Making Sure Data Stays Good

- We have talked about the basics of good data
  - Unique things should be defined as unique
  - Keys need to be unique
  - We should break things into separate tables
- Normal Forms allow us to talk about this more formally
  - Lets us show how robust our schema is
  - Not all schemas are or can be perfect

Functional Dependencies

- In relations some attributes determine other attributes
  - A SSN might determine the name of an employee
  - A student ID, class name, quarter and year might determine a grade
  - Or a student ID and a CRN might determine a grade
- If an attribute X determines some attribute Y, we say Y is functionally dependent on X
- This is noted as X -> Y

Keys and Superkeys, A Review

- A Superkey is a set of attributes that can be used to uniquely determine a row
- A key is a minimal Superkey
  - So if we remove any attribute from a key, it no longer uniquely determines rows

Employee	Project	Hours

  - This table tracks hours employees wok on projects
  - {Employee, Project, Hours} and {Employee, Project} are both Superkeys
  - {Employee, Project} is a key

Normal Forms

- Allow us to prove how robust a database is
- Goes up to 6th normal form
- Most databases go to 3rd or BCNF
  - BCNF is between 3rd and 4th
- Can not determine normal form by looking at populated table
  - Might be able to prove it violates normal form
- Need to know the rules of the database to determine normal form

1st Normal From (1NF)

- Every value must be a non-divisible atomic value
  - No lists
  - No subtables
- Is a string of characters divisible into single letters?
  - For 1NF no. If the field is first name, then it looses meaning when it becomes individual letters
- 1NF is trivially met by most RDBMS

How do We Fix 1NF Issues?

People	Certifications
Lee Adama	{Viper, Raptor, Command}
Galen Tyrol	{Mechanic}

- This is not in 1NF
  - Contains lists of certifications which are non-atomic
- How might we fix this?

Solution 1

- The best way to fix this, add a new table (or 2)

People	Certifications
Lee Adama	{Viper, Raptor, Command}
Galen Tyrol	{Mechanic}

People
Lee Adama
Galen Tyrol

People	Certs
Lee Adama	Viper
Lee Adama	Raptor
Lee Adama	Command
Galen Tyrol	Mechanic

Certifications
Viper
Raptor
Command
Mechanic

Solution 2

- Add more rows
- This introduces problems for higher normal forms
- Looks like previous solution, but has fewer constrains

People	Certifications
Lee Adama	Viper
Lee Adama	Raptor
Lee Adama	Command
Galen Tyrol	Mechanic

Solution 3

- Add more columns
- If you know the max columns, this is OK

People	Cert 1	Cert 2	Cert 3
Lee Adama	Viper	Raptor	Command
Galen Tyrol	Mechanic	NULL	NULL

1NF Review

- Needs to have atomic values
- Can fix by diving values into rows, columns or other tables
- Dividing into separate tables is typically the better way to go
- You probably won't run into this in an existing database
- You will run into having to pick between solutions when designing database

2nd Normal Form (2NF)

- Every nonprime attribute (not part of the primary key) must not be partially dependent on any key
- In other words, if a nonprime attribute depends on only part of something that is a minimal superkey, it is not in 2NF
- You may see this in tables relating 2 entities

Violation of 2NF

People	Certs	CertType
Lee Adama	Viper	Piloting
Lee Adama	Raptor	Piloting
Lee Adama	Command	Leadership
Galen Tyrol	Mechanic	Skill

- CertType is dependent on Cert
- {Person, Cert} is the primary key

Fixing 2NF Issues

People	Certs
Lee Adama	Viper
Lee Adama	Raptor
Lee Adama	Command
Galen Tyrol	Mechanic

Certs	CertType
Viper	Piloting
Raptor	Piloting
Command	Leadership
Mechanic	Skill

- Move attributes to the correct tables or make new tables to allow this to happen

2NF Review

- Attributes must be functionally dependent on entire keys
- Don't forget to look at keys other than the primary key
  - This will cause problems we will see soon
- Solve by breaking attributes that are dependent on only part of the key into separate tables

3rd Normal From (3NF)

- Transitive dependencies
  - If X -> Y and Y -> Z then Z is transitively dependent on X
- 3NF requires that no nonprime attribute be transitively dependent on any key
- This is an easy mistake to make and you will likely run into it in the real world
- No more quick fixes

3NF Violation

ID	People	Role	RolePriority
0	Lee Adama	CAG	3
1	Cara Thrace	Pilot	4
3	Galen Tyrol	Chief Mechanic	8

- Satisfies 2NF
- Violates 3NF
  - Role is dependent on the person and RolePriority is dependent on Role

Solving 3NF Issues

ID	People	Role
0	Lee Adama	CAG
1	Cara Thrace	Pilot
2	Galen Tyrol	Chief Mechanic

Role	RolePriority
CAG	3
Pilot	4
Chief Mechanic	8

- Need to make a new table
- The attributes 'in the middle' of the transitive dependency become the new key
- Joins reconstruct old data

3NF Review

- 3NF is violated when transitive dependencies exist
  - A->B B->C
- Fixed by separating transitive relationship into a new table

Boyce-Codd Normal Form (BCNF)

- Similar to 3NF
- 3NF requires that no nonprime attribute be transitively dependent on any key
- BCNF requires that no ~~nonprime~~ attribute be transitively dependent on any key

Violation of BCNF

Student	Course	Instructor

- An instructor only teaches one course but a course can be taught by many instructors
- A student only takes a course once
- {Student, Course} is the primary key
- {Student, Course} -> Instructor
- Instructor -> Course
- Course is transitively depending on {Student, Course}

Fixing This is Tricky

Student	Course	Instructor

a)

Student	Instructor

Student	Course

b)

Course	Instructor

Course	Student

c)

Instructor	Course

Instructor	Student

- C is the only correct way to break this down
- All three lose the {Student, Course} -> Instructor FD
- The other two will create additional, incorrect rows on a join

BCNF Review

- Slightly more strict than 3NF
- Requires that every attribute, even prime ones, not be transitively dependent on any key
- Fixing it may require completely removing a FD from the schema
- Need to be careful to ensure when it is reconstructed, no additional data will be added

Why Oh Why Do We Do This?

- To prevent data anomalies! (Except 1NF, that is just to make relationships work)

Student	Course	Instructor
Kyle	History	Mr. Garrison
Cartman	History	Mr. Garrison
Kyle	Psychology	Mr. Mackey

- What if we update Kyle's course with Mr. Garrison to Modern History?
- Now Mr. Garrison teaches History and Modern History but teachers only teach one course

How Does BCNF Fix This?

Instructor	Course
Mr. Garrison	History
Mr. Mackey	Psychology

Instructor	Student
Mr. Garrison	Kyle
Mr. Garrison	Cartman
Mr. Mackey	Kyle

- Now if we update the course name to History it is consistent for all students Mr. Garrison teaches

3NF Example

ID	People	Role	RolePriority
0	Lee Adama	CAG	3
1	Cara Thrace	Pilot	4
2	Sharon Valerii	Pilot	4
3	Galen Tyrol	Chief Mechanic	8

- It is easy to see we could change Sharon's role priority in one row and have it no longer match the others

2NF Example

People	Certs	CertType
Lee Adama	Viper	Piloting
Lee Adama	Raptor	Piloting
Lee Adama	Command	Leadership
Galen Tyrol	Mechanic	Skill
Cara Thrace	Viper	Piloting

- Could change CertType for Cara to FighterPiloting and it would not change for Lee's Viper certification