6.1 First, second, and third normal form

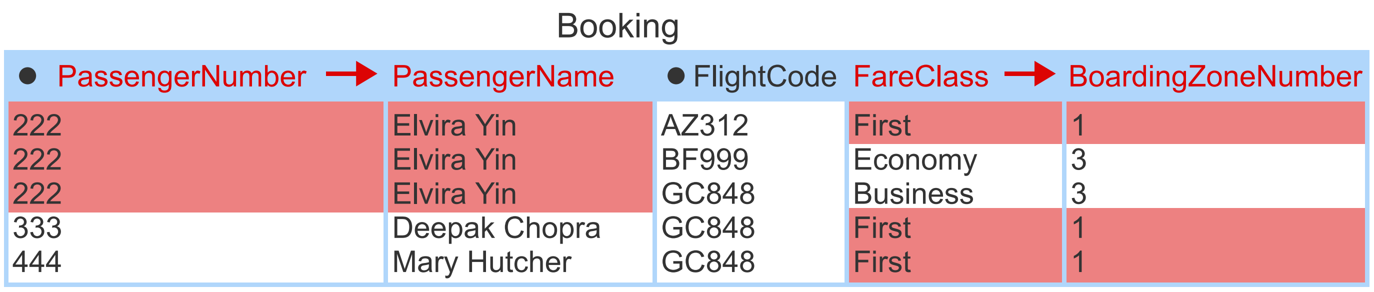
Functional dependence

Column A ***depends on*** column B means each B value is related to at most one A value. Columns A and B may be simple or composite. 'A depends on B' is denoted B → A.

Dependence of one column on another is called ***functional dependence***. Functional dependence reflects business rules. Ex: "Each student receives one letter grade in a course" indicates the Grade column depends on the composite column (StudentID, CourseCode).

Examples in this section illustrate functional dependence with static table data. However, functional dependence cannot be inferred from values in a table at one point in time. Today, each value of column B may relate to at most one value of column A, but future updates may alter the data.

PARTICIPATION ACTIVITY 6.1.1: Functional dependence.


Booking
● PassengerNumber
PassengerName
● FlightCode
FareClass
BoardingZoneNumber

● PassengerNumber	PassengerName	● FlightCode	FareClass	BoardingZoneNumber
222	Elvira Yin	AZ312	First	1
222	Elvira Yin	BF999	Economy	3
222	Elvira Yin	GC848	Business	3
333	Deepak Chopra	GC848	First	1
444	Mary Hutcher	GC848	First	1

Animation content:

Static figure:
Table Booking has columns PassengerNumber, PassengerName, FlightCode, FareClass, and BoardingZoneNumber. (PassengerNumber, FlightCode) is the primary key. An arrow points from PasengerNumber to PassengerName. An arrow points from FareClass to BoardingZoneNumber.
Booking has five rows:
222, Elvira Yin, AZ312, First, 1
222, Elvira Yin, BF999, Economy, 3
222, Elvira Yin, GC848, Business, 3
333, Deepak Chopra, GC848, First, 1
444, Mary Hutcher, GC848, First, 1

The three instances of (222, Elvira Yin) are highlighted. The three instances of (First, 1) are highlighted.

Step 1: Each passenger number always has the same name, so PassengerName depends on PassengerNumber. PassengerNumber, PassengerName, and the arrow between are highlighted.

Step 2: Ex: Passenger 222 is always named Elvira Yin. The three instances of (222, Elvira Yin) are highlighted.

Step 3: Each fare class always has the same boarding zone, so BoardingZoneNumber depends on FareClass. FareClass, BoardingZoneNumber, and the arrow between are highlighted.

Step 4: Ex: First class is always assigned zone 1. The three instances of (First, 1) are highlighted.

Animation captions:

1. Each passenger number always has the same name, so PassengerName depends on PassengerNumber.
2. Ex: Passenger 222 is always named Elvira Yin.
3. Each fare class always has the same boarding zone, so BoardingZoneNumber depends on FareClass.
4. Ex: First class is always assigned zone 1.

Functional dependence is not the only type of dependence. ***Multivalued dependence*** and ***join dependence*** entail dependencies between three or more columns. However, multivalued and join dependencies are complex, uncommon, and not discussed in this material.

PARTICIPATION ACTIVITY 6.1.2: Functional dependence.

Refer to the Booking table in the animation above.

1) B → A means each value of A relates to at most one value of B.

○ True

○ False

2) 'PassengerName depends on PassengerNumber' means each name is related to at most one number.

- True
- False

3) "Column A does *not* depend on column B" can, in some cases, be inferred from static data in columns A and B.
   - True
   - False

4) FareClass depends on BoardingZoneNumber.
   - True
   - False

5) FareClass depends on PassengerNumber.
   - True
   - False

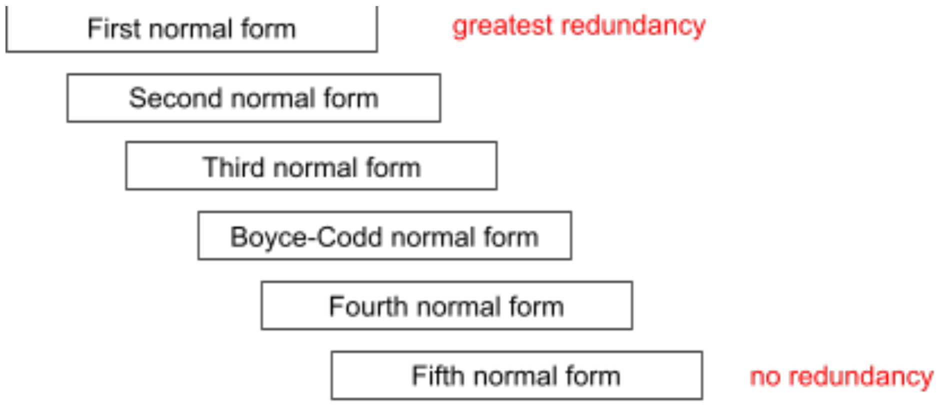
6) FareClass depends on the primary key (PassengerNumber, FlightCode).
   - True
   - False

Normal forms

Redundancy is the repetition of related values in a table. Ex: In the Booking table, above, (222, Elvira Yin) is repeated. Redundancy causes database management problems. When related values are updated, all copies must be changed, which makes queries slow and complex. If copies are not updated uniformly, the copies become inconsistent and the correct version is uncertain.

Normal forms are rules for designing tables with less redundancy. Normal forms are numbered, first through fifth. An additional normal form, Boyce-Codd, is an improved version of third normal form. The six normal forms comprise a sequence, with each successive normal form allowing less redundancy.

Figure 6.1.1: Normal forms.



Redundancy occurs when a dependence is on a column that is not unique. Ex: In the Booking table, (222, Elvira Yin) is repeated because PassengerName depends on PassengerNumber, which is not unique. Boyce-Codd normal form eliminates all dependencies on non-unique columns and, in practice, is the most important normal form.

Fourth and fifth normal forms eliminate multivalued and join dependencies, respectively. Since multivalued and join dependencies are complex and uncommon, fourth and fifth normal forms are primarily of theoretical interest.

First, second, and third normal forms are described below. Boyce-Codd normal form is described elsewhere in this material. Fourth and fifth normal forms are not described in this material.

PARTICIPATION ACTIVITY 6.1.3: Redundancy.

1) Redundancy is the repetition of an individual value.

- True
- False

2) Tables that allow redundancy might contain inconsistent data.

- True
- False

3) Dependence on a primary key can cause redundancy.

- True
- False

PARTICIPATION ACTIVITY 6.1.4: Normal forms.

Match each normal form to a description.

If unable to drag and drop, refresh the page.

Boyce-Codd normal form   First normal form   Fifth normal form

Fourth normal form

	Eliminates multivalued dependencies and associated redundancy.
	Eliminates all redundancy arising from dependencies on non-unique columns.
	Allows the most redundancy of any normal form.
	Eliminates join dependencies and associated redundancy.

Reset

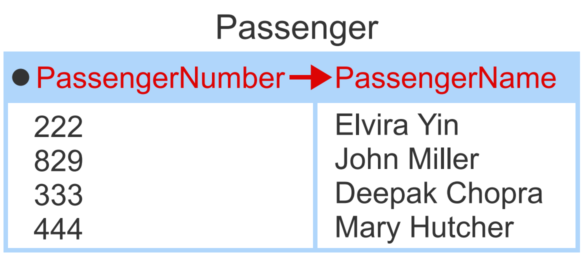
First normal form

Every cell of a table contains exactly one value. A table is in ***first normal form*** when, in addition, the table has a primary key. This definition has two corollaries:

- *In a first normal form table, every non-key column depends on the primary key*. Each primary key value appears in exactly one row, and each non-key cell contains exactly one value. So each primary key value is related to exactly one non-key value.
- *A first normal form table has no duplicate rows*. Every row contains a different primary key value and therefore every row is different.

In practice, databases allow tables with duplicate rows and no primary key. However, such tables are usually temporary. Ex: A database user might load external data with duplicate rows into a temporary table. Normally, when data is moved to a permanent table, duplicate rows are removed and a primary key is created.

PARTICIPATION ACTIVITY | 6.1.5: In a first normal form table, every non-key column depends on the primary key.



First normal form

Animation content:

Static figure:
Table Passenger has columns PassengerNumber and PassengerName. PassengerNumber is a primary key. An arrow points from PassengerNumber to PassengerName. Passenger and has four rows:
222, Elvira Yin
829, John Miller
333, Depak Chopra
444, Mary Hutcher

Passenger is labeled First normal form.

Step 1: The Passenger table has a primary key and therefore is in first normal form. PassengerNumber is highlighted.

Step 2: Since the primary key is unique, each primary key value appears on one row only. The values in the PassengerNumber column are highlighted.

Step 3: Since each cell contains one value, each primary key value is related to one PassengerName value. Each row is highlighted

Step 4: Therefore, PassengerName depends on PassengerNumber. PassengerNumber, PassengerName, and the arrow between are highlighted.

Animation captions:

1. The Passenger table has a primary key and therefore is in first normal form.
2. Since the primary key is unique, each primary key value appears on one row only.
3. Since each cell contains one value, each primary key value is related to one PassengerName value.
4. Therefore, PassengerName depends on PassengerNumber.

Alternative definitions of first normal form

First normal form is commonly defined in several ways:

- *The table has a primary key.*
- *Every non-key column depends on the primary key.*
- *The table cannot have duplicate rows.*
- *Every cell contains exactly one value.*

The first three definitions are equivalent, but the last is different. The last definition is true of any relational table, and allows for duplicate rows and no primary key.

PARTICIPATION ACTIVITY 6.1.6: First normal form.

1) Since a string contains multiple characters, a string represents multiple values in a table cell.

- True
- False

2) In a table that never contains duplicate rows, non-key columns always depend on the primary key.

- True
- False

3) In a first normal form table, non-key columns depend *only* on the primary key.

True

False

4) In relational theory, all tables are in first normal form.

True

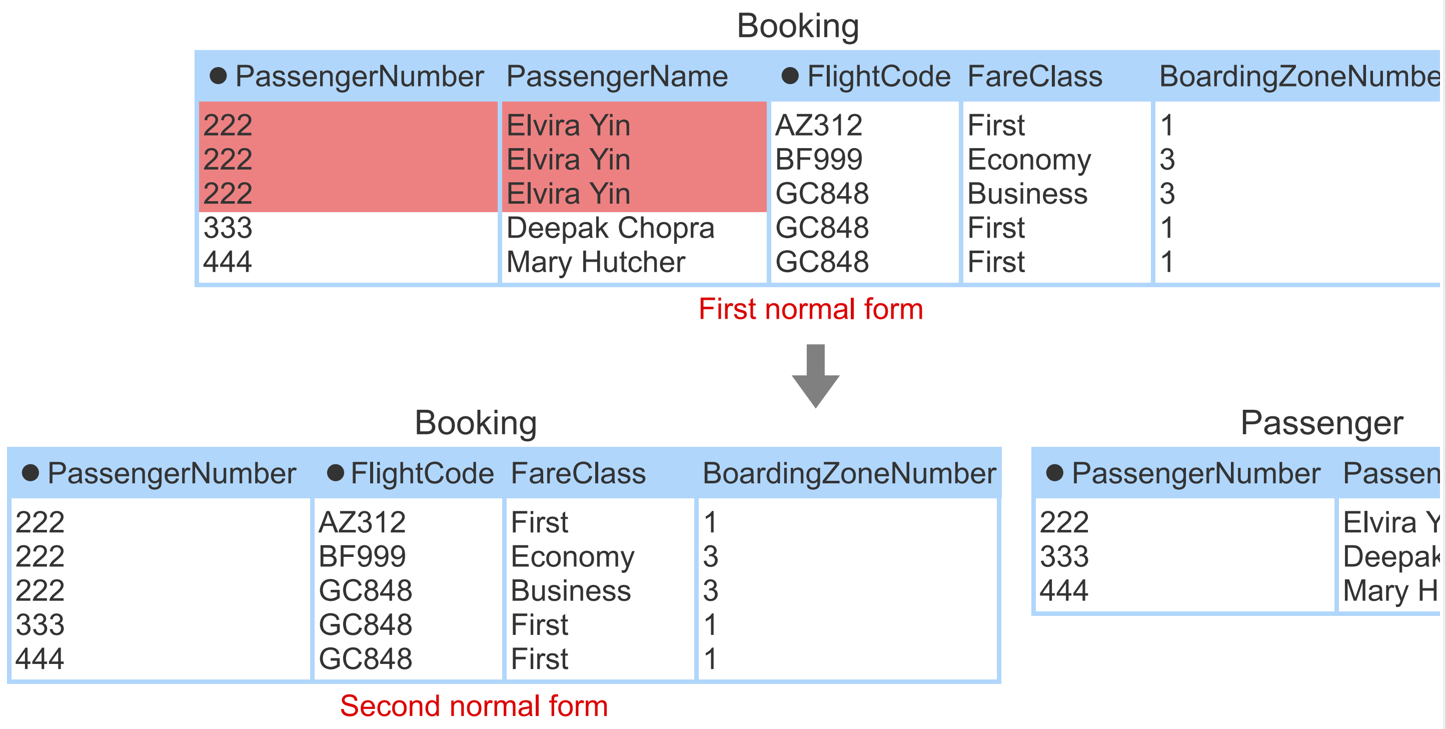
False

Second normal form

A table is in ***second normal form*** when all non-key columns depend on the whole primary key. In other words, a non-key column cannot depend on part of a composite primary key. A table with a simple primary key is automatically in second normal form.

A table in second normal form is also in first normal form, by definition.

PARTICIPATION ACTIVITY 6.1.7: Second normal form eliminates some redundancy.



Booking

● PassengerNumber	PassengerName	● FlightCode	FareClass	BoardingZoneNumbe
222	Elvira Yin	AZ312	First	1
222	Elvira Yin	BF999	Economy	3
222	Elvira Yin	GC848	Business	3
333	Deepak Chopra	GC848	First	1
444	Mary Hutcher	GC848	First	1

First normal form

Booking

● PassengerNumber	● FlightCode	FareClass	BoardingZoneNumber
222	AZ312	First	1
222	BF999	Economy	3
222	GC848	Business	3
333	GC848	First	1
444	GC848	First	1

Second normal form

Passenger

● PassengerNumber	Passen
222	Elvira Y
333	Deepak
444	Mary H

Animation content:

Static figure:
Three tables appear, named Booking, Booking, and Passenger. A large arrow indicates the first

Booking table is decomposed into the second Booking table and the Passenger table.

The first Booking table has columns PassengerNumber, PassengerName, FlightCode, FareClass, and BoardingZoneNumber. (PassengerNumber, FlightCode) is the primary key. Booking has caption First normal form and five rows:
222, Elvira Yin, AZ312, First, 1
222, Elvira Yin, BF999, Economy, 3
222, Elvira Yin, GC848, Business, 3
333, Deepak Chopra, GC848, First, 1
444, Mary Hutcher, GC848, First, 1

The three instances of (222, Elvira Yin) are highlighted.

The second Booking table is the same as the first, except the PassengerName column does not appear. The caption is Second normal form.

The Passenger table has columns PassengerNumber and PassengerName. PassengerNumber is the primary key. Passenger has three rows:
222, Elvira Yin, GC848
333, Deepak Chopra
444, Mary Hutcher

Step 1: PassengerName depends on PassengerNumber. Dependence on part of the primary key causes repetition of (222, Elvira Yin). The first Booking table appears. The three instances of (222, Elvira Yin) are highlighted.

Step 2: Removing PassengerName from Booking eliminates redundancy. Booking is now in second normal form. An X appears above column PassengerName. The second Booking table appears.

Step 3: PassengerName moves to the Passenger table. The Passenger table has no redundancies, since PassengerName depends on the whole primary key. The Passenger table appears.

Animation captions:

1. PassengerName depends on PassengerNumber. Dependence on part of the primary key causes repetition of (222, Elvira Yin).
2. Removing PassengerName from Booking eliminates redundancy. Booking is now in second normal form.
3. PassengerName moves to the Passenger table. The Passenger table has no redundancies, since PassengerName depends on the whole primary key

PARTICIPATION ACTIVITY | 6.1.8: Second normal form.

The table below lists courses taken by students, along with student scores in the course, student email addresses, and letter grades. Each student has exactly one email address.

StudentRecord

● StudentNumber	● CourseNumber	ScoreNumber	EmailAddress	GradeLetter
8034	Math 100	95	john@gmail.com	A
2111	Math 100	73	sammy@icloud.com	C
9930	Spanish 22A	89	abc@hotmail.com	B
8034	History 11	89	john@gmail.com	B
5091	Biology 200B	41	maria@sbc.net	F

1) Which pair of related values is repeated?

- (8034, john@gmail.com)
- (2111, Math 100)
- (Math 100)

2) Which non-key column does not depend on the whole primary key?

- EmailAddress
- StudentNumber
- ScoreNumber

3) Which column must be removed to achieve second normal form?

- ScoreNumber
- CourseNumber
- EmailAddress

4) A table with a simple primary key must be in _____.

- first but not second normal form

○ second normal form or higher

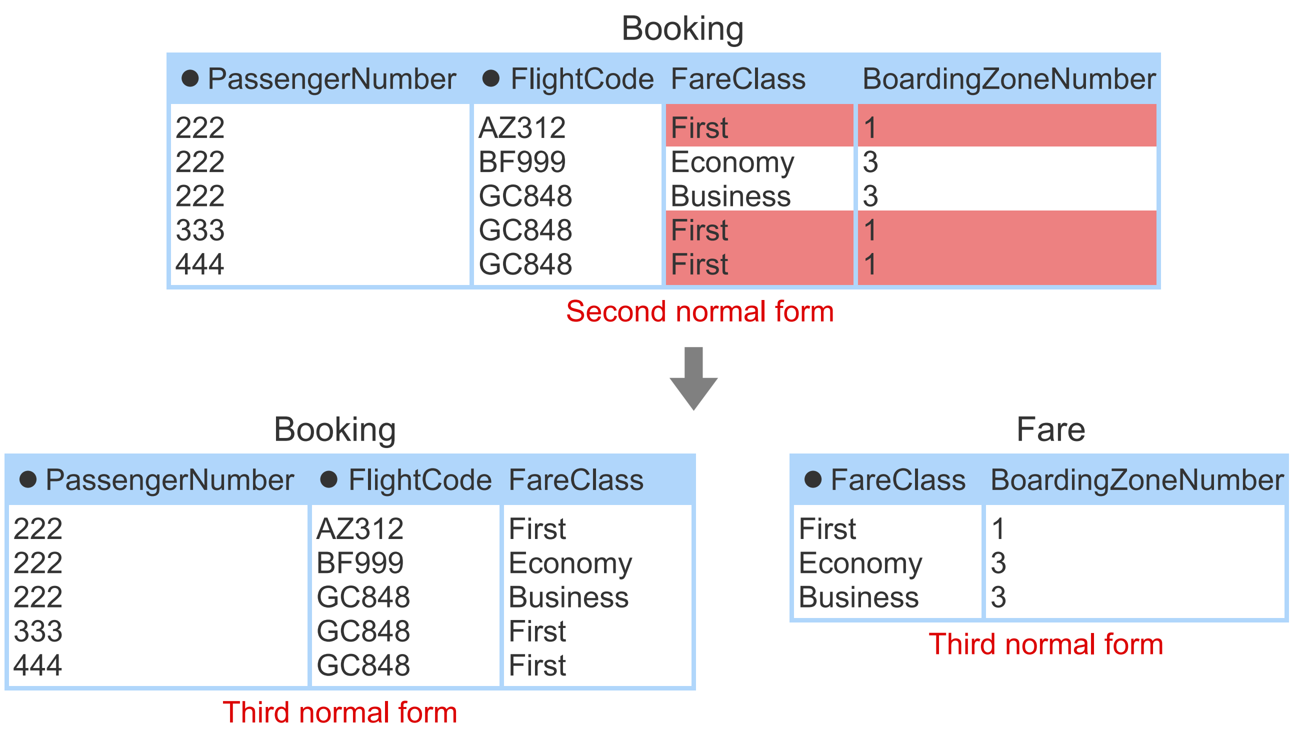
○ third normal form

Third normal form

Redundancy can occur in a second normal form table when a non-key column depends on another non-key column. Informally, a table is in ***third normal form*** when all non-key columns depend on the key, the whole key, and nothing but the key. A formal definition appears elsewhere in this material.

A table in third normal form is also in first and second normal forms, by definition.

PARTICIPATION ACTIVITY 6.1.9: Third normal form eliminates most redundancy.



Booking

● PassengerNumber	● FlightCode	FareClass	BoardingZoneNumber
222	AZ312	First	1
222	BF999	Economy	3
222	GC848	Business	3
333	GC848	First	1
444	GC848	First	1

Second normal form

Booking

● PassengerNumber	● FlightCode	FareClass
222	AZ312	First
222	BF999	Economy
222	GC848	Business
333	GC848	First
444	GC848	First

Third normal form

Fare

● FareClass	BoardingZoneNumber
First	1
Economy	3
Business	3

Third normal form

Animation content:

Static figure:
Three tables appear, named Booking, Booking, and Fare. A large arrow indicates the first Booking table is decomposed into the second Booking table and the Fare table.

The first Booking table has columns PassengerNumber, FlightCode, FareClass, and BoardingZoneNumber. (PassengerNumber, FlightCode) is the primary key. Booking has caption

Second normal form and five rows:
222, AZ312, First, 1
222, BF999, Economy, 3
222, GC848, Business, 3
333, GC848, First, 1
444, GC848, First, 1

The three instances of (First, 1) are highlighted.

The second Booking table is the same as the first, except the BoardingZoneNumber column does not appear. The caption is Third normal form.

The Fare table has columns FareClase and BoardingZoneNumber. FareClass is the primary key. Fare has caption Third normal form and three rows:
First, 1
Economy, 3
Business, 3

Step 1: BoardingZoneNumber depends on FareClass. Dependence on a non-key column causes the redundancy (First, 1). The first Booking table appears. The three instances of (First, 1) are highlighted.

Step 2: Removing BoardingZoneNumber from Booking eliminates redundancy. Booking is now in third normal form. An X appears above the BoardingZoneNumber column. The second Booking table appears.

Step 3: BoardingZoneNumber moves to the Fare table. The Fare table has no redundancies since BoardingZoneNumber depends on the primary key. The Fare table appears.

Animation captions:

1. BoardingZoneNumber depends on FareClass. Dependence on a non-key column causes the redundancy (First, 1).
2. Removing BoardingZoneNumber from Booking eliminates redundancy. Booking is now in third normal form.
3. BoardingZoneNumber moves to the Fare table. The Fare table has no redundancies since BoardingZoneNumber depends on the primary key.

PARTICIPATION ACTIVITY 6.1.10: Third normal form.

Refer to the second normal form table, below. Scores of 90 and above receive an A grade, 80 to 89 a B, and so on.

StudentRecord

● StudentNumber	● CourseNumber	ScoreNumber	GradeLetter
8034	Math 100	95	A
2111	Math 100	88	B
9930	Spanish 22A	72	C
8034	History 11	88	B
5091	Biology 200B	41	F

1) Which fact is repeated?

- (8034)
- (88, B)
- (9930, Spanish 22A)

2) Which non-key column depends on another non-key column?

- GradeLetter
- ScoreNumber
- CourseNumber

3) Which column must be removed to achieve third normal form?

- GradeLetter
- ScoreNumber
- CourseNumber

6.2 Boyce-Codd normal form

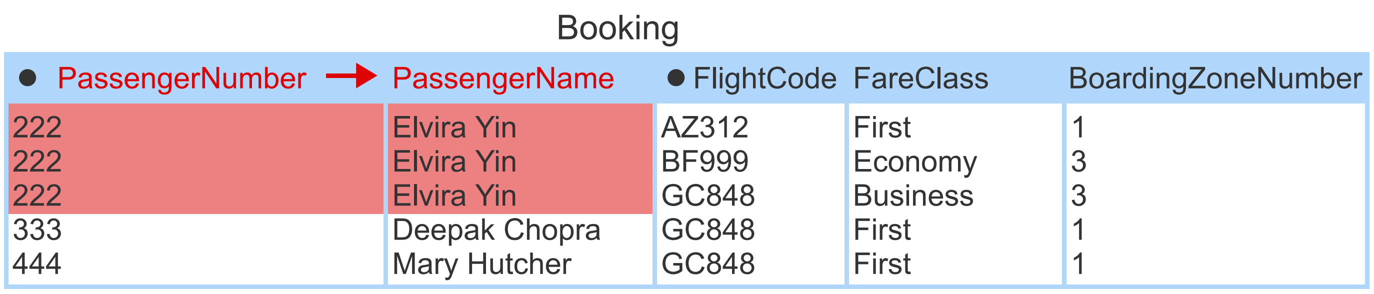
Redundancy and dependence

Column A depends on column B means each B value is related to at most one A value. Columns A and B may be simple or composite. 'A depends on B' is denoted B → A. Dependence of one column on another is called *functional dependence*.

Redundancy occurs when a column depends on another column that is not unique

Redundancy occurs when a column depends on another column that is not unique.

PARTICIPATION ACTIVITY 6.2.1: Redundancy and dependence.



Booking

● PassengerNumber →	PassengerName	● FlightCode	FareClass	BoardingZoneNumber
222	Elvira Yin	AZ312	First	1
222	Elvira Yin	BF999	Economy	3
222	Elvira Yin	GC848	Business	3
333	Deepak Chopra	GC848	First	1
444	Mary Hutcher	GC848	First	1

Animation content:

Static figure:
Table Booking has columns PassengerNumber, PassengerName, FlightCode, FareClass, and BoardingZoneNumber. (PassengerNumber, FlightCode) is the primary key. An arrow points from PasengerNumber to PassengerName. Booking has five rows:
222, Elvira Yin, AZ312, First, 1
222, Elvira Yin, BF999, Economy, 3
222, Elvira Yin, GC848, Business, 3
333, Deepak Chopra, GC848, First, 1
444, Mary Hutcher, GC848, First, 1

The three instances of (222, Elvira Yin) are highlighted.

Step 1: Since a passenger number always identifies the same name, PassengerName depends on PassengerNumber. PassengerNumber, PassengerName, and the arrow between are highlighted.

Step 2: Since a passenger may have bookings on several flights, PassengerNumber is not unique. The three instances of 222 are highlighted.

Step 3: The dependence of PassengerName on a non-unique column creates redundancy. The three instances of (222, Elvira Yin) are highlighted.

Animation captions:

1. Since a passenger number always identifies the same name, PassengerName depends on PassengerNumber.

2. Since a passenger may have bookings on several flights, PassengerNumber is not unique.
3. The dependence of PassengerName on a non-unique column creates redundancy.

In a Boyce-Codd normal form table, all dependencies are on unique columns. Dependence on a unique column never creates redundancy, so Boyce-Codd normal form eliminates all redundancy arising from functional dependence.

PARTICIPATION ACTIVITY 6.2.2: Redundancy and dependence.

Refer to the Booking table in the animation above.

1) What column does BoardingZoneNumber depend on?
   - PassengerName
   - FlightCode
   - FareClass

2) Is FareClass unique?
   - Yes
   - No
   - Cannot determine from table data

3) Does the dependency FareClass → BoardingZoneNumber create redundancy?
   - Yes
   - No
   - Cannot determine from table data

4) A table with a dependency on a non-unique column _____ contains redundant data.
   - always
   - often
   - never

Third normal form

Informally, a table is in third normal form when all non-key columns depend on the key, the whole key, and nothing but the key. This definition is accurate when the primary key is the only unique column. The formal definition, below, accounts for tables with several unique columns.

A ***candidate key*** is a simple or composite column that is unique and minimal. ***Minimal*** means all columns are necessary for uniqueness. A table may have several candidate keys. The database designer designates one candidate key as the primary key.

A ***non-key*** column is a column that is not contained in a candidate key.

A table is in ***third normal form*** if, whenever a non-key column A depends on column B, then B is unique. Columns A and B may be simple or composite. Although B is unique, B is not necessarily minimal and therefore is not necessarily a candidate key.

PARTICIPATION ACTIVITY 6.2.3: Third normal form.

Refer to the following table. Each department has one chair, but the same person occasionally chairs several departments. Course names can be repeated in different departments, but never within the same department.

Course

●CourseCode	DepartmentCode	CourseName	DepartmentChair
8451	MATH	Discrete Mathematics	Azim Rafiq
8452	CS	Discrete Mathematics	Je Soomin
2391	SPAN	Introduction to Spanish	Susan Williams
5505	BIO	Genetics and Evolution	Susan Williams
8449	MATH	Calculus I	Azim Rafiq
8036	MATH	Differential Equations	Azim Rafiq

1) Which fact is repeated?

- ○ Discrete Mathematics
- ○ (MATH, Differential Equations)
- ○ (MATH, Azim Rafiq)

2) What are the candidate keys?

- ○ CourseCode only
- ○ CourseCode and (DepartmentCode,

CourseName)

○ CourseCode, (DepartmentCode, CourseName), and (CourseName, DepartmentChair)

3) What are the non-key columns?

○ None

○ DepartmentChair only

○ CourseName and DepartmentChair

4) What dependency violates the definition of third normal form?

○ DepartmentChair depends on DepartmentCode

○ DepartmentChair depends on CourseName

○ CourseName depends on CourseCode

5) Which column should be removed to achieve third normal form?

○ DepartmentChair

○ CourseCode

○ DepartmentCode

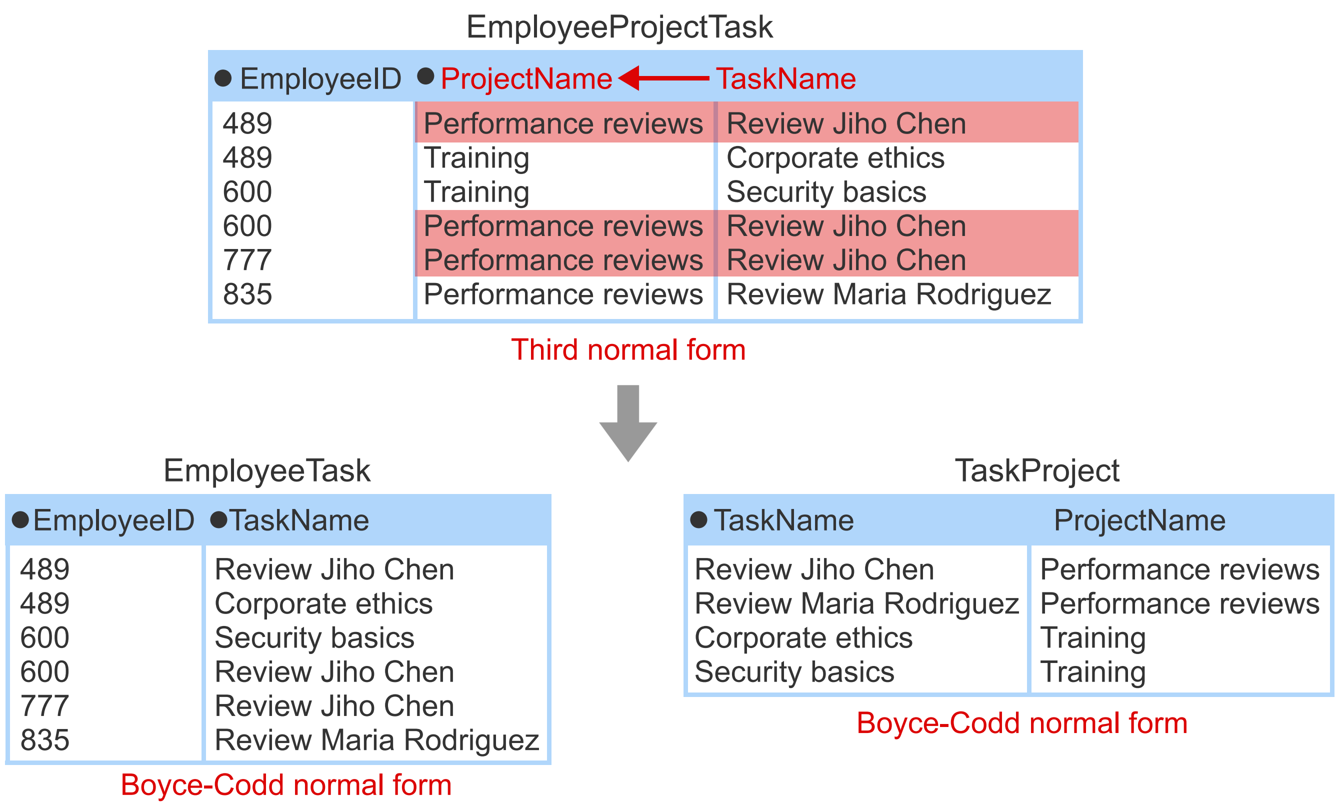
Boyce-Codd normal form

The definition of third normal form applies to *non-key* columns only, which allows for occasional redundancy. Boyce-Codd normal form applies to *all* columns and eliminates this redundancy.

A table is in ***Boyce-Codd normal form*** if, whenever column A depends on column B, then B is unique. Columns A and B may be simple or composite. This definition is identical to the definition of third normal form with the term 'non-key' removed.

Boyce-Codd normal form is considered the gold standard of table design. Although fourth and fifth normal forms remove additional types of redundancy, these redundancies are uncommon and of little practical concern.

PARTICIPATION ACTIVITY | 6.2.4: Boyce-Codd normal form eliminates all common redundancy.


EmployeeProjectTask

● EmployeeID	● ProjectName	TaskName
489	Performance reviews	Review Jiho Chen
489	Training	Corporate ethics
600	Training	Security basics
600	Performance reviews	Review Jiho Chen
777	Performance reviews	Review Jiho Chen
835	Performance reviews	Review Maria Rodriguez

Third normal form

EmployeeTask

●EmployeeID	●TaskName
489	Review Jiho Chen
489	Corporate ethics
600	Security basics
600	Review Jiho Chen
777	Review Jiho Chen
835	Review Maria Rodriguez

Boyce-Codd normal form

TaskProject

● TaskName	ProjectName
Review Jiho Chen	Performance reviews
Review Maria Rodriguez	Performance reviews
Corporate ethics	Training
Security basics	Training

Boyce-Codd normal form

Animation content:

Static figure:
Tables EmployeeProjectTask, EmployeeTask, and TaskProject appear. A large arrow indicates EmployeeProjectTask is decomposed to EmployeeTask and TaskProject.

The EmployeeProjectTask table has columns EmployeeID, ProjectName, and TaskName. (EmployeeID, ProjectName) is the primary key. An arrow points from TaskName to ProjectName. EmployeeProjectTask has caption Third normal form and six rows:
489, Performance reviews, Review Jiho Chen
489, Training, Corporate ethics
600, Training, Security basics
600, Performance reviews, Review Jiho Chen
777, Performance reviews, Review Jiho Chen
835, Performance reviews, Review Maria Rodriguez

The three instances of (Performance reviews, Review Jiho Chen) are highlighted.

The EmployeeTask table has columns EmployeeID and TaskName. (EmployeeID, TaskName) is the primary key. EmployeeTask has caption Boyce-Codd normal form and six rows:
489, Review Jiho Chen
489, Corporate ethics
600, Security basics
600, Review Jiho Chen
777, Review Jiho Chen
835, Review Maria Rodriguez

The TaskProject table has columns TaskName, ProjectName. (TaskName, ProjectName) is the primary key. TaskProject has caption Boyce-Codd normal form and four rows:
Review Jiho Chen, Performance reviews
Review Maria Rodriguez, Performance reviews
Corporate ethics, Training
Security basics, Training

Step 1: Each employee is assigned at most one task on each project. The EmployeeTaskProject table appears.

Step 2: TaskName depends on the primary key (EmployeeID, ProjectName) but not on EmployeeID or ProjectName. Parentheses surround EmployeeID and ProjectName. An arrow points from (EmployeeID, ProjectName) to TaskName.

Step 3: Since every non-key column depends on a unique column, EmployeeProjectTask is in third normal form. The caption Third normal form appears below EmployeeProjectTask.

Step 4: In this company, TaskNames are never repeated on different projects. So ProjectName depends on TaskName, which is not unique. TaskName, ProjectName, and the arrow between are highlighted.

Step 5: Since ProjectName depends on a non-unique column, EmployeeProjectTask is not in Boyce-Codd normal form and contains redundancy. The three instances of (Performance reviews, Review Jiho Chen) are highlighted.

Step 6: Decomposing to EmployeeTask and TaskProject eliminates the redundancy. The EmployeeTask and TaskProject tables appear.

Step 7: EmployeeTask contains no dependencies. TaskProject contains one dependency, ProjectName on the unique column TaskName. The caption Boyce-Codd normal form appears below the EmployeeTask and TaskProject tables.

Animation captions:

1. Each employee is assigned at most one task on each project.
2. TaskName depends on the primary key (EmployeeID, ProjectName) but not on EmployeeID or ProjectName.
3. Since every non-key column depends on a unique column, EmployeeProjectTask is in third normal form.
4. In this company, TaskNames are never repeated on different projects. So ProjectName depends on TaskName, which is not unique.
5. Since ProjectName depends on a non-unique column, EmployeeProjectTask is not in Boyce-Codd normal form and contains redundancy.
6. Decomposing to EmployeeTask and TaskProject eliminates the redundancy.
7. EmployeeTask contains no dependencies. TaskProject contains one dependency, ProjectName on the unique column TaskName.

PARTICIPATION ACTIVITY | 6.2.5: Boyce-Codd normal form.

Refer to the following table.

Each postal code, in United States *zip+4* format, is located in one state only.

Employee

● EmployeeID	DriversLicenseNumber	DriversLicenseState	Name	PostalCode
489	AB7325	IL	Lisa Ellison	60415-1922
517	N3259211	CA	Sam Snead	94015-1648
600	B16629045	CA	Malia Efrenza	94105-1648
777	8242103	TX	Nadia Shah	78758-4414
929	8242103	NY	Maria Rodriguez	10028-4316
933	AX493200	NY	Jiho Chen	10028-4316

1) EmployeeID is the only candidate key.

○ True

○ False

2) Name and PostalCode are the only non-key columns.

○ True

○ False

3) All non-key columns depend only on unique columns.

○ True

○ False

4) Employee is in third normal form.

○ True

○ False

5) All columns depend only on unique columns.

○ True

○ False

6) Employee is in Boyce-Codd normal form.

○ True

○ False

7) Removing DriversLicenseState eliminates all redundancy from Employee.

○ True

○ False

Trivial dependencies

*When the columns of A are a subset of the columns of B, A always depends on B. Ex: FareClass depends on (FlightCode, FareClass). These dependencies are called **trivial**.*

*Technically, trivial dependencies must be excluded in definitions of normal form: A table is in Boyce-Codd normal form if, for all **non-trivial** dependencies B → A, B is unique.*

CHALLENGE ACTIVITY 6.2.1: Normal form.

544874.3500394.qx3zqy7

Start

Country

● ISO	CountryName	IndependenceDate	IndependenceYear
SL	Sierra Leone	1961-04-27	1961
KM	Comoros	1975-07-06	1975
SS	South Sudan	2011-07-09	2011
TL	East Timor	2002-05-20	2002
LC	Saint Lucia	1979-02-22	1979

What columns depend on CountryName?

☐ ISO ☐ IndependenceDate ☐ IndependenceYear

What columns depend on IndependenceDate?

☐ ISO ☐ CountryName ☐ IndependenceYear

1 2 3

Check Next

Exploring further:

- Boyce-Codd normal form

6.3 Applying normal form

Normalization

Occasionally, implementing entities, relationships, and attributes generates tables that contain redundancy. This redundancy is eliminated with normalization, the last step of logical design.

Normalization eliminates redundancy by decomposing a table into two or more tables in higher normal form.

Database designers usually normalize tables to Boyce-Codd normal form. In a ***Boyce-Codd normal form*** table, if column A depends on column B, then B must be unique. Normalizing a table to Boyce-Codd normal form involves three steps:

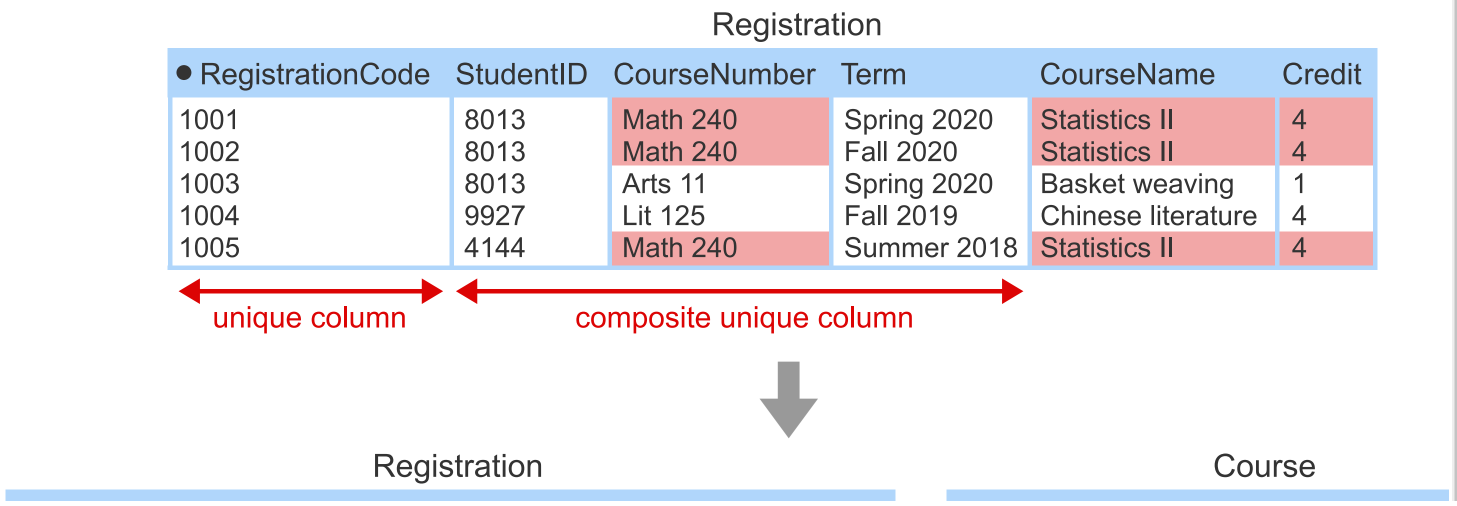
1. *List all unique columns*. Unique columns may be simple or composite. Composite columns must be minimal - remove any columns that are not necessary for uniqueness.
2. *Identify dependencies on non-unique columns*. Non-unique columns are either *external* to all unique columns or *contained within* a composite unique column.
3. *Eliminate dependencies on non-unique columns*. If column A depends on a non-unique column B, A is removed from the original table. A new table is created containing A and B. B is a primary key in the new table and a foreign key in the original table.

Since the data relating columns A and B is recorded in a new table, no information is lost when A is removed from the original table.

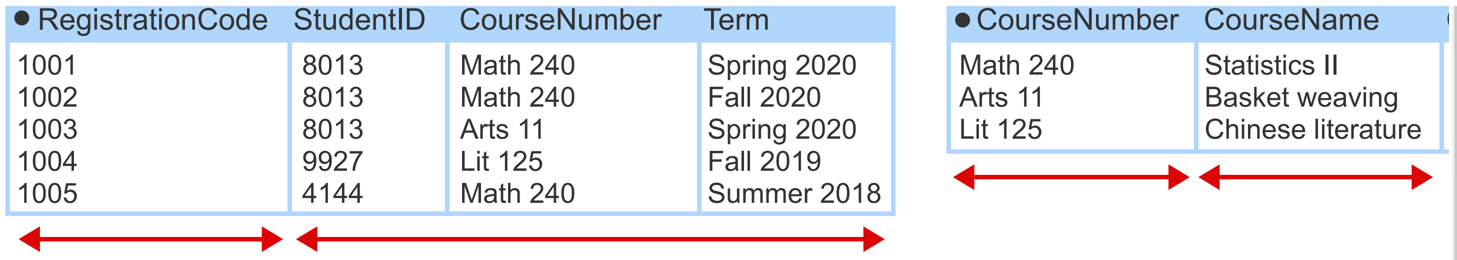
Terminology

In E. F. Codd's original paper on the relational model, ***normalization*** *meant achieving first normal form. Over time, however, normalization has come to mean achieving higher normal forms.*

PARTICIPATION ACTIVITY 6.3.1: Normalizing tables.



● RegistrationCode	StudentID	CourseNumber	Term	CourseName	Credit
1001	8013	Math 240	Spring 2020	Statistics II	4
1002	8013	Math 240	Fall 2020	Statistics II	4
1003	8013	Arts 11	Spring 2020	Basket weaving	1
1004	9927	Lit 125	Fall 2019	Chinese literature	4
1005	4144	Math 240	Summer 2018	Statistics II	4



● RegistrationCode	StudentID	CourseNumber	Term
1001	8013	Math 240	Spring 2020
1002	8013	Math 240	Fall 2020
1003	8013	Arts 11	Spring 2020
1004	9927	Lit 125	Fall 2019
1005	4144	Math 240	Summer 2018

● CourseNumber	CourseName
Math 240	Statistics II
Arts 11	Basket weaving
Lit 125	Chinese literature

Animation content:

Static figure:
Tables Registration, Registration, and Course appear. A large arrow indicates that the first Registration table is decomposed into the second Registration table and the Course table.

The first Registration table has columns RegistrationCode, StudentID, CourseNumber, Term, CourseName, and Credit. RegistrationCode is the primary key. RegistrationCode is labeled unique column. (StudentID, CourseNumber, Term) is labeled composite unique column. Registration has five rows:
1001, 8013, Math 240, Spring 2020, Statistics II, 4
1002, 8013, Math 240, Fall 2020, Statistics II, 4
1003, 8013, Arts 11, Spring 2020, Basket weaving, 1
1004, 9927, Lit 125, Fall 2019, Chinese literature, 4
1005, 4144, Math 240, Summer 2018, Statistics II, 4

The three instances of (Math 240, Statistics II, 4) are highlighted.

The second Registration table has columns RegistrationCode, StudentID, CourseNumber, and Term. RegistrationCode is the primary key. A highlight line spans RegistrationCode. Another highlight line spans (StudentID, CourseNumber, Term). The second Registration table has five rows:
1001, 8013, Math 240, Spring 2020
1002, 8013, Math 240, Fall 2020
1003, 8013, Arts 11, Spring 2020
1004, 9927, Lit 125, Fall 2019
1005, 4144, Math 240, Summer 2018

The Course table has columns CourseNumber, CourseName, and Credit. CourseNumber is the primary key. A highlight line spans CourseNumber. Another highlight line spans CourseName. Course has three rows:
Math 240, Statistics II, 4
Arts 11, Basket weaving, 1
Lit 125, Chinese literature, 4

Step 1: The Registration table lists student registration for courses by term. The first Registration table appears.

Step 2: RegistrationCode is a unique column. The RegistrationCode column is highlighted. The unique column caption appears..

Step 3: (StudentID, CourseNumber, Term) is a composite unique column. The (StudentID, CourseNumber, Term) columns are highlighted. The composite unique column caption appears.

Step 4: CourseName and Credit depend on CourseNumber, which is not unique. Registration is not in Boyce-Codd normal form. The three instances of (Math 240, Statistics II, 4) are highlighted.

Step 5: Redundancy is eliminated by removing CourseName and Credit. The new table is in Boyce-Codd normal form. The second Registration table appears with highlight lines.

Step 6: CourseNumber, CourseName, and Credit are tracked in a new Course table. The Course table appears.

Step 7: All dependencies in Course are on a unique column. Course is in Boyce-Codd normal form. The highlight lines for the Course table appear.

Animation captions:

1. The Registration table lists student registration for courses by term.
2. RegistrationCode is a unique column.
3. (StudentID, CourseNumber, Term) is a composite unique column.
4. CourseName and Credit depend on CourseNumber, which is not unique. Registration is not in Boyce-Codd normal form.
5. Redundancy is eliminated by removing CourseName and Credit. The new table is in Boyce-Codd normal form.
6. CourseNumber, CourseName, and Credit are tracked in a new Course table.
7. All dependencies in Course are on a unique column. Course is in Boyce-Codd normal form.

PARTICIPATION ACTIVITY | 6.3.2: Normalization.

Refer to the Shipment table:

Shipment

●TrackingID	OrderCode	ProductNumber	ProductName	ShipperCode	ShipperName

201	A37	8024	Slip Sandal	FX	Federal Express
202	A37	9119	Zephyr T-Shirt	UPS	United Parcel Service
203	C28	9119	Zephyr T-Shirt	FX	Federal Express
204	R43	3558	Cool Hat	FX	Federal Express

1) Which simple or composite column(s) are unique and minimal?

- ○ TrackingID only
- ○ TrackingID and (OrderCode, ProductNumber)
- ○ TrackingID and (OrderCode, ProductNumber, ShipperCode)

2) Which column depends on part of the composite unique column (OrderCode, ProductNumber)?

- ○ ProductName
- ○ ShipperCode
- ○ ShipperName

3) Which column depends on a column that is external to unique columns?

- ○ ProductName
- ○ TrackingID
- ○ ShipperName

4) When Shipment is normalized, which columns remain?

- ○ TrackingID, OrderCode, ProductNumber, ShipperCode
- ○ TrackingID, OrderCode, ProductNumber
- ○ OrderCode, ProductNumber, ShipperCode

5) How many tables are in the normalized design?

- ○ 2

○ 3

○ 4

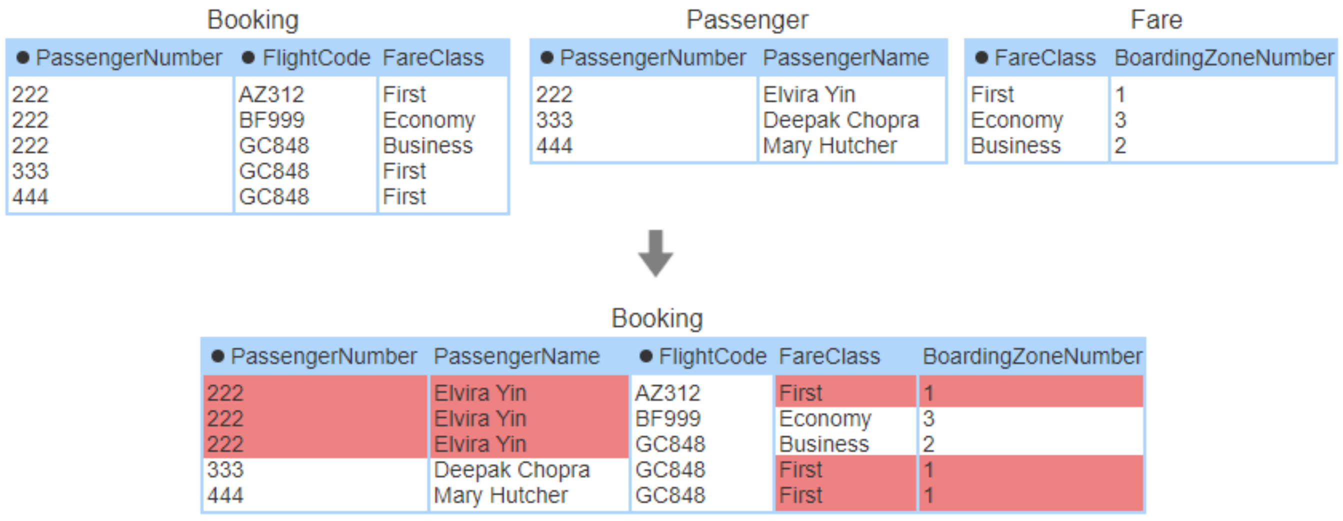
Denormalization

Boyce-Codd normal form is ideal for tables with frequent inserts, updates, and deletes. In a database used primarily for reporting, changes are infrequent and redundancy is acceptable. In fact, redundancy can be desirable in reporting databases, as processing is faster and queries are simpler. Therefore, reporting databases may contain tables that, by design, are not in third normal form.

Denormalization means intentionally introducing redundancy by merging tables. Denormalization eliminates join queries and therefore improves query performance. Denormalization results in first and second normal form tables and should be applied selectively and cautiously.

In the figure below, the Booking, Passenger, and Fare tables are denormalized into a single Booking table. The red highlight indicates redundancy in the denormalized table.

Figure 6.3.1: Denormalization example.



Booking

● PassengerNumber	● FlightCode	FareClass
222	AZ312	First
222	BF999	Economy
222	GC848	Business
333	GC848	First
444	GC848	First

Passenger

● PassengerNumber	PassengerName
222	Elvira Yin
333	Deepak Chopra
444	Mary Hutcher

Fare

● FareClass	BoardingZoneNumber
First	1
Economy	3
Business	2

Booking

● PassengerNumber	PassengerName	● FlightCode	FareClass	BoardingZoneNumber
222	Elvira Yin	AZ312	First	1
222	Elvira Yin	BF999	Economy	3
222	Elvira Yin	GC848	Business	2
333	Deepak Chopra	GC848	First	1
444	Mary Hutcher	GC848	First	1

PARTICIPATION ACTIVITY | 6.3.3: Denormalization.

1) Denormalization never results in second-normal-form tables.

○ True

○ False

2) Denormalization accelerates all

SELECT queries.

- ○ True
- ○ False

3) Occasionally, tables are denormalized in a frequently updated database.

- ○ True
- ○ False

Database design

As tables and keys are specified, the database designer reviews each table for Boyce-Codd normal form. Dependencies and unique columns are identified. If any dependencies are not on unique columns, the table is decomposed into smaller tables in Boyce-Codd normal form. Tables that experience infrequent inserts, updates, and deletes may be denormalized to simplify and accelerate join queries.

Table 6.3.1: Applying normal form.

Step	Activity
8A	Identify dependencies on non-unique columns.
8B	Eliminate redundancy by decomposing tables.
8C	Consider denormalizing tables in reporting databases.

PARTICIPATION ACTIVITY 6.3.4: Applying normal form.

Match the activity with the resulting normal form.

If unable to drag and drop, refresh the page.

First or second normal form **Boyce-Codd normal form** **Second normal form**

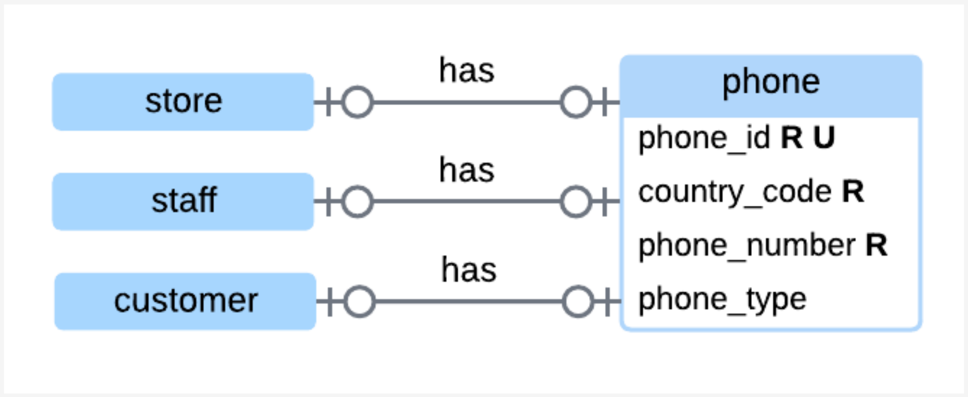
Consider denormalizing tables in

	Consider denormalizing tables in reporting databases.
	Eliminate dependencies on columns contained within a composite unique column.
	Eliminate dependencies on non-unique columns.

Reset

6.4 LAB - Implement strong entity (Sakila)

Implement a new strong entity `phone` in the Sakila database. Attributes and relationships are shown in the following diagram:



The diagram uses Sakila naming conventions. Follow the Sakila conventions for your table and column names:

- All lower case
- Underscore separator between root and suffix
- Foreign keys have the same name as referenced primary key

Write CREATE TABLE and ALTER TABLE statements that:

1. Implement the entity as a new `phone` table.
2. Implement the `has` relationships as foreign keys in the Sakila `customer`, `staff`, and `store` tables.
3. Remove the existing `phone` column from the Sakila `address` table.

Step 2 requires adding a foreign key constraint to an existing table. Ex:

```
ALTER TABLE customer
   ADD FOREIGN KEY (phone_id) REFERENCES phone(phone_id)
   ON DELETE SET NULL
   ON UPDATE CASCADE;
```

Specify data types as follows:

- phone_id, phone_number, and country_code have data type INT.
- phone_type has data type VARCHAR(12) and contains strings like 'Home', 'Mobile', and 'Other'.

Apply these constraints:

- NOT NULL and UNIQUE constraints correspond to cardinalities on the diagram above.
- Foreign key actions are SET NULL for delete rules and CASCADE for update rules.
- Specify a suitable column as the `phone` table primary key.

544874.3500394.qx3zqy7

LAB ACTIVITY 6.4.1: LAB - Implement strong entity (Sakila) 10 / 10

Main.sql

Load default template...

```sql
-- Your CREATE TABLE and ALTER TABLE statements go here
CREATE TABLE phone (
phone_id INT NOT NULL,
country_code INT NOT NULL,
phone_number INT NOT NULL,
phone_type VARCHAR(12) CHECK (phone_type = 'Home' OR 'Mobile' OR 'Other'),
PRIMARY KEY (phone_id)
);
ALTER TABLE customer
ADD COLUMN phone_id INT,
ADD FOREIGN KEY (phone_id) REFERENCES phone(phone_id)
ON DELETE SET NULL
ON UPDATE CASCADE;

ALTER TABLE staff
ADD COLUMN phone_id INT
```

Develop mode **Submit mode**

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql
(Your program)

Output (shown below)

Program output displayed here

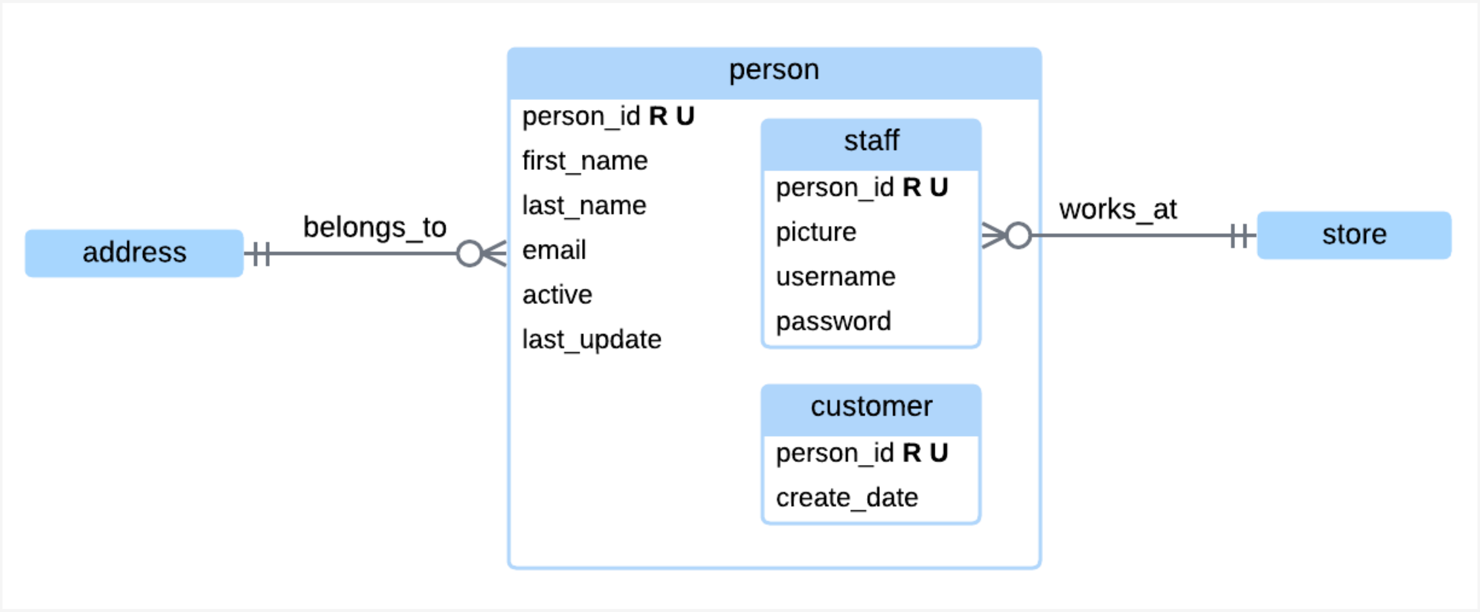
Coding trail of your work What is this?

```
2/5 M----------10 min:7
```

6.5 LAB - Implement supertype and subtype entities (Sakila)

Similar entities have many common attributes and relationships. Similar entities are often converted into subtypes of a supertype entity, as illustrated in this lab.

In the Sakila database, the `customer` and `staff` tables have several common columns. Convert these tables into subtypes of `person`. Specifically, write CREATE TABLE statements for `person`, `customer`, and `staff` that implement this ER diagram:



Follow Sakila conventions for table and column names:

- All lower case
- Underscore separator between root and suffix
- Foreign keys have the same name as referenced primary key

Implement attributes as columns:

- The primary key of all three tables is `person_id` with data type `SMALLINT UNSIGNED`.
- The `last_update` and `create_date` columns have data type `TIMESTAMP`.
- The `picture` column has data type `BLOB`.
- All other columns have data type `VARCHAR(20)`.

Implement the `belongs_to` and `works_at` relationships as foreign keys:

- `belongs_to` becomes an `address_id` foreign key in `person` with data type `SMALLINT UNSIGNED`.
- `works_at` becomes a `store_id` foreign key in `staff` with data type `TINYINT UNSIGNED`.
- Specify RESTRICT actions for both foreign keys.
- Required relationships become NOT NULL foreign keys.

Subtype entities have an `IsA` relationship to the supertype. Implement these relationships as foreign keys:

- The `person_id` columns of `customer` and `staff` become foreign keys referring to `person`.
- Specify CASCADE actions for both foreign keys.

NOTE: If you execute your solution with the Sakila database, you must first drop `customer`, `staff`, and all constraints that refer to these tables. Use the following statements with Sakila only, not in the zyLab environment:

```
ALTER TABLE payment
   DROP CONSTRAINT fk_payment_customer,
   DROP CONSTRAINT fk_payment_staff;
ALTER TABLE rental
   DROP CONSTRAINT fk_rental_customer,
   DROP CONSTRAINT fk_rental_staff;
ALTER TABLE store
   DROP CONSTRAINT fk_store_staff;
DROP TABLE customer, staff;
```

544874.3500394.qx3zqy7

LAB ACTIVITY | 6.5.1: LAB - Implement supertype and subtype entities (Sakila) | 10 / 10 ✓

Main.sql

```
Loading latest submission...
```

Develop mode | Submit mode

Explore the database and run your program as often as you'd like, before submitting for grading. Click **Run program** and observe the program's output in the second box.

Run program

Main.sql
(Your program)

→ Output (shown below)

Program output displayed here

Coding trail of your work [What is this?]

Retrieving signature