Relational Database Design

- One single, large table
- Simple ?
- Good ? or Bad? Or just Ugly?

Normal Forms

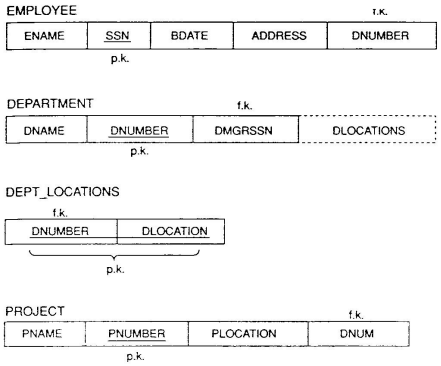
- We discussed how to fix 'bad' schemas
- but what is a 'good' schema?
- Informally: "we want tables where the attributes depend on the primary key, on the whole key, and nothing but the key"
- Formally: 'good', if it obeys a 'normal form'
- Typically: Boyce-Codd Normal form or the 3NF
- Normal forms are defined in terms of FDs

Functional dependency

- Definition: Let $R=(A_1, A_2,..., A_n)$ and $X \subseteq R$ and $Y \subseteq R$
  $X \rightarrow Y$ if the value of X **uniquely** determines a value of Y
- A functional dependency is a property of the meaning or semantic of the attributes in a relation schema.
- We use our understanding of the semantics of the attributes of R – that is, how they relate to one another – to specify the FD that should hold an all relational instances.
- Functional dependence is a semantic notion.
  - Recognizing the FDs is part of the process of understanding what data means.

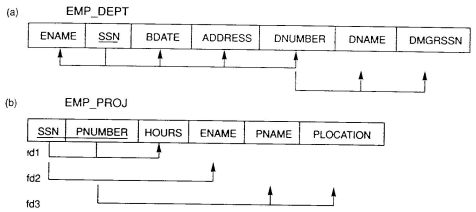
Good Database Schema

- Relations should have simple meaning



Bad Database Schema

- Relations should not have multiple meanings



Types of Functional Dependencies

- Trivial dependency: X→Y is *trivial* if it is true for any X and Y of any relation, regardless of X and Y semantics.
  - Ex1: A -> A
  - Ex2: If {A,B} a Key, then {A,B} → A (Y⊆X, X→Y is trivial)
- Partial dependency: X→Y is *partial* if there is an attribute A in X that can be removed from X and the dependency can still hold: X–{A} → Y
  - E.g., SUPPLY (SID,PID,DID, SCity, DCity, Qty)
    {SID, PID, DID} → Scity SID → SCity
    {SID, PID, DID} → Dcity DID → Dcity
- Full dependency: ??

Types of Functional Dependencies…

- Transitive dependency: X→Y is transitive in R if there is a set of attributes Z that is not a subset of any key of R and both X → Z and Z → Y hold
  - E.g., EMP (SSN, EName, DeptID, MGRSSN)
    (fd.1) SSN → DeptId
    (fd.2) DeptId → MGRSSN
    (from fd.1 & fd.2) SSN → MGRSSN
- Multivalued dependency : X→Y is multivalued dependency in R if X is a key, Z in R and Z→Y
  - E.g., DJP (DeptID, ProjectID, part)
    (fd.1) DeptId → part
    (fd.2) ProjectID → part

Non-Relational Table

- Unnormalized table/ NoSQL

MemID	Full Names	Address	Movies Rented	Awards
1	Janet Jones	First Street Plot No 4	(Pirates of the Caribbean, Clash of the Titans)	(15, 10)
2	Robert Phil	3rd Street 34	(Forgetting Sarah Marsal, Daddy's Little Girls)	(16, 2)
3	Robert Phil	5th Avenue	Clash of the titans	10

Order Lists/ Arrays

First Normal Form

- 1NF: First Normal Form
  Every attribute has a single atomic value.
- Example in 1NF

MemID	Full Names	Address	Movies Rented	Awards
1	Janet Jones	First Street Plot No 4	Pirates of the Caribbean	15
1	Janet Jones	First Street Plot No 4	Clash of the Titans	10
2	Robert Phil	3rd Street 34	Forgetting Sarah Marsal	16
2	Robert Phil	3rd Street 34	Daddy's Little Girls	2
3	Robert Phil	5th Avenue	Clash of the titans	10

- PK: MemID, Movies Rented
- FD1: MemID → (Full Names, Address)
- FD2: Movies Rented → Awards

Not in Second Normal Form

- PK: memID, Movies Rented
- FD1: memID → (Full Names, Address)
- FD2: Movies Rented → Awards

MemID	Full Names	Address	Movies Rented	Awards
1	Janet Jones	First Street Plot No 4	Pirates of the Caribbean	15
1	Janet Jones	First Street Plot No 4	Clash of the Titans	10
2	Robert Phil	3rd Street 34	Forgetting Sarah Marsal	16
2	Robert Phil	3rd Street 34	Daddy's Little Girls	2
3	Robert Phil	5th Avenue	Clash of the titans	10

Second Normal Forms

- 2NF: Second Normal Form
  It is in 1NF and does not have partial dependencies.
  - Counter Example: 1NF but not 2NF
    SUPPLY (SID, PID, DID, SCity, DCity, Qty)
- Example in 2NF

MemID	Full Names	Address
1	Janet Jones	First Street Plot No 4
2	Robert Phil	3rd Street 34
3	Robert Phil	5th Avenue

FD1:
MemID → (Full Names, Address)

MemID	Movies Rented	Awards
1	Pirates of the Caribbean	15
1	Clash of the Titans	10
2	Forgetting Sarah Marsal	16
2	Daddy's Little Girls	2
3	Clash of the titans	10

Third Normal Forms

- 3NF: Third Normal Form
  It is in 2NF and does not have transitive dependencies to attributes that are not part of a key.
  - If X→A is an FD then (a) it is trivial, or (b) X is a superKey, or (c) A is a subset of a **candidate** Key.
  - Counter Example: 2NF but not 3NF
    EMP (SSN, EName, Bdate, Salary, DID, DName, MGRSSN)
- Example 3NF

3NF Example

MemID	Full Names	Address
1	Janet Jones	First Street Plot No 4
2	Robert Phil	3rd Street 34
3	Robert Phil	5th Avenue

FD1:
MemID → (Full Names, Address)

MemID	Movies Rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marsal
2	Daddy's Little Girls
3	Clash of the titans

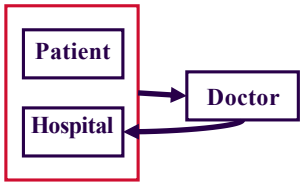
Movies Rented	Awards
Pirates of the Caribbean	15
Clash of the Titans	10
Forgetting Sarah Marsal	16
Daddy's Little Girls	2
Clash of the titans	10

PK: MemID, Movies Rental

FD2:
Movies Rented → Awards

Normal Forms - 3NF

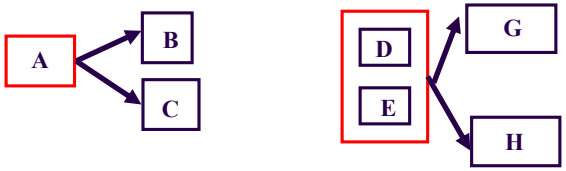
- Example:
  PATIENT-VISIT (PATIENT,HOSPITAL,DOCTOR)
  fd.1 (Patient, Hospital) → Doctor
  fd.2 Doctor → Hospital
- Pictorially:



- *3NF may not be adequate*. Why??
- May force NULL values to attributes of the primary key.

Normal Forms - BCNF

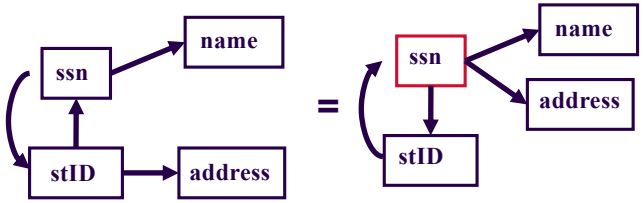
- Boyce-Codd Normal Form: A relation is in 3NF and has not transitive dependencies
  - if X → A is an FD, then
    (a) it is trivial, or (b) X is a superKey.
  - Informally: everything depends on the **full key**, and nothing but the key
- Pictorially: we want a 'star' shape



Normal Forms - BCNF

- or a star-like: (e.g., 2 cand. keys):

  STUDENT(ssn, stID, name, address)



Normal Forms - BCNF

- Theorem:
  given a schema R and a set of FD 'F', we can always decompose it to schemas R1, … Rn,
  so that
  - R1, …, Rn are in BCNF and
  - the decompositions is lossless
- But, some decompositions might lose dependencies ⇒ use 3NF
  - 3NF always loseless
  - 3NF always preserves dependencies

BCNF & Dependency Preservation

- BCNF is not always dependency preserving
- Example: 3NF but not BCNF
  PATIENT-VISIT (PATIENT, HOSPITAL,DOCTOR)
  fd.1 (Patient, Hospital) → Doctor
  fd.2 Doctor → Hospital
- Possible Decomposition 1:
  - Doctor-Hospital (Doctor, Hospital) {Doctor-> Hospital}
  - Patient-Doctor (Patient, Doctor) {Patient → Doctor}
- Possible Decomposition 2:
  - Doctor-Hospital (Doctor, Hospital) {Doctor-> Hospital}
  - Patient-Doctor (Patient, Hospital) {Patient → Hospital}
- BUT these decompositions lose fd.1

Normal Forms - 4NF

- Fourth Normal Form: A relation is in BCNF and has no Multivalue Dependencies
- Example: **(FACULTY, Dept, Committee)**

1. A faculty member can belong to more than one dept.
2. A faculty can be on several college-wide committees.
3. There is no relation between dept. and committee.

FacultyID	Dept	Committee
F101	CS	Budget
F101	CoE	Budget
F101	CS	Curriculum
F101	CoE	Curriculum
F221	Bio	Library
F330	Math	Budget
F330	Math	Admissions

- Anomalies? Change F101 from Budget to Admissions

More Normal Forms…

- 5NF: Fifth Normal Form
  - No Join Dependencies
- 6NF: ….