Normalization

Sections to read:

- “Design of relational database schemas” of the textbook.

Key concepts from the relational design

- A candidate key of a relation *R (a set of attributes)* :
  - $K \rightarrow R$ and K is minimal: K is a candidate key of R.
- FD inference rules:
  - transitivity: $X \rightarrow Y$, $Y \rightarrow Z$ so $X \rightarrow Z$.
  - augmentation: $X \rightarrow Y$ so $X \cup Z \rightarrow Y \cup Z$.
- BCNF:
  - For *any* FD $X \rightarrow Y$ for relation R where X is a minimal set of attributes, and Y is not a subset of X, X is a key of R.

Revision on BCNF

Consider Class(cno, title, day, time, room, type)

cno → title; room → type

Is Class in BCNF?

- The first step is to work out the candidate keys.

{cno}+ = {cno, title}

{room}+ = {room, type}

{day, time, room, cno}+ = {day, time, room, cno, title, type}

So {day, time, room, cno} is a candidate key for the Class relation (and the only one).

- Check if all FDs conform to the BCNF definition.

cno → title violates BCNF. room → type also violates BCNF.

So Class is not in BCNF.

Example …

Class is normalised (decomposed) into the following three relations (process to be explained later):

CourseInfo(<u>cno</u>, title)

Classroom(<u>room</u>, type)

CourseClass(<u>cno, day, time, room</u>)

Note that the relations after decomposition are suitably named to reflect meaning of the relation.

- CourseInfo is in BCNF.
  - The only projected FD: cno → title
- Classroom is in BCNF.
  - The only projected FD: room → type
- CourseClass is in BCNF.
  - No FDs.

Example …

Class

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1055	Database Concepts	Wed	1730	12.05.02	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Thur	1430	07.02.23	lect

CourseInfo

cno	title
isys1057	Database Concepts
isys1055	Database Concepts
acct1009	Another Course

Classroom

room	type
14.04.27	lect
12.05.02	lect
14.10.30	tute
14.09.23	tute
07.02.23	lect

CourseClass

cno	day	time	room
isys1057	Wed	1030	14.04.27
isys1057	Thur	1330	14.04.27
isys1055	Wed	1730	12.05.02
isys1057	Wed	1130	14.10.30
isys1057	Wed	1330	14.09.23
acct1009	Wed	1130	14.04.27
acct1009	Thur	1330	07.02.23
acct1009	Thur	1430	07.02.23

After decomposition, redundancy and anomalies are removed.

Normalization

- Normalisation: *decompose* a relation not in BCNF into a set of relations in BCNF.
  - normalisation is also called decomposition.
- Data (tuples) of the original relation will be projected onto two or more relations after decomposition, but data redundancy will be removed.
- Normalization is not level by level -- A relation in 1NF can be directly decomposed into relations in BCNF.

Conditions for correct normalization

- Lossless join
- Dependency preservation

A bad decomposition

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1055	Database Concepts	Wed	1730	12.05.02	lect
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Thur	1430	07.02.23	tute

cno	title
isys1057	Database Concepts
isys1055	Database Concepts
acct1009	Another Course

day	time	room	type
Wed	1030	14.04.27	lect
Thur	1330	14.04.27	lect
Wed	1730	12.05.02	lect
Wed	1130	14.10.30	tute
Wed	1330	14.09.23	tute
Wed	1130	14.04.27	lect
Thur	1330	07.02.23	lect
Thur	1430	07.02.23	tute

A bad decomposition …

- With the original relation Class, the information on what courses are timetabled as what classes is available.
- From the 2 relations after decomposition, can the same information be obtained?
  - In decomposing relations, there should be at least some common attributes in the relations after decomposition.

A Good Decomposition

Class

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1055	Database Concepts	Wed	1730	12.05.02	lect
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Thur	1430	07.02.23	tute

Tuples projected to {cno, title}.

Tuples projected to {cno, day, time, room, type}.

CourseInfo

cno	title
isys1057	Database Concepts
isys1055	Database Concepts
acct1009	Another Course

CourseClass

cno	day	time	room	type
isys1057	Wed	1030	14.04.27	lect
isys1057	Thur	1330	14.04.27	lect
isys1057	Wed	1130	14.10.30	tute
isys1057	Wed	1330	14.09.23	tute
isys1055	Wed	1730	12.05.02	lect
acct1009	Wed	1130	14.04.27	lect
acct1009	Thur	1330	07.02.23	lect
acct1009	Thur	1430	07.02.23	tute

Decomposition – Lossless Join

- Relation R is decomposed into R_1 and R_2. Tuples in an instance for R can be reconstructed from joining the projected tuples for R_1 and R_2 on common attributes. The decomposition is a lossless-join decomposition, or simply lossless.
  - R_1 and R_2 must at least have common attributes.
  - For any instance of R, combining tuples in R_1 and R_2 surely can produce rightful tuples for R.
  - For any instance of R, combining tuples in R_1 and R_2 should not produce spurious tuples not in R.

Lossless/Lossy Join: Example

Class(cno, title, day, time, room, type)

cno → title

room → type

- **The following decomposition is lossless.**

  CourseInfo(cno, title)

  CourseClass(cno, day, time, room, type)

- **The following decomposition is lossy.**

  Class1(cno, title, type)

  Class2(day, time, room, type)

Common attribute Cno

Common attribute Type

Lossless Join: Example ...

cno	title
isys1057	Database Concepts
isys1055	Database Concepts
acct1009	Another Course

cno	day	time	room	type
isys1057	Wed	1030	14.04.27	lect
isys1057	Thur	1330	14.04.27	lect
isys1057	Wed	1130	14.10.30	tute
isys1057	Wed	1330	14.09.23	tute
isys1055	Wed	1730	12.05.02	lect
acct1009	Wed	1130	14.04.27	lect
acct1009	Thur	1330	07.02.23	lect
acct1009	Thur	1430	07.02.23	tute

Join on common attribute cno

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1055	Database Concepts	Wed	1730	12.05.02	lect
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Thur	1430	07.02.23	tute

Lossy Join: Example

cno	title	type
isys1057	Database Concepts	lect
isys1055	Database Concepts	lect
isys1057	Database Concepts	tute
acct1009	Another Course	lect
acct1009	Another Course	tute

day	time	room	type
Wed	1030	14.04.27	lect
Thur	1330	14.04.27	lect
Wed	1730	12.05.02	lect
Wed	1130	14.10.30	tute
Wed	1330	14.09.23	tute
Wed	1130	14.04.27	lect
Thur	1330	07.02.23	lect
Thur	1430	07.02.23	tute

Join on common attribute Type

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1730	12.05.02	lect
isys1057	Database Concepts	Wed	1130	14.04.27	lect
isys1057	Database Concepts	Thur	1330	07.02.23	lect
isys1055	Database Concepts	Wed	1030	14.04.27	lect
isys1055	Database Concepts	Thur	1330	14.04.27	lect
isys1055	Database Concepts	Wed	1730	12.05.02	lect
isys1055	Database Concepts	Wed	1130	14.04.27	lect
isys1055	Database Concepts	Thur	1330	07.02.23	lect
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1057	Database Concepts	Thur	1430	07.02.23	tute
acct1009	Another Course	Wed	1030	14.04.27	lect
acct1009	Another Course	Thur	1330	14.04.27	lect
acct1009	Another Course	Wed	1730	12.05.02	lect
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1330	07.02.23	lect
acct1009	Another Course	Wed	1130	14.10.30	tute
acct1009	Another Course	Wed	1330	14.09.23	tute
acct1009	Another Course	Thur	1430	07.02.23	tute

isys1057 should have two lectures but now has 5 lectures!!!

Highlighted tuples are **spurious** tuples representing **false** information !!!

Lossless Join Decomposition Process

- Decomposed relations must at least have common attributes. Not enough! The common attributes must also be a key in at least one relation after decomposition.
- Examples:
  - Lossless Join: “cno” is a candidate (the primary) key for CourseInfo.
    CourseInfo(cno, title)
    CourseClass(cno, day, time, room, type)

  -- Lossy decomposition: “type” is not a candidate key in any relation.
    Class1(cno, title, type)
    Class2(day, time, room, type)
- The Chase test for lossless Join --- see Sections 3.4.2 and 3.4.3 of the textbook for details

Decomposition: Preserving FDs

The decomposition

Class1(<u>cno, day, time, room</u>)

Class2(<u>title, day, time, type</u>)

does not preserve the FDs:

cno → title, room → type

"cno” is in Class1 whereas “title” is in relation Class2. “room” is in Class1 whereas “type” is Class2.

Decomposition: Preserving FDs …

Joining Class1 and Class 2 generates spurious tuples, as highlighted below.

cno	title	day	time	room	type
isys1057	Database Concepts	Wed	1030	14.04.27	lect
isys1057	Database Concepts	Thur	1330	14.04.27	lect
isys1057	Database Concepts	Wed	1330	14.09.23	tute
isys1057	Database Concepts	Wed	1130	14.10.30	tute
isys1057	Another Course	Thur	1330	14.04.27	lect
isys1057	Another Course	Wed	1130	14.10.30	lect
isys1055	Database Concepts	Wed	1730	12.05.02	lect
acct1009	Database Concepts	Thur	1330	07.02.23	lect
acct1009	Database Concepts	Wed	1130	14.04.27	tute
acct1009	Another Course	Wed	1130	14.04.27	lect
acct1009	Another Course	Thur	1430	07.02.23	lect
acct1009	Another Course	Thur	1330	07.02.23	lect

The Ideal BCNF Decomposition

- Ideally decomposition into BCNF should satisfy:
  - Lossless join: decomposed relations should be joined to produce the original relation.
  - FD preservation: all FDs hold in the original relation should also be enforced in the decomposed relations.

Example: Decomposition to BCNF

Class(cno, title, day, time, room, type)

cno → title

room → type

Decomposition:

Class1(<u>cno</u>, title)

Class2(<u>room</u>, type)

Class3(<u>day, time, room*, cno*</u>)

But the Ideal Decomposition may not always be achieved

The ideal BCNF decomposition can not always be achieved. Consider relation Program(show, studio, host) for TV programs, and FDs:

- An on-air show in a studio has only one host.
  - show, studio → host
- Each host has a "home" studio:
  - host → studio

The Program relation has two keys:

{show, studio} and {show, host}

The Program relation is not in BCNF. host → studio violates BCNF.

Ideal BCNF Decomposition …

If we decompose Program using

host → studio as follows:

HostStudio(<u>host</u>, studio): host → studio,

HostShow(<u>host, show</u>): Nil FDs,

It is impossible to enforce the FD

show, studio → host

in the decomposed relations – they are in separate relations!

Solution — 3NF

- Third Normal Form (3NF) loosens the BCNF condition so we do not have to decompose in this problematic situation.
- A relation R is in 3NF if for any FD *X* ->*A* on *R,* where *X* is a set of attributes without redundant attributes and *A* is an attribute, either *X* is a candidate key of *R* or *A* is part of a (possibly different) candidate key.
- A relation in BCNF is also in 3NF.

3NF: Example

- Program(show, studio, host)
  - FD1: show, studio → host
  - FD2: host → studio

There are two keys

{show, studio} and {host, show}.

FD1: the left is a key.

FD2: the left is not a key, but studio on the right is part of the key {show, studio}.

Program is not in BCNF, but in 3NF.

3NF ...

- Our definition of 3NF simplifies the 3NF definition in the textbook, which is based on all FDs, including FDs with redundant attributes on the left hand side.
- Given an FD

  student-no → student-name

  the FD below also holds but contains redundant attribute student-age:

  student-no, student-age → student-name.

Decomposition into BCNF/3NF

- Lossless Join (LJ) and Dependency Preservation (DP) are desirable properties of decomposition into BCNF/3NF.
- The decomposition algorithm should ensure LJ and DP.
  - The decomposition often results in BCNF relations. This is because very often the real-world constraints between attributes are not that complex.

Decomposition into BCNF/3NF: Algorithm

Given a relation and FDs on its attributes,

1. Construct a minimal basis for given FDs.
2. Find all candidate keys for the relation.
3. One relation for each FD in the minimal basis.
   – Schema is the union of the left and right sides.
4. If no candidate key from Step 2 is in any relation from Step 3, add a relation whose schema is some key.

Proof that the process is lossless join and dependency preserving and produces 3NF (which very often are indeed BCNF) relations is in the textbook.

Decomposition into BCNF/3NF: Example

Let us revisit the example relation Class:

Class(cno, title, prog, progtype, progleader, day, time, room, classtype, capacity)

cno → title, prog, progtype
prog → progtype, progleader
day, time, room → classtype
room → classtype, capacity

1. The minimal basis for FDs:

cno → title
cno → prog
prog → progtype
prog → progleader
room → classtype
room → capacity

2. The only candidate key for Class is {cno, day, time, room}.

Decomposition into 3NF: Example …

3. Constructing relations:

R1(cno, title)
R2(cno, prog)
R3(prog, progtype)
R4(prog, progleader)
R5(room, classtype)
R6(room, capacity)

Combining relations:

Class1(<u>cno</u>, title, prog)
Class2(<u>prog</u>, progtype, progleader)
Class3(<u>room</u>, classtype, capacity)

4. The Key of relation Class is {cno, day, time, room}. So an additional relation is needed.

Class4(<u>cno*, day, time, room*</u>)

The resulting relations are in 3NF, and indeed also in BCNF.

Exercise

Given relation $R(A, B, C, D)$ with FDs

$$A \rightarrow B,\ A \rightarrow D$$

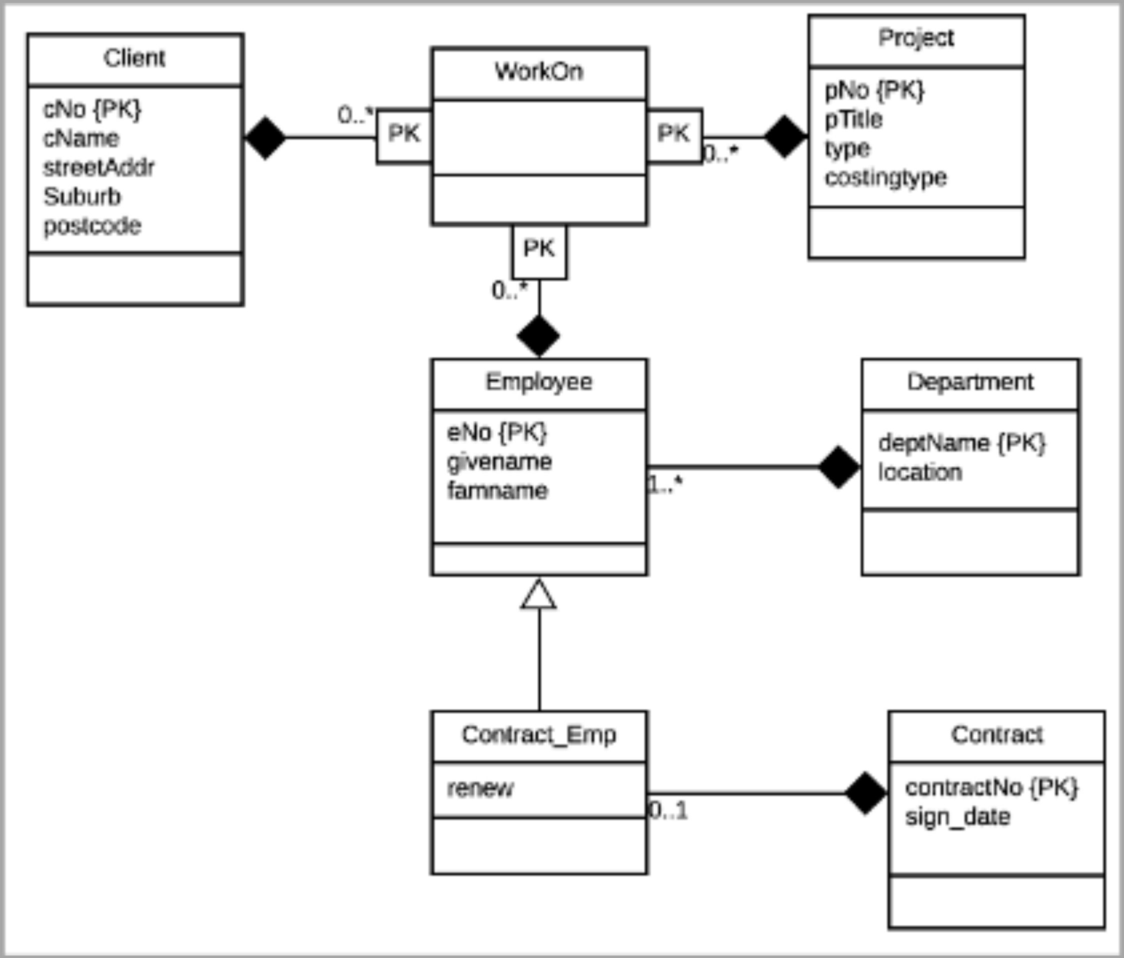
- Is R in BCNF?
- Is R in 3NF?
- If not in BCNF or 3NF, decompose R into BCNF/3NF relations.

Relational Database Design: summary

Relational database design practice:

1. Design an ER diagram.
2. Map the ER diagram into a relational database schema (following the mapping rules).
   - If mapped correctly, the resulting relation schemas are most likely in BCNF or 3NF as entity primary key and relationship multiplicity constraints are captured in the ER diagram.
3. Examine the normal form of resulting relations, using FDs based on constraints from the ER diagram and any FDs not in the ER diagram.
4. If a relation is not in BCNF/3NF, decompose it into BCNF/3NF.

Example ERD: The Employee database



Mapping to a relational database schema

Client(<u>cNo</u>, cName, streetAddr, suburb, postcode)

Project(<u>pNo</u>, pTitle, type, costingtype)

Department(<u>deptName</u>, location)

Employee(<u>eNo</u>, givename, famname, deptName*)

Contract_Emp(<u>eNo*</u>, renew, contractNo*)

Contract(<u>contractNo</u>, sign_date)

WorkOn(<u>cNo*, pNo*, eNo*</u>)

Mapping into a relational schema: is every relation in BCNF/3NF?

Consider two FDs:

FD1: suburb→ postcode

FD2: Type → costingtype

- Client(cNo, cName, streetAddr, suburb, postcode) is not in BCNF or 3NF, due to FD1. Decomposition:

  Client(cNo, cName, streetAddr, suburb*)

  ClientSuburb(suburb, postcode)

- Project(pNo, pTitle, type, costingtype) is not in BCNF or 3NF, due to FD2. Decomposition:

  Project(pNo, pTitle, type*)

  ProjectType(type, costingtype)

Mapping to a relational database schema and normalization into BCNF/3NF

Client(cNo, cName, streetAddr, suburb*)

ClientSuburb(suburb, postcode)

Project(pNo, pTitle, type*)

ProjectType (type, costingtype)

Department(deptName, location)

Employee(eNo, givename, famname, deptName*)

Contract_Emp(eNo*, renew, contractNo*)

Contract(contractNo, sign_date)

WorkOn(cNo*, pNo*, eNo*)