Databases and information management

Lecture 8

Piotr Syga

2022-11-17

1NF

First normal form

Relation R is in 1NF **iff** every attribute is atomic and there are no duplicate rows (i.e., there is a key).

2NF

Second normal form

Relation R is in 2NF **iff** R is in 1NF and no non-prime attribute depends functionally on a proper subset of a candidate key.

It means that if A is a non-prime attribute and X is a candidate key in R, then every f.d. $Y \to A$, where $Y \subset X$ violates 2NF.

Normalization

1. We select a f.d. $X \to Y$ violating 2NF
2. We decompose R w.r.t. the f.d. $U_1 = \{X\}^+$, $U_2 = (U \setminus \{X\}^+) \cup X$.
3. We end when no f.d. violates 2NF.

3NF

Third normal form

Relation R is in 3NF **iff** R is in 2NF and all non-prime attributes depend only on a complete candidate key.

It means that for any non-prime attributes X, Y any f.d. $X \rightarrow Y$ violates 3NF.

3NF

Third normal form

Relation R is in 3NF **iff** R is in 2NF and all non-prime attributes depend only on a complete candidate key.

It means that for any non-prime attributes X, Y any f.d. $X \to Y$ violates 3NF.

In other words

Relation R is in 3NF **iff** for any f.d. $(X \to Y) \in \mathcal{F}$ for R, at least one condition is fulfilled:

- $Y \subseteq X$
- X is a key
- every attribute $A \in Y \setminus X$ is a primary attribute.

Decomposition of a relation

ID	FName	LName	RoomNo	DeptID
1	John	Doe	107	D11/2
2	Jane	Doe	107	D13/4
3	John	Smith	387	D4/1
4	Jane	Smith	41	D11/6

DeptID	FacID	DeptName
D11/2	11	Computer Science
D13/4	13	Mathematics
D4/1	4	Computer Science
D11/6	11	Theoretical Physics

Lecturer	CID	CName
1	I04	Data bases
2	T06	Graph Theory
3	Z17	System architecture
4	I23	Complex Systems

Example

$R = ABCDEFG$,
$\mathcal{F} =$
$\{DG \to CE,\ B \to FA,\ ED \to CBG,\ G \to FA,\ EG \to BA,\ CDF \to E\}$
We start with splitting the right sides (canonical form):
$\mathcal{F} = \{DG \to E,\ DG \to C,\ B \to F,\ B \to A,\ ED \to C,\ ED \to B,$
$ED \to G,\ G \to F,\ G \to A,\ EG \to B,\ EG \to A,\ CDF \to E\}$
We're looking for surplus f.d.:
$DG \to C$: $DG^{+\setminus\{DG\to C\}} = DGE^{+\setminus\{DG\to C\}} = DGEC^{+\setminus\{DG\to C\}}$ – surplus
$DG \to E$: $DG^{+\setminus\{DG\to E\}} = DGAF$ – required

...
$\mathcal{F}_{\min} =$
$\{DG \to E,\ B \to F,\ B \to A,\ ED \to C,\ ED \to G,\ G \to F,\ G \to A,$
$EG \to B,\ CDF \to E\}$

Example

$R = ABCDEFG$,
$\mathcal{F} =$
$\{DG \to CE,\ B \to FA,\ ED \to CBG,\ G \to FA,\ EG \to BA,\ CDF \to E\}$
$\mathcal{F}_{\min} =$
$\{DG \to E,\ B \to F,\ B \to A,\ ED \to C,\ ED \to G,\ G \to F,\ G \to A,$
$EG \to B,\ CDF \to E\}$
Looking for candidate keys – checking LHS and RHS of f.d.
D i A, note that $D^+ \neq R$
$DE^+ = DECG^+ = DECGAFB = R$, since $D^+ \neq R$ i $E^+ \neq R$ – a candidate key
Since we have $DG \to E$, let us check $DG^+ = R$, after verification of G^+ we get another candidate key.
Additionally, $CDF \to E$, hence we check $CDF^+ = CDFE^+ = R$.
C can be learned only with DE, so it cannot be modified into a new key, yet F can be learned with $B \to F$ and $G \to F$.
$CDG^+ = R$ – is a key, yet it has DG subset. Since $CDB^+ = R$ i $CD^+ \neq R$, $DB^+ = DBAF \neq R$ and D has to be part of every key, we know that we got a candidate key.

Example

$R = ABCDEFG$,
$\mathcal{F} =$
$\{DG \to CE,\ B \to FA,\ ED \to CBG,\ G \to FA,\ EG \to BA,\ CDF \to E\}$
We check the normal form of the relation, note that A is non-primary and all other attributes are primary. Hence $B \to A$ and $G \to A$ are the only f.d. that can violate 2NF/3NF (and they both do). Yet splitting w.r.t. any of them gives as R again.
Hence we split directly into 3: $R_1 = B^+$, $R_2 = G^+$, R_3 – compliment.
$R_1 = BAF$, $\mathcal{F}_1 = \{B \to A,\ B \to F\}$ – 3NF, key: B
$R_2 = GAF$, $\mathcal{F}_2 = \{G \to A,\ G \to F\}$ – 3NF, key: G
$\mathcal{F}_3' = \mathcal{F}_{\min} \setminus (\mathcal{F}_1 \cup \mathcal{F}_2) =$
$\{DG \to E,\ ED \to C,\ ED \to G,\ EG \to B,\ CDF \to E\}$
$R_3 = BCDEFG$, $\mathcal{F}_3 = \mathcal{F}_3' \cup \{B \to F,\ G \to F\}$

Example

$R = ABCDEFG$,
$\mathcal{F} =$
$\{DG \to CE,\ B \to FA,\ ED \to CBG,\ G \to FA,\ EG \to BA,\ CDF \to E\}$
$R_3 = BCDEFG$, $\mathcal{F}_3 = \mathcal{F}_3' \cup \{B \to F,\ G \to F\}$
D, again, has to be part of each key.
Checking $DG^+ = R$, $DE^+ = R$, $CDF^+ = R$ and $CBD^+ = R$ results in a set of primary attributes: $BCDEFG = R_3$. Since all attributes are primary we have a relation in 3NF.

A Simple Guide to 5 Normal Forms in Relational DB Theory. W. Kent

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.)

BCNF

Boyce–Codd Normal Form

Relation R is in BCNF **iff** for every f.d. $(X \to Y) \in \mathcal{F}$ of R at least one condition is fulfilled:

- $Y \subseteq X$
- X is a key

Note that **not every relation** may be normalized to BCNF without the loss of information (attributes or functional dependencies). Minimal (not)working example is a relation $R = ABC$ with f.d. $\{AB \to C,\ C \to A\}$.

Relational databases

- based on mathematical model
- reflect the relationships between data sets
- easy normalization for removing the anomalies
- consistency checks(triggers, transactions)
- popularity, multiple dialects following a common standard

Motivation for a change

- SQL databases do not allow flexible data – common schema in tables
- object-oriented concepts added artificially to relational databases
- requirement of atomic types in normalized SQL databases further hinders storing multimedia files
- SQL often duplicates some of the capabilities of programming languages
- SQL was designed with non-distributed environment in mind, having some significant problems in distributed setting

ACID – limitations

Transactions

For keeping the proper state of the database after each transactions we require that all operation are A*tomic* – cannot be performed partialy, that the data is C*onsistent* after the transaction, even if multiple datanodes exist, that all transactions are I*solated* and do not interweave and that all the changes after the transaction ended are D*urable*.

Potential problems

- scalability with respect to the size of data
- scalability with respect to number of users
- synchronization in distributed environments
- vulnerability to errors or access denial

NotOnly SQL

- designed with distributed data in mind
- capability of handling large number of 'records'
- adaptability to cope with lightweight as well as multimedia files
- no consistent schema required
- providing significant horizontal scalability (adding data nodes)

CAP

We replace the ACID requirements with ones adjusted to distributed environment:

- (C*onsistency*) after performing the operation,
- (A*vailability*) of the data without interrupts,
- (P*artition tolerance*) when data nodes cannot exchange messages in specific time periods.

In practice we fulfill two of the requirements, e.g., relational databases(CA), MongoDB/HBase(CP), CouchDB/Cassandra(AP).