Exercise 2

Functional dependencies
Multivalued dependencies
Normal forms (BCNF & 4NF)

Functional dependencies

In a relation, $R=(A_1,\ldots,A_n,B,\ldots)$,

- $A_1\ldots A_n \rightarrow B$ means that

  If two tuples have the same value for $A_1\ldots A_n$, then they also have the same value for B.

Think of it as:

R – a table

$A_1,\ldots,A_n,B,\ldots$ - column labels

Tuples - rows

NB!

- The ”B” on the right hand side can NOT be a set of values.
- Hence, Student→PassedCourses is NOT a FD.
- Also, ”B” must be a column in the table.
- Student→EligibleForGraduation is NOT a FD.

Keys

- IF $A_1 \ldots A_n$ determines all other attributes/columns in the relation/table *R*
- AND this does not hold for any subset of $A_1 \ldots A_n$,
- THEN $A_1 \ldots A_n$ is a **key** of *R*
- Any set of attributes that contains a key (including the set that contains nothing but the key) is a **superkey** of *R*
- You always have to pick a key as your **primary key**

Problems that can arise

Movie	Length	Actor
Star Wars	124	Carrie Fisher
Star Wars	124	Mark Hamill
Star Wars	124	Harrison Ford
Mighty Ducks	104	Emilio Estevez
Wayne's World	95	Dana Carvey
Wayne's World	95	Mike Myers

- Redundancy
- Update anomalies
- Deletion anomalies

Solution - BCNF

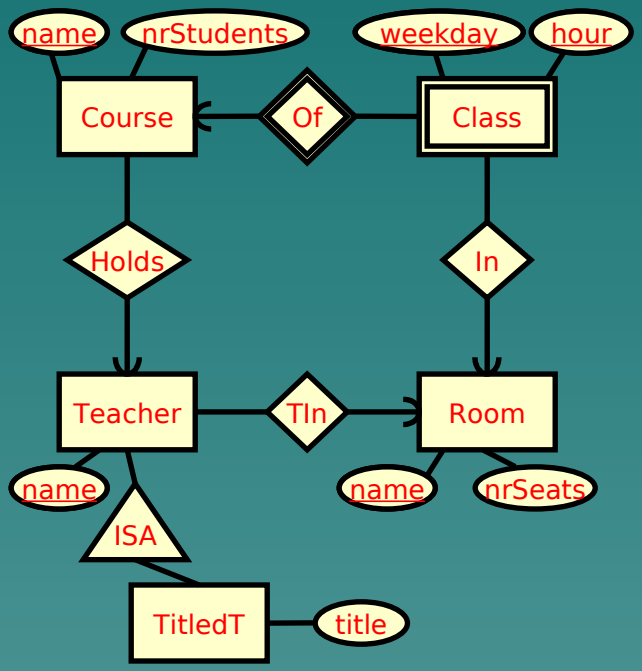
- IF whenever $A_1 \ldots A_n \rightarrow B$ holds on *R*, $A_1 \ldots A_n$ must be a *superkey* of *R*
- THEN *R* is in BCNF

How to get there

- Find $A_1 \ldots A_n \rightarrow B$ that violates BCNF
- Make two new tables/relations:
  - $\{A_1 \ldots A_n\}^+$, that is everything that is known as soon as you know $A_1 \ldots A_n$
  - *R* **minus** $\{A_1 \ldots A_n\}^+$ **plus** $A_1 \ldots A_n$
  - A reference
- Repeat until no more violations

Let's try

- Course names
- Teacher names
- Teacher titles (optional, e.g. Professor)
- Class room names
- Class room capacity
- Number of students taking a course
- Day and time of classes



Classes(courseName, teacherName, teacherTitle, roomName, #students, weekday, time, #seats)

FDs

- **courseName → teacherName** (a course has only one responsible teacher)
- **courseName → #students** (a course has only one number of students)
- **teacherName → teacherTitle** (a teacher has at most one title, could be argued differently)
- **teacherName → roomName** (a teacher holds all classes in the same room, by the domain description)
- **roomName → #seats** (a room has only one number of seats)
- **courseName, weekday, hour → roomName** (a course has only one class at the same time)
- **roomName, weekday, hour → courseName** (only one course at a time can be in a room)

- *Note that the second to last is actually not needed, since we have courseName → teacherName, teacherName → roomName.*

Keys

- To find possible keys for the full relation, we need to compute the closures of all attributes.
- A trick is that we don't need to look at attributes that never appear on the left-hand side of a FD, since these can never give anything new.
- Another trick is that we only need to look at attribute sets that are supersets of some left-hand side for some FD, since if the set was not a superset of some left-hand side then there would be no FDs to follow!

Closures

- **{courseName}+** = {courseName, teacherName, #students, teacherTitle, roomName, #seats}
- **{teacherName}+** = {teacherName, teacherTitle, roomName, #seats}
- **{roomName}+** = {roomName, #seats}
- **{courseName, teacherName}+** = {courseName}+
- **{courseName, roomName}+** = {courseName}+
- **{teacherName, roomName}+** = {teacherName}+
- **{courseName, weekday, hour}+** = all attributes (only weekday and hour missing from {courseName}+)
- **{roomName, weekday, hour}+** = all attributes ({roomName, weekday, hour} gives courseName, from there the rest)
- **{teacherName, weekday, hour}+** = all attributes (teacherName gives roomName, from there the rest)

Closures cont.

- The full set of FDs for this relation, i.e. the closure of F (F+), is thus:
  - courseName → teacherName, #students, teacherTitle, roomName, #seats
  - teacherName → teacherTitle, roomName, #seats
  - roomName → #seats
  - roomName, weekday, hour → courseName, teacherName, #students, teacherTitle
  - teacherName, weekday, hour → courseName, #students

Now we decompose

- Violation 1: courseName → teacherName
  - R1(courseName, teacherName, #students, teacherTitle, roomName, #seats) (i.e. the attributes in {courseName}+).
  - R2(courseName, weekday, hour) (i.e. the remaining attributes, plus courseName).
  - A reference from R2.courseName to R1.courseName
- Violation 2 (in R1): roomName → #seats
  - R11(roomName, #seats) (i.e. {roomName}+)
  - R12(courseName, teacherName, #students, teacherTitle, roomName) (i.e. the rest, plus roomName)
  - A reference from R12.roomName to R11.roomName

More decomposition

- Violation 3 (in R12): teacherName → teacherTitle
  - R121(teacherName, teacherTitle, roomName) (i.e. {teacherName}+ restricted to R11 (meaning #seats no longer exists in R11)).
  - R122(courseName, teacherName, #students) (i.e. the rest, plus teacherName)
  - A reference from R122.teacherName to R121.teacherName

- We're done!

- Rooms(roomName, #seats) (R11)
- Teachers(teacherName, teacherTitle, roomName) (R121)
  roomName → Rooms.roomName
- Courses(courseName, teacherName, #students)
  teacherName → Teachers.teacherName
- Classes(courseName, weekday, hour) (R2)
  courseName → Courses.courseName

Multivalued dependencies

- $A_1 \ldots A_n \twoheadrightarrow B_1 \ldots B_m$
- Let's call all the attributes that are NOT in the As or Bs $C_1 \ldots C_k$
- Then if two tuples have the same $A_1 \ldots A_n$, you can switch the Bs or Cs "blockwise" between the tuples, and the result will be an existing tuple

A	B	C
A	B	C

MVD cont.

- $A_1 \ldots A_n \twoheadrightarrow B_1 \ldots B_m$ would therefore also mean $A_1 \ldots A_n \twoheadrightarrow C_1 \ldots C_k$

- And all FDs are MVDs

Actor	Address	Movie
Carrie Fisher	Malibu	Star Wars
Carrie Fisher	Hollywood	Star Wars
Carrie Fisher	Malibu	Return of the Jedi
Carrie Fisher	Hollywood	Return of the Jedi

4NF

- Sharpen BCNF a bit
- IF whenever $A_1 \ldots A_n \twoheadrightarrow B$ is nontrivial on *R*, $A_1 \ldots A_n$ must be a *superkey* of *R*
- THEN *R* is in 4NF

- *Nontrivial* basically means that the As, Bs and Cs are **non-overlapping** and **non-empty** sets

How to do it?

- Very similar procedure to making BCNF
- …but I won’t steal the fun exercise from you ☺