Database Design

Shifting Gears:

From ***Introductory Concepts*** to ***Design Steps***

Or,

"Now that we've learned a bit about relational databases, their history, and conceptual principles, let's look into how one goes about designing a database for an organization."

Design

Design Steps:

Collect all the information you can about the organization's data (i.e. "requirements")

- What do they call their entities and attributes?
- Data types and lengths
- How and when do they use each piece of data?
- Where does it come from? Where does it go to?
- Who uses it for what purpose?
- How many occurrences/instances of each entity do they deal with?

Design

How?

1. Conduct interviews
2. Review documentation
3. Review current systems and processes
   - User data entry screens
   - Paper forms
   - Paper reports
   - Computer reports
   - Process flows

Design

Design Steps:

Once data requirements are collected and documented, organize the data:

- Identify all Functional Dependencies
- Put the data into 3rd Normal Form
- Create a logical data model

Functional Dependency

Definition: (Ullman text, page 68)

"If the values of one or more attributes (A1, A2, A3, etc.) of a relation functionally determine the value of another attribute (B) of that relation, then we can say that (B) is functionally dependent on (A1, A2, A3, etc.)"

In other words,

If I know the value of an attribute (or set of attributes), I can determine the value of another attribute.

Functional Dependency

Examples: (from the text)

The “Movies1” relation

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne’s World	1992	95	comedy	Paramount	Dana Carvey
Wayne’s World	1992	95	comedy	Paramount	Mike Meyers

Functional Dependency

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

FD = title + year ➔ length + genre + studioName

If 2 tuples have the same values in title + year,
those 2 tuples will have the same values in length, genre and studioName

TRUE

Functional Dependency

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

FD = title + year ➔ starName

If 2 tuples have the same values in title + year,
those 2 tuples will have the same values in starName

FALSE

Functional Dependency

A formal definition of a Key

1. “A set of attributes (A1, A2, A3,etc.) is a KEY for a relation if that set of attributes functionally determines all the other attributes of the relation.

And

2. No subset of those attributes (A1, A2, A3,etc.) functionally determines all the other attributes of the relation.”

(FD doesn’t differentiate Primary Key versus Candidate Key)

Functional Dependency

Apply some FD tests to this relation.

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

title
title+year
title+starName
title+year+starName

- *The Manchurian Candidate* (1962 film), starring Frank Sinatra
- *The Manchurian Candidate* (2004 film), starring Denzel Washington
- *Blade Runner (1982 film) staring Harrison Ford*
- *Blade Runner (2017 film) staring Harrison Ford*

Functional Dependency

Combining + Splitting Rule

In a relation R(A, B, C, D)

if A+B ➔ C, and A+B ➔ D, then A+B ➔ C, D

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

title+year ➔ length
title+year ➔ genre
title+year ➔ studioName

title+year ➔ length, genre, studioName (combine the right side)

Functional Dependency

Combining + Splitting Rule

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

But these are false: (can't split the left side)
title ➔ length
year ➔ length

Functional Dependency

Let’s practice

A relation of people in the U.S.
name, SSN, address, city, state, ZIP, area code, 7-digit phone

FD is a relationship of one attribute or field in a record to another.

e.g. one field defines the other. SSN defines name.

IF we know SSN, we can find name so name is functionally dependend on SSN. SSN ➔ Name

Functional Dependency

More practice

Project Code	Project Title	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Pensions System	M Phillips	24500	S10001	A Smith	L004	IT	22.00
PC010	Pensions System	M Phillips	24500	S10030	L Jones	L023	Pensions	18.50
PC010	Pensions System	M Phillips	24500	S21010	P Lewis	L004	IT	21.00
PC045	Salaries System	H Martin	17400	S10010	B Jones	L004	IT	21.75
PC045	Salaries System	H Martin	17400	S10001	A Smith	L004	IT	18.00
PC045	Salaries System	H Martin	17400	S31002	T Gilbert	L028	Database	25.50
PC045	Salaries System	H Martin	17400	S13210	W Richards	L008	Salary	17.00
PC064	HR System	K Lewis	12250	S31002	T Gilbert	L028	Database	23.25
PC064	HR System	K Lewis	12250	S21010	P Lewis	L004	IT	17.50
PC064	HR System	K Lewis	12250	S10034	B James	L009	HR	16.50

Functional Dependency

Transitive Rule

In a relation R(A, B, C)

if A ➔ B, and B ➔ C, then A➔C

Example:

Project Code	Project Title	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Pensions System	M Phillips	24500	S10001	A Smith	L004	IT	22.00
PC010	Pensions System	M Phillips	24500	S10030	L Jones	L023	Pensions	18.50
PC010	Pensions System	M Phillips	24500	S21010	P Lewis	L004	IT	21.00
PC045	Salaries System	H Martin	17400	S10010	B Jones	L004	IT	21.75
PC045	Salaries System	H Martin	17400	S10001	A Smith	L004	IT	18.00
PC045	Salaries System	H Martin	17400	S31002	T Gilbert	L028	Database	25.50
PC045	Salaries System	H Martin	17400	S13210	W Richards	L008	Salary	17.00
PC064	HR System	K Lewis	12250	S31002	T Gilbert	L028	Database	23.25
PC064	HR System	K Lewis	12250	S21010	P Lewis	L004	IT	17.50
PC064	HR System	K Lewis	12250	S10034	B James	L009	HR	16.50

EmployeeNo ➔ EmployeeName

EmployeeName ➔ DepartmentNo, therefore

EmployeeNo ➔ DepartmentNo

Functional Dependency

Augmentation Rule

In a relation R(A, B, C)

if A ➔ C, then A+B ➔ C+B

Example:

Project Code	Project Title	Project Manager	Project Budget	Employee No.	Employee Name	Department No.	Department Name	Hourly Rate
PC010	Pensions System	M Phillips	24500	S10001	A Smith	L004	IT	22.00
PC010	Pensions System	M Phillips	24500	S10030	L Jones	L023	Pensions	18.50
PC010	Pensions System	M Phillips	24500	S21010	P Lewis	L004	IT	21.00
PC045	Salaries System	H Martin	17400	S10010	B Jones	L004	IT	21.75
PC045	Salaries System	H Martin	17400	S10001	A Smith	L004	IT	18.00
PC045	Salaries System	H Martin	17400	S31002	T Gilbert	L028	Database	25.50
PC045	Salaries System	H Martin	17400	S13210	W Richards	L008	Salary	17.00
PC064	HR System	K Lewis	12250	S31002	T Gilbert	L028	Database	23.25
PC064	HR System	K Lewis	12250	S21010	P Lewis	L004	IT	17.50
PC064	HR System	K Lewis	12250	S10034	B James	L009	HR	16.50

EmployeeNo + DepartmentNo ➔ EmployeeName + DepartmentNo

Functional Dependency

Some FD rules

If I know a true FD for a relation, I can deduce other true FD's for that relation.

project (project_code, project_title, project_manager, project_budget)

project employee (project_code, employee_no, Hourly_rate)

employee (Employee_no, Employee_name, Department_No)

Department (Department_no, Department_name)

Why do we care about all this?

- If I am designing a database for an organization, I must consider all their data. I must organize that data into relations.
- I need to understand Functional Dependencies so I can determine keys for every relation.
- I need keys so that I can normalize the data.
- I must normalize the data to design the database properly.