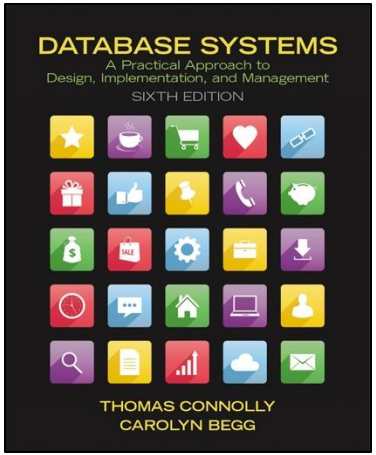
Reducing Data Redundancy

Topic 2 Lesson 7
Schema Refinement

Chapter 14 14.1-14.4 Connolly and Begg



Functional Dependencies

Reducing Data Redundancy

Major aim of relational database design is to group attributes into relations to minimize data redundancy.

Benefits:

Updates to the data stored in the database are achieved with a minimal number of operations thus reducing the opportunities for **data inconsistencies**.

Reduction in the file storage space required by the base relations thus **minimizing costs**.

What is wrong with this table?

Staff Branch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Data redundancy of the branch address

Data redundancy leads to update anomalies

Relations that contain redundant information may potentially suffer from **update anomalies.**

What are update anomalies?

- **Insertion anomaly**
  - Tuple being inserted may contain data fields that are inconsistent with data in other tuples in the table
- **Deletion anomaly**
  - Deleting a tuple leads to loss of information other than the tuple
- **Modification anomaly**
  - Modification of one tuple is dependent on the modifications of other tuples

Functional dependency

Important concept associated with normalization.

Functional dependency describes a relationship between attributes.

For example, if A and B are attributes of relation R,
B is functionally dependent on A (denoted $A \rightarrow B$),
if each value of A in R is associated with
exactly one value of B in R.

Types of Functional Dependencies

Full functional dependency indicates that if A and B are attributes of a relation,

B is fully functionally dependent on A,

if B is functionally dependent on A,

but not on any proper subset of A.

Partial dependency if B is also functionally dependent on a subset of A.

Transitive dependency

Important to recognize a transitive dependency because its existence in a relation can potentially cause update anomalies.

Transitive dependency describes a condition
where A, B, and C are attributes of a relation
such that if $A \rightarrow B$ and $B \rightarrow C$,
then C is transitively dependent on A via B
(if A is not functionally dependent on B or C).

Let’s remove the redundancy

Staff Branch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

A staff table and a branch table

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

Reconfigure tables

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

A staff table and a branch table

Redundancy only for the foreign key

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

Example of transitive dependency

Consider functional dependencies in the StaffBranch relation

StaffNo → sName, position, salary, **branchNo**, bAddress

branchNo → bAddress

Transitive dependency, StaffNo → branchNo
branchNo → bAddress
Transitive dependency staffNo to bAddress via branchNo.

Redundancy leads to anomalies

UPDATE ANOMALIES

Modification anomaly: Can we change W in just the first tuple?

Deletion anomaly: Can we delete tuple 3 and 4?

Insertion anomaly: What if we insert another tuple where the rating equals 8 but the wage is not equal to 10? How do we track the wage associated with ratings not stored in the employee table?

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

There is a functional dependency between Rate and Wage. This functional dependency limits the operations I can do on my data if I want to keep my data consistent.

Identifying functional dependencies

Functional dependencies, can be used to identify schemas with such problems and to suggest schema refinements.

Each relation is dependent on the primary key since the primary key identifies the values for the other attributes

In our prior example, we had 2 functional dependencies (FDS)

S → {S,N,L,R,W,H}

R → {W} - each value of R is associated with exactly 1 value of W

W → {R} Determinant on left hand side of → dependent set on right

If no attributes are dependent on another (not including the primary key) then there is no redundancy

How to remove a functional dependency?

S	N	L	R	W	H
123-22-3666	Attishoo	48	8	10	40
231-31-5368	Smiley	22	8	10	30
131-24-3650	Smethurst	35	5	7	30
434-26-3751	Guldu	35	5	7	32
612-67-4134	Madayan	35	8	10	40

Decompose the original relation into 2 relations.

S	N	L	R	H
123-22-3666	Attishoo	48	8	40
231-31-5368	Smiley	22	8	30
131-24-3650	Smethurst	35	5	30
434-26-3751	Guldu	35	5	32
612-67-4134	Madayan	35	8	40

R	W
8	10
5	7

Wage

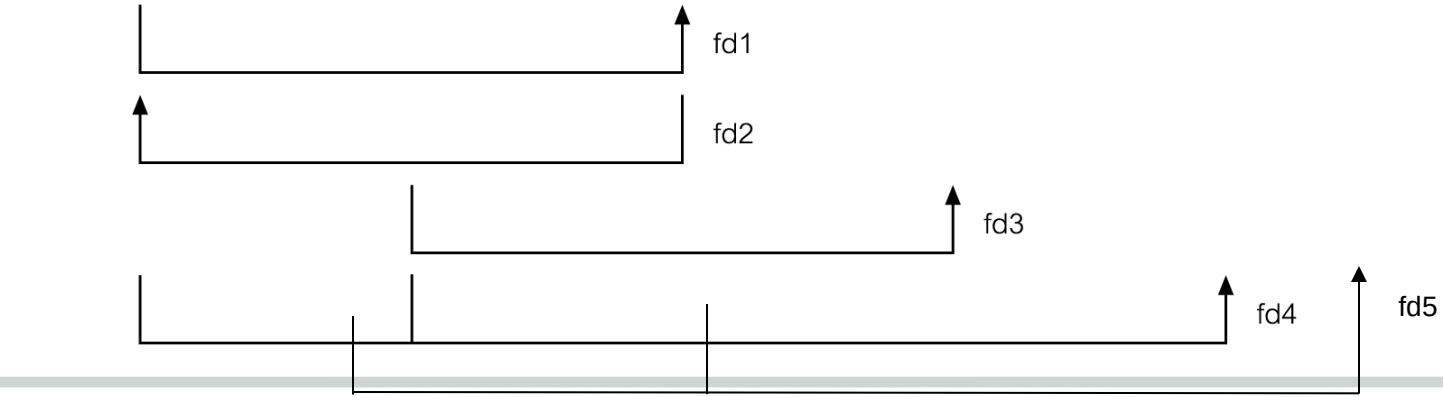
Example: Find the functional dependencies

Sample Relation

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t

Given these values:

A	B	C	D	E
a	b	z	w	q
e	b	r	w	p
a	d	z	w	t
e	d	r	w	q
a	f	z	s	t
e	f	r	s	t



Identified functional dependencies

Functional dependencies between attributes A to E in the Sample relation.

$A \rightarrow C$	(fd1)
$C \rightarrow A$	(fd2)
$B \rightarrow D$	(fd3)
$A, B \rightarrow E$	(fd4)
$B, C \rightarrow E$	(fd5)

Candidate keys for sample relation

A candidate key must provide the irreducibility and uniqueness property. We do not have 1 attribute that can determine all attributes.

A,B determines C, D, and E

B,C determines A, D, and E

So, we have 2 candidate keys.

Decomposition 1 AB→ E

A	B	E
a	b	q
e	b	p
a	d	t
e	d	q
a	f	t
e	f	t

B	D
b	w
d	w
f	s

A	C
a	z
e	r

Decomposition 2 CB → E (Alternative)

C	B	E
z	b	q
r	b	p
z	d	t
r	d	q
z	f	t
r	f	t

B	D
b	w
d	w
f	s

A	C
a	z
e	r

Identifying functional dependencies

Main characteristics of functional dependencies used in normalization:

- There is a **one-to-one** relationship between the attribute(s) on the left-hand side (determinant) and those on the right-hand side (dependency set) of a functional dependency.
- Holds for **all** time.
- The determinant has the **minimal** number of attributes necessary to maintain the dependency with the attribute(s) on the right hand-side.

Decomposition properties

Decomposition process must ensure:

- **Lossless-join property** enables us to find any instance of the original relation from corresponding instances in the smaller relations.
  - We can still generate the original table from the decomposed tables (no loss of information)
- **Dependency preservation property** enables us to enforce a constraint on the original relation by enforcing some constraint on each of the smaller relations
  - No functional dependency is lost in the process

Lossless Join Property guarantees that

- The union of the attributes in the 2 smaller tables must be equal to the attributes in the larger table
- The intersection of the attributes in the 2 smaller tables must not be empty
- A common attribute in one of the relations must be a key in the other table

Summary

- Functional dependencies not involving a candidate key can lead to update anomalies
- Update anomalies can lead to data inconsistencies
- Schema refinement using the normalization process can resolve update anomalies