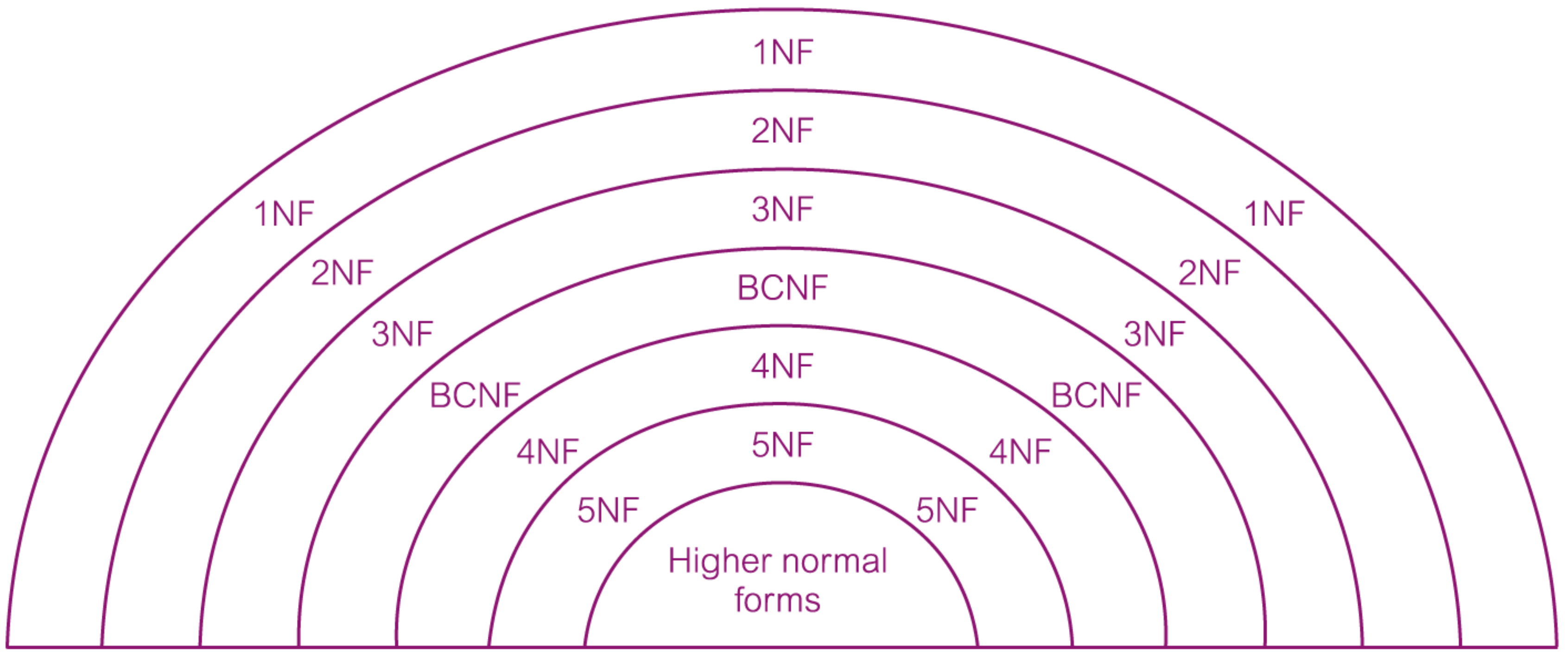
Advanced Normalization

Learning Outcomes

- Understand there are normal forms that go beyond Third Normal Form (3NF).
- Understand and be able to identify Boyce-Codd Normal Form (BCNF).
- Be able to decompose relations to BCNF.
- Understand and be able to explain Multi-valued Dependency (MVD)
- Understand and be able to identify Fourth Normal Form (4NF).
- Be able to decompose relations to 4NF.

The Process of Normalization



Boyce–Codd Normal Form (BCNF)

- Based on functional dependencies that take into account all candidate keys in a relation, however BCNF also has additional constraints compared with the general definition of 3NF.

- Boyce–Codd normal form (BCNF)
  - A relation is in BCNF if and only if every determinant is a candidate key.

BCNF

- Difference between 3NF and BCNF:

  For a functional dependency A -> B

  – 3NF allows this dependency in a relation if B is a candidate-key attribute and A is not a candidate key.
  – BCNF requires A must be a candidate key.

General Definitions of 2NF and 3NF

- Second normal form (2NF)
  - A relation that is in first normal form and every *non-candidate-key* attribute is fully functionally dependent on any *candidate key*.

- Third normal form (3NF)
  - A relation that is in first and second normal form and in which no *non-candidate-key* attribute is transitively dependent on any *candidate key*.

BCNF

Client (clientNo, cName)

clientNo -> cName (primary key)

This relation is in BCNF

Rental(clientNo, propertyNo, rentStart, rentFinish)

clientNo, propertyNo -> rentStart, rentFinish (primary key)

clientNo, rentStart -> propertyNo, rentFinish (candidate key)

propertyNo, rentStart -> clientNo, rentFinish (candidate key)

All determinants are candidate key, this relation is in BCNF

PropertyForRent(propertyNo, pAddress, rent, ownerNo)

propertyNo -> pAddress, rent, ownerNo (primary key)

This relation is in BCNF

Owner(ownerNo, oName)

ownerNo -> oName (primary key)

This relation is in BCNF

BCNF example

Example:

- Client is interviewed by members of staff in *DreamHome* case.
- The members of staff involved in interviewing clients are allocated to a specific room on the day of interview.
- However a room may be allocated to several members of staff as required throughout a working day.
- A client is interviewed only once on a given date, but may be requested to attend further interviews at later dates.

BCNF example

ClientInterview

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-05	10.30	SG5	G101
CR56	13-May-05	12.00	SG5	G101
CR74	13-May-05	12.00	SG37	G102
CR56	1-Jul-05	10.30	SG5	G102

- Functional dependencies:

clientNo, interviewDate → interviewTime, staffNo, roomNo **(Primary key)**
staffNo, interviewDate, interviewTime → clientNo **(Candidate key)**
roomNo, interviewDate, interviewTime → staffNo, clientNo **(Candidate key)**
staffNo, interviewDate → roomNo

BCNF example

ClientInterview

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-05	10.30	SG5	G101
CR56	13-May-05	12.00	SG5	G101
CR74	13-May-05	12.00	SG37	G102
CR56	1-Jul-05	10.30	SG5	G102

- Candidate keys:
  (clientNo, interviewDate)
  (staffNo, interviewDate, interviewTime)
  (roomNo, interviewDate, interviewTime)

General Definitions of 2NF and 3NF

- Second normal form (2NF)
  - A relation that is in first normal form and every *non-candidate-key* attribute is fully functionally dependent on any *candidate key*.

- Third normal form (3NF)
  - A relation that is in first and second normal form and in which no *non-candidate-key* attribute is transitively dependent on any *candidate key*.

BCNF example

ClientInterview

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-05	10.30	SG5	G101
CR56	13-May-05	12.00	SG5	G101
CR74	13-May-05	12.00	SG37	G102
CR56	1-Jul-05	10.30	SG5	G102

- Update anomaly: to change the room number for staff SG5 on 13-May-05, two tuples must be updated.

BCNF example

- To transform the ClientInterview into BCNF, we remove the violating functional dependency by creating two new relations:
  - Interview (clientNo, interviewDate, interviewTime, staffNo)
  - StaffRoom (staffNo, interviewDate, roomNo)

Interview

clientNo	interviewDate	interviewTime	staffNo
CR76	13-May-05	10.30	SG5
CR56	13-May-05	12.00	SG5
CR74	13-May-05	12.00	SG37
CR56	1-Jul-05	10.30	SG5

StaffRoom

staffNo	interviewDate	roomNo
SG5	13-May-05	G101
SG37	13-May-05	G102
SG5	1-Jul-05	G102

Boyce–Codd Normal Form (BCNF)

- Every relation in BCNF is also in 3NF. However, a relation in 3NF is not necessarily in BCNF.

- For a relation that is in 3NF, violation of BCNF is quite *rare*.

- The potential of 3NF relation to violate BCNF may occur in a relation that:
  - contains **two (or more)** composite candidate keys;
  - the candidate keys **overlap**, that is, they have at least one attribute in common.

Exercise 1

The relation below contains information about student applying for colleges on majors.

Apply(studentNo, collegeName, cCity, date, major)

Suppose college names are unique and students may

apply to each college only once, so we have two FDs:

fd1: collegeName $\rightarrow$ cCity

fd2: studentNo, collegeName $\rightarrow$ date, major

Is Apply in BCNF?

Exercise 2

- Relation Z(A, B, C, D, E) has functional dependencies:

  A, B → C, D, E

  B, C → D

  Is Z in BCNF?

Algorithm for decomposing relations into BCNF

- Relation R with FDs
- Compute keys for R
- Repeat until all relations are in BCNF:
  – Pick any R’ with A->B that violates BCNF
  – Decompose R’ into R1(A, B) and R2(A, rest)
  – Compute FDs for R1 and R2
  – Compute keys for R1 and R2

Exercise 3

- Relation Z(A, B, C, D, E) has functional dependencies:

  fd1: A, B $\rightarrow$ C, D, E

  fd2: B, C $\rightarrow$ D

Decompose Z to relations in BCNF.

BCNF problem

- Consider following relation StudentLabTime:

Student	courseLab	time
111	Database	9:00
112	Database	9:00
113	Database	11:00
111	Multimedia	13:00
113	Multimedia	15:00

- Each course has several labs
- Only one lab (of any course at all) takes place at any given time
- Each student taking a course is assigned to a single lab for it

BCNF problem

Student	courseLab	time
111	Database	9:00
112	Database	9:00
113	Database	11:00
111	Multimedia	13:00
113	Multimedia	15:00

- FDs:

  Student, courseLab $\rightarrow$ time

  time $\rightarrow$ courseLab
- Candidate keys: (Student, courseLab) and (Student, time)

BCNF problem

- To change StudentLabTime to BCNF:

Student	time
111	9:00
112	9:00
113	11:00
111	13:00
113	15:00
111	**11:00**

time	courseLab
9:00	Database
11:00	Database
13:00	Multimedia
15:00	Multimedia

However the decomposition is not acceptable because it allows us to record multiple times of the same courseLab against the same student. That is, we have lost the FD: Student, courseLab $\rightarrow$ Time

BCNF problem

- A set of functional dependencies ***{AB → C, C → B}*** cannot be represented by a BCNF schema
- A design that eliminates all of these anomalies (but does not conform to BCNF) is possible. This design introduces a new normal form, know as Elementary Key Normal Form (EKNF).

What have we learned?

- BCNF definition
- How to decompose a relation to BCNF
- Situation where a relation cannot be represented in BCNF

Fourth Normal Form (4NF)

- Although BCNF removes anomalies due to functional dependencies, another type of dependency called a multi-valued dependency (MVD) can also cause data redundancy.

4NF - Example

BranchStaffOwner

branchNo	sName	oName
B003	Ann Beech	Carol Farrel
B003	David Ford	Carol Farrel
B003	Ann Beech	Tina Murphy
B003	David Ford	Tina Murphy

- This relation shows the names of staff who works at each branch and names of property owners registered at each branch office.
- Assume sName uniquely identifies each staff and oName uniquely identifies each owner.
- No relationship between members of staff and owners at a given branch.

Fourth Normal Form (4NF)

- Multi-valued Dependency (MVD)
  - Dependency between attributes (for example, A, B, and C) in a relation, such that for each value of A there is a set of values for B and a set of values for C. However, the set of values for B and C are independent of each other.
- MVD between attributes A, B, and C in a relation using the following notation:

  A –>> B

  A –>> C

Fourth Normal Form (4NF)

- A multi-valued dependency can be further defined as being trivial or nontrivial.
  - A MVD A −>> B in relation R is defined as being trivial if (a) B is a subset of A *or* (b) A $\cup$ B = R.
  - A MVD is defined as being nontrivial if neither (a) nor (b) are satisfied.
  - A trivial MVD does not specify a constraint on a relation, while a nontrivial MVD does specify a constraint.

Fourth Normal Form (4NF)

- 4NF: a relation is in 4NF if and only if for every nontrivial multi-valued dependency A->>B, A is a candidate key of the relation.

4NF - Example

BranchStaffOwner

branchNo	sName	oName
B003	Ann Beech	Carol Farrel
B003	David Ford	Carol Farrel
B003	Ann Beech	Tina Murphy
B003	David Ford	Tina Murphy

Two nontrivial dependencies:

branchNo ->> sName

branchNo ->> oName

Not in 4NF

Decompose to 4NF

BranchStaff

branchNo	sName
B003	Ann Beech
B003	David Ford

BranchOwner

branchNo	oName
B003	Carol Farrel
B003	Tina Murphy

Exercise 4

Student

studentID	sport	subject
45	Football	English
45	Football	Maths
45	Tennis	English
45	Tennis	Maths
46	Football	English
46	Football	Maths

- Is student relation in 4NF?

Decomposition Properties

- **Lossless**: Data should not be lost or created when splitting relations up

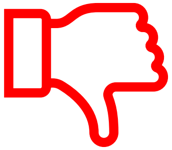
- **Dependency preservation**: It is desirable that FDs are preserved when splitting relations up

Decomposition Properties

- Normalization to 3NF is always lossless and dependency preserving
- Normalization to BCNF is lossless, but may not preserve all dependencies

- Normalization
  - Removes data redundancy
  - Solves INSERT, UPDATE, and DELETE anomalies
  - This makes it easier to maintain the information in the database in a consistent state

- However
  - It leads to more tables in the database
  - Often these need to be joined back together, which is expensive to do



  - So sometimes (not often) it is worth ‘**denormalizing**’

Denormalization

- You *might* want to denormalize if
  - Database speeds are unacceptable (not just a bit slow)
  - There are going to be very few INSERTs, UPDATEs, or DELETEs
  - There are going to be lots of SELECTs that involve the joining of tables

What have we learned?

- BCNF definition
- How to decompose a relation to BCNF
- MVD (trivial, nontrivial)
- 4th Normal Form
- How to decompose a relation to 4NF
- Denormalization