Design Theory
Anomaly and Functional Dependencies

Hui Chen [a]

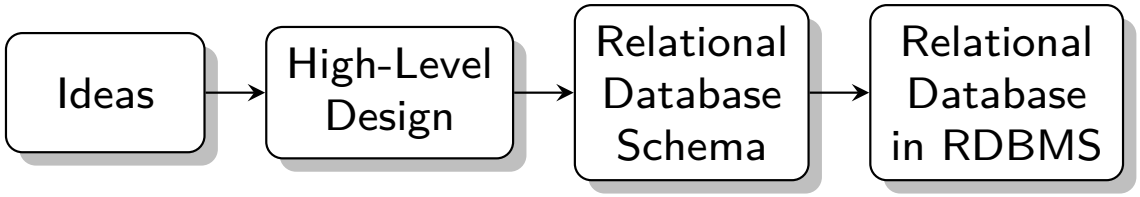
[a]CUNY Brooklyn College, Brooklyn, NY, USA

March 27, 2025

Outline

Outline

Overview


Ideas
High-Level Design
Relational Database Schema
Relational Database in RDBMS

A Design Challenge

There are a variety of ways that we can design relational schema – there is a space for improvement.

- Problem. we are trying to combine too much into one relation $\rightarrow$ maintenance problems called *anomaly*.
- Problem. we are trying to create to many relations $\rightarrow$ *difficult to answer queries or retrieve the data*

How do we identify such design problem and make improvements? – a design trade-off must be made.

- A well developed theory – dependencies and normalization
- *Normalization* – the process of converting a relation into a normal form.
  - The process usually consists of decomposing a table into two or more tables with fewer attributes
  - When normalizing relations, we are generally sacrificing retrieval speed to prevent data maintenance problems – a trade-off

Outline

Anomaly

Anomalies – Undesirable side effects that can occur if relations are not in the proper form (what is it?).

- Insertion anomalies – It occurs when we can not insert a tuple Usually when the primary key of the tuple is unknown
- Deletion Anomalies – valid fact is lost when a tuple is deleted. It occurs when 3 circumstances exist
  - when we delete a tuple from a table
  - the tuple that we delete contains an important piece of information
  - the tuple that we delete is the last one that contains that piece of information
- Update Anomalies – one occurrence of a fact is changed, but not all occurrences.
- Redundancy – update anomalies occur when we have unnecessary redundancy in the data

Insertion Anomaly: Example

Given the relational instances as follows,

Departments

DeptNo	DeptName
10	Production
20	Supplies
30	Marketing

Employees

EmpNo	Name	DeptNo
101	Sasha	10
102	LaTasha	20
103	John	30

what if we wish to do,

```
INSERT INTO
Employees(EmpNo, Name, DeptNo)
VALUES(104, 'Jane', 40);
```

Deletion Anomaly: Example

Consider that we design the `Department` database with a single relation that capture all the information,

Departments

EmpNo	Name	DeptNo	DeptName
101	Sasha	10	Production
102	LaTasha	20	Supplies
103	John	30	Marketing

What if we delete `Sasha`, `LaTasha`, and `John` from the relation?

Update Anomaly: Example

Consider that we design the `Department` database with a single relation that capture all the information,

Departments

EmpNo	EmpName	DeptNo	DeptName
101	Sasha	10	Production
102	LaTasha	20	Supplies
103	John	30	Marketing
104	Jane	10	Production

What if we change department name `Production` to `Manufacturing` by updating `Sasha`'s department to `Manufacturing`?

Outline

Functional Dependencies

$X \to Y$ is an assertion about a relation R that whenever two tuples of R agree on all the attributes of X, then they must also agree on all attributes in set Y.

- Say "$X \to Y$ holds in R.", which reads,
- "X functionally determines Y"
- For convenience, we use the following convention:
  - no set forms for sets of attributes, just $A\ B\ C$, rather than $\{A, B, C\}$.

Functional Dependencies (FD): Example 1

Consider that we design the `Department` database with a single relation that capture all the information,

Departments

EmpNo	EmpName	DeptNo	DeptName
101	Sasha	10	Production
102	LaTasha	20	Supplies
103	John	30	Marketing
104	Jane	10	Production

- FD holds: `DeptNo` $\rightarrow$ `DeptName`
- FD holds: `DeptName` $\rightarrow$ `DeptNo`
- FD does not holds: `DeptNo` $\rightarrow$ `EmpNo EmpName`
- FD does not holds: `DeptName` $\rightarrow$ `EmpNo EmpName`

Functional Dependencies (FD): Example 2

Consider that we design the `Movies` database with a single relation that capture all the information as follows,

Movies

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
I, Robot	2004	115	SciFi	Fox	Will Smith
I, Robot	2004	115	SciFi	Fox	Bridget Moynahan

- FD holds: title year $\rightarrow$ length genre studioName
- FD does not hold: title year $\rightarrow$ starName

Superkeys and Keys of Relations

We can define keys of relations using functional dependencies.

- K a *superkey* for relation R if K functionally determines all of R, i.e., $K \rightarrow R$.
- K is a key for R if K is a superkey, but no proper subset of K is a superkey.

Examples of Super Keys

Consider that the `Movies` database example discussed before,

Movies

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
I, Robot	2004	115	SciFi	Fox	Will Smith
I, Robot	2004	115	SciFi	Fox	Bridget Moynahan

- Super Key: $\{title, year, starName\}$
- Super Key: $\{title, year, starName, length\}$
- Any others super keys?

Examples of Keys

Consider that the `Movies` database example discussed before,

Movies

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
I, Robot	2004	115	SciFi	Fox	Will Smith
I, Robot	2004	115	SciFi	Fox	Bridget Moynahan

- Key: $\{title, year, starName\}$
- Why is it a key?

Outline

Assignment

Let's work on an assignment using paper and pencil/pen ...