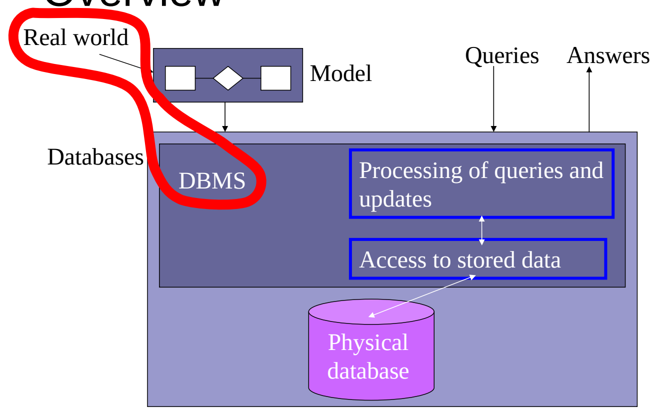
Functional Dependencies and Normalization

Jose M. Peña
jospe@ida.liu.se

* slides kindly provided by Vaida Jakonienė

Overview



Good Design

- Can we be sure that a translation from EER-diagram to relational tables results in good database design?
- Confronted with a deployed database, how can we be sure that it is well-designed?
- What *is* good database design?
  - Four informal measures
  - Formal measure: normalization

Informal design guideline

- Easy to explain semantics of the relation schema
- Reducing redundant information in tuples

  Redundancy causes update anomalies:
  - Insertion anomalies
  - Deletion anomalies
  - Modification anomalies

EMP(	EMPID,	EMPNAME,	DEPTNAME,	DEPTMGR)
	123	Smith	Research	999
	333	Wong	Research	999
	888	Borg	Administration	null

Informal design guideline

- Sometimes, it may be desirable to have redundancy to gain in runtime, i.e. trade space for time.
- In that case and to avoid update anomalies
  - either, use triggers or stored procedures to update the base tables
  - or, keep the base tables free of redundancy and use views (assuming that the views are materialized).

Informal design guideline

- Reducing NULL values in tuples

  Why
  - Efficient use of space
  - Avoid costly outer joins
  - Ambiguous interpretation (unknown vs. doesn't apply).
- Disallow the possibility of generating spurious tuples
  - Figures 10.5 and 10.6: cartesian product results in incorrect tuples
  - Only join on foreign key/primary key-attributes
  - Lossless join property: guarantees that the spurious tuple generation problem does not occur

Functional dependencies (FD)

- Let R be a relational schema with the attributes $A_1,...,A_n$ and let X and Y be subsets of $\{A_1,...,A_n\}$.
- Let r(R) denote a relation in relational schema R.

> We say that X *functionally determines* Y,
> X → Y
> if for each pair of tuples $t_1, t_2 \in r(R)$ and for all relations in r(R):
> If $t_1[X] = t_2[X]$ then we must also have $t_1[Y] = t_2[Y]$

- Despite the mathematical definition an FD cannot be determined automatically. It is a property of the semantics of attributes.

Inference rules

1. If X ⊇ Y then X→Y, or X → X (reflexive rule)
2. X→Y |= XZ → YZ (augmentation rule)
3. X → Y, Y → Z |= X → Z (transitive rule)
4. X → YZ |= X → Y (decomposition rule)
5. X → Y, X → Z |= X → YZ (union or additive rule)
6. X → Y, WY → Z |= WX → Z (pseudotransitive rule)

Inference rules

- Textbook, page 341:
  ”… X → A, and Y → B does *not* imply that XY → AB.”
  Prove that this statement is wrong.

- Prove inference rules 4, 5 and 6 by using **only** inference rules 1, 2 and 3.

Definitions

For any relation extension or state

- **Superkey:** a set of attributes uniquely (but not minimally!) identifying a tuple of a relation.
- **Key:** A ***set of attributes*** that uniquely and minimally identifies a tuple of a relation.
- **Candidate key:** If there is more than one ***key*** in a relation, the keys are called candidate keys.
- **Primary key:** One ***candidate key*** is chosen to be the primary key.
- **Prime attribute:** An attribute ***A*** that is part of a ***candidate*** key ***X*** (vs. nonprime attribute)

Normal Forms

- 1NF, 2NF, 3NF, BCNF (4NF, 5NF)
- ***Minimize redundancy***
- ***Minimize update anomalies***
- Normal form ↑ = redundancy and update anomalies ↓ and relations become smaller.
- Join operation to recover original relations.

1NF

- 1NF: The relation should have no non-atomic values.

R_{non1NF}

ID	Name	LivesIn
100	Pettersson	{Stockholm, Linköping}
101	Andersson	{Linköping}
102	Svensson	{Ystad, Hjo, Berlin}

What about multi-valued composite attributes ?

Normalization

$R1_{1NF}$

ID	Name
100	Pettersson
101	Andersson
102	Svensson

$R2_{1NF}$

ID	LivesIn
100	Stockholm
100	Linköping
101	Linköping
102	Ystad
102	Hjo
102	Berlin

2NF

- 2NF: *no **nonprime** attribute* should be functionally dependent on a ***part*** of a candidate key (= partial dependency).

R_{non2NF}

EmpID	Dept	Work%	EmpName
100	Dev	50	Baker
100	Support	50	Baker
200	Dev	80	Miller

Normalization

$R1_{2NF}$

EmpID	EmpName
100	Baker
200	Miller

$R2_{2NF}$

EmpID	Dept	Work%
100	Dev	50
100	Support	50
200	Dev	80

2NF

- No 2NF: A part of a candidate key can have repeated values in the relation and, thus, so can have the nonprime attribute, i.e. redundancy + insertion and modification anomalies.
- An FD X→Y is a **full functional dependency (FFD)** if removal of any attribute A_i from X means that the dependency does not hold any more.
- 2NF: Every **nonprime** attribute is fully functionally dependent on every candidate key.

3NF

- 3NF: **2NF +** no **nonprime** attribute should be functionally dependent on a set of nonprime attributes

R_{non3NF}

ID	Name	Zip	City
100	Andersson	58214	Linköping
101	Björk	10223	Stockholm
102	Carlsson	58214	Linköping

Normalization

$R1_{3NF}$

ID	Name	Zip
100	Andersson	58214
101	Björk	10223
102	Carlsson	58214

$R2_{3NF}$

Zip	City
58214	Linköping
10223	Stockholm

3NF

- No 3NF (but 2NF): A set of nonprime attributes can have repeated values in the relation and, thus, so can have the nonprime attribute, i.e. redundancy + insertion and modification anomalies.
- An FD X→Y is a **transitive dependency** if there is a set of **nonprime** attributes Z such that both X→Z and Z→Y hold.
- 3NF: 2NF + no **nonprime** attribute is transitively dependent on any candidate key.

Little summary

- X → A
- 2NF and 3NF do nothing if A is prime.
- Assume A is nonprime.
- 2NF = decompose if X is part of a candidate key.
- 3NF = decompose if X is part of a candidate key or X is nonprime, i.e. if X → A is partial or transitive.
- 3NF = X is a superkey or A is prime.
- Should A be discriminated for being prime ?

Boyce-Codd Normal Form

- BCNF: **Every determinant is a superkey**
  (in practice: every determinant is a candidate key)
- BCNF = decompose if X → A is such that X is not a superkey and A is a prime attribute.
- Example: Given R(A,B,C,D) and AB→CD, C→B. Then R is in 3NF but not in BCNF
  - C is a determinant but not a superkey (tuples are not uniquely identified in R)

BCNF: Example

At a gym, an instructor is leading an activity in a certain room at a certain time.

$R_{nonBCNF}$

<u>Time</u>	<u>Room</u>	Instructor	Activity
Mon 17.00	Gym	Tina	IronWoman
Mon 17.00	Mirrors	Anna	Aerobics
Tue 17.00	Gym	Tina	Intro
Tue 17.00	Mirrors	Anna	Aerobics
Wed 18.00	Gym	Anna	IronWoman

Properties of decomposition

- Keep all attributes from the universal relation R.
- Preserve the identified functional dependencies.
- Lossless join
  - It must be possible to join the smaller tables to arrive at composite information without spurious tuples.

Normalization: Example

Given universal relation

R(PID, PersonNamn, Land, Kontinent, KontinentYta, AntalBesökILandet)

- Functional dependencies?
- Keys?

Normalization: Example

PID → PersonNamn

PID, Land → AntalBesökILandet

Land → Kontinent

Kontinent → KontinentYta

- Based on FDs, what are keys for R?
- Use inference rules

Normalization: Example

Land → Kontinent, Kontinent → KontinentYta,
then
Land → Kontinent, KontinentYta (transitive rule)

PID, Land → Kontinent, KontinentYta (augmentation rule),
PID, Land → PersonNamn (augmentation rule),
PID, Land → AntalBesökILandet,
then
PID, Land → Kontinent, KontinentYta, PersonNamn, AntalBesökILandet (additive rule)

<u>Person, Land</u> is the key for R.

Normalization: Example

Is
R (<u>PID, Land</u>, Kontinent, KontinentYta, PersonNamn, AntalBesökILandet)
in 2NF?

No, *PersonNamn* depends on a part of the key (*PID*), then
R1(<u>PID</u>, PersonNamn)
R2(<u>PID, Land</u>, Kontinent, KontinentYta, AntalBesökILandet)

Is R2 in 2NF?
No, *Kontinent* and *KontinentYta* depend on a part of the key (*Land*), then
R1(<u>PID</u>, PersonNamn)
R21(<u>Land</u>, Kontinent, KontinentYta)
R22(<u>PID, Land</u>, AntalBesökILandet)
→ R1, R21, R22 are in 2NF

> 2NF: *no nonprime attribute* should be functionally dependent on a ***part*** of a candidate key.

Are R1, R21, R22 in 3NF?

3NF: 2NF + no nonprime attribute should be functionally dependent on a set of nonprime attributes (= no transitive dependency)

R22(<u>PID, Land</u>, AntalBesökILandet),
R1(<u>PID</u>, PersonNamn):
Yes, a single nonprime attribute, no transitive dependencies.

R21(<u>Land</u>, Kontinent, KontinentYta):
No, Kontinent defines KontinentYta, then
R211(<u>Land</u>, Kontinent)
R212(<u>Kontinent</u>, KontinentYta)

→ R1, R22, R211, R212 are in 3NF

Are R1, R22, R211, R212 in BCNF?

BCNF: Every determinant is a superkey

R22(<u>PID, Land</u>, AntalBesökILandet),
R1(<u>PID</u>, PersonNamn):
R211(<u>Land</u>, Kontinent)
R212(<u>Kontinent</u>, KontinentYta)

→ Yes **(don't be confused by candidate keys!)**

Can the universal relation R be reproduced from R1, R22, R211 and R212 without spurious tuples?

Summary and open issues

- Good design ... properties of relations
- Functional ... normal forms, are ... = real-world know... only be automated
- Are high no... when it comes to pe...
  - No, denormalization may be required.

1. Which normal form?

- The database contains data about cars, their owners and when the car was registered for that owner.

<u>PersonID</u>	FirstName	LastName	<u>LicensePlate</u>	RegistrationDate	Birthdate
1000	Ann	Anderson	ABC123	2004-10-12	1981-04-04
1010	Ben	Benson	DEF234	2003-02-12	1945-12-12
1000	Ann	Anderson	ABC123	2001-04-23	1981-04-04

2. Which normal form?

- A database contains data about registered cars and their make (type).

<u>LicensePlate</u>	Type	Maker
ABC123	C70	Volvo
DEF234	S40	Volvo
FGH345	Corolla	Toyota

3. Which normal form?

- The database contains data about flights, aircrafts and their pilots. Flights use different aircrafts depending on the number of booked passengers.

<u>Date</u>	<u>Flight</u>	<u>Aircraft</u>	Pilot
13-Jan-2005	TGU7	Airbus 300	John
14-Jan-2005	TGU7	Boeing 747	Daniel
12-Jan-2005	SKX6	Airbus 300	John
13-Jan-2005	SKX6	Boeing 747	Ann
14-Jan-2005	SKX6	Fokker 50	Mary