Data Redundancy & Normal Form

Adapted from Chapter 14 (Connolly & Begg)

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 be 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

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)

$S \rightarrow \{S,N,L,R,W,H\}$

$R \rightarrow \{W\}$ - each value of R is associated with exactly 1 value of W

Determinant on left hand side of $\rightarrow$

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

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

Decompose the original relation into 2 relations.

R	W
8	10
5	7

Wage

Normal form to addresses dependencies

- Free the collection of relations from undesirable insertion, modification and deletion **dependencies**
  - If schema has duplicated data in multiple rows
    - Forced to update/delete all copies of a piece of data
    - How do you know you got all copies of it?
- Reduce the need for restructuring the collection of relations
  - Build an extensible design now as opposed to later
- Make the relational model more informative to users
  - Cleaner model should be easier to understand
- Make the collection of relations neutral to the query statistics
  - Designed for general purpose querying

Normalization

We use the normalization process as a validation technique for the defined relations.

It has been used as a method for conceptual database design.

Helps us identify redundancy that leads to functional dependency.

The normalization process is a series of test that help identify the optimal grouping of attributes to relations.

Reduce data redundancy

Unnormalized form

- No primary key or NULL values in the primary key fields
- A table that contains an attribute with one or more repeating groups (set)
  - Attributes need not be atomic

Unnormalized form

UNNORMALIZED FORM (UNF) – DUPLICATES ENTITIES					
Mother Id	Mother Name	Child1	Child2	Child3	Child4
1	Elsa	Mary	Alice	NULL	NULL
2	Golda	George	Fred	NULL	NULL
3	Viola	Ava	NULL	NULL	NULL
4	Iris	Kayla	NULL	NULL	NULL
5	Daisy	Harry	NULL	NULL	NULL

Table to track mother to child

First normal form

- Tuples in a relation must contain the same number of fields
- The domain of each attribute must be the same
- The value of each attribute contains only a single value
  - No attributes are sets

Relational Model

1st normal form

1st Normal Form

Mother Id	Mother Name
1	Elsa
2	Golda
3	Viola
4	Iris
5	Daisy

Child Id	Name	Mother
11	Mary	1
12	Alice	1
13	George	2
14	Fred	2
15	Ava	3
16	Kayla	4
17	Harry	5

Decompose table, remove repeating attributes

Second normal form

- Requirement for tables that have composite key
- Table is in first normal form
- Every non-primary key attribute is fully functionally dependent on the (entire) primary key
- A table in first normal form and having a primary key with only one field is also in 2nd normal form
- To get a table in 1st normal form into 2nd normal form, remove partial key dependencies by table decomposition

Example 2NF with a composite key

1st Normal Form but NOT 2nd NORMAL FORM

Mother Id	First Name	Last Name	Hospital	Hospital Address
1	Elsa	General	BIDMC	Boston
2	Golda	Major	MGH	Boston
3	Viola	Funt	TMC	Cambridge
4	Iris	Batter	Mayo	Allston
5	Daisy	Mae		

2nd NORMAL FORM

Hospital ID	Hospital	Hospital Address
1	BIDMC	Boston
2	MGH	Boston
3	TMC	Cambridge
4	Mayo	Allston

2nd NORMAL FORM

Mother Id	First Name	Last Name	Hospital Id
1	Elsa	General	1
2	Golda	Major	2
3	Viola	Funt	3
4	Iris	Batter	1
5	Daisy	Mae	4

Third normal form

- Table is in first and second normal form
- No dependencies between 2 non-key attributes
- No non-primary-key attribute is transitively dependent on the primary key
- Solution: decompose the table so that the offending attribute is in a separate table
- Attribute is fully functionally dependent on the primary key

Example: to 3rd normal form

2nd NORMAL FORM

Mother Id	First Name	Last Name	Hospital Id	Room Number
1	Elsa	General	1	36
2	Golda	Major		
3	Viola	Funt		
4	Iris	Batter		
5	Daisy	Mae		

3rd Normal Form

Register Id	Hospital Id	Room Id
1	1	36
2	2	48
3	3	36
4	1	41
5	4	32

3rd NORMAL FORM

Mother Id	First Name	Last Name	Register Id
1	Elsa	General	1
2	Golda	Major	2
	Viola	Funt	3
	Iris	Batter	4
	Daisy	Mae	5

Bill Kent's quote:

Every non-key attribute must provide a fact :

about the key,

the whole key

and nothing but the key

Example to model

A university consists of a number of departments. Each department offers several majors. A number of courses make up each major. Students declare a particular major and take courses towards the completion of that major. Each course is taught by a lecturer from the appropriate department, and each lecturer tutors a group of students

Example: entities

- A **university** consists of a number of **departments**. Each department offers several **majors**. A number of **courses** make up each **major.** **Students** declare a particular major and take courses towards the completion of that major. Each course is taught by a **lecturer** from the appropriate department, and each lecturer tutors a group of students

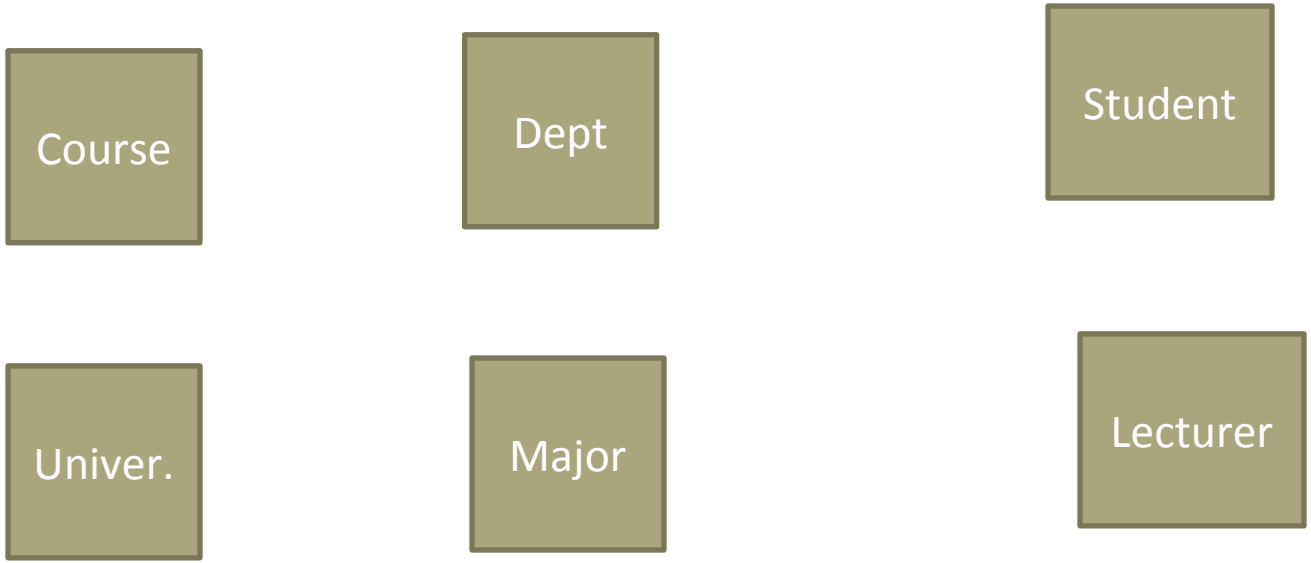
Example: relationships

- A **university** consists of a number of departments. Each department offers several **majors**. A number of **courses** make up each major. **Students** declare a particular major and take courses towards the completion of that major. Each course is taught by a **lecturer** from the appropriate department, and each lecturer tutors a group of students

Entities:

How do we add:

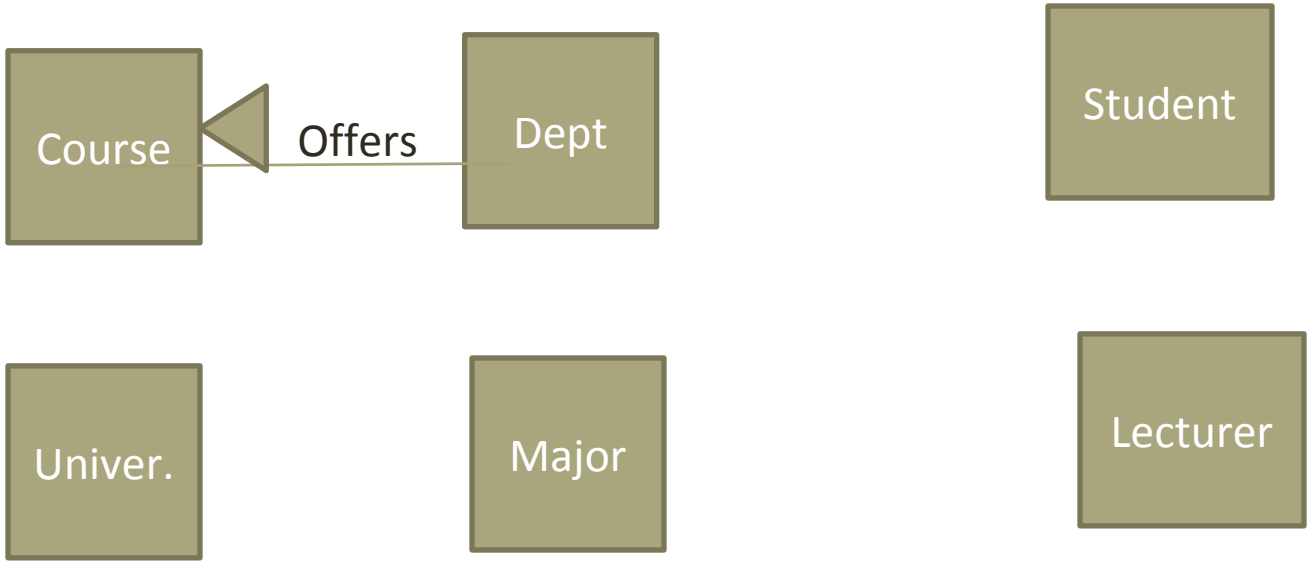
Department offers courses



Entities:

How do we add:

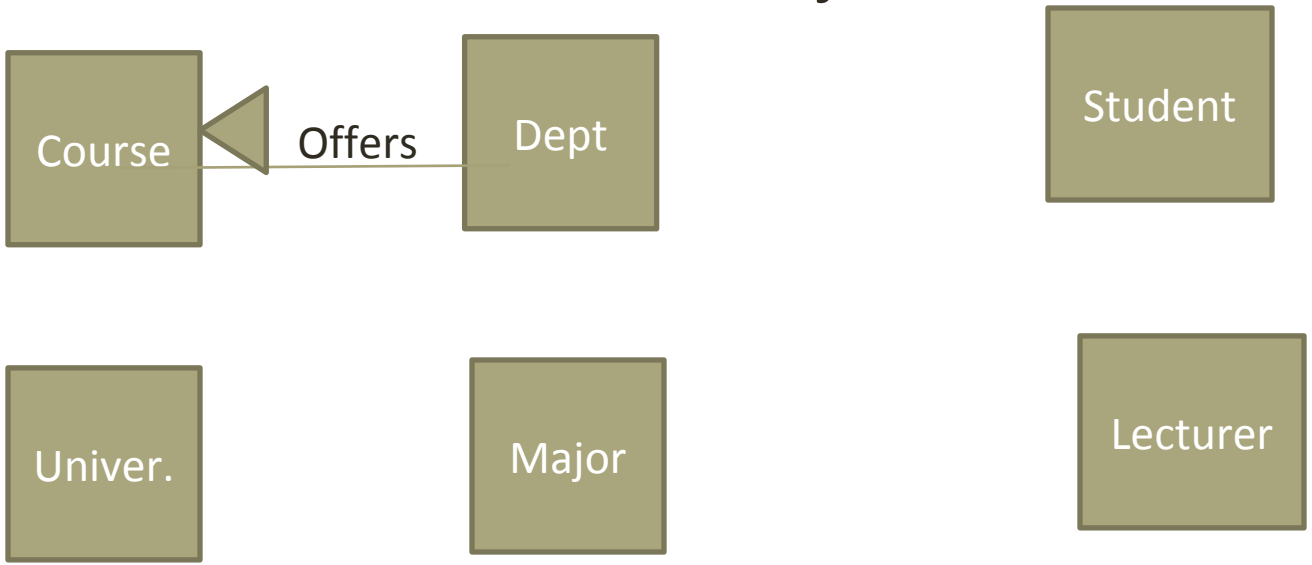
Department offers courses



Relationships:

How do we add:

Department offers several majors



Model to SQL tables?

- Identify the tables for the entities
- Identify the tables for the relationships
- Identify the table name, field names and data types
- Identify the primary keys
- Identify the foreign keys
  - Determine behavior for DELETE/UPDATE operations
- Represent other column and table constraints
  - NULL allowed for field?
  - Default value for a field?