Normalization

Databases and Interfaces

Yuan Yao & Matthew Pike

Recall: Relational model

Attributes are: ID, Name, Salary & Department

The degree of the relation is 4

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts

Schema is { ID, Name, Salary, Department }

Tuples, e.g.
{ (ID, A368),
(Name, Jane Brown),
(Salary, 22,000),
(Department, Accounts) }

The cardinality of the relation is 5

Warm up: An example database

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI, FAI	Databases and Interfaces, Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA, DBI	Programming and Algorithms, Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

- Each staff belongs to one department.
- Each department has its own building.

Warm up: Non-atomic values

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI, FAI	Databases and Interfaces, Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA, DBI	Programming and Algorithms, Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

Warm up: A table with atomic values

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

Warm up: Data Redundancy

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

Overview

- Normalization
  - Data Redundancy
  - Unnormalized Data → Normal Forms
  - Functional Dependencies
  - Normal Forms

What is Normalization?

- Normalization: A technique for producing a set of relations with desirable properties, given the data requirements.
- Desirable relations:
  - The minimal number of attributes necessary to support the data requirement.
  - Attributes with a close logical relationship in the same relation.
  - Minimal redundancy.
- Why normalized data:
  - Easy to access and maintain
  - Minimal storage space.
- Normalization steps: 1NF, 2NF, 3NF.

1NF

Normalization to 1NF

- To convert any relation into 1NF, split any non-atomic values

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI, FAI	Databases and Interfaces, Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA, DBI	Programming and Algorithms, Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

1NF

Any problems ?

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

INSERT Anomalies

- If we want to add a new staff David Ford from Computer Science with ID 005

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

INSERT Anomalies

- If we want to add a new staff David Ford from Computer Science with ID 005

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists
005	David Ford	Computer Science	B1	Null	Null

DELETION Anomalies

- If we want to delete module MCS

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

DELETION Anomalies

- If we want to delete module MCS

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
~~004~~	~~David Jones~~	~~Mathematics~~	~~A1~~	~~MCS~~	~~Mathematics for Computer Scientists~~

UPDATE Anomalies

- Mary Jones is now transferred to the department of Mathematics

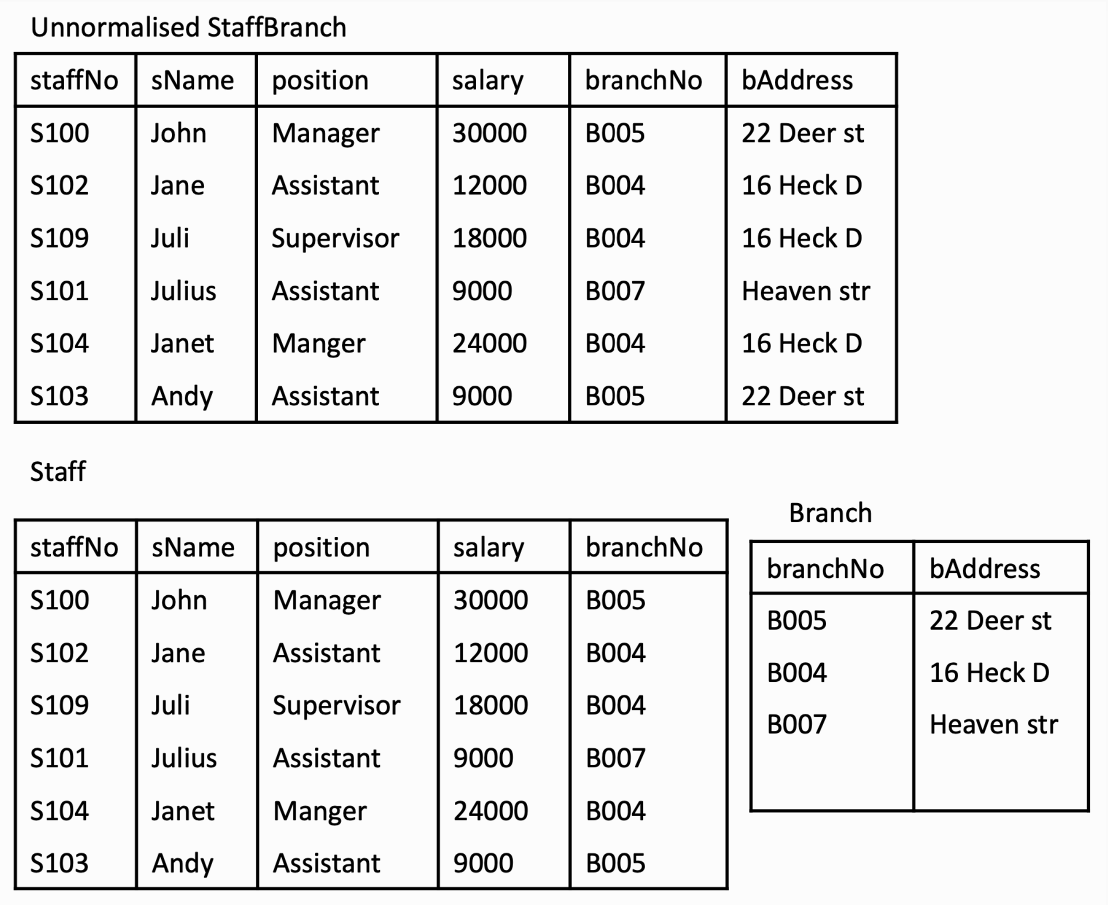
staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

UPDATE Anomalies

- Mary Jones is now transferred to the department of Mathematics

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Mathematics	B1	PGA	Programming and Algorithms
003	Mary Jones	Mathematics	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

Decomposition



Unnormalised StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
S100	John	Manager	30000	B005	22 Deer st
S102	Jane	Assistant	12000	B004	16 Heck D
S109	Juli	Supervisor	18000	B004	16 Heck D
S101	Julius	Assistant	9000	B007	Heaven str
S104	Janet	Manger	24000	B004	16 Heck D
S103	Andy	Assistant	9000	B005	22 Deer st

Staff

staffNo	sName	position	salary	branchNo
S100	John	Manager	30000	B005
S102	Jane	Assistant	12000	B004
S109	Juli	Supervisor	18000	B004
S101	Julius	Assistant	9000	B007
S104	Janet	Manger	24000	B004
S103	Andy	Assistant	9000	B005

Branch

branchNo	bAddress
B005	22 Deer st
B004	16 Heck D
B007	Heaven str

- Decomposition
  - Decompose a large relation into smaller relations
- Properties:
  - Lossless-join: any instance of the original relation can be identified in the smaller relations
  - Dependency preservation: All constraints still remain.

Functional Dependencies

- Functional Dependencies (FD) is a link between two sets of attributes in a relation.

- A set of attributes A functionally determines another set B, (A → B):
    - If whenever two rows of the relation have the same value for all the attributes in A, then they also have the same values for all the attributes in B.

-

ID	First	Last
…	…	…

{ID} → {First}

{ID} → {First, Last}

{First} → {Last}

{ID, First} → {Last}

{First, Last} → {ID}

Exercise 1: FDs

staffID	Name	Department	Building	ModuleCode	Module
001	John Smith	Computer Science	B1	DBI	Databases and Interfaces
001	John Smith	Computer Science	B1	FAI	Foundation of Artificial Intelligence
002	Mark Brown	Computer Science	B1	FAI	Foundation of Artificial Intelligence
003	Mary Jones	Computer Science	B1	PGA	Programming and Algorithms
003	Mary Jones	Computer Science	B1	DBI	Databases and Interfaces
004	David Jones	Mathematics	A1	MCS	Mathematics for Computer Scientists

Why we care about FDs?

staffID	Name	Department	Building	ModuleCode	Module
…	…	…		…	…

Redundancy is often caused by a functional dependency:

- {StaffID} → {Name, Department}
- {Department} → {Building}
- {ModuleCode} → {Module}

Normal Forms (e.g., 1NF, 2NF, 3NF):

- Each Normal Form has fewer FDs than the last. (what does that mean?)
- Not all FDs cause a problem
- Each NF removes a type of FD that is a problem
- Need a way to remove FDs

Properties of FDs

- In any relation:
    - The candidate keys functionally determine any set of attributes in that relation.
        - K → X

          where K is a candidate key
    - Any set of attributes is FD on itself
        - X → X

- Rules:
    - Reflexivity:
        - If B ⊆ A, then A → B
    - Augmentation:
        - If A → B, then A ∪ C → B ∪ C
    - Transitivity:
        - If A → B and B → C, then A → C

Functional Dependencies

staffID	Name	Department	Building	ModuleCode	Module
…	…	…		…	…

- Module is the Primary key:
  - {staffID, ModuleCode} → {Name, Department, Building, Module}, …
- Reflexivity:
  - {Name, Department} → {Name}, …
- Augmentation:
  - {ModuleCode, StaffID} → {Module, StaffID}
- Transitivity:
  - {staffID} → {Department}, {Department} → {Building}, {staffID} → {Building}

Full vs Partial functional dependency

staffID	Name	Department	Building	ModuleCode	Module
…	…	…		…	…

- Full FD:
  - $A \rightarrow B$ is full FD, if there is no such $C \subset A$, $C \rightarrow B$
  - E.g., {ModuleCode} → {Module}
- Partial FD:
  - $A \rightarrow B$ is partial FD, if $C \subset A$, such that $C \rightarrow B$
  - E.g., {staffID, Name} → {Department}

2NF

Second Normal Form

- 2NF:
  - It is in 1NF
  - There is no non-key attributes B is partially dependent on a candidate key.
  - No C → B, where B is a set of non-key attributes, C is a strict subset of candidate key.

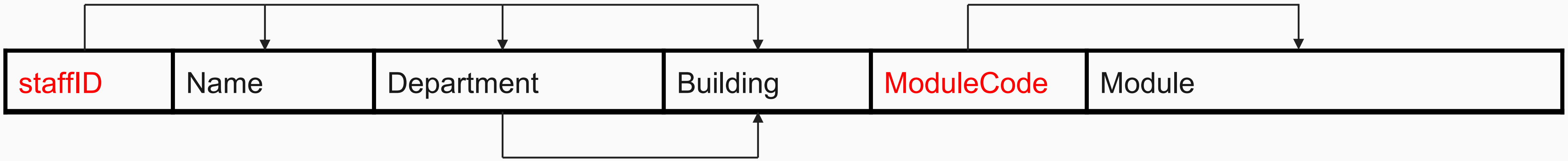
- Attributes:
  - Key attributes: part of some candidate key.
  - Non-key attributes: the rest non-key attributes.

Exercise 2: 2NF

staffID	Name	Department	Building	ModuleCode	Module
…	…	…		…	…

Is this relation in 2NF? If not, find the partial FDs on a candidate key.

Exercise 2: 2NF



Candidate key: {staffID, ModuleCode}

Partial FD: {staffID, ModuleCode} → {Module}
{staffID, ModuleCode} → {Name, Department, Building}

Removing FDs

Suppose we have a relation R with schema S,
a FD $A \rightarrow B$, where:

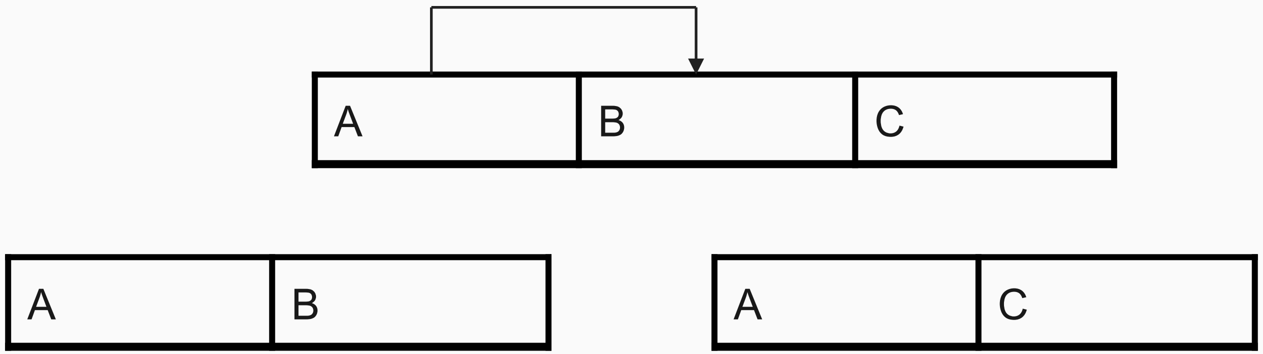
- A is a key attribute.
- $A \cap B = \{\}$

Let $C = S – (A \cup B)$, i.e., all other attributes

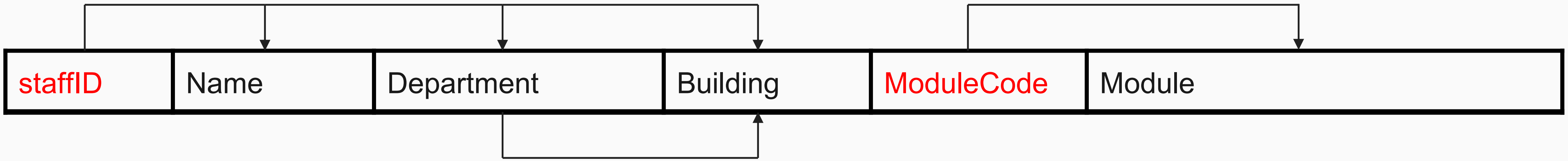
We could divide S into two parts:

- R1, with schema $C \cup A$
- R2, with schema $A \cup B$

$S = R1 \bowtie R2$



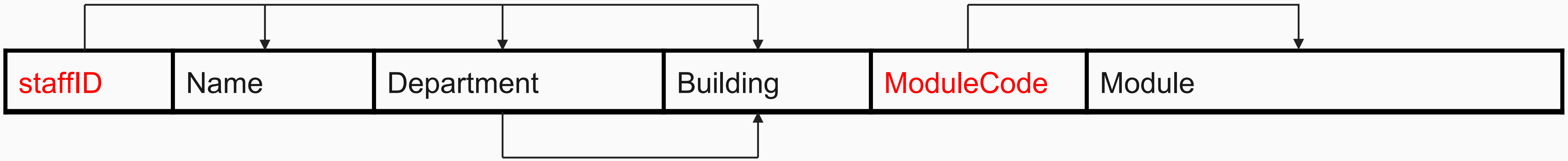
Exercise 3: Removing FDs



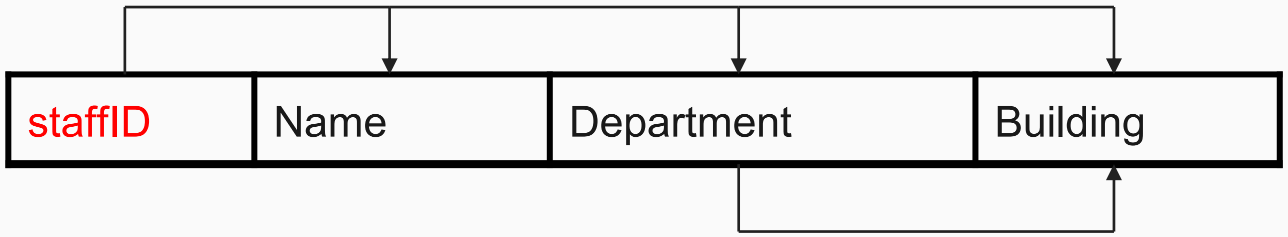
staffID	Name	Department	Building	ModuleCode	Module

Exercise 3: Removing FDs

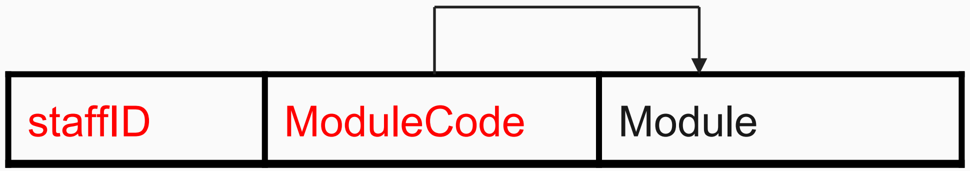
A : staffID B: {Name, Department, Building} C: {ModuleCode, Module}



staffID	Name	Department	Building	ModuleCode	Module



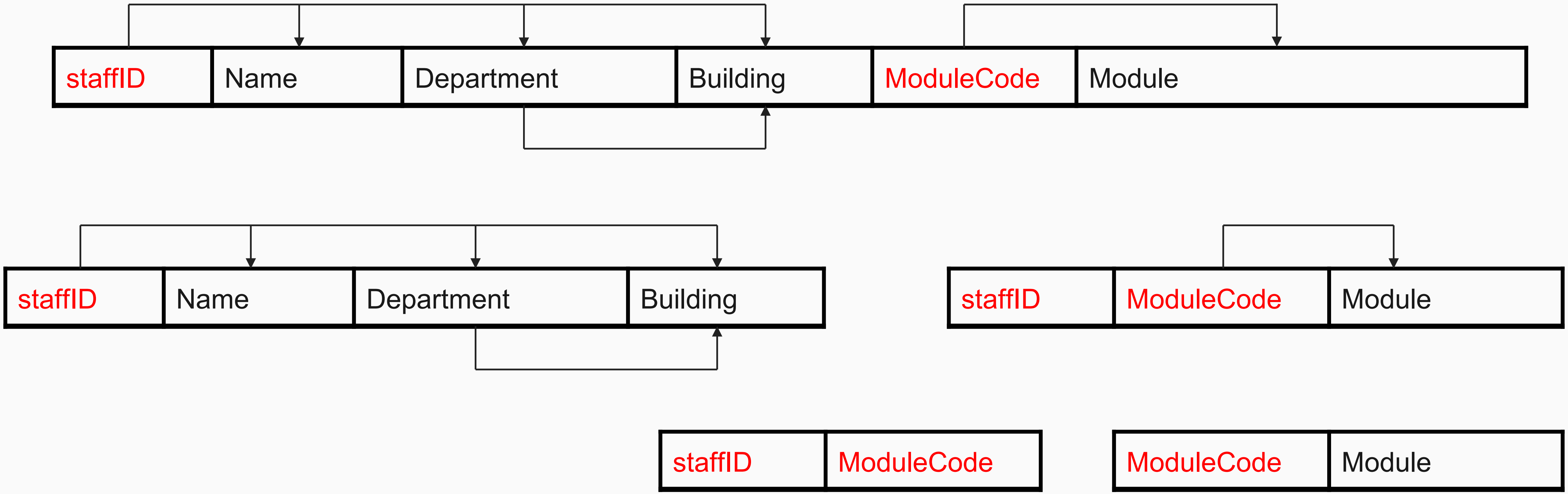
staffID	Name	Department	Building



staffID	ModuleCode	Module

Exercise 3: Removing FDs

A : staffID B: {Name, Department, Building} C: {ModuleCode, Module}



staffID	Name	Department	Building	ModuleCode	Module

staffID	Name	Department	Building

staffID	ModuleCode	Module

staffID	ModuleCode

ModuleCode	Module

Example Database in 2NF

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

Recall: the problems we have in 1NF

- If we want to add a new staff David Ford from Computer Science with ID 005

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

Insertion anomalies in 1NF

- If we want to add a new staff David Ford from Computer Science with ID 005

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1
005	David Ford	Computer Science	B1?

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

Insertion anomalies in 2NF

- If we want to add the department of Chemistry that is in building C1

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1
null	null	Chemistry	C1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

Deletion anomalies in 1NF

- If we want to delete module MCS

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

Deletion anomalies in 2NF

- If we want to delete module MCS

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
~~MCS~~	~~Mathematics for Computer Scientists~~

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
~~004~~	~~MCS~~

Deletion anomalies in 2NF

- What if we want to delete David Jones?

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
~~004~~	~~David Jones~~	~~Mathematics~~	~~A1~~

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
~~004~~	~~MCS~~

Update anomalies in 1NF

- Mary Jones is now transferred to the department of Mathematics

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

Update anomalies in 2NF

- Mary Jones is now transferred to the department of Mathematics

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Mathematics	B1?
004	David Jones	Mathematics	A1

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

3NF

3NF

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

- A special type of FDs: transitive FD.
  - A → C is a transitive FD, if there is some set B such that A → B and B → C.
  - For example, {staffID} → {Department} → {Building}
- Third Normal Form (3NF):
  - It is in 2NF.
  - No non-key attribute is transitively dependent on a candidate key.

3NF

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

- If A → B and B → C, S is the schema.
- Divide S into two parts:
  - R1, with schema $B \cup C$.
  - R2, with schema $S - C$.

Exercise 4: Removing transitive FDs

staffID	Name	Department	Building
001	John Smith	Computer Science	B1
002	Mark Brown	Computer Science	B1
003	Mary Jones	Computer Science	B1
004	David Jones	Mathematics	A1

Exercise 4: Removing transitive FDs

staffID	Name	Department
001	John Smith	Computer Science
002	Mark Brown	Computer Science
003	Mary Jones	Computer Science
004	David Jones	Mathematics

Department	Building
Computer Science	B1
Mathematics	A1

Example in 3NF

staffID	Name	Department
001	John Smith	Computer Science
002	Mark Brown	Computer Science
003	Mary Jones	Computer Science
004	David Jones	Mathematics

Department	Building
Computer Science	B1
Mathematics	A1

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

Insertion anomalies in 3NF

- If we want to add the department of Chemistry that is in building C1

staffID	Name	Department
001	John Smith	Computer Science
002	Mark Brown	Computer Science
003	Mary Jones	Computer Science
004	David Jones	Mathematics

Department	Building
Computer Science	B1
Mathematics	A1
Chemistry	C1

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

Deletion anomalies in 3NF

- What if we want to delete David Jones?

staffID	Name	Department
001	John Smith	Computer Science
002	Mark Brown	Computer Science
003	Mary Jones	Computer Science
~~004~~	~~David Jones~~	~~Mathematics~~

Department	Building
Computer Science	B1
Mathematics	A1

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

Update anomalies in 3NF

- Mary Jones is now transferred to the department of Mathematics

staffID	Name	Department
001	John Smith	Computer Science
002	Mark Brown	Computer Science
003	Mary Jones	Mathematics
004	David Jones	Mathematics

Department	Building
Computer Science	B1
Mathematics	A1

staffID	ModuleCode
001	DBI
001	FAI
002	FAI
003	PGA
003	DBI
004	MCS

ModuleCode	Module
DBI	Databases and Interfaces
FAI	Foundation of Artificial Intelligence
PGA	Programming and Algorithm
MCS	Mathematics for Computer Scientists

Summary

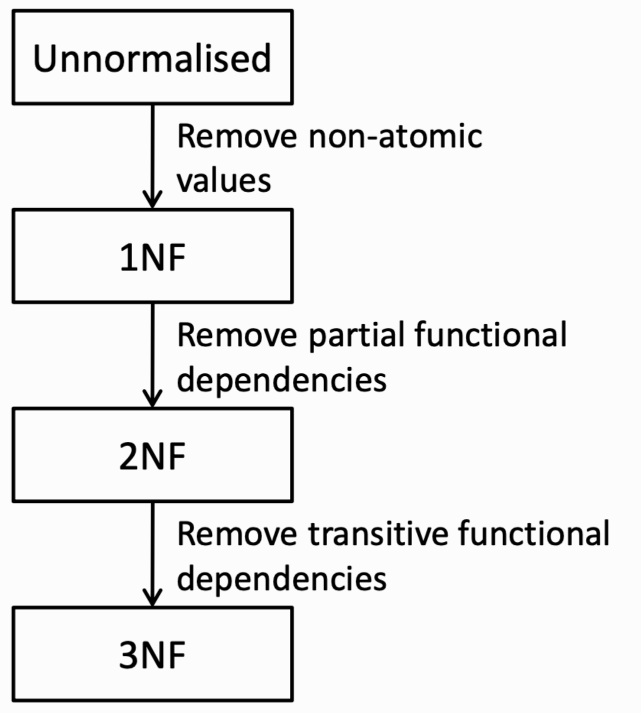
Normalization and Design

- Normalization is related to Database design
  - A database should normally be in 3NF at least
  - If your design leads to a non-3NF database, then you might want to revise it

- When you find you have a non-3NF database
  - Identify the FDs that are causing a problem
  - Think if they will lead to any insert, update, or delete anomalies
  - Try to remove them

Summary

- Normalization
  - Data Redundancy
  - Functional Dependencies
  - Normal Forms
  - First, Second and Third Normal Forms



Unnormalised

Remove non-atomic values

1NF

Remove partial functional dependencies

2NF

Remove transitive functional dependencies

3NF

Relational Algebra VS SQL

Recall: Relational model VS E/R model

Relational Model	E/R Model
Relation	Entity Type
Tuple	Entity Instance
Attribute	Attribute
Primary Key	Attribute
Foreign Key	1:M relationship

Relational model VS E/R model VS SQL

Relational Model	E/R Model	SQL
Relation	Entity Type	Table
Tuple	Entity Instance	Row
Attribute	Attribute	Column
Primary Key	Attribute	Primary Key
Foreign Key	1:M relationship	Foreign Key

Recall: Relational Algebra

- Selection (σ) : $\sigma_{\alpha(R)}$, select tuples that satisfy condition α from relation *R.*
- Projection (π) : $\pi_X(R)$, project a set of attributes *X* from relation *R*.
- Union (∪) : A ∪ B, set-union, A and B must be compatible.
- Difference (−) : A − B, set-difference, A and B must be compatible.
- Intersection (∩) : A ∩ B, set-intersection, A and B must be compatible.
- Cartesian product (×) : A × B, all possible combinations of tuples from A and B.
- Natural join (⋈) : A ⋈ B, enforces equality on all attributes with the same name
- Theta join ($\bowtie_\theta$) : A $\bowtie_\theta$ B, Cartesian product satisfying certain properties θ.
- Rename (ρ) : ρR(A_1, A_2, …A_n)(E), create a copy of E with relation name R, and attributes name A_1, A_2, …A_n

SQL and Data manipulation

- SQL is relationally complete: it implements all operations of relational algebra.
- However, some implementation of SQL operations do not exist in relational algebra:

  e.g., counting, finding maximum value, sum …
- SQL does things like formatting.
- Most importantly, SQL contains active operations which update data in database, while relational algebra operations are passive: they just retrieve the information which is there

SQL Select

General syntax of the select-statement:

```
SELECT [DISTINCT] column [, column]* | *
FROM table [,table]*
[Where          test]
[GROUP BY column [, column]*
[HAVING function-list]]
[ORDER BY column [ASC|DESC] [, column [ASC|DESC]]*]
```

Restricted Select statement

```
SELECT DISTINCT column [, column]* | *
FROM   table [,table]*
[Where          test]
```

We assume there are no aggregate functions, ORDER BY or GROUP BY.

Converting basic select statements

Simple retrieval (one table), no where

SQL:

```
SELECT Name
FROM Student;
```

Relational Algebra:

π_{Name} Student

Simple retrieval (one table), with where clause

SQL:

```
SELECT Name
FROM Student
Where Age > 18;
```

Relational Algebra:

$\pi_{\text{Name}} (\sigma_{\text{Age>18}})$Student

Converting basic select statements

Multiple tables

SQL:

```
SELECT Name, Lecturer
FROM Student, Course;
```

Relational Algebra:

$\pi_{\text{Name, Lecturer}}$ (Student × Course)

Natural Join

SQL:

```
SELECT Name, Module
FROM Student NATRUAL JOIN
Mark;
```

Relational Algebra:

$\pi_{\text{Name, Module}}$ (Student $\bowtie$ Mark)

Converting basic select statements

Inner Join

SQL:

```
SELECT Name, Module
FROM Student INNER JOIN
Mark ON Student.ID = Mark.ID;
```

Relational Algebra:

$$\pi_{\text{Name, Module}} \text{ (Student } \bowtie_{\text{(Student.ID = Mark.ID)}} \text{ Mark)}$$

There are special operators for left join and right join

⟕ ⟖

Union

SQL:

```
SELECT Name, Module
FROM Student NATURAL JOIN Mark
WHERE Marks > 70
UNION
SELECT Name, Module
FROM Student NATURAL JOIN Mark
WHERE Marks < 40;
```

Relational Algebra:

$\pi_{\text{Name, Module}}\ (\sigma_{\text{Marks} > 70})(\text{Student} \bowtie \text{Mark})$

U

$\pi_{\text{Name, Module}}\ (\sigma_{\text{Marks} < 40})(\text{Student} \bowtie \text{Mark})$

Exercise 5: SQL and Relational Algebra

Student

sID	name
…	…

Grade

sID	mCode	grade
…	…	…

Module

mCode	title	credits
…	…	…

```
SELECT name, title, grade

FROM Student NATURAL JOIN Grade NATURAL JOIN Module

WHERE credits > 10 AND grade > 70;
```

Exercise 5: SQL and Relational Algebra

Student

sID	name
…	…

Grade

sID	mCode	grade
…	…	…

Module

mCode	title	credits
…	…	…

SELECT name, title, grade

FROM Student NATURAL JOIN Grade NATURAL JOIN Module

WHERE credits > 10 AND grade > 70;

$$\pi_{\text{name,title,grade}}(\sigma_{\text{credits>10 AND grade>70}})(\text{Student} \bowtie \text{Grade} \bowtie \text{Module})$$