Week 11

Database Systems -

Introduction to Databases and Data Warehouses

CHAPTER 4 - Update Operations, Update Anomalies, and Normalization (Part 2 V2)

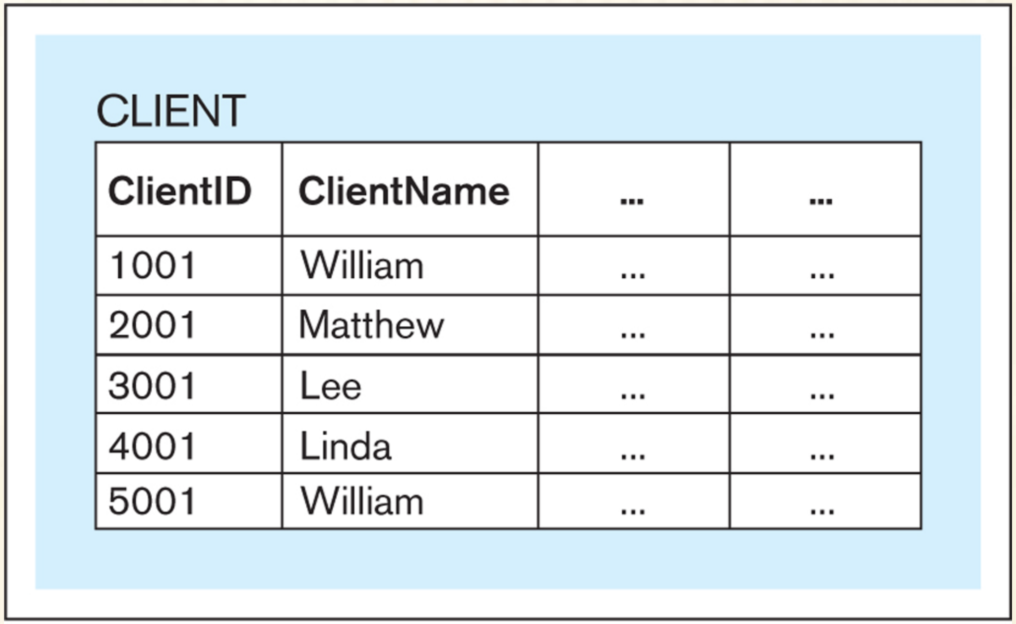
MAIN TOPICS

- Functional Dependencies
  - Notation and Example 1
- Streamline Functional Dependencies
- Types of Functional Dependencies
- Normalization forms – 1NF, 2NF, 3NF
- Normalization Example 1 – Ad Campaign
- Normalization to 1NF
- Normalization to 2NF
- Normalization to 3NF
- Eliminate Redundancy and Resolve Update Anomalies
- Functional Dependencies – Example 2

FUNCTIONAL DEPENDENCIES

- Functional dependency
  - Occurs when the value of one (or more) column(s) in each record of a relation uniquely determines the value of another column in that same record of the relation
  - Example - Relation CLIENT (ClientID, ClientName, ...)

- ClientID determines ClientName
- ClientName does NOT determine ClientID



CLIENT

ClientID	ClientName	...	...
1001	William	...	...
2001	Matthew	...	...
3001	Lee	...	...
4001	Linda	...	...
5001	William	...	...

Copyright ©2014 Pearson Education

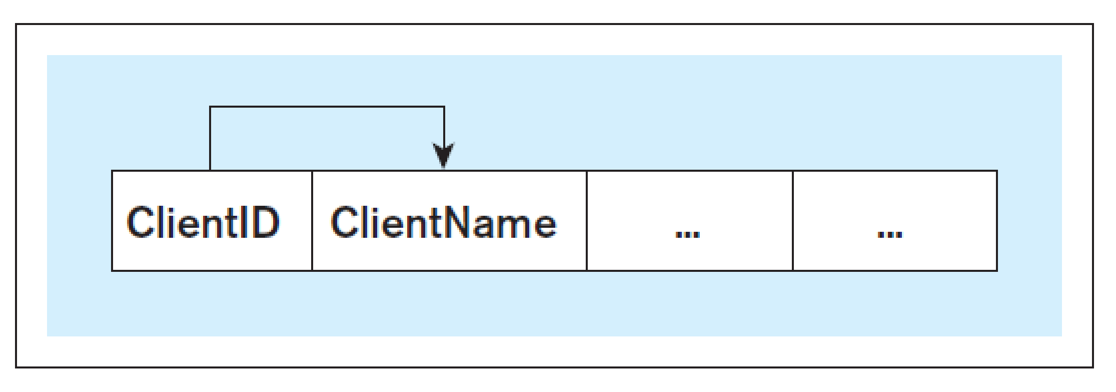
FUNCTIONAL DEPENDENCIES

- Two functional dependency notations
- Symbolic notation: ClientID determines ClientName

$$\mathbf{A \longrightarrow B}$$

$$\mathbf{ClientID \longrightarrow ClientName}$$

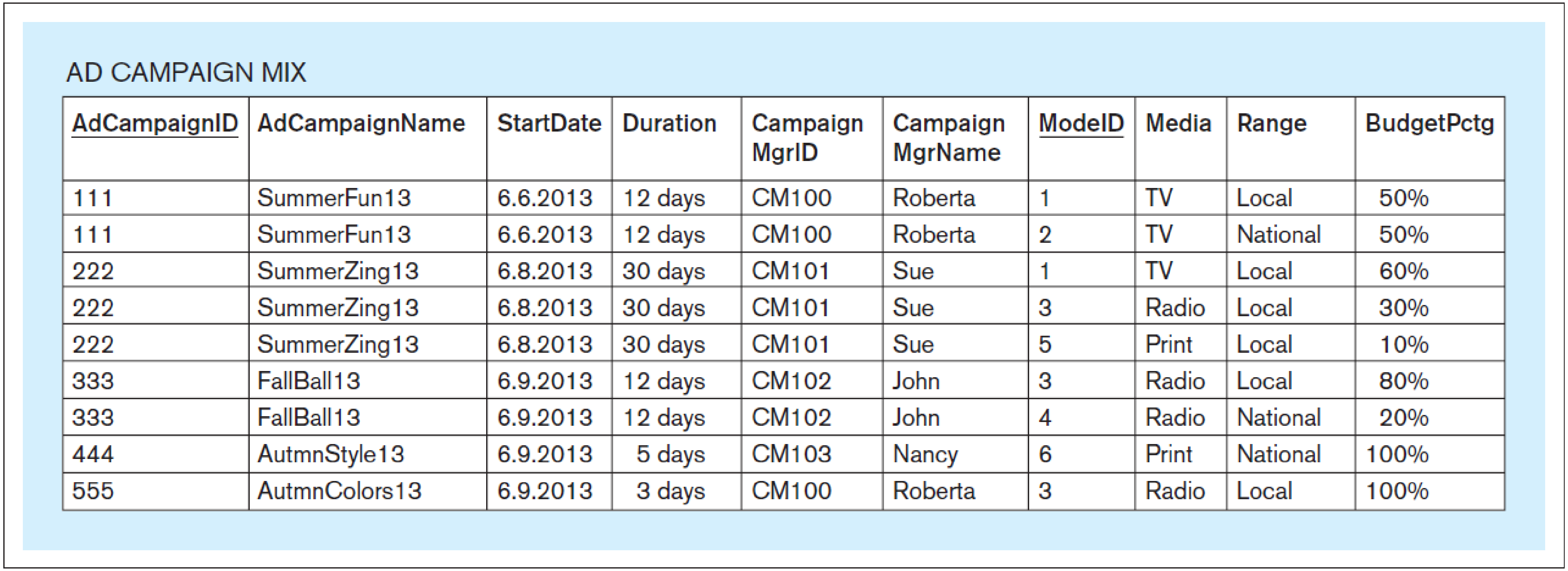
- Graphical notation: ClientID determines ClientName



ClientID ClientName ... ...

Example: FDs of relation AD CAMPAIGN MIX

Sample Data:



AD CAMPAIGN MIX

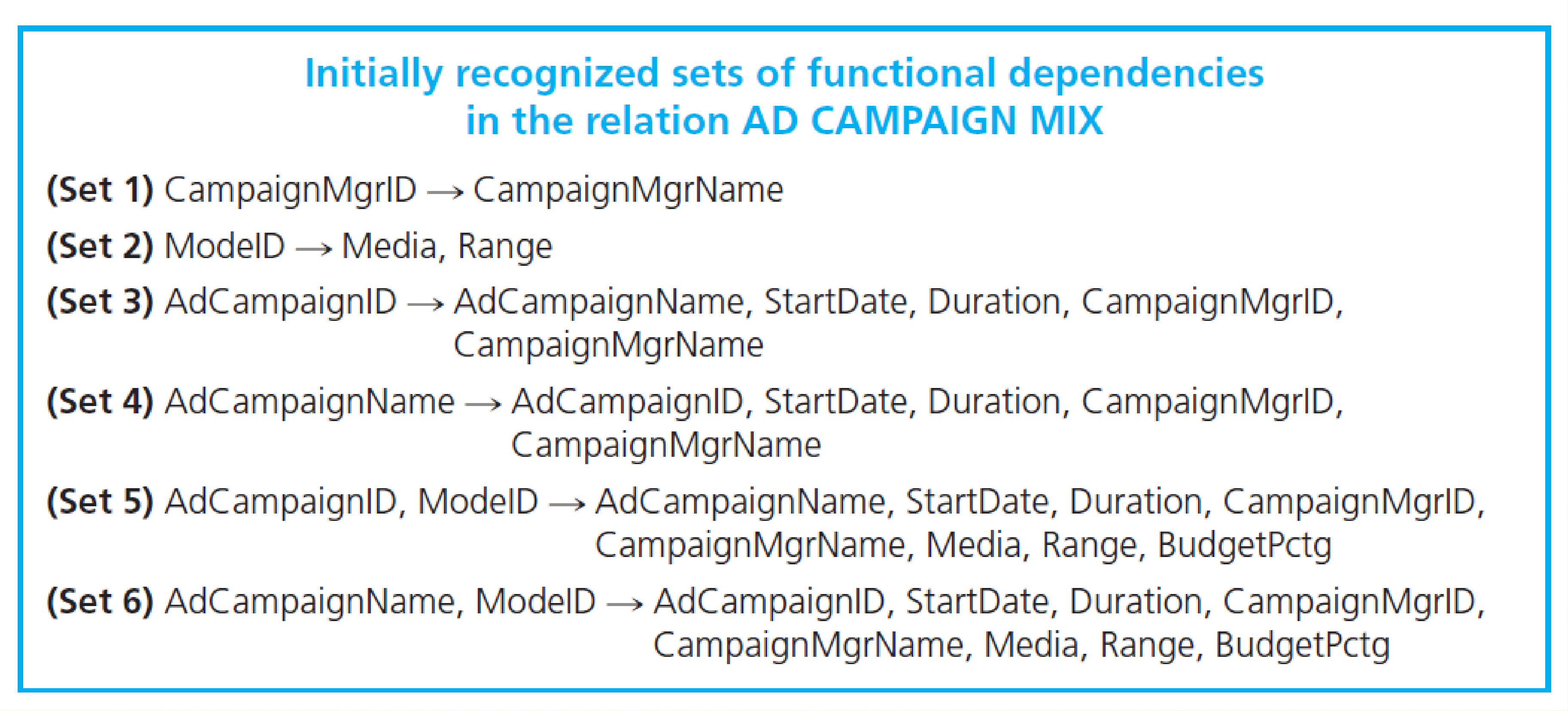
AdCampaignID	AdCampaignName	StartDate	Duration	Campaign MgrID	Campaign MgrName	ModelID	Media	Range	BudgetPctg
111	SummerFun13	6.6.2013	12 days	CM100	Roberta	1	TV	Local	50%
111	SummerFun13	6.6.2013	12 days	CM100	Roberta	2	TV	National	50%
222	SummerZing13	6.8.2013	30 days	CM101	Sue	1	TV	Local	60%
222	SummerZing13	6.8.2013	30 days	CM101	Sue	3	Radio	Local	30%
222	SummerZing13	6.8.2013	30 days	CM101	Sue	5	Print	Local	10%
333	FallBall13	6.9.2013	12 days	CM102	John	3	Radio	Local	80%
333	FallBall13	6.9.2013	12 days	CM102	John	4	Radio	National	20%
444	AutmnStyle13	6.9.2013	5 days	CM103	Nancy	6	Print	National	100%
555	AutmnColors13	6.9.2013	3 days	CM100	Roberta	3	Radio	Local	100%

Identify the functional dependencies in the relation AD CAMPAIGN MIX.

FUNCTIONAL DEPENDENCIES

Example

– initially recognized sets of functional dependencies in relation AD CAMPAIGN MIX



Initially recognized sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 4) AdCampaignName → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 5) AdCampaignID, ModelD → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg

(Set 6) AdCampaignName, ModelD → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg

FUNCTIONAL DEPENDENCIES

- **Streamlining functional dependencies**
  - Not all functional dependencies need to be depicted
  - These types of functional dependencies can be omitted in functional dependency analysis:
    - **Trivial functional dependencies**
    - **Augmented functional dependencies**
    - **Equivalent functional dependencies**

FUNCTIONAL DEPENDENCIES

- Trivial functional dependency
  - *Occurs when an attribute (or a set of attributes) functionally determines itself or its subset*
  - Examples:

    A → A

    A, B → A, B

    A, B → A

    CampaignMgrID, CampaignMgrName → CampaignMgrName

  - Trivial functional dependencies are not depicted when analyzing functional dependencies.

FUNCTIONAL DEPENDENCIES

- Augmented functional dependency
  - *Functional dependency that contains an existing functional dependency*
  - Example

    If the functional dependency in a relation exists:

    $$A \longrightarrow B$$

    Then the *augmented functional dependency exists*:

    $$A, C \longrightarrow B$$

  - Does not add new information to what is already described by the existing functional dependency
  - Usually omitted during functional dependency analysis

FUNCTIONAL DEPENDENCIES

- AD CAMPAIGN MIX example
  - Functional dependencies (Set 3):

    AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

  - Augmented functional dependencies (in Set 5) due to Set 3:

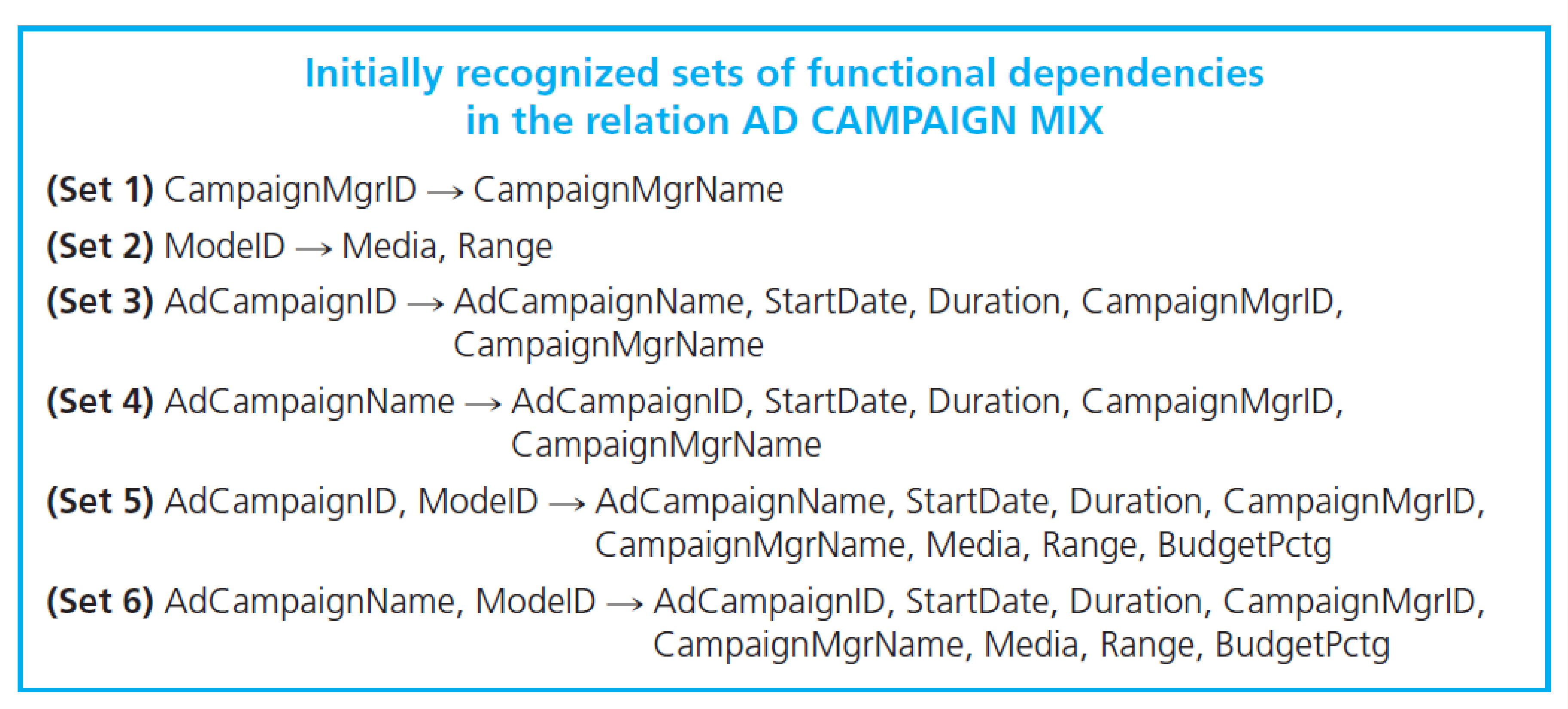
    AdCampaignID, ModeID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

    can be omitted

FUNCTIONAL DEPENDENCIES

Example

– initially recognized sets of functional dependencies in relation AD CAMPAIGN MIX



Initially recognized sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 4) AdCampaignName → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 5) AdCampaignID, ModelD → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg

(Set 6) AdCampaignName, ModelD → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg

FUNCTIONAL DEPENDENCIES

- AD CAMPAIGN MIX example
  - Functional dependencies (Set 2):

    ModelD → Media, Range

  - Augmented functional dependencies (in Set 5) due to Set 2:

    AdCampaignID, ModelD → Media, Range

  can be omitted

FUNCTIONAL DEPENDENCIES

- Equivalent functional dependency
  - Two functional dependencies are equivalent
  - *Occurs when two columns (or sets of columns) that functionally determine each other determine other columns,* Example:
  - If functional dependencies in a relation exists:

    A → B
    B → A

    Then these are equivalent functional dependencies

    A → B
    B → A

    and these are equivalent functional dependencies:

    A → B, X
    B → A, X

    and these are equivalent functional dependencies:

    Y,A → B, X
    Y,B → A, X

  - Depict only one of the equivalent functional dependencies, omit the other

FUNCTIONAL DEPENDENCIES

- Equivalent Functional Dependency -- AD CAMPAIGN MIX example
  - These two are equivalent functional dependencies:

AdCampaignID → AdCampaignName

AdCampaignName → AdCampaignID

So:

Set 3 and Set 4 are *equivalent sets.*
Set 5 and Set 6 are *equivalent sets.*

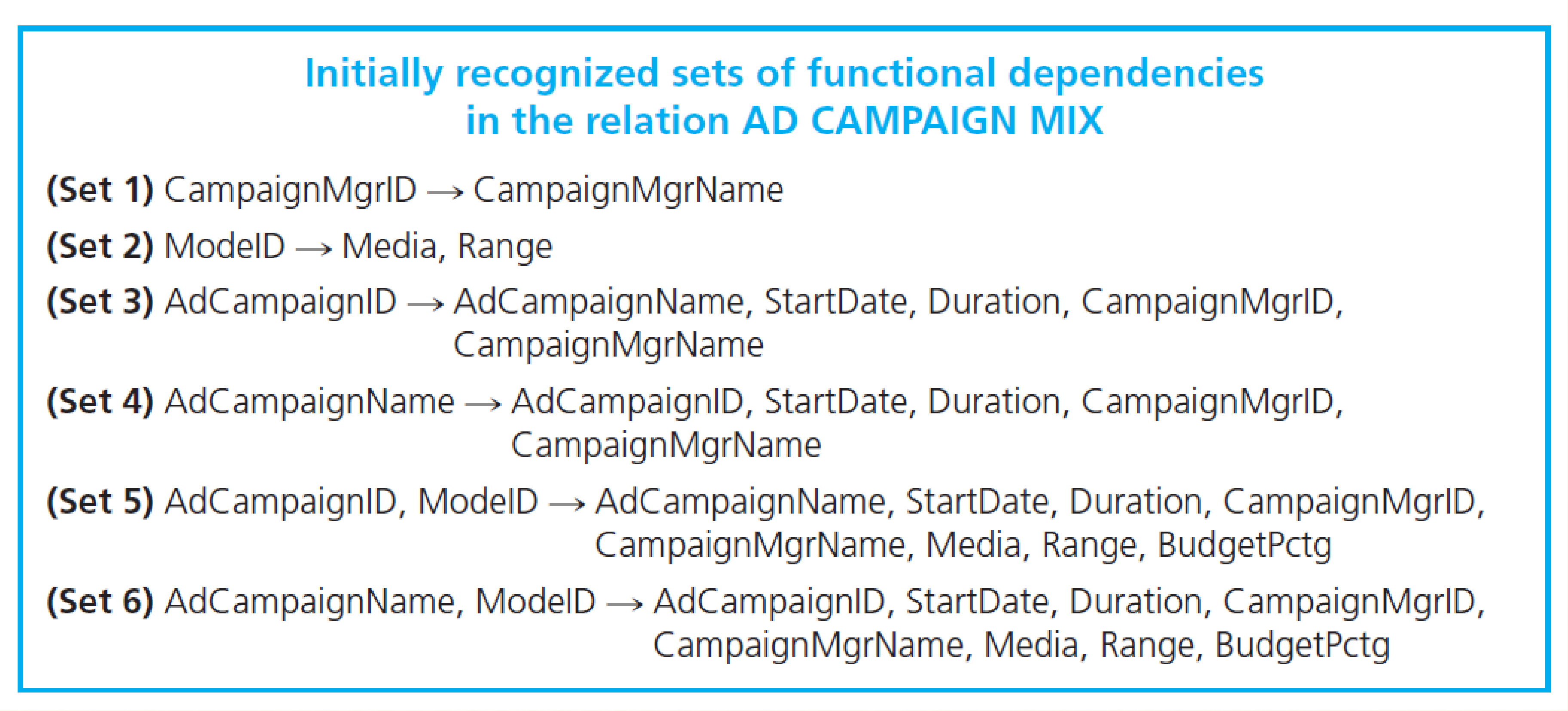
So

- Can depict only Set 3 and omit Set 4
- Can depict only Set 5 and omit Set 6

FUNCTIONAL DEPENDENCIES

Example

– initially recognized sets of functional dependencies in relation AD CAMPAIGN MIX



Initially recognized sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 4) AdCampaignName → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName

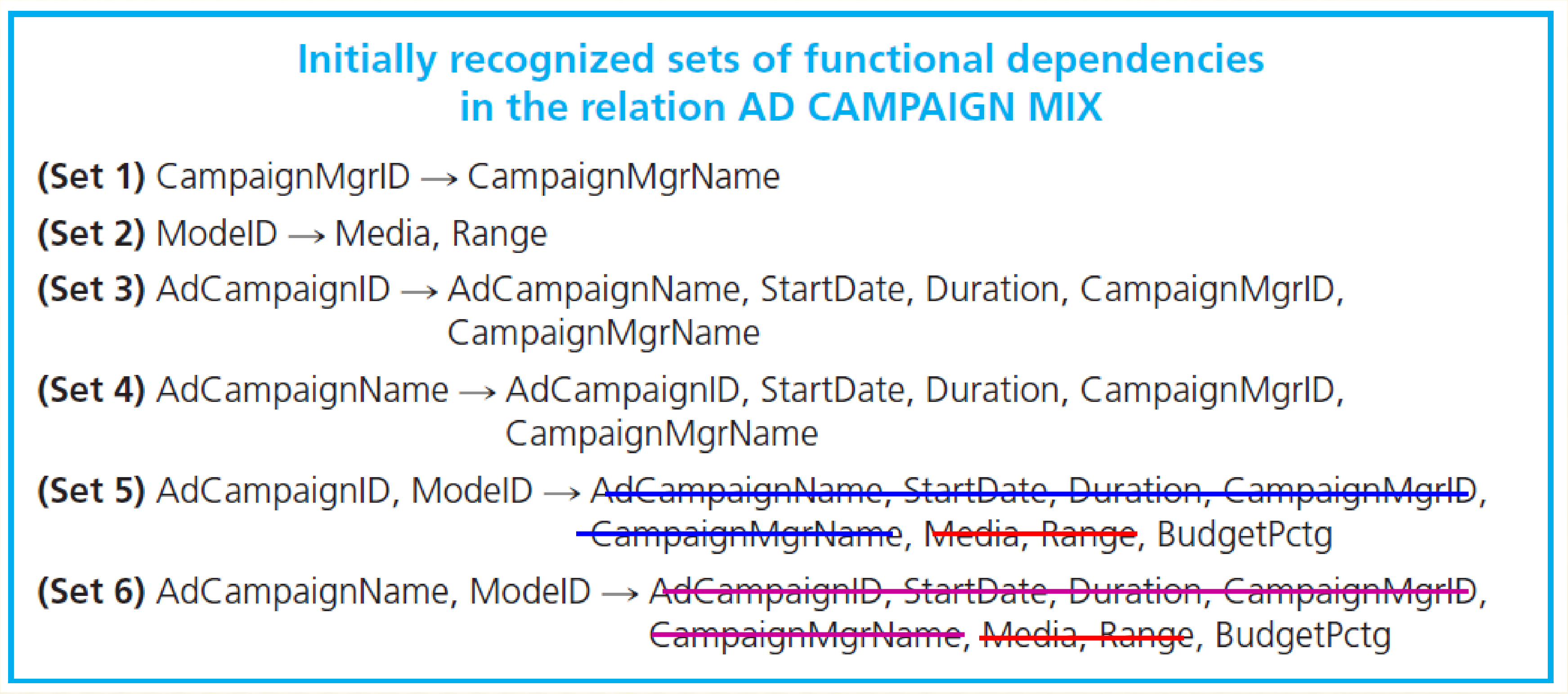
(Set 5) AdCampaignID, ModelD → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg

(Set 6) AdCampaignName, ModelD → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg

FUNCTIONAL DEPENDENCIES (FD)

Example - **streamlining** functional dependencies

1. First remove augmented FDs from original FD sets



Initially recognized sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 4) AdCampaignName → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName

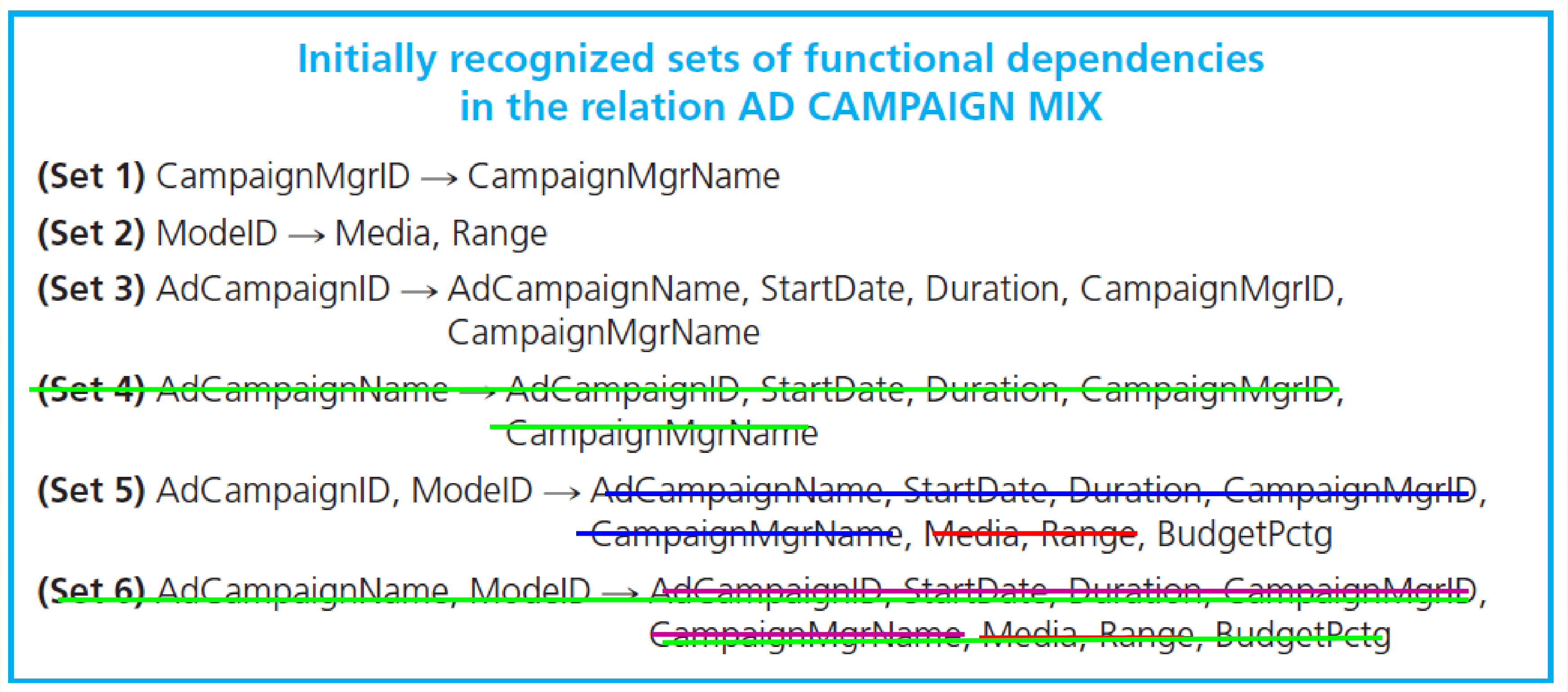
(Set 5) AdCampaignID, ModelD → ~~AdCampaignName, StartDate, Duration, CampaignMgrID~~, ~~CampaignMgrName~~, ~~Media, Range~~, BudgetPctg

(Set 6) AdCampaignName, ModelD → ~~AdCampaignID, StartDate, Duration, CampaignMgrID~~, ~~CampaignMgrName~~, ~~Media, Range~~, BudgetPctg

FUNCTIONAL DEPENDENCIES (FD)

Example - **streamlining** functional dependencies

2. Then keep only one of the equivalent FDs in original FD sets
   - Green lines



Initially recognized sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

~~**(Set 4)** AdCampaignName → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName~~

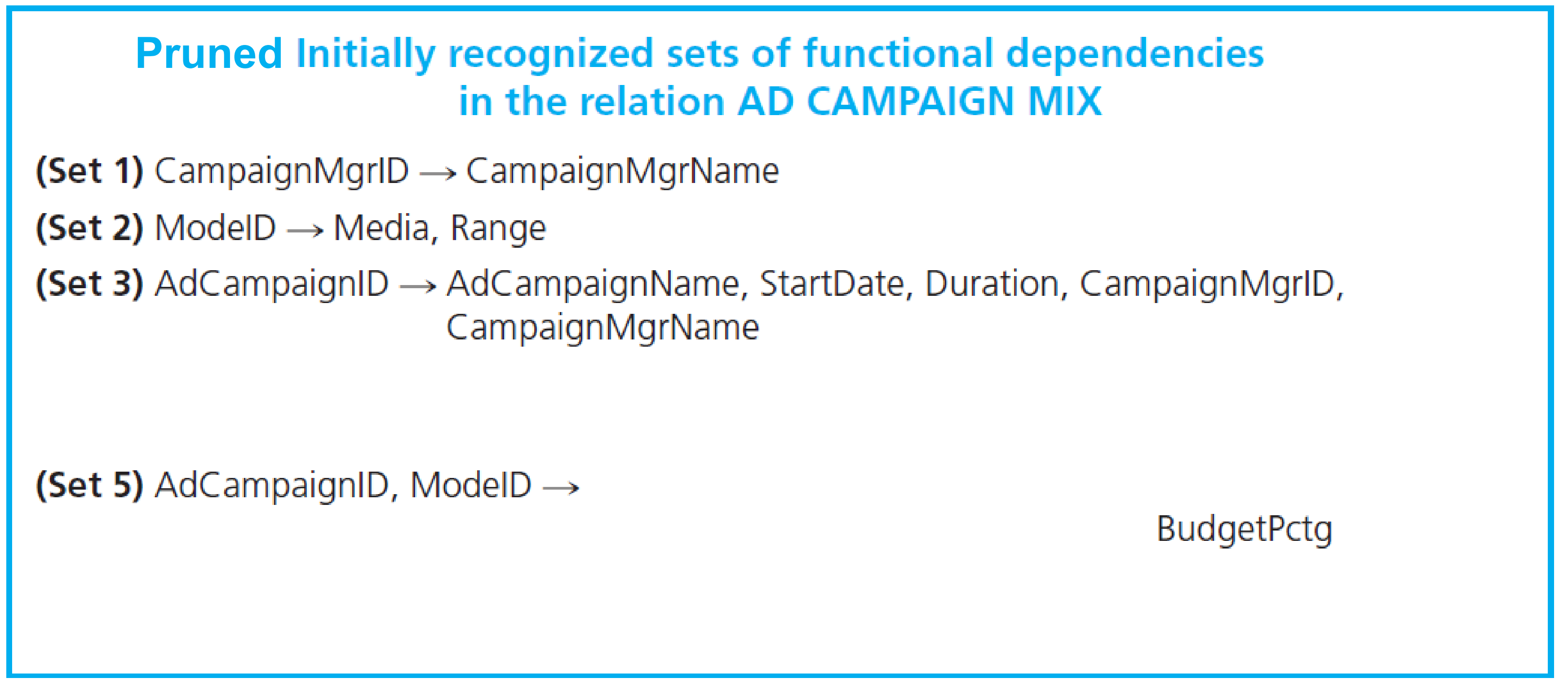
(Set 5) AdCampaignID, ModelD → ~~AdCampaignName, StartDate, Duration, CampaignMgrID~~, ~~CampaignMgrName~~, ~~Media, Range~~, BudgetPctg

~~**(Set 6)** AdCampaignName, ModelD → AdCampaignID, StartDate, Duration, CampaignMgrID, CampaignMgrName, Media, Range, BudgetPctg~~

FUNCTIONAL DEPENDENCIES

Example - **streamlining** functional dependencies

- Original FD sets after being pruned



Pruned Initially recognized sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

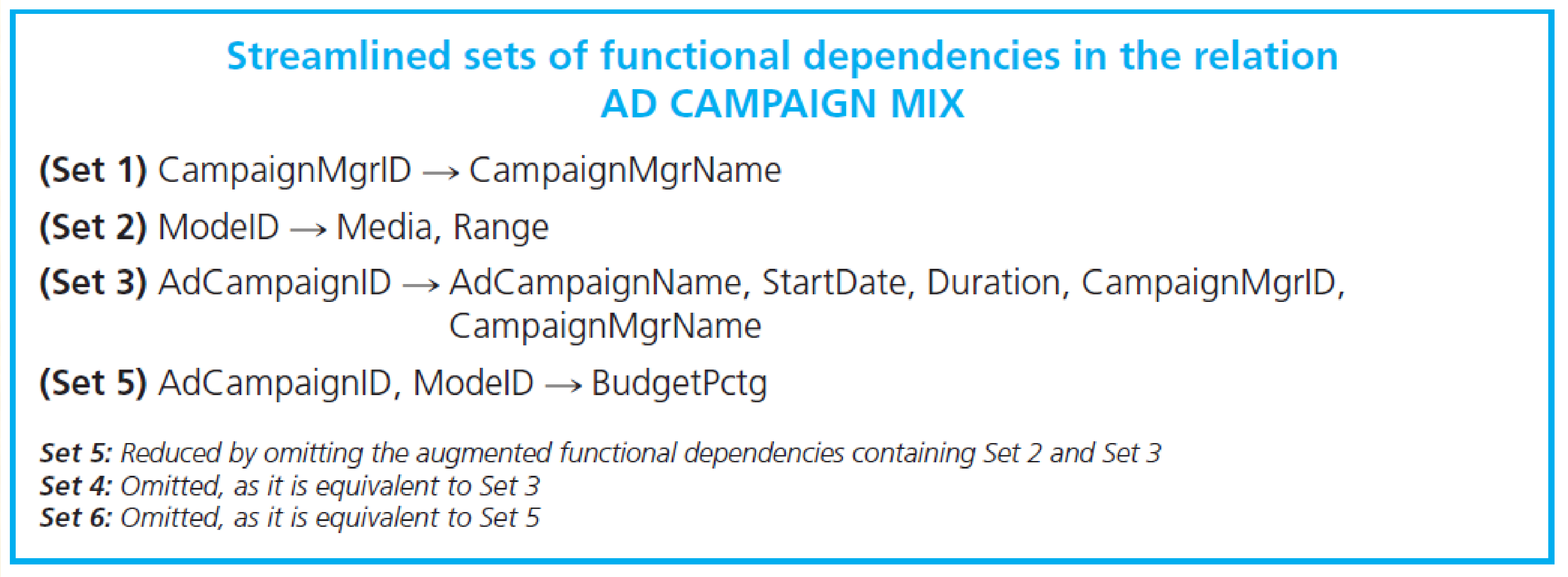
(Set 5) AdCampaignID, ModelD →

BudgetPctg

FUNCTIONAL DEPENDENCIES

Example – **streamlined** functional dependencies

- Re-arranged to look neat



Streamlined sets of functional dependencies in the relation AD CAMPAIGN MIX

(Set 1) CampaignMgrID → CampaignMgrName

(Set 2) ModelD → Media, Range

(Set 3) AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

(Set 5) AdCampaignID, ModelD → BudgetPctg

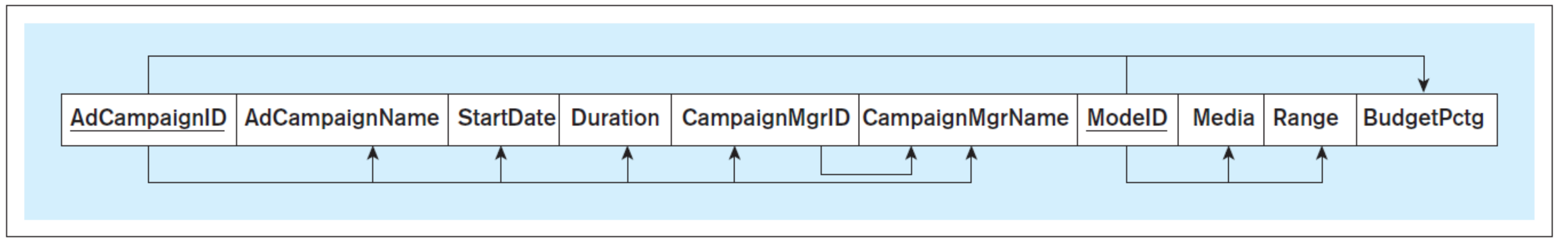
Set 5: *Reduced by omitting the augmented functional dependencies containing Set 2 and Set 3*
Set 4: *Omitted, as it is equivalent to Set 3*
Set 6: *Omitted, as it is equivalent to Set 5*

Visualize these FDs using a figure?

FUNCTIONAL DEPENDENCIES

Example – **streamlined** functional dependencies

- Graphical depiction of the **pruned** FD sets



FUNCTIONAL DEPENDENCIES

- Types of functional dependencies
  - three categories of functional dependencies are used as the basis for typical normalization process:
    - Partial functional dependency
    - Full key functional dependency
    - Transitive functional dependency

FUNCTIONAL DEPENDENCIES

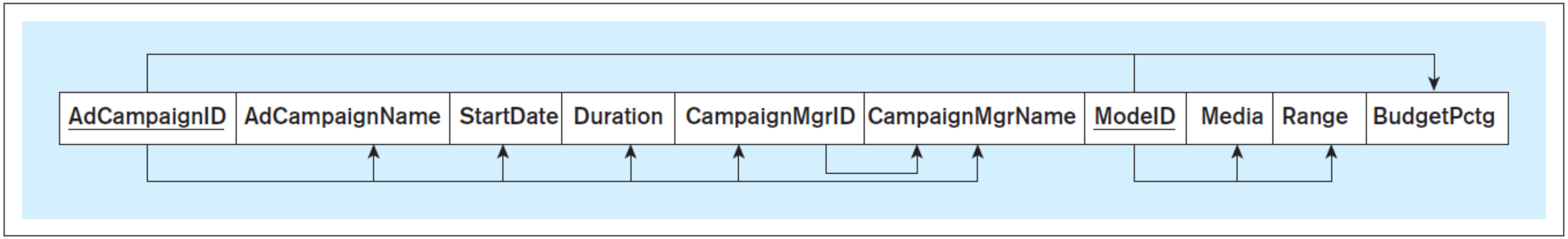
- Partial functional dependency
  - *occurs when a column of a relation is functionally **dependent on a component of a composite primary key***
  - Can occur only in cases when a relation has a composite primary key
    - Only composite primary keys have separate components

FUNCTIONAL DEPENDENCIES

- Partial functional dependencies (in AD CAMPAIGN MIX example)

AdCampaignID → AdCampaignName, StartDate, Duration, CampaignMgrID, CampaignMgrName

ModelID → Media, Range



FUNCTIONAL DEPENDENCIES

- Full key functional dependency
  - *Occurs when*
    - *A **primary key** functionally **determines the column** of a relation and*
    - ***No separate component** of the primary key **partially determines** the **same column***
  - For the single-column primary key of a relation
    - PK always fully functionally determines all the other columns
  - For the composite primary key of a relation
    - If portions of the composite primary key partially determine some columns, then the PK does not fully functionally determine these columns

FUNCTIONAL DEPENDENCIES

- Full key functional dependency (in AD CAMPAIGN MIX example)

AdCampaignID, ModeID → BudgetPctg

FUNCTIONAL DEPENDENCIES

- Transitive functional dependency
  - *occurs when nonkey columns functionally determine other nonkey columns of a relation*
    - Nonkey column
      - Neither a primary key column nor a candidate key column

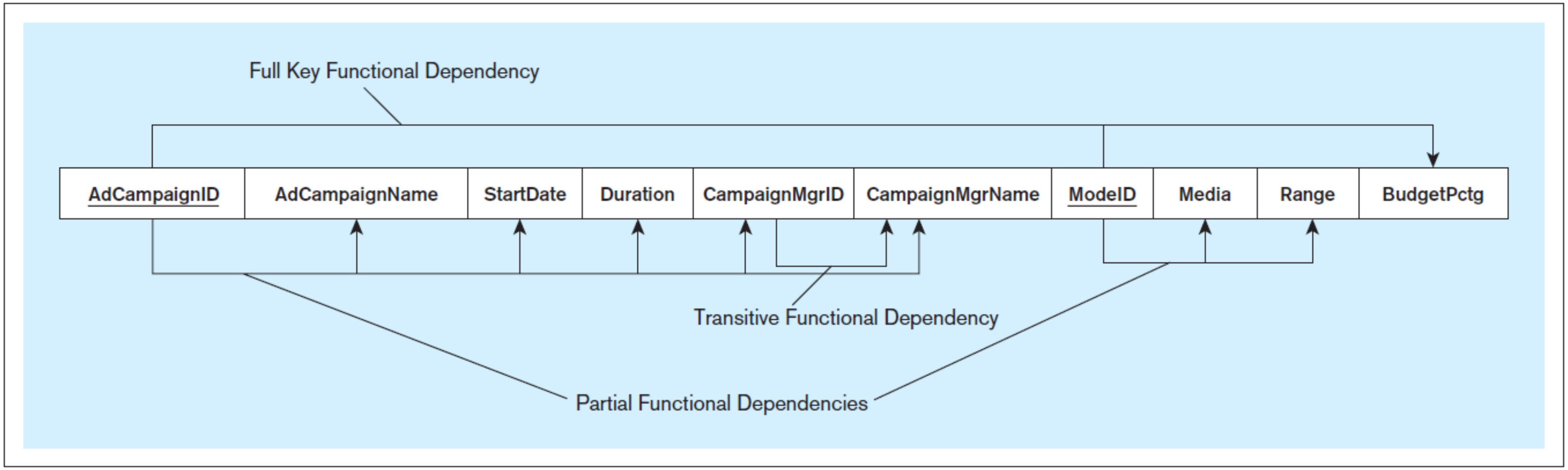
FUNCTIONAL DEPENDENCIES

- Transitive functional dependency (in AD CAMPAIGN MIX example)

CampaignMgrID → CampaignMgrName

FUNCTIONAL DEPENDENCIES

Functional dependencies in the relation AD CAMPAIGN MIX (types indicated)



Full Key Functional Dependency

AdCampaignID	AdCampaignName	StartDate	Duration	CampaignMgrID	CampaignMgrName	ModelID	Media	Range	BudgetPctg

Transitive Functional Dependency

Partial Functional Dependencies

NORMALIZATION

- **Normalization**
  - process used to improve the design of relational databases
- **Normal form** - term representing a set of particular conditions (whose purpose is reducing data redundancy) that a table has to satisfy
  - From a lower to a higher normal form, these conditions are increasingly stricter and leave less possibility for redundant data

NORMALIZATION

- Most fundamental normal forms
  - First normal form (1NF)
  - Second normal form (2NF)
  - Third normal form (3NF)

NORMALIZATION

- The normalization process
  - Examine each table and verify if it satisfies a particular normal form
  - If a table **satisfies** a particular normal form,
    - verify if that relation satisfies the next higher normal form
  - If a table **does not satisfy** a particular normal form,
    - convert the table into a set of tables that satisfy the particular normal form

NORMALIZATION

- Normalizing to *first normal form*
  - *D*one on non-relational tables
  - Purpose: to convert non-relational tables to relational tables
- Normalizing to *subsequent normal forms* (e.g., 2NF, 3NF)
  - Improve the design of relational tables containing redundant information
  - Alleviate the problem of update anomalies

NORMALIZATION

- **First Normal Form (1NF)** - *A table is in 1NF if* ***each row is unique*** *and* ***no column in any row*** *contains* ***multiple*** *values*
  - Every relational table is, by definition, in 1NF
    - Conditions for a table to be a relation
  - Normalizing to 1NF
    - not a part of the normalization process of relational databases
    - done only when a non-relational table is being converted to a relation

NORMALIZATION

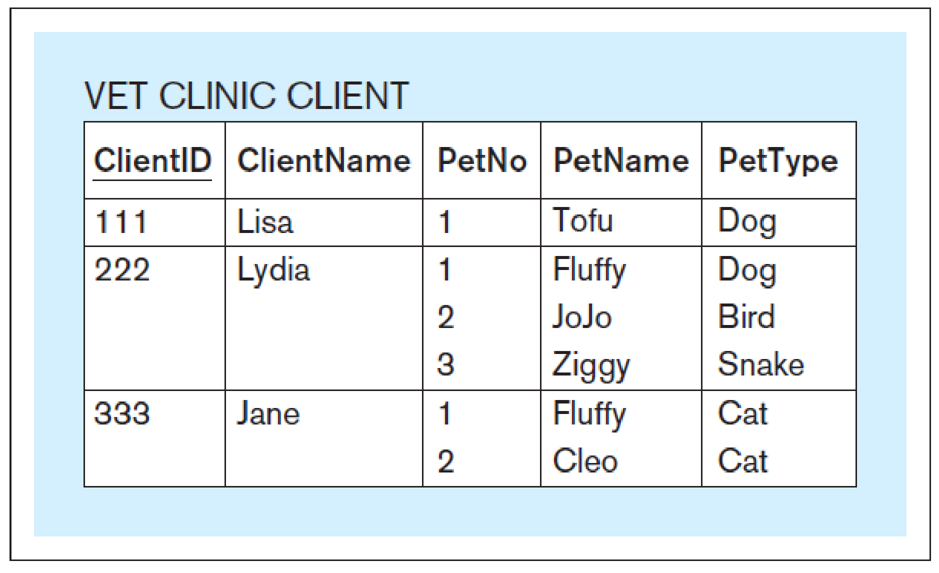
- First Normal Form (1NF)
  - Related multivalued columns
    - Columns in a table referring to the **same real-world concept (entity)** and
    - Have multiple values per record
  - Normalizing to 1NF
    - Eliminate **groups** of related multi-valued columns

NORMALIZATION

- Normalizing to 1NF
  - Eliminate **groups** of related multi-valued columns
  - 2 methods
    - Add new records
      - Add to the original table one record for each occurrence of related multi-valued columns
    - Add new tables
      - Add a new separate table for related multi-valued columns
      - Add to the new table one record for each occurrence of related multi-valued columns

Example: Normalizing a table to **1NF**

Non-relational table (not in 1NF).

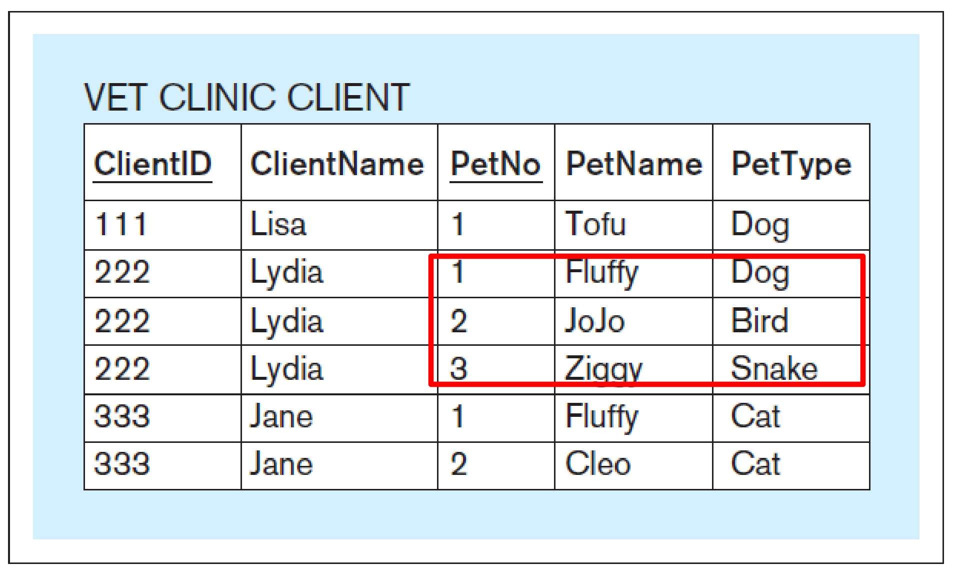


VET CLINIC CLIENT

ClientID	ClientName	PetNo	PetName	PetType
111	Lisa	1	Tofu	Dog
222	Lydia	1 2 3	Fluffy JoJo Ziggy	Dog Bird Snake
333	Jane	1 2	Fluffy Cleo	Cat Cat

Normalizing the table to 1NF by **adding new records**

- **Duplicate ClientID, ClientName**
- **Composite Primary Key**

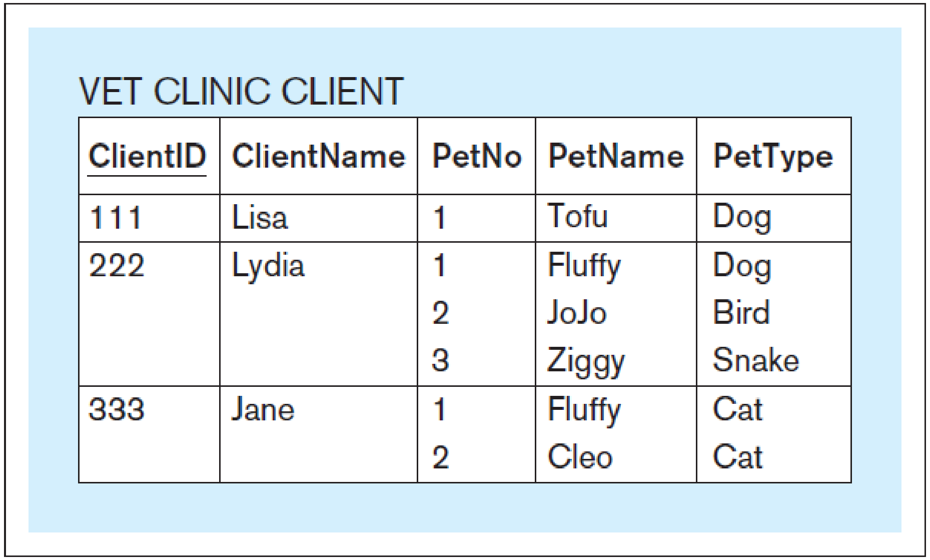


VET CLINIC CLIENT

ClientID	ClientName	PetNo	PetName	PetType
111	Lisa	1	Tofu	Dog
222	Lydia	1	Fluffy	Dog
222	Lydia	2	JoJo	Bird
222	Lydia	3	Ziggy	Snake
333	Jane	1	Fluffy	Cat
333	Jane	2	Cleo	Cat

Example: Normalizing a table to **1NF**

Non-relational table (not in 1NF).

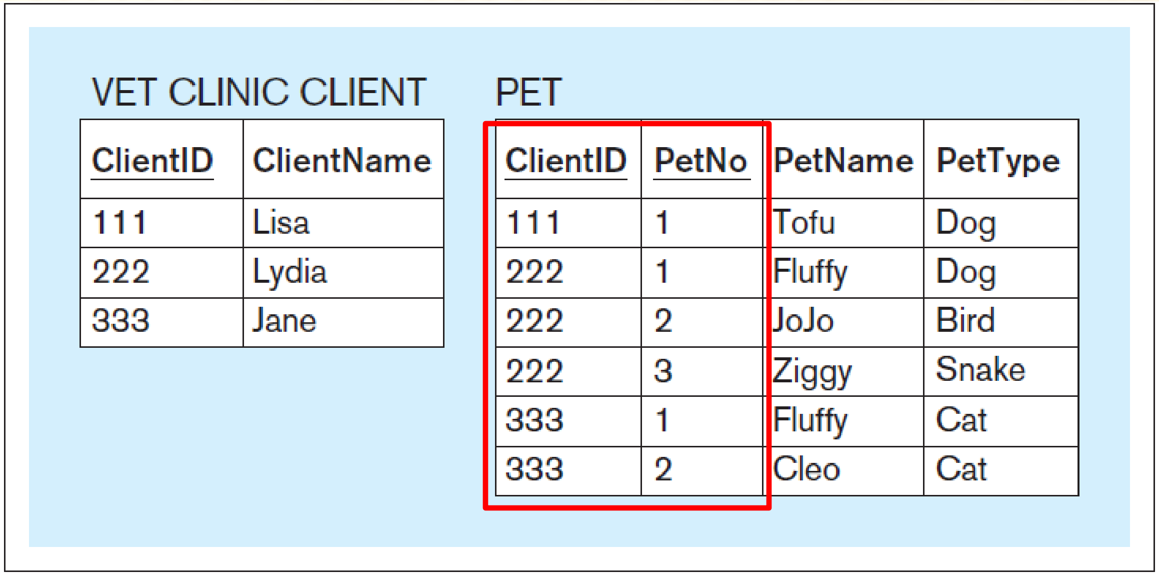


VET CLINIC CLIENT

ClientID	ClientName	PetNo	PetName	PetType
111	Lisa	1	Tofu	Dog
222	Lydia	1	Fluffy	Dog
		2	JoJo	Bird
		3	Ziggy	Snake
333	Jane	1	Fluffy	Cat
		2	Cleo	Cat

Normalizing the table to 1NF by **creating a new, separate table**

- **No duplicate data**
- **Composite PK**



VET CLINIC CLIENT

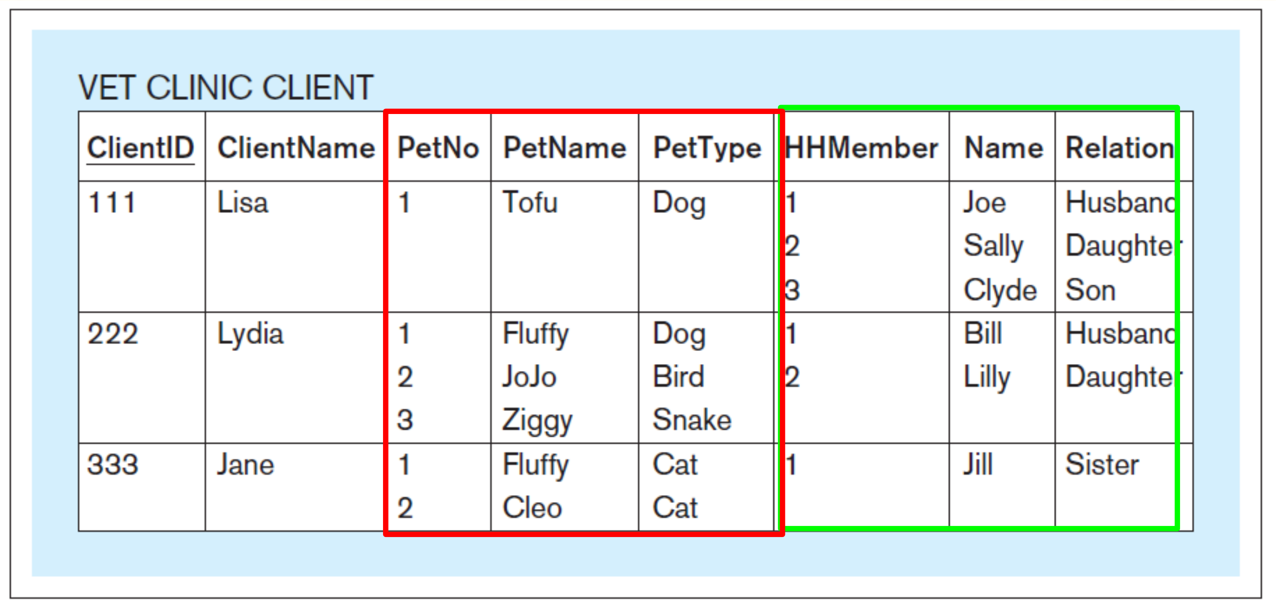
ClientID	ClientName
111	Lisa
222	Lydia
333	Jane

PET

ClientID	PetNo	PetName	PetType
111	1	Tofu	Dog
222	1	Fluffy	Dog
222	2	JoJo	Bird
222	3	Ziggy	Snake
333	1	Fluffy	Cat
333	2	Cleo	Cat

Example: Normalizing a table to **1NF**

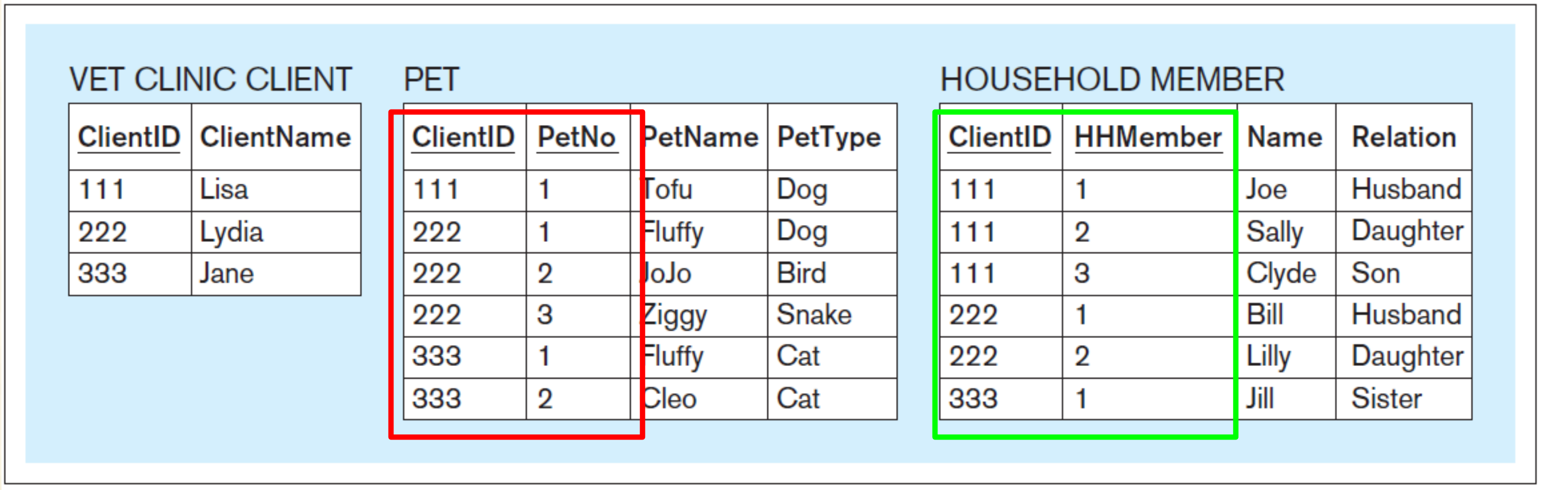
Non-relational table (not in 1NF) with two groups of related multivalued columns



VET CLINIC CLIENT

ClientID	ClientName	PetNo	PetName	PetType	HHMember	Name	Relation
111	Lisa	1	Tofu	Dog	1 2 3	Joe Sally Clyde	Husband Daughter Son
222	Lydia	1 2 3	Fluffy JoJo Ziggy	Dog Bird Snake	1 2	Bill Lilly	Husband Daughter
333	Jane	1 2	Fluffy Cleo	Cat Cat	1	Jill	Sister

Normalizing the table to 1NF – add one table per group, each with composite PK



VET CLINIC CLIENT

ClientID	ClientName
111	Lisa
222	Lydia
333	Jane

PET

ClientID	PetNo	PetName	PetType
111	1	Tofu	Dog
222	1	Fluffy	Dog
222	2	JoJo	Bird
222	3	Ziggy	Snake
333	1	Fluffy	Cat
333	2	Cleo	Cat

HOUSEHOLD MEMBER

ClientID	HHMember	Name	Relation
111	1	Joe	Husband
111	2	Sally	Daughter
111	3	Clyde	Son
222	1	Bill	Husband
222	2	Lilly	Daughter
333	1	Jill	Sister

NORMALIZATION

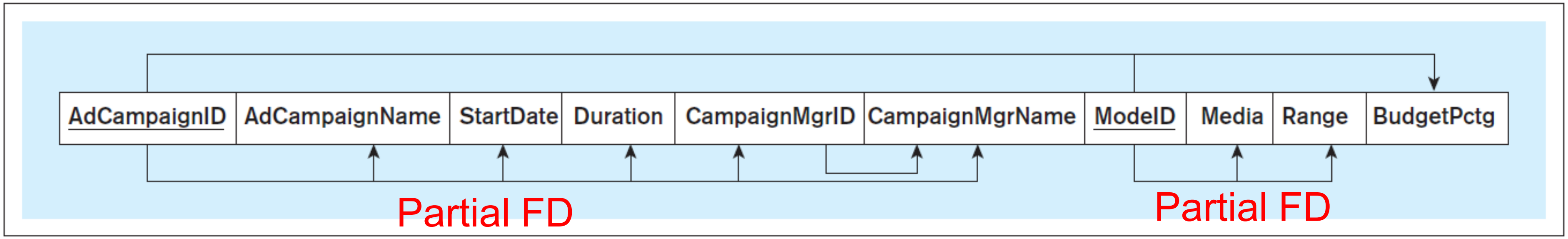
- Second Normal Form (2NF)
  - *A table is in 2NF if it is in 1NF and if it **does not contain partial functional dependencies***
  - If a relation has a single-column primary key, then
    - No possibility of partial functional dependencies
    - Must be in 2NF
    - No normalization to 2NF
  - If a relation has a composite primary key, then
    - If no partial dependencies, then
      - In 2NF and no normalization to 2NF
    - If having partial dependencies, then
      - Not in 2NF; Must be normalized to 2NF

NORMALIZATION

- **Normalization of a relation to 2NF**
  - Create an additional relation for each set of partial dependencies
    - Primary key of the additional relation
      - Part of the primary key that functionally determines the columns in the original relation
      - i.e. where the arrow coming from
    - Other columns in the additional relation
      - The partially dependent columns in original relation
      - i.e. where the arrow going to
  - Keep the original table but remove all partially dependent columns (i.e. where the arrow going to)
    - Keep all components of primary key in original relation

Example: Normalizing a table to **2NF**

Pressly Ad Agency - relation AD CAMPAIGN MIX

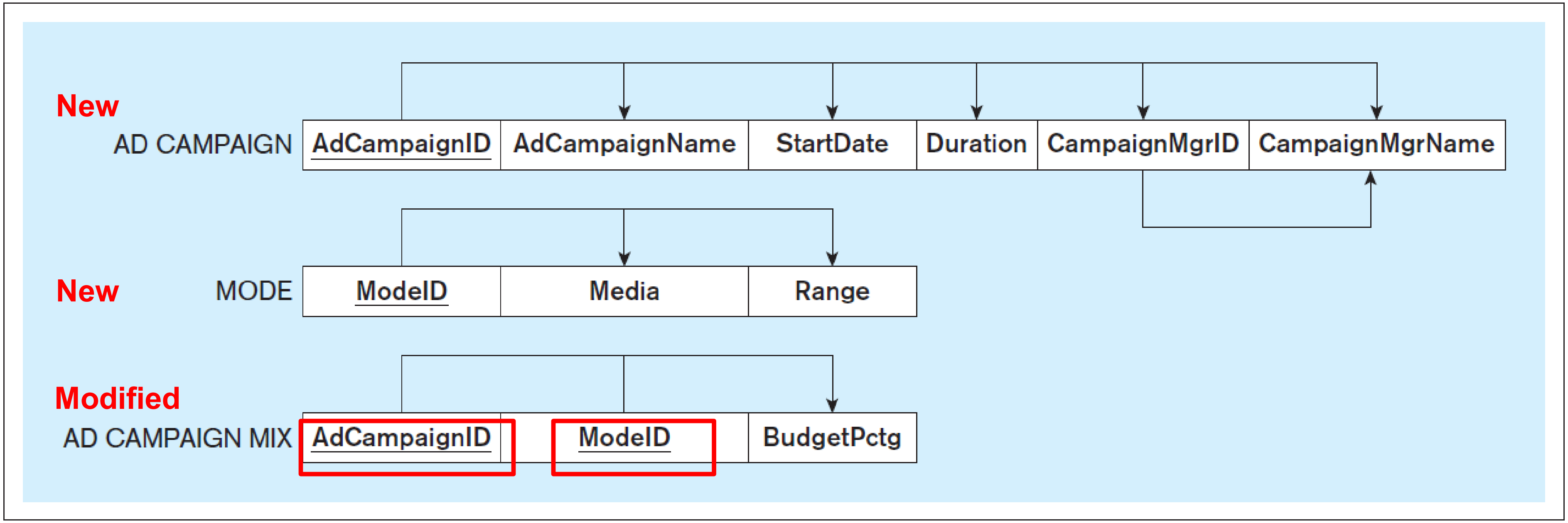


AdCampaignID AdCampaignName StartDate Duration CampaignMgrID CampaignMgrName ModelID Media Range BudgetPctg

Partial FD

Partial FD

In 2NF:



New

AD CAMPAIGN: AdCampaignID AdCampaignName StartDate Duration CampaignMgrID CampaignMgrName

New

MODE: ModelID Media Range

Modified

AD CAMPAIGN MIX: AdCampaignID ModelID BudgetPctg

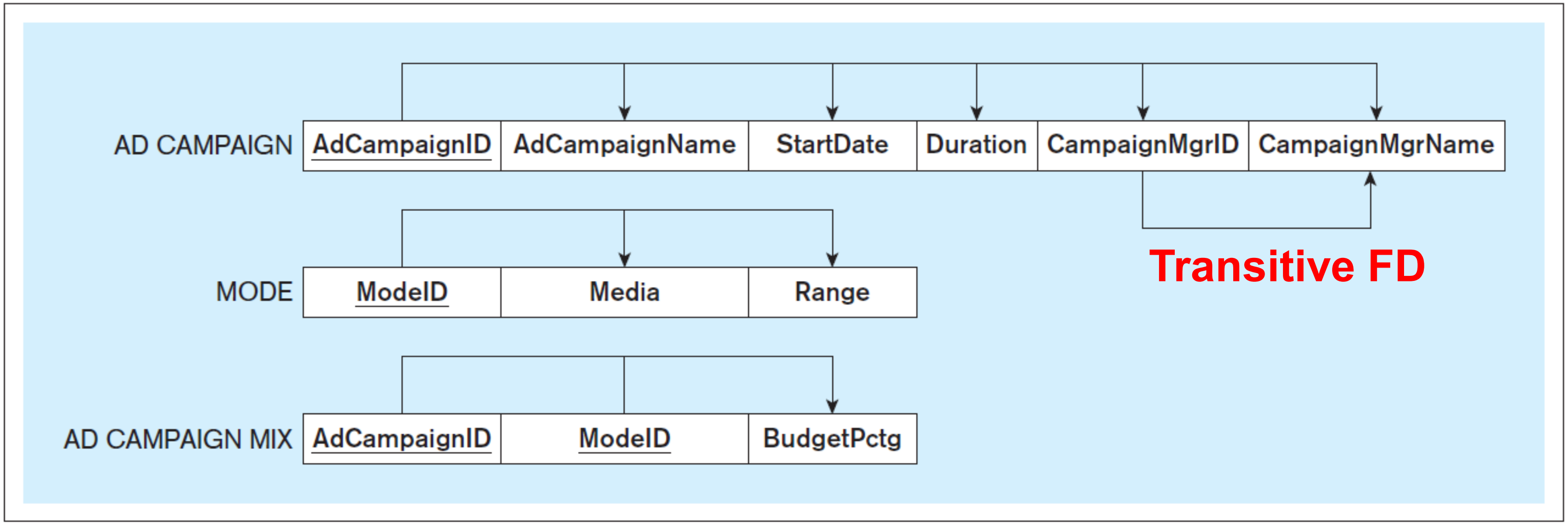
NORMALIZATION

- Third Normal Form (3NF)
  - *A table is in 3NF if*
    - *Is in 2NF and*
    - ***Does not contain transitive functional dependencies***
  - For relations that are in 2NF
    - Continue to normalize them to 3NF.

NORMALIZATION

- Normalization of a relation to 3NF
  - Create one additional relation for each set of transitive dependencies
    - Primary key of the additional relation
      - Nonkey column (or columns) that functionally determine other nonkey columns in original relation
      - i.e. where the arrow coming from
    - Other columns in the additional relation
      - Transitively dependent nonkey columns in original relation
      - i.e. where the arrow going to
  - Keep original table but remove transitively dependent columns (i.e. where the arrow going to)
    - Keep nonkey column(s) that functionally determined other nonkey columns in original relation

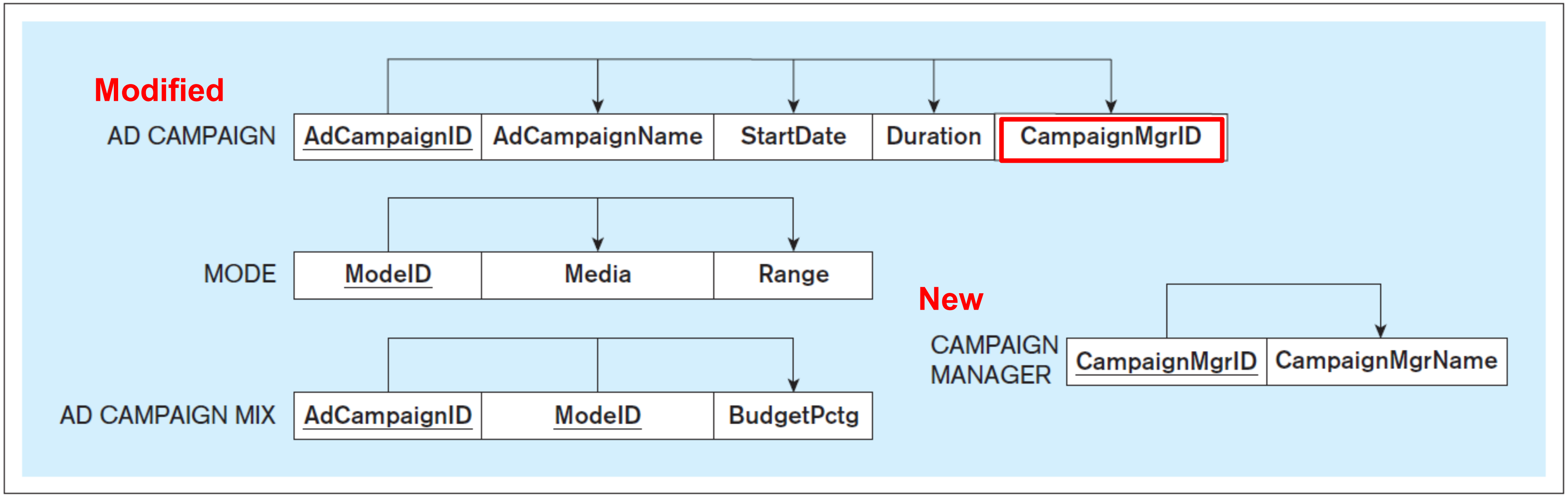
Example: Normalizing a table to 3NF



AD CAMPAIGN	AdCampaignID	AdCampaignName	StartDate	Duration	CampaignMgrID	CampaignMgrName
MODE	ModelD	Media	Range			
AD CAMPAIGN MIX	AdCampaignID	ModelD	BudgetPctg			

Transitive FD

Pressly Ad Agency example - normalized to 3NF



Modified

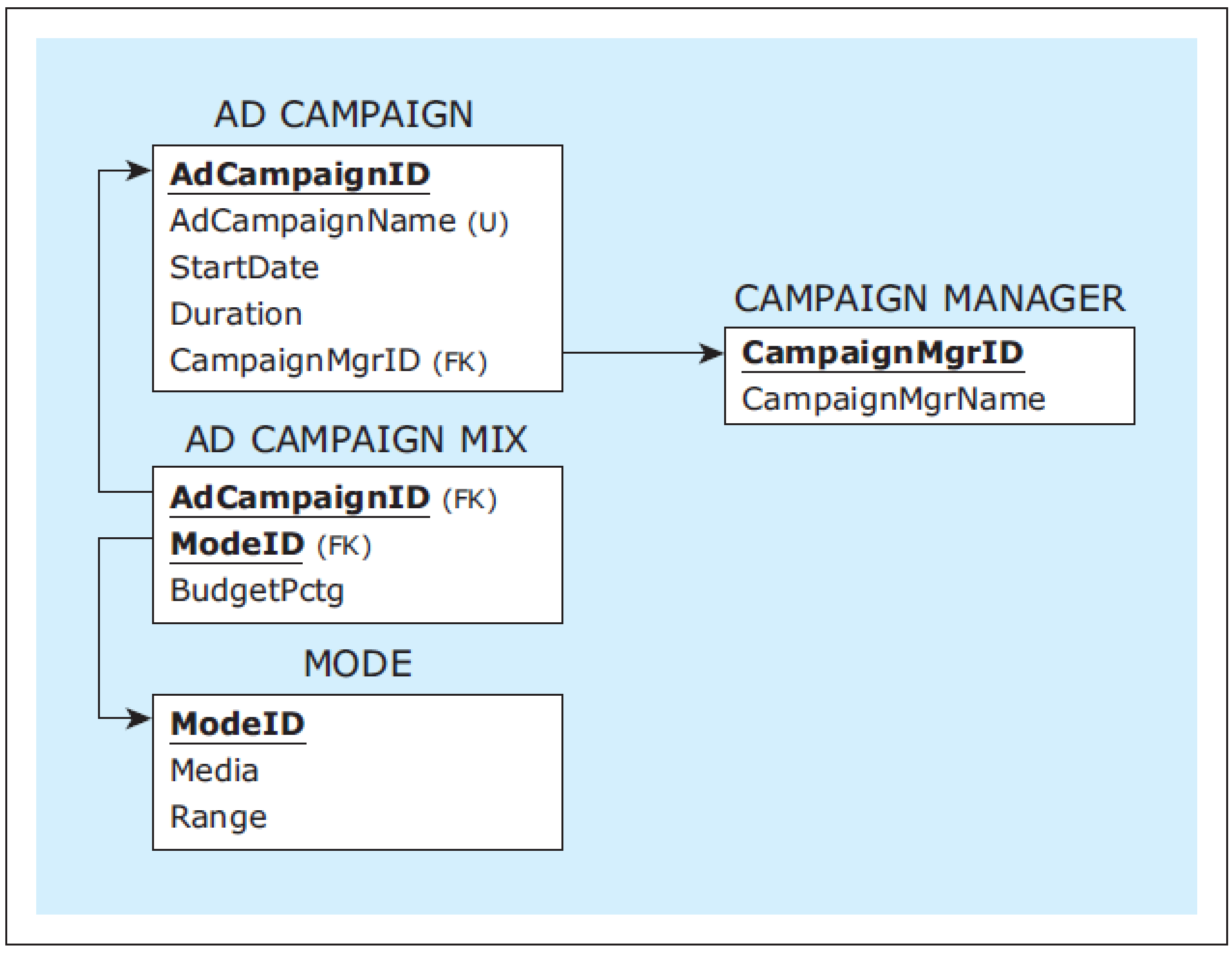
AD CAMPAIGN	AdCampaignID	AdCampaignName	StartDate	Duration	CampaignMgrID
MODE	ModelD	Media	Range		
AD CAMPAIGN MIX	AdCampaignID	ModelD	BudgetPctg		

New

CAMPAIGN MANAGER	CampaignMgrID	CampaignMgrName

Example: Normalizing a table to 3NF

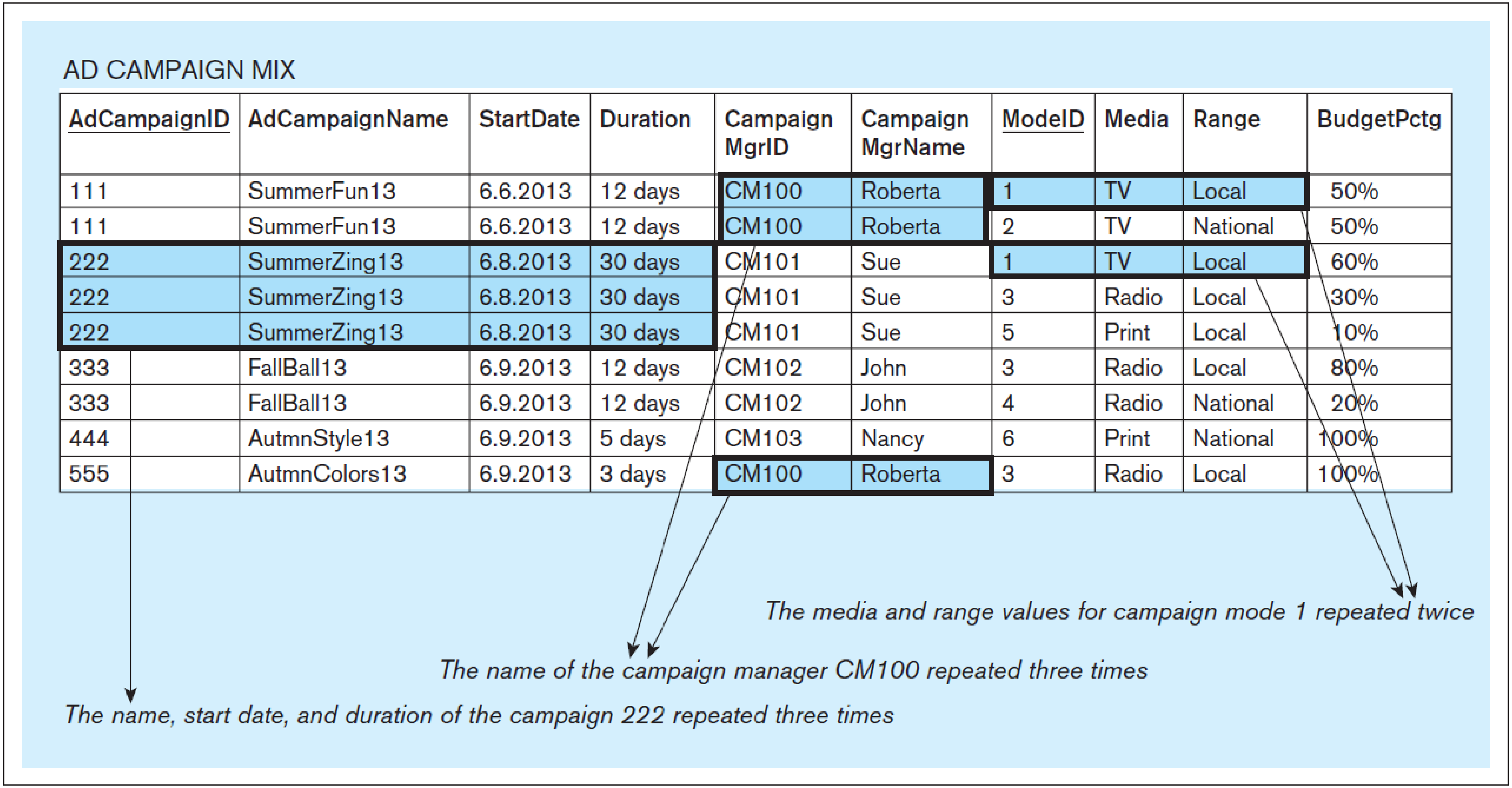
Pressly Ad Agency example – relational schema of 3NF relations



NORMALIZATION

- **Eliminate Redundancy and Resolve Update Anomalies**
  - Normalizing to 3NF eliminates unnecessary duplicates (redundancy)
    - Example:
      - Ad Campaign Agency
      - Duplicate foreign key values are necessary to connect relations.

Example relation AD CAMPAIGN MIX (contains redundant data)



AD CAMPAIGN MIX

AdCampaignID	AdCampaignName	StartDate	Duration	Campaign MgrID	Campaign MgrName	ModelD	Media	Range	BudgetPctg
111	SummerFun13	6.6.2013	12 days	CM100	Roberta	1	TV	Local	50%
111	SummerFun13	6.6.2013	12 days	CM100	Roberta	2	TV	National	50%
222	SummerZing13	6.8.2013	30 days	CM101	Sue	1	TV	Local	60%
222	SummerZing13	6.8.2013	30 days	CM101	Sue	3	Radio	Local	30%
222	SummerZing13	6.8.2013	30 days	CM101	Sue	5	Print	Local	10%
333	FallBall13	6.9.2013	12 days	CM102	John	3	Radio	Local	80%
333	FallBall13	6.9.2013	12 days	CM102	John	4	Radio	National	20%
444	AutmnStyle13	6.9.2013	5 days	CM103	Nancy	6	Print	National	100%
555	AutmnColors13	6.9.2013	3 days	CM100	Roberta	3	Radio	Local	100%

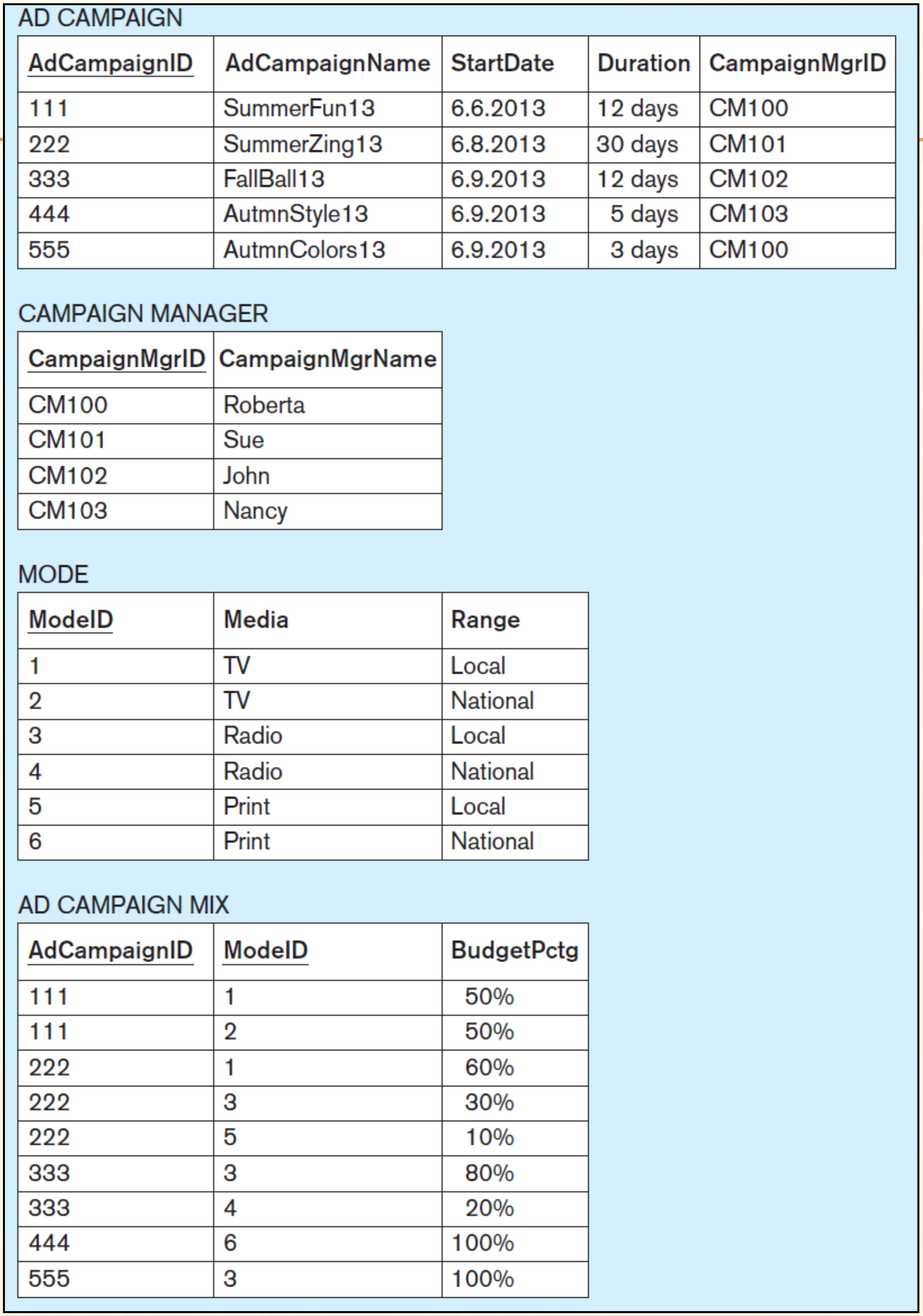
The media and range values for campaign mode 1 repeated twice

The name of the campaign manager CM100 repeated three times

The name, start date, and duration of the campaign 222 repeated three times

Pressly Ad Agency example — **normalized relations with data**

- No unnecessarily duplicate data
- May have necessary duplicate foreign key values



AD CAMPAIGN

AdCampaignID	AdCampaignName	StartDate	Duration	CampaignMgrID
111	SummerFun13	6.6.2013	12 days	CM100
222	SummerZing13	6.8.2013	30 days	CM101
333	FallBall13	6.9.2013	12 days	CM102
444	AutmnStyle13	6.9.2013	5 days	CM103
555	AutmnColors13	6.9.2013	3 days	CM100

CAMPAIGN MANAGER

CampaignMgrID	CampaignMgrName
CM100	Roberta
CM101	Sue
CM102	John
CM103	Nancy

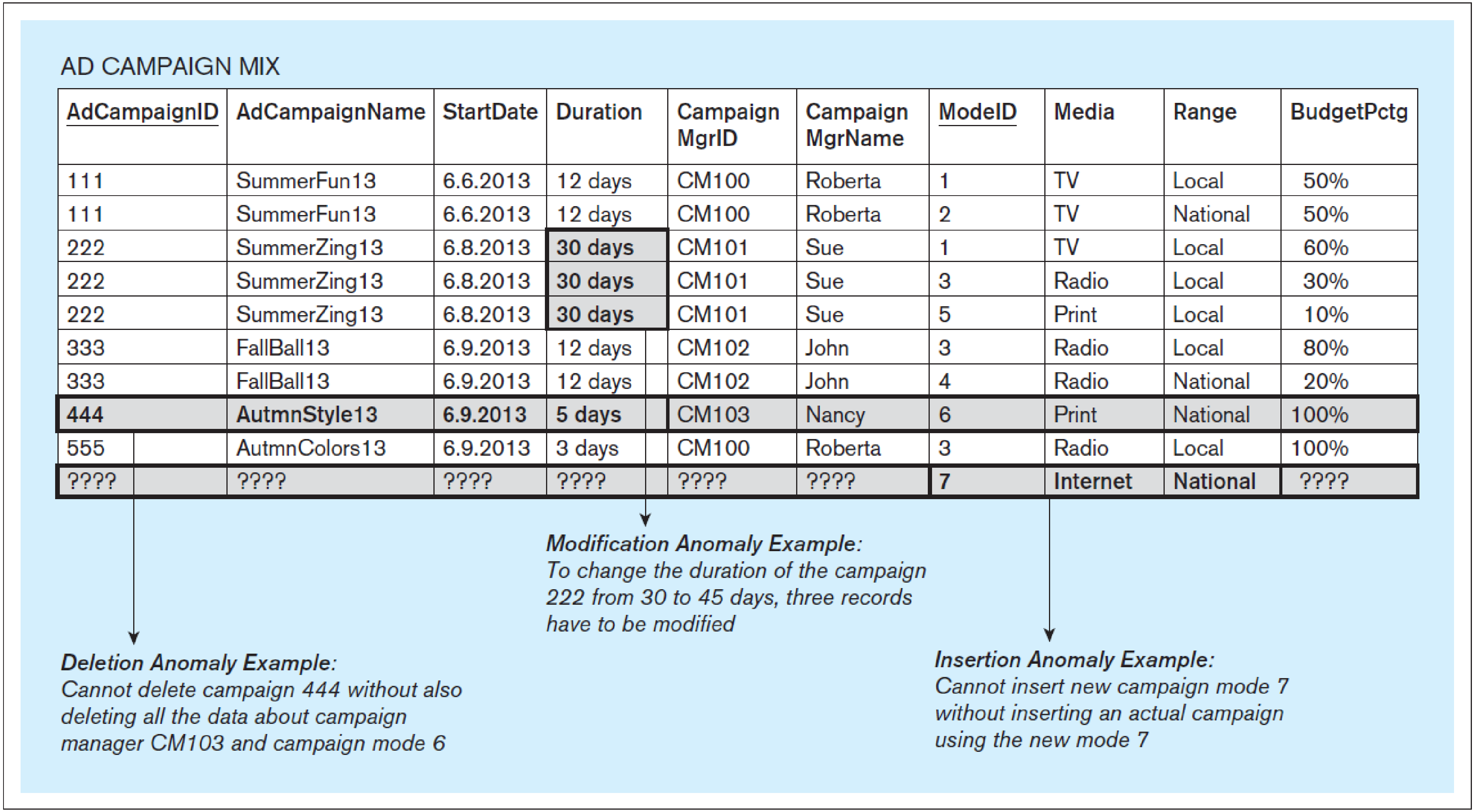
MODE

ModeID	Media	Range
1	TV	Local
2	TV	National
3	Radio	Local
4	Radio	National
5	Print	Local
6	Print	National

AD CAMPAIGN MIX

AdCampaignID	ModeID	BudgetPctg
111	1	50%
111	2	50%
222	1	60%
222	3	30%
222	5	10%
333	3	80%
333	4	20%
444	6	100%
555	3	100%

Examples of update anomalies in the relation AD CAMPAIGN MIX



AD CAMPAIGN MIX

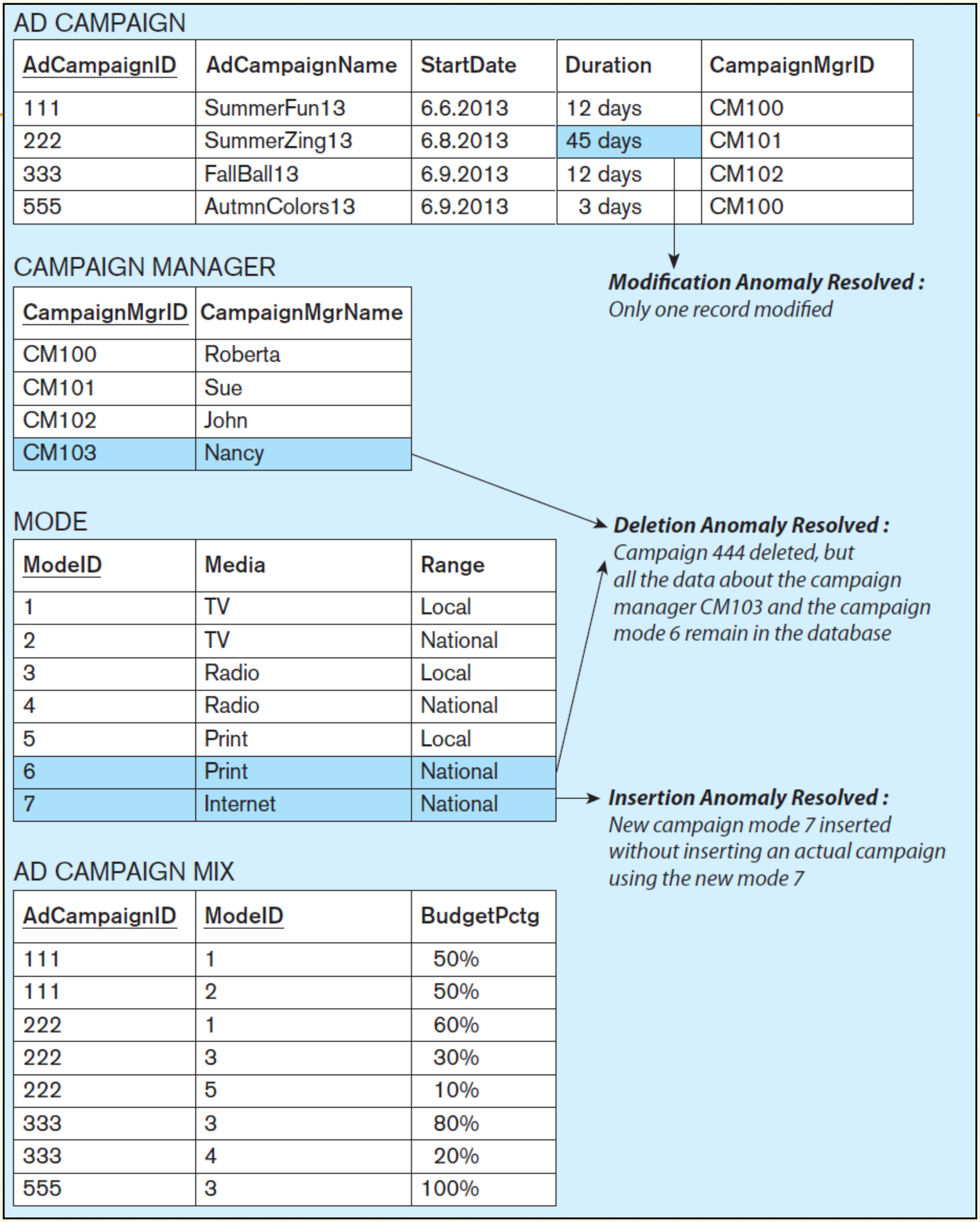
AdCampaignID	AdCampaignName	StartDate	Duration	Campaign MgrID	Campaign MgrName	ModelD	Media	Range	BudgetPctg
111	SummerFun13	6.6.2013	12 days	CM100	Roberta	1	TV	Local	50%
111	SummerFun13	6.6.2013	12 days	CM100	Roberta	2	TV	National	50%
222	SummerZing13	6.8.2013	**30 days**	CM101	Sue	1	TV	Local	60%
222	SummerZing13	6.8.2013	**30 days**	CM101	Sue	3	Radio	Local	30%
222	SummerZing13	6.8.2013	**30 days**	CM101	Sue	5	Print	Local	10%
333	FallBall13	6.9.2013	12 days	CM102	John	3	Radio	Local	80%
333	FallBall13	6.9.2013	12 days	CM102	John	4	Radio	National	20%
444	**AutmnStyle13**	**6.9.2013**	**5 days**	CM103	Nancy	6	Print	National	100%
555	AutmnColors13	6.9.2013	3 days	CM100	Roberta	3	Radio	Local	100%
????	????	????	????	????	????	**7**	**Internet**	**National**	????

Modification Anomaly Example:
To change the duration of the campaign 222 from 30 to 45 days, three records have to be modified

Deletion Anomaly Example:
Cannot delete campaign 444 without also deleting all the data about campaign manager CM103 and campaign mode 6

Insertion Anomaly Example:
Cannot insert new campaign mode 7 without inserting an actual campaign using the new mode 7

Pressly Ad Agency example—normalized relations with data, update anomalies resolved



AD CAMPAIGN

AdCampaignID	AdCampaignName	StartDate	Duration	CampaignMgrID
111	SummerFun13	6.6.2013	12 days	CM100
222	SummerZing13	6.8.2013	45 days	CM101
333	FallBall13	6.9.2013	12 days	CM102
555	AutmnColors13	6.9.2013	3 days	CM100

Modification Anomaly Resolved :
Only one record modified

CAMPAIGN MANAGER

CampaignMgrID	CampaignMgrName
CM100	Roberta
CM101	Sue
CM102	John
CM103	Nancy

MODE

ModeID	Media	Range
1	TV	Local
2	TV	National
3	Radio	Local
4	Radio	National
5	Print	Local
6	Print	National
7	Internet	National

Deletion Anomaly Resolved :
Campaign 444 deleted, but all the data about the campaign manager CM103 and the campaign mode 6 remain in the database

Insertion Anomaly Resolved :
New campaign mode 7 inserted without inserting an actual campaign using the new mode 7

AD CAMPAIGN MIX

AdCampaignID	ModeID	BudgetPctg
111	1	50%
111	2	50%
222	1	60%
222	3	30%
222	5	10%
333	3	80%
333	4	20%
555	3	100%

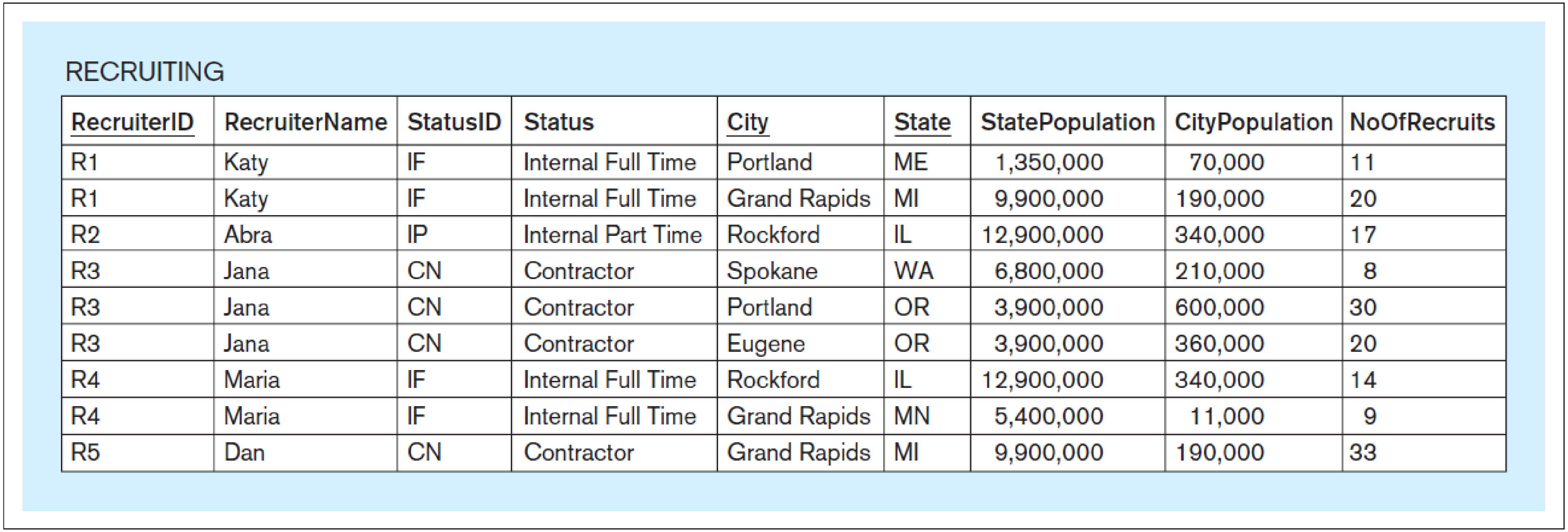
Functional Dependencies: Example 2: relation RECRUITING

- **Example: A university recruiting data**
  - Use one relation RECRUITING to represent and store all recruiters, recruiting cities and states, and recruiting results
  - All columns in relation RECRUITING
    - RecruiterID
    - RecruiterName
    - StatusID
    - Status – text description
    - City
    - State
    - StatePopulation
    - CityPopulation
    - NoOfRecruits – from one city

- A recruiter can recruit from multiple cities.
- The same city can have more than one recruiters working there.
- There may be same city names in different states, but not in the same state.

Functional Dependencies: Example 2: relation RECRUITING

Table columns and data



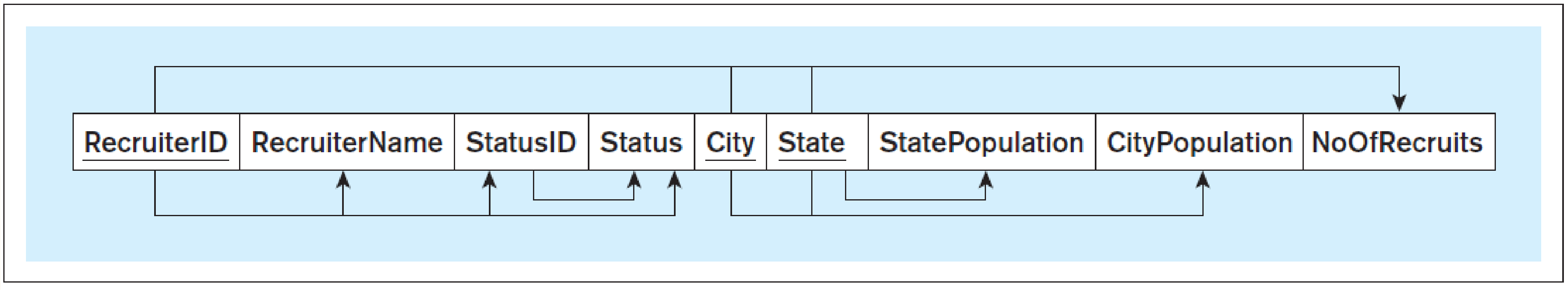
RECRUITING

RecruiterID	RecruiterName	StatusID	Status	City	State	StatePopulation	CityPopulation	NoOfRecruits
R1	Katy	IF	Internal Full Time	Portland	ME	1,350,000	70,000	11
R1	Katy	IF	Internal Full Time	Grand Rapids	MI	9,900,000	190,000	20
R2	Abra	IP	Internal Part Time	Rockford	IL	12,900,000	340,000	17
R3	Jana	CN	Contractor	Spokane	WA	6,800,000	210,000	8
R3	Jana	CN	Contractor	Portland	OR	3,900,000	600,000	30
R3	Jana	CN	Contractor	Eugene	OR	3,900,000	360,000	20
R4	Maria	IF	Internal Full Time	Rockford	IL	12,900,000	340,000	14
R4	Maria	IF	Internal Full Time	Grand Rapids	MN	5,400,000	11,000	9
R5	Dan	CN	Contractor	Grand Rapids	MI	9,900,000	190,000	33

- Composite primary key
  - RecruiterID, City, State
- Each row
  - A specific recruiter’s recruiting result in one city

Functional Dependencies: Example 2: relation RECRUITING

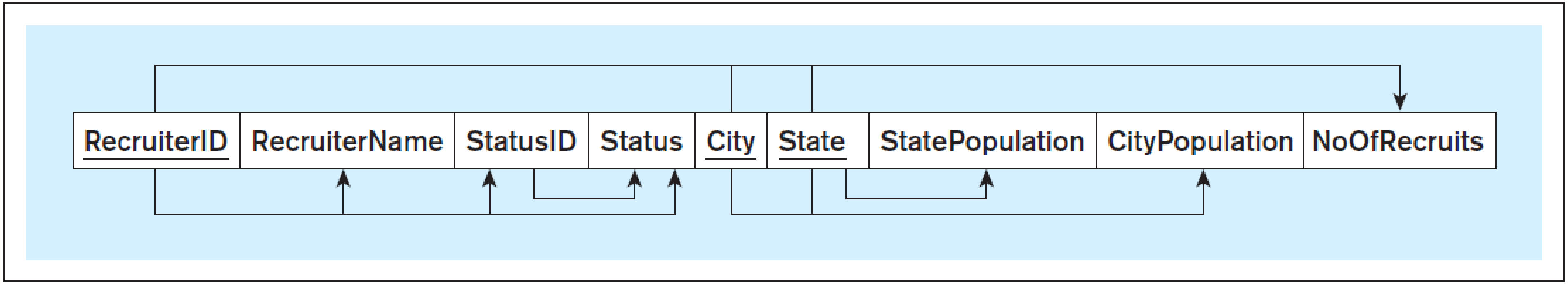
Functional dependencies in the table



RecruiterID -> RecuriterName, StatusID, Status
StatusID -> Status
City, State -> CityPopulation
State -> StatePopulation
RecruiterID, City, State -> NoOfRecruits

Functional Dependencies:
Example 2: relation RECRUITING

Types of functional dependencies in the table:



RecruiterID	RecruiterName	StatusID	Status	City	State	StatePopulation	CityPopulation	NoOfRecruits

Type	Functional Dependency
Partial	RecruiterID -> RecuriterName, StatusID, Status
Transitive	StatusID -> Status
Partial	City, State -> CityPopulation
Partial	State -> StatePopulation
Full	RecruiterID, City, State -> NoOfRecruits