Database Management

COMP 3010E FALL 2025

LECTURE 5 NORMAL FORMS

Why Do We Need Foreign Keys?

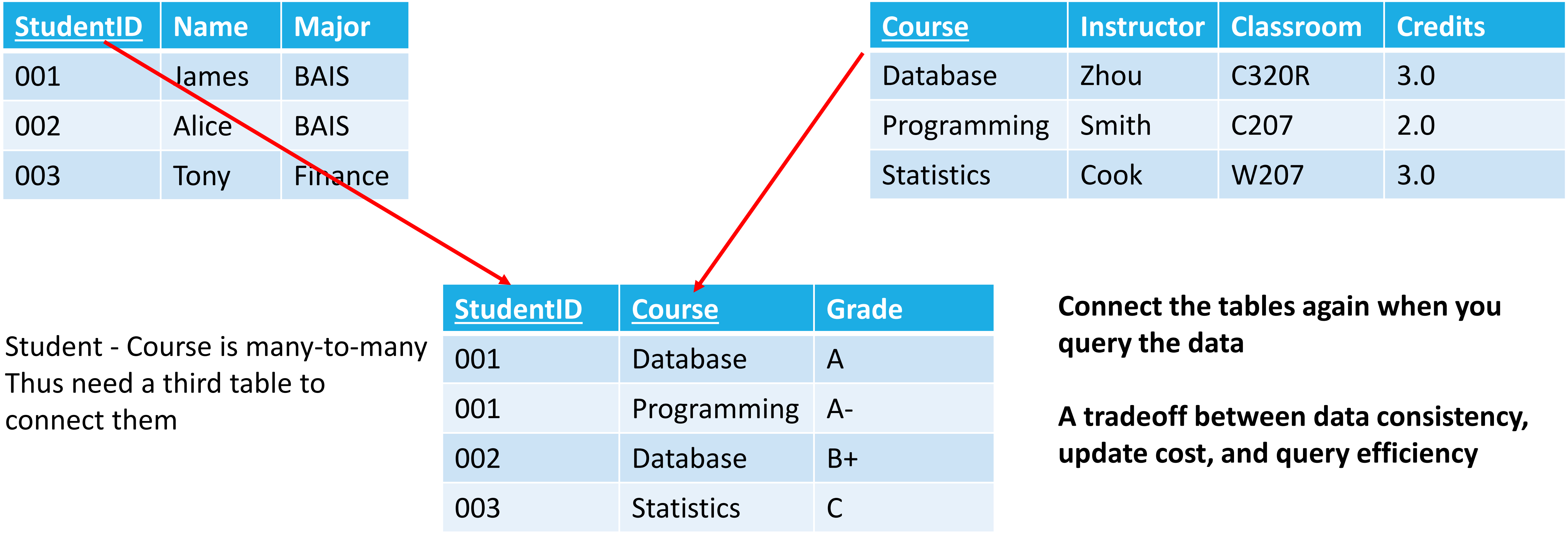
- Why don’t we connect all the tables and make a big, wide table with many columns?
- Anomalies: risks of data inconsistency, loss of information, or redundancy

When inserting a student who has not taken any course…
When deleting the only student who took Programming …
When updating the major of a student …

StudentID	Name	Major	Course	Instructor	Classroom	Credits	Grade
001	James	BAIS	Database	Zhou	C320R	3.0	A
001	James	BAIS	Programming	Smith	C207	2.0	A-
002	Alice	BAIS	Database	Zhou	C320R	3.0	B+
003	Tony	Finance	Statistics	Cook	W207	3.0	C

Why Do We Need Foreign Keys?

- ❑ Store information about each different "entity" in a separate table



StudentID	Name	Major
001	James	BAIS
002	Alice	BAIS
003	Tony	Finance

Course	Instructor	Classroom	Credits
Database	Zhou	C320R	3.0
Programming	Smith	C207	2.0
Statistics	Cook	W207	3.0

StudentID	Course	Grade
001	Database	A
001	Programming	A-
002	Database	B+
003	Statistics	C

Student - Course is many-to-many
Thus need a third table to connect them

Connect the tables again when you query the data

A tradeoff between data consistency, update cost, and query efficiency

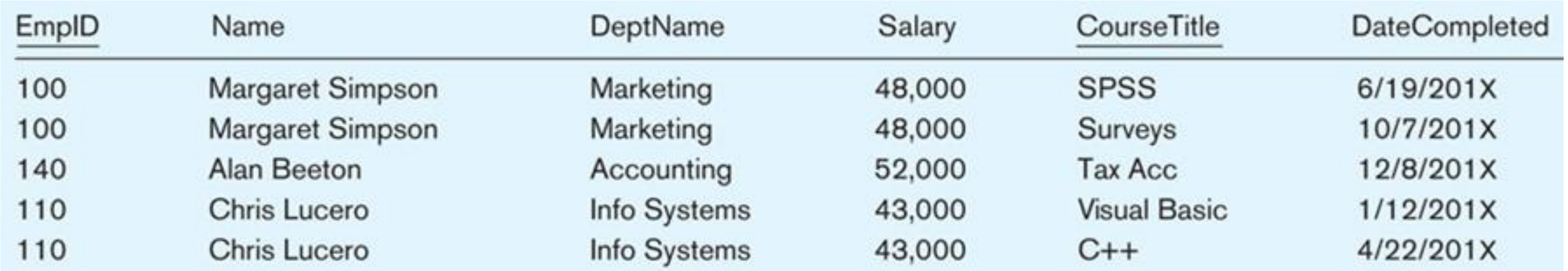
Introduction to Normalization

❑ Well-structured Relation

- Contain minimal redundancy
- Allow users to insert/modify/delete the rows in a table without errors or inconsistencies

❑ Anomaly

- An error or inconsistency that may result when updating a table that contains redundant data
- Insertion/deletion/modification anomalies



EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/201X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/201X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/201X
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/201X
110	Chris Lucero	Info Systems	43,000	C++	4/22/201X

ill-structured

❑ Normalization

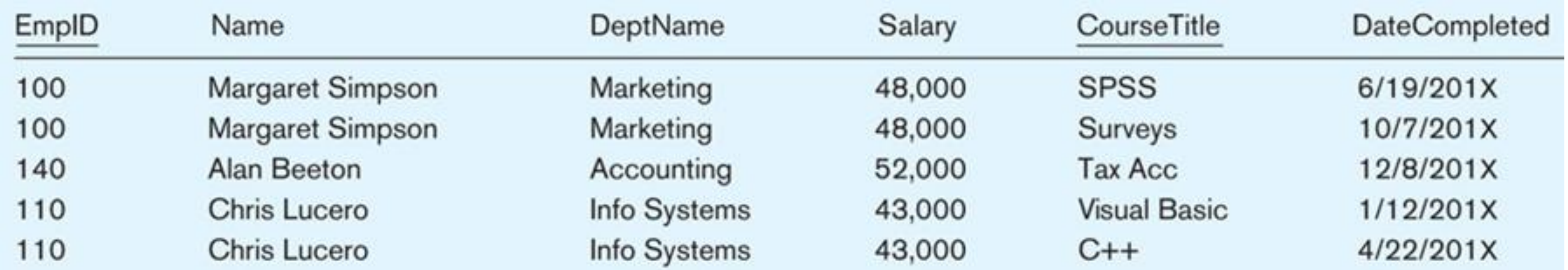
- The process of successively reducing relations with anomalies to produce smaller, well-structured relations

Anomalies

Normalization: a formal process for grouping attributes into different relations so that all anomalies are removed

- **Errors or inconsistencies due to data redundancy**

- **Three Types**
  - **Insertion** Anomaly (e.g., add a new employee who does not complete any courses – null value for PK)
  - **Deletion** Anomaly (e.g., delete Beeton – lost info about the Tax Acc course)
  - **Modification** Anomaly (e.g., increase the salary of Simpson – multiple rows)



EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/201X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/201X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/201X
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/201X
110	Chris Lucero	Info Systems	43,000	C++	4/22/201X

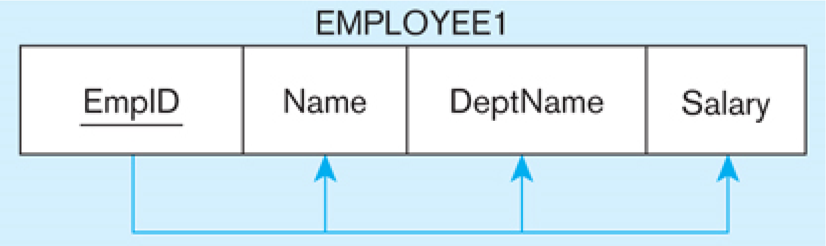
Functional Dependency and Candidate Key

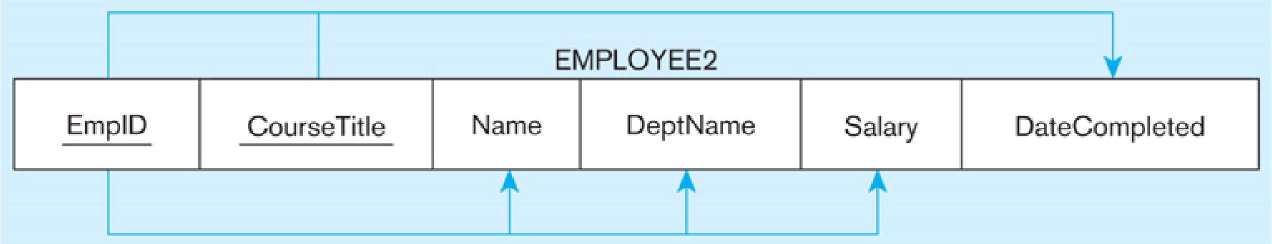
Functional Dependency

- A constraint between two (or two set of) attributes
- A→ B: the value of A determines uniquely the value of B (A is called a determinant).
- B is a subset of A: A→B is a **trivial functional dependency**.

EmpID → Name, DeptName, Salary

EmpID → Name, DeptName, Salary
EmpID, CourseTitle → DateCompleted

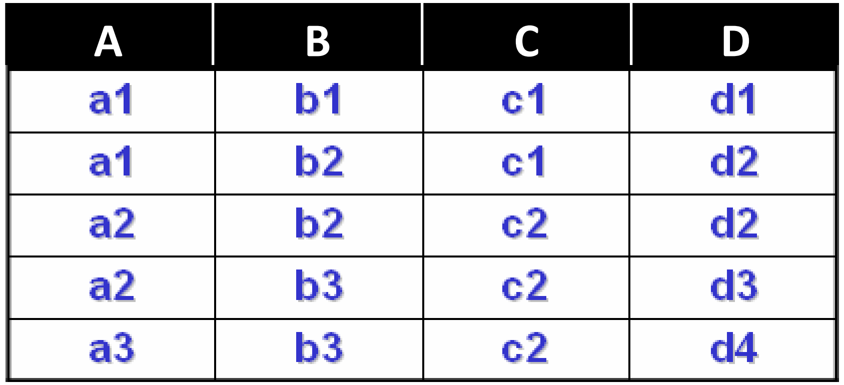




Candidate Key

- An attribute (or combination of attributes) that uniquely identifies a row
  - Unique identification: each nonkey attribute is functionally dependent on the key
  - Non-redundancy: No attribute in the key can be removed (e.g., EmpID+CourseTitle)

Example



A	B	C	D
a1	b1	c1	d1
a1	b2	c1	d2
a2	b2	c2	d2
a2	b3	c2	d3
a3	b3	c2	d4

What functional dependencies might exist according to this relation instance?

A→C, D→B

What if we have no duplicate values for A?

Then A determines all the columns. A→ U
(U represents all the columns in the relational schema)

Normalization

Higher-level NF is built upon lower-level NF!

Normal Form

- A state of a relation that requires certain rules regarding functional dependencies are satisfied

Steps in Normalization

- First normal form (1NF)
  - A primary key has bee defined
  - Each attribute is single-valued (no *multivalued* attributes)
- Second normal form (2NF)
  - Satisfy 1st normal form
  - Nonkey attributes are identified by the whole PK (no *partial* dependencies)
- Third normal form (3NF)
  - Satisfy 2nd normal form
  - Nonkey attributes are identified by only the PK (no *transitive* dependencies)

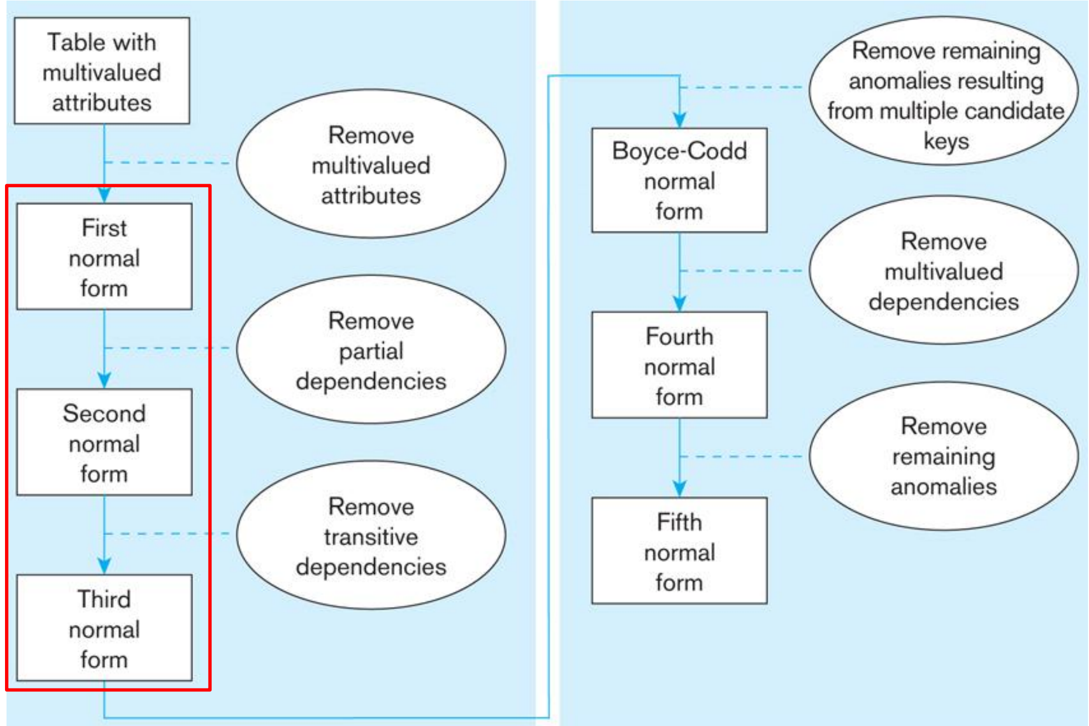
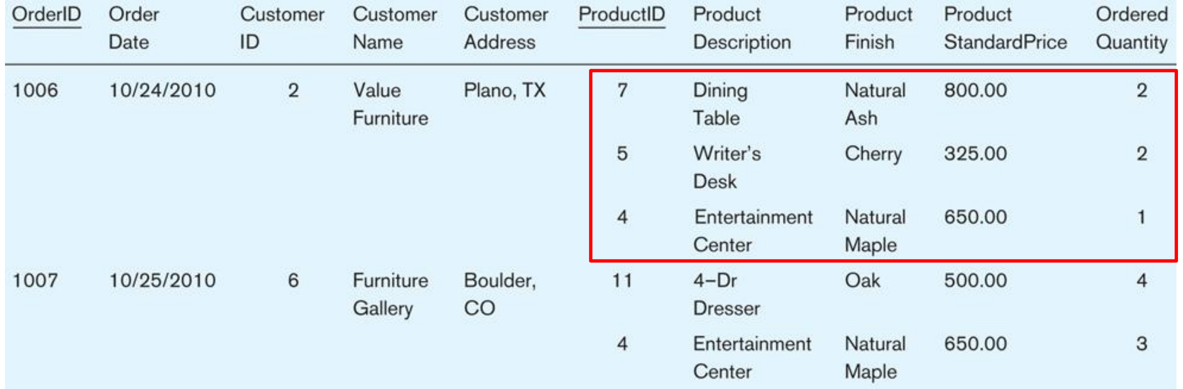


Table with multivalued attributes
Remove multivalued attributes
First normal form
Remove partial dependencies
Second normal form
Remove transitive dependencies
Third normal form
Remove remaining anomalies resulting from multiple candidate keys
Boyce-Codd normal form
Remove multivalued dependencies
Fourth normal form
Remove remaining anomalies
Fifth normal form

First Normal Form (1NF)

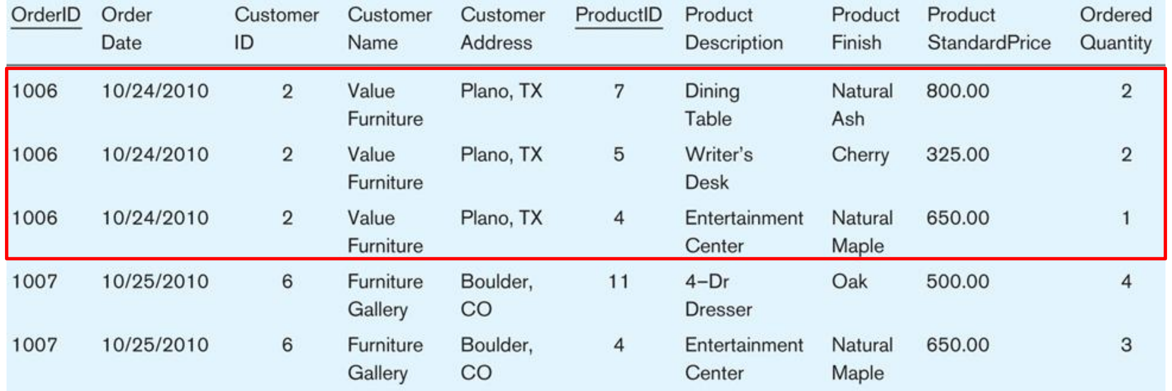
Two constraints

- There are no repeating rows (or multivalued attributes)
- Has a primary key (uniquely identifying each row; each nonkey attribute is functionally dependent on the PK)



OrderID	Order Date	Customer ID	Customer Name	Customer Address	ProductID	Product Description	Product Finish	Product StandardPrice	Ordered Quantity
1006	10/24/2010	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2010	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
					4	Entertainment Center	Natural Maple	650.00	3

OrderID → OrderDate, CustID, CustName, CustAddress
CustID → CustName, CustAddress
ProductID → ProdDesciption, ProdFinish, ProdPrice
OrderID, ProductID → OrderedQuantity



OrderID	Order Date	Customer ID	Customer Name	Customer Address	ProductID	Product Description	Product Finish	Product StandardPrice	Ordered Quantity
1006	10/24/2010	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2010	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2010	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2010	6	Furniture Gallery	Boulder, CO	11	4–Dr Dresser	Oak	500.00	4
1007	10/25/2010	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

Why 1NF?

1. Relation property: each entry is atomic
2. Entity integrity: every relation has a primary key

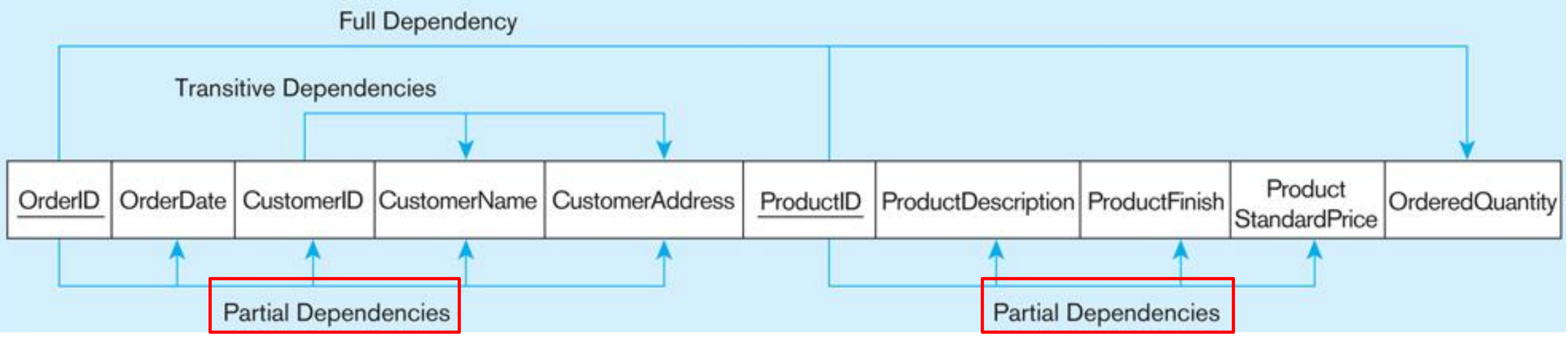
Second Normal Form (2NF)

Why 2NF?

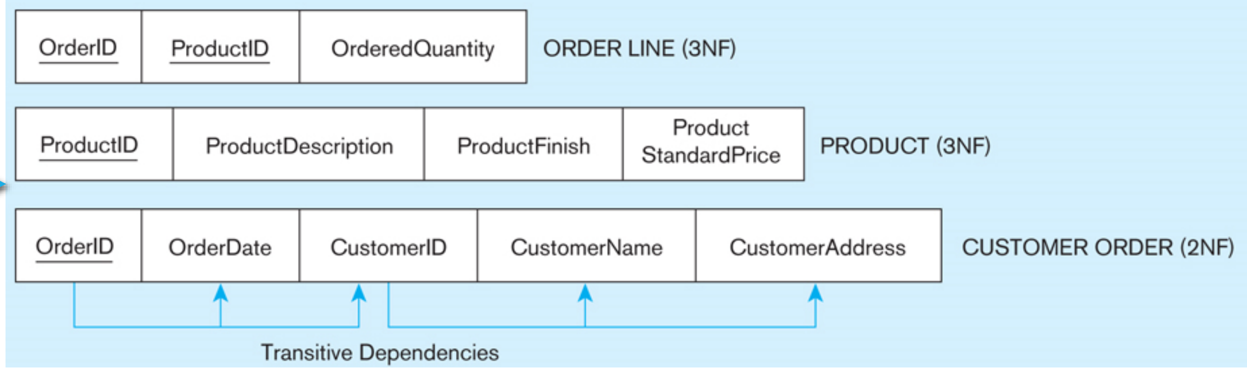
Reduce data redundancy

- **Two constraints**
  - In 1st normal form
  - No partial dependencies *(i.e., Every nonkey attribute is fully functionally dependent on the PK)*

Partial functional dependency: One or more nonkey attributes are functionally dependent on part (but not all) of the PK



1. Create a new relation for each PK attribute that is a determinant in a partial dependency. This attribute is the PK in the new relation;
2. Move all nonkey attributes that are only dependent on this PK attribute to the new relation.



Third Normal Form (3NF)

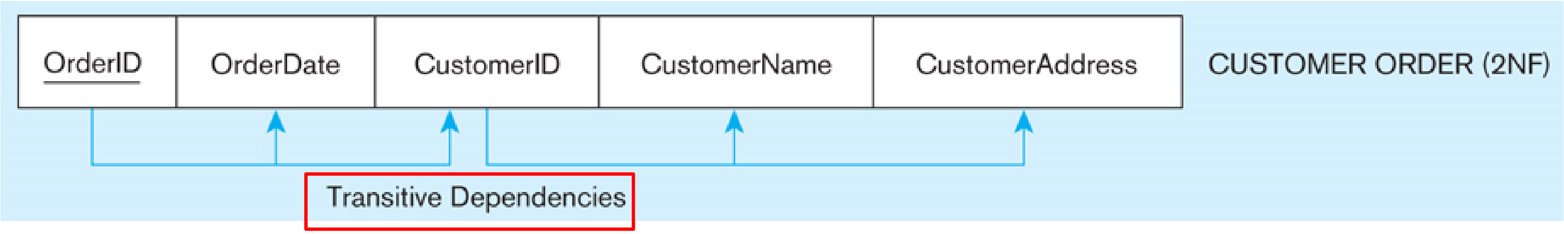
Why 3NF?

Further reduce data redundancy

❑ Two constraints

- ○ In 2nd normal form
- ○ No transitive dependencies

Transitive dependency: One or more nonkey attributes are functionally dependent on the PK via another nonkey attribute (or attribute combinations)

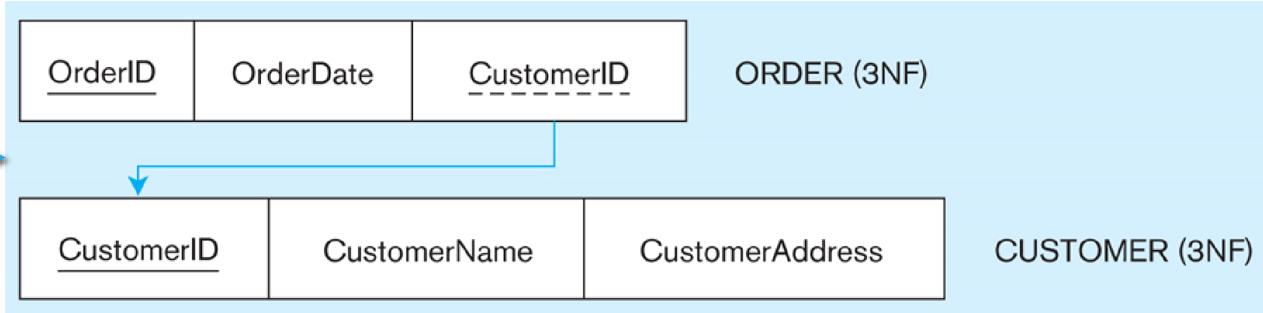


OrderID | OrderDate | CustomerID | CustomerName | CustomerAddress

CUSTOMER ORDER (2NF)

Transitive Dependencies

1. Create a new relation for each nonkey attribute that is a determinant in a transitive dependency. This attribute (D) is the PK in the new relation;
2. Move all nonkey attributes that are directly dependent on this PK attribute to the new relation;
3. Leave this attribute (D) in the old relation as a FK.



OrderID | OrderDate | CustomerID

ORDER (3NF)

CustomerID | CustomerName | CustomerAddress

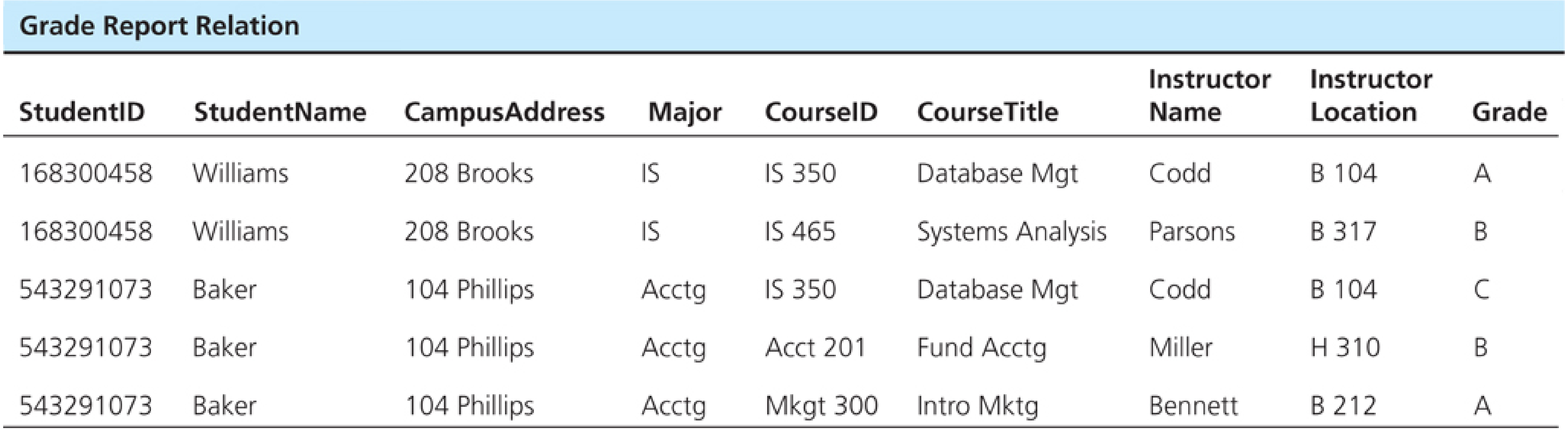
CUSTOMER (3NF)

Transitive Functional Dependencies

- **X→Y, Y→Z:**
  - X is a candidate key, Y is not
  - Y is not subset of X, Z is not subset of Y or X
  - Y→X Not true
- Then X→Z is a transitive functional dependency (via Y)

Exercise

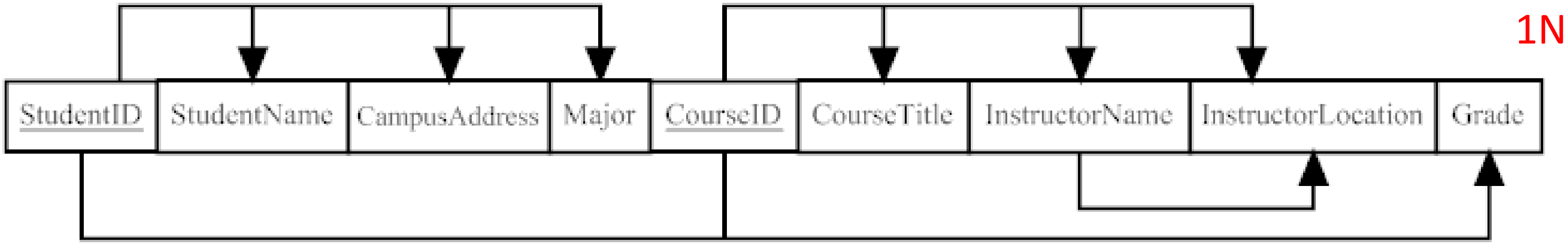
1. Draw a relational schema of this relation and diagram the functional dependencies;
2. In what normal form is this relation?
3. If it is not in 3NF, decompose it into a set of 3NF relations.



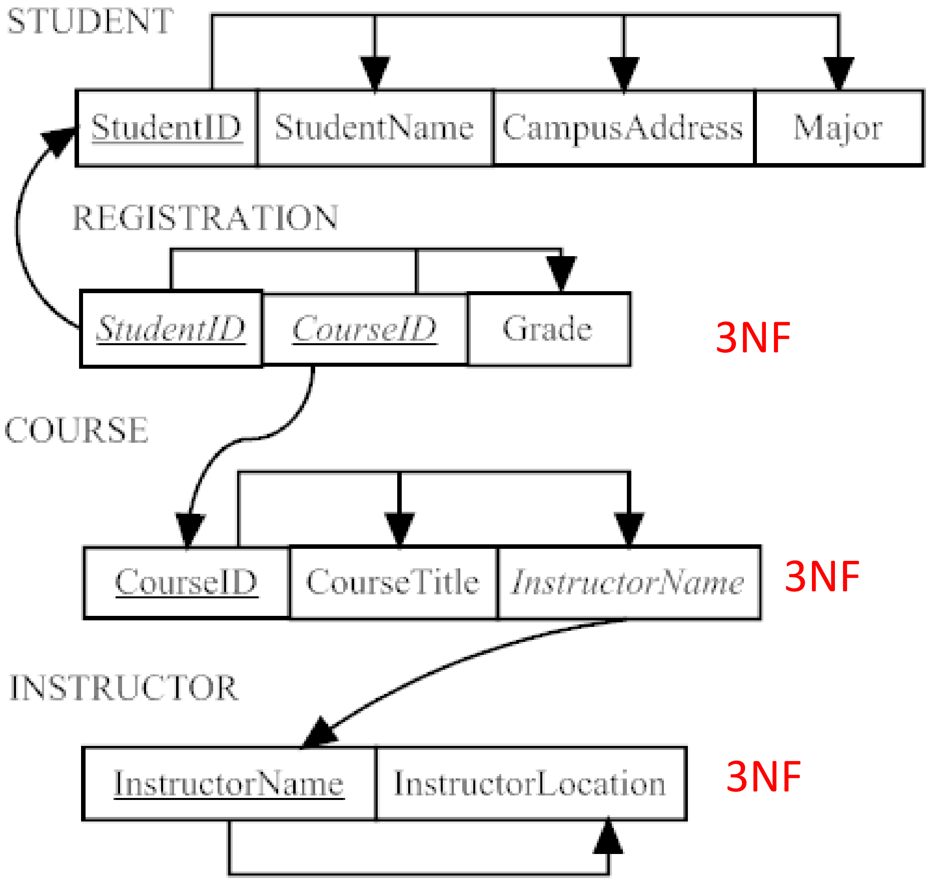
Grade Report Relation

StudentID	StudentName	CampusAddress	Major	CourseID	CourseTitle	Instructor Name	Instructor Location	Grade
168300458	Williams	208 Brooks	IS	IS 350	Database Mgt	Codd	B 104	A
168300458	Williams	208 Brooks	IS	IS 465	Systems Analysis	Parsons	B 317	B
543291073	Baker	104 Phillips	Acctg	IS 350	Database Mgt	Codd	B 104	C
543291073	Baker	104 Phillips	Acctg	Acct 201	Fund Acctg	Miller	H 310	B
543291073	Baker	104 Phillips	Acctg	Mkgt 300	Intro Mktg	Bennett	B 212	A

Solution



1NF



3NF

3NF

3NF

3NF

Other Normal Forms

Boyce-Codd Normal Form (Stronger than 3NF)

- The Relation is in 3NF
- All primary attributes (those in any candidate key) only depend on the whole PK (or candidate keys), rather than other attributes (excluding trivial dependencies like A→A)

- Example:

Postal (<u>City</u>, <u>Street</u>, ZipCode)

(City, Street) → ZipCode

ZipCode → City (Primary attribute “City” is functionally dependent on non-key attribute ZipCode!)

This relation is in 3NF but not BCNF. Not always possible in normalization without losing information.

Summary

❑ **Relational Data Modeling** --

- What is a relation, a relational schema (primary/foreign/composite/candidate key)
- Know how to express relational schema using text statements and graphical representation
- Constraints (domain, entity integrity, referential integrity)
- Transform an EER diagram to a set of relations
- Normalization analysis (up to 3NF)