Database Management System (DBMS)

L-7:

Normalization

Prof. Dr. Kamruddin Nur

Kamruddin.nur@gmail.com

Lecture Content

- Normalization
- Why Normalization
- Normal Forms

Reading: Chapter - 13 Text Book

Recap

- *The main objective of relational database is to create an accurate representation of data, relationships between data, and constraints*
- *To achieve this objective,*

  - We must identify a suitable set of relations

- *A technique that helps such (accurate) relations is* ***Normalization***

Normalization

"A technique to produce / design a set of relations that is optimal from the point of view of database updating."

- Series of tests on a relation to determine whether it satisfies or violates the requirements of a given normal form

- 3 Normal forms are initially proposed by **E. F. Codd (1972)**
  - **First Normal Form (1NF)**
  - **Second Normal Form (2NF)**
  - **Third Normal Form (3NF)**

- Subsequently, R. Boyce and E. F. Codd (Codd, 1974) introduced a stronger definition of 3NF, Called **Boyce-Codd Normal Form (BCNF)**.
- Later,
  - **4NF** and **5NF** was introduced, Fagin (1977, 1979)

Normalization

- ✔ Formal method that identifies relations based on their primary or candidate keys and the functional dependencies among their Attributes.
- ✔ *Series of tests, which can be applied on individual relations so that a relational schema can be normalized to a specific form to prevent the possible update anomalies*
- ✔ *Update anomalies are insertion, deletion, or modification anomalies*

↳ abnormalities / problems

Data Redundancy

- Major aim of relational database design is to group attributes into relations to minimize data redundancy and reduce file storage space required by base relations.
- Problems associated with data redundancy are illustrated by comparing the following Staff and Branch relations with the StaffBranch

Reduction of Data redundancy / Removal

Data Redundancy

StaffBranch

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

Staff

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

Branch

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

Data Redundancy

- StaffBranch relation has redundant data: details of a branch are repeated for every member of staff.

- In contrast, branch information appears only once for each branch in Branch relation and only branchNo is repeated in Staff relation, to represent where each member of staff works.

Update Anomalies

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

- Types of update anomalies include:
  - ✔ Insertion
  - ✔ Deletion
  - ✔ Modification / edit / update

Data 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

S123 Mr. X " 10000 B005 163 Main St, Glasgow

X NULL — — — B008 57 Princess St, Edinbu

Insertion Anomalies

- **New member of staff joins branch B005**
  - Insert new row into StaffBranch table
  - Type wrong address: 163 Main St, Glasgow.
  - Database is now inconsistent!

- **Establish new branch with no members of staff**
  - B008, 57 Princes St, Edinburgh
  - No staff members, so staffNo must be NULL
  - But staffNo is the primary key of the StaffBranch table, so cannot be NULL!

Deletion Anomalies

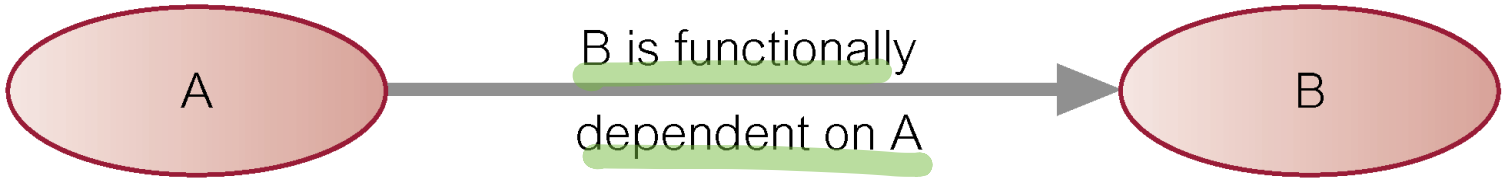
- **Mary Howe, staffNo SA9, leaves the company**
  - Delete the appropriate row of StaffBranch
  - This also deletes details of branch B007 where Mary Howe works
  - But no-one else works at branch B007, so we no longer know the address of this branch!

Modification Anomalies

- ✔ **Branch B003 has transferred to a new location**
  - New address is 145 Main St, Glasgow
  - Must change *three* rows of the StaffBranch relation

Functional Dependency

- Main concept associated with Normalization
- Describes relationships between attributes in a relation
- If **A** and **B** are attributes of relation ***R***
  if each value of **A** in ***R*** is associated with exactly one value of **B** in ***R*** then **A** → **B**



- Left hand side of a functional dependency is called a determinant. Here, A is the determinant

Functional Dependency cont.

Let A, B, and C be subsets of the attributes of relation R.

Armstrong's axioms are as follows:

1. Reflexivity

If B is a subset of A, then A →B

B is functionally dependent on A

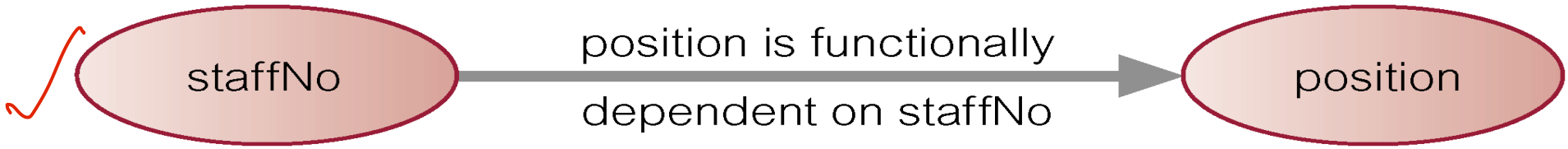
2. Augmentation

If A →B, then A,C →C

3. Transitivity

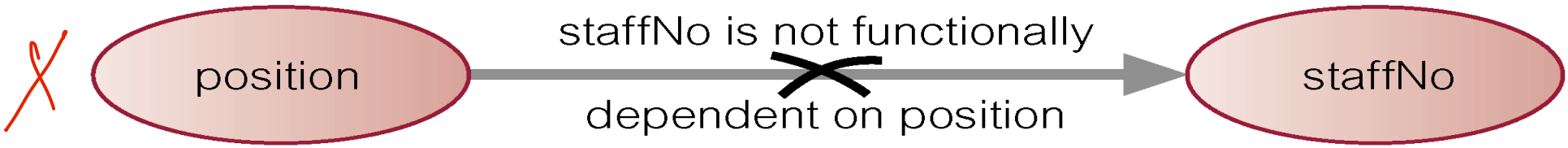
If A →B and B →C, then A →C

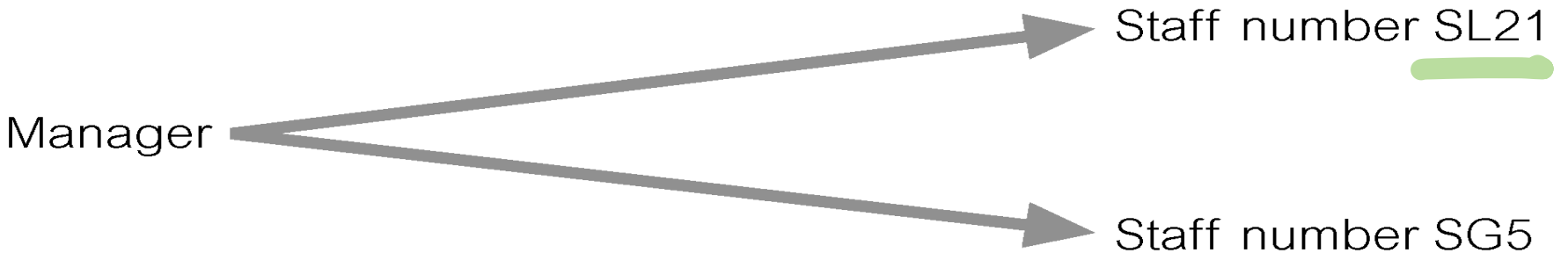
Example: Functional Dependency



Staff number SL21 → Manager

(a)





(b)

Identifying Candidate Keys

- ✔ **A candidate key is an attribute, or set of attributes, that uniquely identifies a row**
  - Must be *irreducible*
  - *No part of a candidate can ever be NULL*

- ✔ **An attribute A that functionally determines *every other* attribute of the relation is a candidate key**
  - For each value of A there is exactly one value of each of the other attributes
  - So each value of A must identify a single row

ID → stud name, address

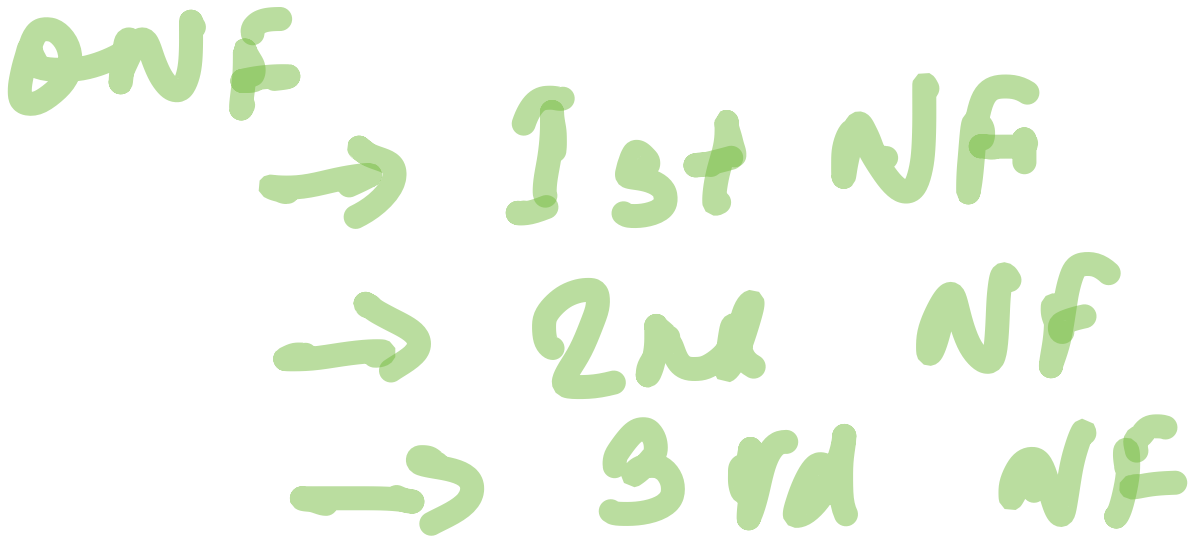
Identifying Primary Keys

- **A primary key is a candidate key chosen to identify rows uniquely within a table**
  - Other candidate keys called *alternate keys*

- **Some guidelines on choosing the primary key**
  - Pick the candidate key with fewest attributes
  - Pick the candidate key with shortest length
  - Pick the candidate key that makes most sense

Why Normalization?

- ✔ The main objective of relational database is to create an accurate representation of data, its relationships and constraints.
- ✔ The achieve the above objective,
  We must identify a suitable set of relations.
- ✔ Normalization process helps identifying such relations.

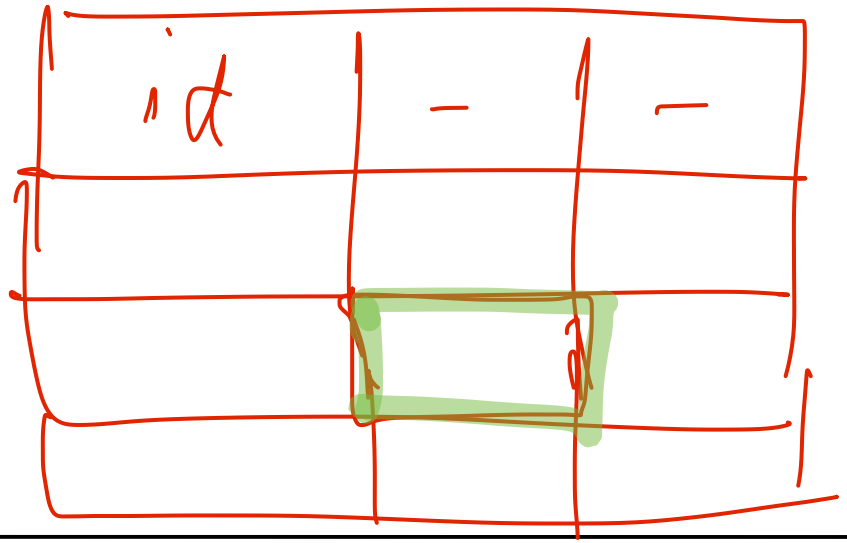


1st Normal Form

A relation in which intersection of each row and column contains one and only one value.

How to achieve:

➔ if required break table into different entity table to minimize redundant data (update anomalies).



ONF to 1NF

0NF

Module	Dept	Lecturer	Texts
M1	D1	L1	T1, T2
M2	D1	L1	T1, T3
M3	D1	L2	T4
M4	D2	L3	T1, T5
M5	D2	L4	T6

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

Problems with 1NF

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

M6

INSERT anomalies:

Can not add a module with no texts

UPDATE anomalies:

To change lecturer for M1, we have to change two rows

DELETE anomalies:

If we remove M3, we remove L2 as well

2nd Normal Form

A relation that is in 1st Normal Form and every non-primary-key attribute is fully functionally dependent on the primary key.

Full Functional Dependency:

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**

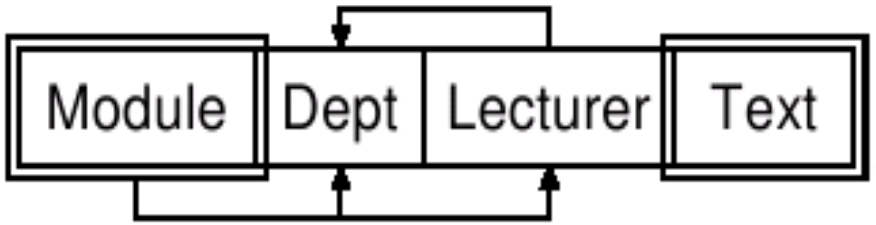
How To Achieve:

- ➔ Break into tables by removing non-primary-attributes along with a copy of part of primary key on which they are fully functionally dependent.
- ➔ In other word, making attributes fully functional dependent on primary keys.

Finding Functional Dependencies (FD)

1NF

Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6



The primary key is {Module, Text}

so,

{Module, Text} → {Dept, Lecturer}

{Module}→ {Dept}

{Module}→ {Lecturer}

{Lecturer}→ {Module}

But also,

{Module} → {Dept, Lecturer}

So, Lecturer and Dept are partially dependent on primary key!

1NF to 2NF

1NF

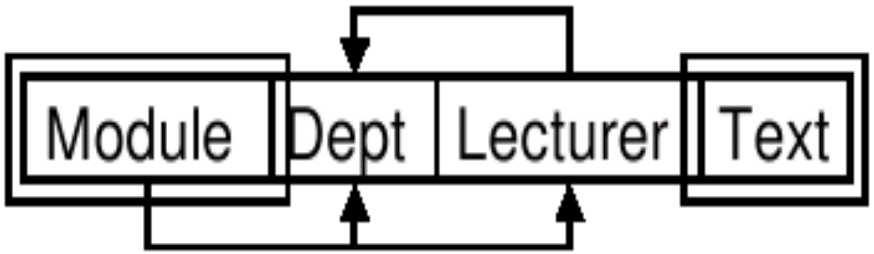
Module	Dept	Lecturer	Text
M1	D1	L1	T1
M1	D1	L1	T2
M2	D1	L1	T1
M2	D1	L1	T3
M3	D1	L2	T4
M4	D2	L3	T1
M4	D2	L3	T5
M5	D2	L4	T6

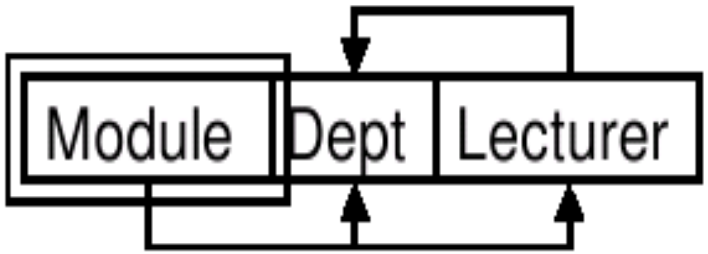
2NFa

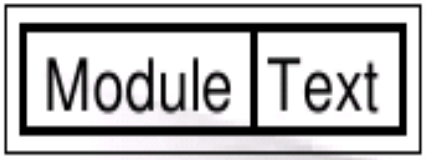
Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4

2NFb

Module	Text
M1	T1
M1	T2
M2	T1
M2	T3
M3	T4
M4	T1
M4	T5
M1	T6







Problems Resolved in 2NF

Problems in 1NF:

INSERT anomalies

Can not add a module with no texts

UPDATE anomalies:

To change lecturer for M1, we have to change two rows

DELETE anomalies:

If we remove M3, we remove L2 as well

In 2NF:

the first two problems (**INSERT** and **UPDATE**)are resolved but not **DELETE**

2NFa

Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4

3rd Normal Form

A relation which is in 1st and 2nd Normal Form, and in which no non-primary-key attribute is transitively dependent on the primary key.

Transitive Dependency:

if **A →B, B →C**

Then A →C

if and only if B ↛A and C ↛A

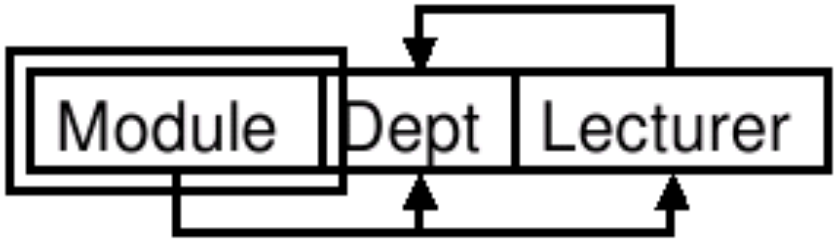
How To Achieve:

- In the above transitive dependency, A is not functionally dependent on any of B or C
- Which means B or C are not a part of a relation which has attribute A.
- So create a table with B and C.

2NF not In 3NF

2NFa

Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4



2NFa is not in 3NF

Because,

{Module}→ {Lecturer}

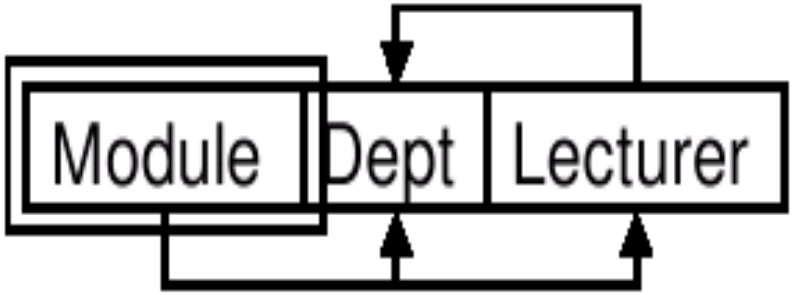
{Lecturer} → {Dept}

So, there is a transitive FD form the primary key {Module} to {Dept}

2NF to 3NF

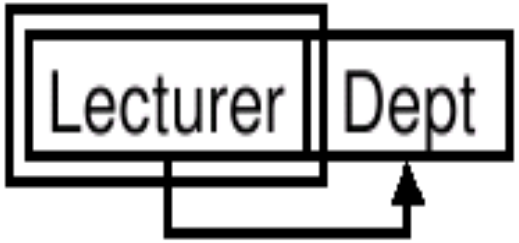
2NFa

Module	Dept	Lecturer
M1	D1	L1
M2	D1	L1
M3	D1	L2
M4	D2	L3
M5	D2	L4



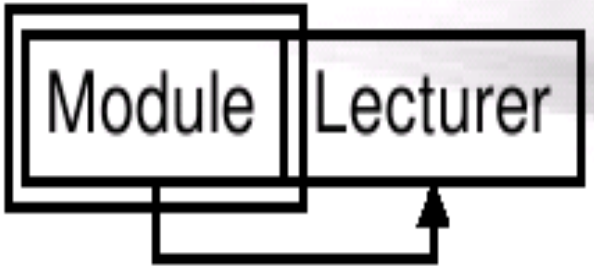
3NFa

Lecturer	Dept
L1	D1
L2	D1
L3	D2
L4	D2



3NFb

Module	Lecturer
M1	L1
M2	L1
M3	L2
M4	L3
M5	L4



Summary

From this lecture we have learned the details of

- Normalization
- Data Redundancy
- Functional Dependencies
- Insert, Update, Delete Anomalies
- 1NF, 2NF and 3NF
- A database should be at least in 3NF