Design a Relational Model

http://goo.gl/lEQwnx

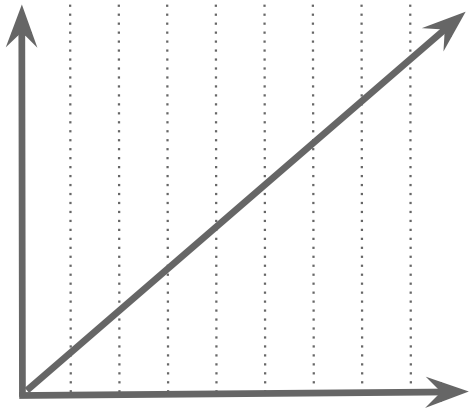
CS5200 DBMS
Bruce Chhay

Design a Relational Model

L1: Functional Dependencies

Functional Dependencies

- Given a relation R, a **functional dependency** is a constraint between two sets of attributes in R, let's say X and Y, such that each value of X is associated with exactly one value of Y.



Functional Dependencies

- Given a relation R, a **functional dependency** is a constraint between two sets of attributes in R, let's say X and Y, such that each value of X is associated with exactly one value of Y.
- Written as $X \rightarrow Y$, or $f(x) = y$.
- The function establishes a relationship between X and Y, where X is determinant set, Y is dependent set.
- Constraint that enforces integrity of records.

Example

Relation: Student Grades

StudentId	Grade
1	3.9
2	4.0
3	3.9

Example

Relation: Student Grades

StudentId	Grade
1	3.9
2	4.0
3	3.9

- StudentId → Grade

Example

Relation: Student Grades

StudentId	CourseId	Grade
1	5200	4.0
2	5600	3.8
3	5010	3.9
1	5600	3.9

Example

Relation: Student Grades

StudentId	CourseId	Grade
1	5200	4.0
2	5600	3.8
3	5010	3.9
1	5600	3.9

- StudentId, CourseId → Grade

Example

Relation: Top 5 Favorite Blog Sites

UserName	BloggerStatus	Rank	BlogId	BlogTitle
Tony	Advanced	1	4	DBMS Cliff Notes
Dan	Intermediate	3	4	DBMS Cliff Notes
James	Intermediate	3	7	Food for Thought
Tony	Advanced	2	9	Food I Crave

Example

Relation: Top 5 Favorite Blog Sites

UserName	BloggerStatus	Rank	BlogId	BlogTitle
Tony	Advanced	1	4	DBMS Cliff Notes
Dan	Intermediate	3	4	DBMS Cliff Notes
James	Intermediate	3	7	Food for Thought
Tony	Advanced	2	9	Food I Crave

- UserName → BloggerStatus
- UserName, Rank → BlogId
- BlogId → BlogTitle

Problems

- **Redundancy**: BloggerStatus and BlogTitle dependent values appear multiple times since the determinant values appear multiple times.

UserName	BloggerStatus	Rank	BlogId	BlogTitle
Tony	Advanced	1	4	DBMS Cliff Notes
Dan	Intermediate	3	4	DBMS Cliff Notes
James	Intermediate	3	7	Food for Thought
Tony	Advanced	2	9	Food I Crave

- UserName → BloggerStatus
- UserName, Rank → BlogId
- BlogId → BlogTitle

Problems

- **Inconsistency**: updating BloggerStatus or BlogTitle for a single record could cause inconsistency by integrity constraints.

UserName	BloggerStatus	Rank	BlogId	BlogTitle
Tony	Expert	1	4	One Weird DBMS Trick
Dan	Intermediate	3	4	DBMS Cliff Notes
James	Intermediate	3	7	Food for Thought
Tony	Advanced	2	9	Food I Crave

- UserName → BloggerStatus
- UserName, Rank → BlogId
- BlogId → BlogTitle

Canonical Set

- **Canonical set**, or **minimal set**, of FDs cannot be further reduced when:
  - Each dependent set, Y, contains only one attribute.
  - Removing an attribute from the determinant set, X, would alter the relation, R.
  - Eliminating a FD from the set would alter the relation, R.

Relation: BlogUsers (Normalized)

UserName	BloggerStatus	FirstName	LastName	DoB
Tony	3	Tony	Davidson	1980-06-10
Dan	2	Daniel	Kwan	1992-02-16
James	2	James	Marks	1991-01-25

Canonical set:
UserName → BloggerStatus
UserName → FirstName
UserName → LastName
UserName → DoB

Canonical Set

- **Canonical set**, or **minimal set**, of FDs cannot be further reduced when:
  - Each dependent set, Y, contains only one attribute.
  - Removing an attribute from the determinant set, X, would alter the relation, R.
  - Eliminating a FD from the set would alter the relation, R.

Relation: Top 5 Favorite Blog Sites (Normalized)

UserName	Rank	BlogId
Tony	1	4
Dan	3	4
James	3	7
Tony	2	9

UserName~~, Rank~~ → BlogId

Canonical Set

- **Canonical set**, or **minimal set**, of FDs cannot be further reduced when:
  - Each dependent set, Y, contains only one attribute.
  - Removing an attribute from the determinant set, X, would alter the relation, R.
  - Eliminating a FD from the set would alter the relation, R.

Relation: BlogUsers (Normalized)

UserName	BloggerStatus	FirstName	LastName	DoB
Tony	3	Tony	Davidson	1980-06-10
Dan	2	Daniel	Kwan	1992-02-16
James	2	James	Marks	1991-01-25

Canonical set:
UserName → BloggerStatus
UserName → FirstName
UserName → LastName
~~UserName → DoB~~

Canonical Set

- **Canonical set**, or **minimal set**, of FDs cannot be further reduced when:
  - Each dependent set, Y, contains only one attribute.
  - Removing an attribute from the determinant set, X, would alter the relation, R.
  - Eliminating a FD from the set would alter the relation, R.
- Use a canonical set to reduce redundancies and inconsistencies.

FD Properties

Armstrong's Axioms:

- **Augmentation**: if $X \rightarrow Y$, then $XZ \rightarrow YZ$
- **Reflexivity**: (subset is related to superset, e.g. identity) if $Y <= X$, then $X \rightarrow Y$.
- **Transitivity**: if $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

Secondary rules:

- **Union**: if $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.
- **Decomposition**: if $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.
- **Composition**: if $X \rightarrow Y$ and $Z \rightarrow W$, then $XZ \rightarrow YW$.

Closure

- The closure, F+, of a set of functional dependencies, F, is the set of all functional dependencies logically implied by F.
- For the previous example, the closure is UserName -> BloggerStatus,FirstName,LastName,DoB.
- When the closure of a relation is one FD, then the determinate set X is a good choice for a primary key.

Closure

- The closure, F+, of a set of functional dependencies, F, is the set of all functional dependencies logically implied by F.
- For the previous example, the closure is UserName → BloggerStatus,FirstName,LastName,DoB.

Deductive proof:

Given the canonical set:

UserName → BloggerStatus
UserName → FirstName
UserName → LastName
UserName → DoB

Apply union (if X → Y and X → Z, then X → YZ) multiple times:

UserName → BloggerStatus,FirstName,LastName,DoB

Closure

- The closure, F+, of a set of functional dependencies, F, is the set of all functional dependencies logically implied by F.
- For the previous example, the closure is UserName → BloggerStatus,FirstName,LastName,DoB.

Deductive proof:

Given the closure:

UserName → BloggerStatus,FirstName,LastName,DoB

Apply decomposition (if X → YZ, then X → Y and X → Z) multiple times:

UserName → BloggerStatus
UserName → FirstName
UserName → LastName
UserName → DoB

Closure

- The closure, F+, of a set of functional dependencies, F, is the set of all functional dependencies logically implied by F.
- For the previous example, the closure is UserName -> BloggerStatus,FirstName,LastName,DoB.
- When the closure of a relation is one FD, then the determinate set X is a good choice for a primary key.

Design a Relational Model

L2: Normalization

Normalization

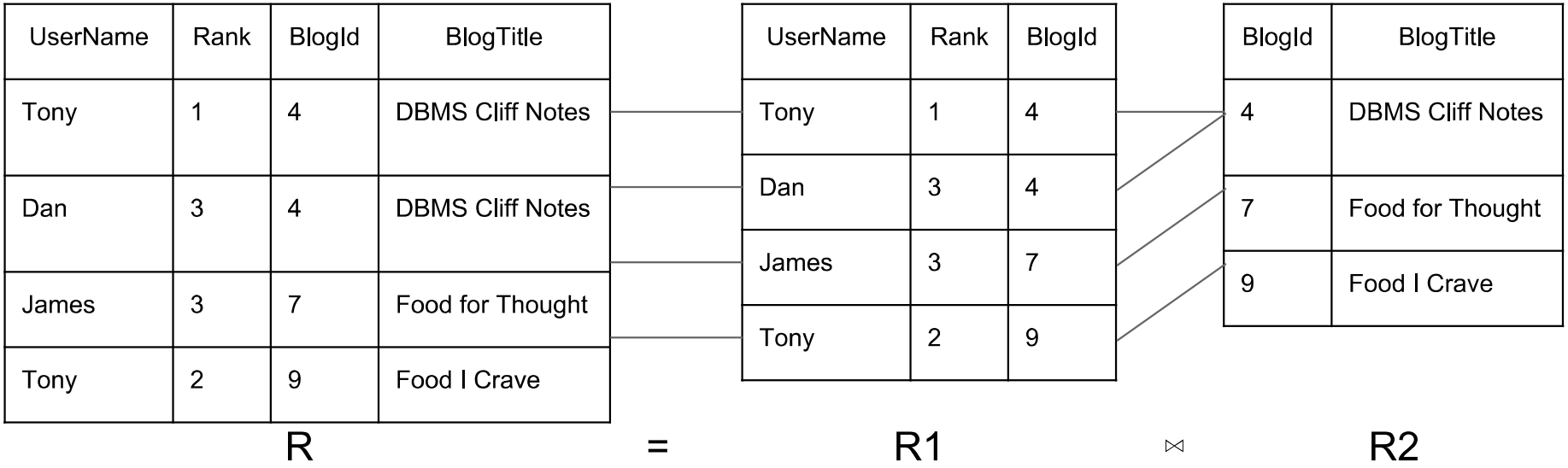
- Process of minimizing redundancy so that modification of an attribute only needs to be done in one table and will be persisted through the database by relationships.
- EF Codd, father of relational dbs, normalization attempts to:
  - Eliminate insert, update, delete inconsistencies.
  - Reduce need for redesign when new attributes/relations are added.
  - Increase query usability.
  - Ensure query usability in the future.
- Your relational model is considered normalized if it reaches a certain normal form level and thus is free of modification irregularities.

Heath's Theorem

- **Heath's Theorem**: the join of a decomposition (if $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$) yields the original relation.
- Normalization: reorganize relation (to minimize redundancy and eliminate inconsistencies).
- The original relation can be recreated through a join, which will be lossless.

Decomposition (Heath's Theorem)

Relation: Top 5 Favorite Blog Sites



UserName	Rank	BlogId	BlogTitle
Tony	1	4	DBMS Cliff Notes
Dan	3	4	DBMS Cliff Notes
James	3	7	Food for Thought
Tony	2	9	Food I Crave

R = R1 ⋈ R2

UserName	Rank	BlogId
Tony	1	4
Dan	3	4
James	3	7
Tony	2	9

BlogId	BlogTitle
4	DBMS Cliff Notes
7	Food for Thought
9	Food I Crave

Normalization

- Process of minimizing redundancy so that modification of an attribute only needs to be done in one table and will be persisted through the database by relationships.
- EF Codd, father of relational dbs, normalization attempts to:
  - Eliminate insert, update, delete inconsistencies.
  - Reduce need for redesign when new attributes/relations are added.
  - Increase query usability.
  - Ensure query usability in the future.
- Your relational model is considered normalized if it reaches a certain normal form level and thus is free of modification irregularities.

Terminology

- **Superkey**: set of attributes that can uniquely identify a record.
- **Candidate key**: a minimal superkey.
- **Prime attribute**: attribute that occurs in a candidate key (i.e. in the determinant set).
- **Non-prime attribute**: attribute that does not occur in any candidate keys (i.e. in the dependent set).
- **Primary key**: superkey chosen for the relation.
- **Alternate key**: superkeys that are not the primary key.

Types of Primary Keys

- **Surrogate**: artificial key chosen for the relation as a substitute for a candidate key.
- **Natural**: naturally occurring outside the db.
- **Foreign**: primary key from another relation.
- **Simple** vs. **compound**: single attribute vs. two or more.
- **Concatenated**: two or more attributes combined into one.

Normalization

- Process of minimizing redundancy so that modification of an attribute only needs to be done in one table and will be persisted through the database by relationships.
- EF Codd, father of relational dbs, normalization attempts to:
  - Eliminate insert, update, delete inconsistencies.
  - Reduce need for redesign when new attributes/relations are added.
  - Increase query usability.
  - Ensure query usability in the future.
- Your relational model is considered normalized if it reaches a certain **normal form** level and thus is free of modification irregularities.

First Normal Form (1NF)

- What: each attribute contains a single value and is atomic.
- How: reorganize attributes that contain repeated (i.e. array) and/or nested (i.e. nested record) values.

1NF Example

Repeated

StudentId	FirstName	RegisteredCourses
2	Tony	5200, 5010, 5600

1NF Example

Repeated

StudentId	FirstName	RegisteredCourses
2	Tony	5200, 5010, 5600

1NF [1]

StudentId	FirstName	RegisteredCourse
2	Tony	5200
2	Tony	5010
2	Tony	5600

1. 1NF does not require a candidate key, so duplicate StudentId values can exist. So what's the 3NF of this relation?

1NF Example

Nested

StudentId	FirstName	Grade	
2	Tony	5200	4.0

1NF Example

Nested

StudentId	FirstName	Grade	
2	Tony	5200	4.0

1NF

StudentId	FirstName
2	Tony

StudentId	Course	Grade
2	5200	4.0

Second Normal Form (2NF)

- What: in addition to 1NF, every non-prime attribute in the relation is dependent on the *whole* of a candidate key. [1]
- How: reorganize to ensure no non-prime attribute is dependent on a (proper) subset of a candidate key.

1. Also known as a “full functional dependency”: dependent set, Y, cannot be functionally dependent on a proper subset of determinant, X. Stated differently, this is also equivalent to rule #2 of canonical set: removing an attribute from the determinant set, X, would alter the relation, R.

2NF Example

CourseNumber	Semester	Instructor	Title
5200	Spring	Bruce	DBMS
5010	Spring	Ezra	PDP
5600	Spring	Nate	Systems
5200	Fall	Jose	DBMS
5010	Fall	Jim	PDP

- FDs: CourseNumber,Semester → Instructor, CourseNumber → Title.
- CourseNumber is a proper subset of the primary key CourseNumber,Semester.

Advanced example: Given a candidate key that is not the primary key, and a non-prime attribute is dependent on a subset of that candidate key.

2NF Example

2NF

CourseNumber	Semester	Instructor
5200	Spring	Bruce
5010	Spring	Ezra
5600	Spring	Nate
5200	Fall	Jose
5010	Fall	Jim

CourseNumber	Title
5010	PDP
5200	DBMS
5600	Systems

CourseNumber,Semester → Instructor

CourseNumber → Title

Third Normal Form (3NF)

- What: in addition to 2NF, all attributes in the relation are dependent on the primary key and only the primary key.
- How: reorganize to eliminate non-prime attributes that are dependent on other non-prime attributes (i.e. remove non-prime attributes that have a transitive dependency on a superkey).

3NF Example

CourseNumber	Semester	Instructor	InstructorHanded
5200	Spring	Bruce	Right
5010	Spring	Ezra	Right
5600	Spring	Nate	Left
8003	Fall	Bruce	Right
5200	Fall	Jim	Right

- FDs:
  CourseNumber,Semester → Instructor
  CourseNumber,Semester → InstructorHanded
  Instructor → InstructorHanded
- InstructorHanded, a non-prime attribute, is dependent on Instructor, another non-prime attribute.

3NF Example

3NF

CourseNumber	Semester	Instructor
5200	Spring	Bruce
5010	Spring	Ezra
5600	Spring	Nate
8003	Fall	Bruce
5200	Fall	Jim

Instructor	InstructorHanded
Bruce	Right
Ezra	Right
Nate	Left
Jim	Right

CourseNumber,Semester → Instructor

Instructor → InstructorHanded

Boyce-Codd (BCNF)

- What: stronger version of 3NF, also known as 3.5NF -- every functional dependency in a relation is a dependency on a superkey.
- How: reorganize to remove prime attributes that have a functional dependency.

Normal Form in Practice

- 3NF is the standard since it eliminates modification irregularities.
- 3NF is usually BCNF.
- Design for 3NF from the start (instead of intentionally working through all normal forms).

Normal Form in Practice

The relational db oath:
"Every non-key attribute must provide a fact about the key *(1NF)*, the whole key *(2NF)*, and nothing but the key *(3NF)*, so help me Codd." [1].

1. Affirmation for the oath of truth: I solemnly swear to tell the truth, the whole truth, and nothing but the truth, so help me God.