V1

EMP_NUM	ORG_CODE
10123	RC
10123	UW
10123	WF

The relational diagram



The multivalued dependency is resolved and eliminated by creating the ASSIGNMENT and SERVICE_V1 tables. Those tables are said to be in 4NF.

5NF - Fifth Normal Form

- **Fifth normal form (5NF)**, also known as **project join normal form (PJNF)**, addresses the issue in which a table cannot be decomposed anymore without losing data or creating incorrect information.
- In other words, a table is in 4NF and cannot have further **lossless** decompositions.
- Reduce 4NF to 5 NF, by removing pairwise cyclic dependencies (appearing within composite primary keys with three or more component attributes) to three or more parent entities

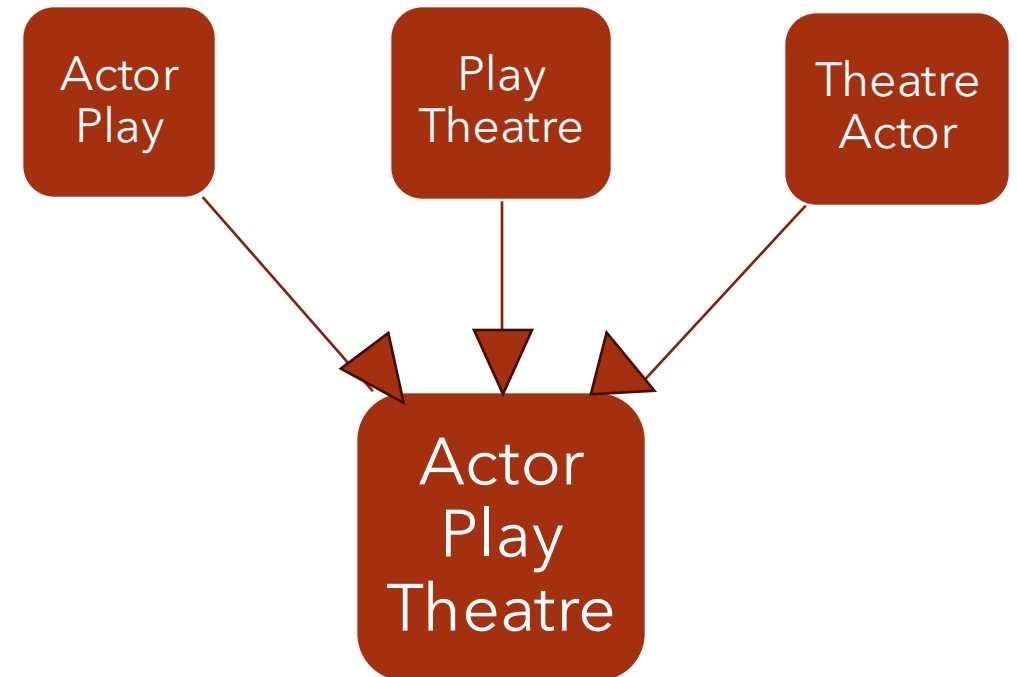
5 NF

- This addresses problems that arise from representing associations between multiple entities with interdependencies.
- *Making it 5NF consists of adding parent tables, one for each meaningful combination that has children in the original table.*
- A table with such information is 5NF if the information cannot be represented in multiple smaller entities alone.

5 NF - EXAMPLE

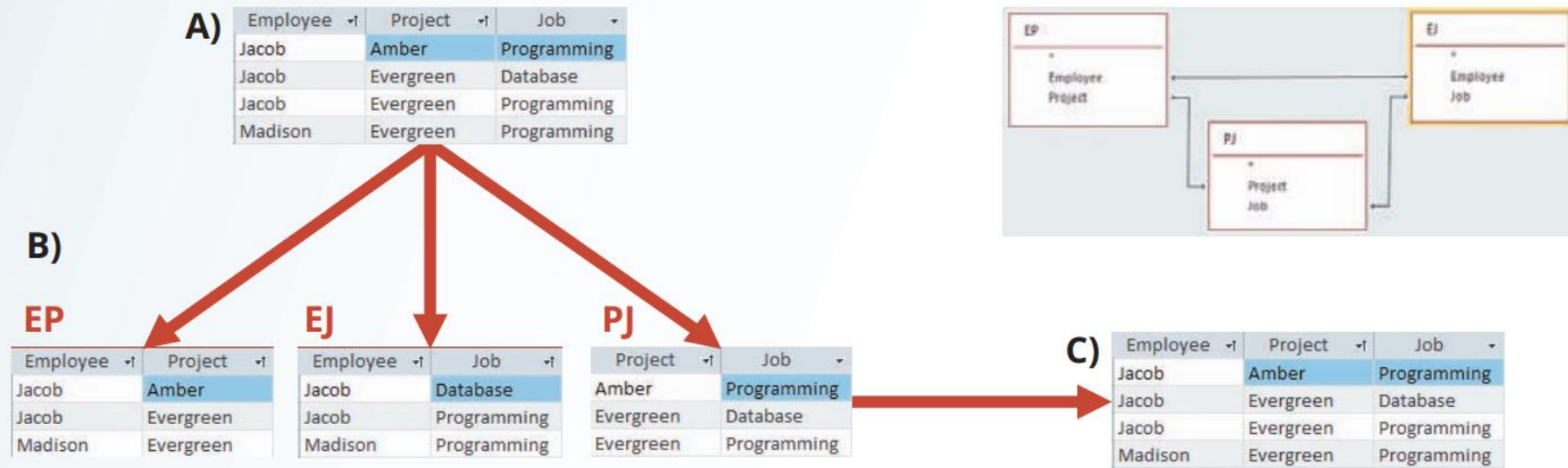
The entities relate to each other cyclically. To resolve this, we would need to establish parent tables with Actor - Play, Play - Theater, and Theater - Actor. These would each contain a portion of the Primary Key in the Actor, Play, and Theater table.

Actor	Play	Theater
Billy Bob	Catcher in the Rye	West 42nd
Ann	Catcher in the Rye	West 42nd
John	Catch-22	Broadway
Lily	Hamlet	Broadway
Lisa	Cats	West 42nd
Andy	Cats	Darlington



Lossless Decomposition when the decomposed tables are joined it recreates the original table. There is no missing data nor new erroneous data.

Figure 6.13 5NF Conversion



A table is in **fifth normal form (5NF)** when it is in 4NF and it cannot have further lossless decompositions.
Here the original table is not in 5NF, but the decomposed tables are in 5NF

Normalization Forms

Normal Form	Characteristic
First normal form (1NF)	Table format, no repeating groups, and PK identified
Second normal form (2NF)	1NF and no partial dependencies
Third normal form (3NF)	2NF and no transitive dependencies
Boyce-Codd normal form (BCNF)	3NF and every determinant is a candidate key (special case of 3NF)
Fourth normal form (4NF)	BCNF and no independent multivalued dependencies
Fifth normal form (5NF or PJNF)	4NF and cannot have lossless decomposition into smaller tables

Why Denormalization

- Normalization is one of many database design goals
- Normalized table requirements
 - Additional processing – process together decomposed tables
 - Loss of system speed – lots of joins
- Normalization purity is sometimes difficult to sustain due to conflict in:
 - Design efficiency
 - Information requirements
 - Processing

Data Modeling Checklist

Business Rules

- Properly document and verify all business rules with the end users.
- Ensure that all business rules are written precisely, clearly, and simply. The business rules must help identify entities, attributes, relationships, and constraints.
- Identify the source of all business rules, and ensure that each business rule is justified, dated, and signed off by an approving authority.

Data Modeling

Naming conventions: All names should be limited in length (database-dependent size).

- Entity names:
 - Should be nouns that are familiar to business and should be short and meaningful.
 - Should document abbreviations, synonyms, and aliases for each entity.
 - Should be unique within the model.
 - Composite entities may include a combination of abbreviated names of the entities linked through the composite entity.
- Attribute names:
 - Should be unique within the entity.
 - Should use the entity abbreviation as a prefix.
 - Should be descriptive of the characteristic.
 - Should use suffixes such as _ID, _NUM, or _CODE for the PK attribute.
 - Should not be a reserved word.
 - Should not contain spaces or special characters such as @, !, or &.
- Relationship names:
 - Should be active or passive verbs that clearly indicate the nature of the relationship.

Entities:

- Each entity should represent a single subject.
- Each entity should represent a set of distinguishable entity instances.
- All entities should be in 3NF or higher. Any entities below 3NF should be justified.
- The granularity of the entity instance should be clearly defined.
- The PK should be clearly defined and support the selected data granularity.

Attributes:

- Should be simple and single-valued (atomic data).
- Should document default values, constraints, synonyms, and aliases.
- Derived attributes should be clearly identified and include source(s).
- Should not be redundant unless this is required for transaction accuracy, performance, or maintaining a history.
- Nonkey attributes must be fully dependent on the PK attribute.

Relationships:

- Should clearly identify relationship participants.
- Should clearly define participation, connectivity, and document cardinality.

ER model:

- Should be validated against expected processes: inserts, updates, and deletions.
- Should evaluate where, when, and how to maintain a history.
- Should not contain redundant relationships except as required (see attributes).
- Should minimize data redundancy to ensure single-place updates.
- Should conform to the minimal data rule: All that is needed is there, and all that is there is needed.

- ER-Model design process usually caters to the normalization process but often things get missed out.
- In most cases for OLTP systems, a normalized ERD up to 3 NF suffice also ensuring data integrity and avoidance of anomalies.