Databases
Lectures 4, 5, and 6

Timothy G. Griffin

Computer Laboratory
University of Cambridge, UK

Databases, Lent 2009

Lecture 04: Database Updates

Outline

- Transactions
- Short review of ACID requirements

Transactions — ACID properties

Should be review from Concurrent Systems and Applications

Atomicity Either all actions are carried out, or none are

- logs needed to undo operations, if needed

Consistency If each transaction is consistent, and the database is initially consistent, then it is left consistent

- This is very much a part of applications design.

Isolation Transactions are isolated, or protected, from the effects of other scheduled transactions

- Serializability, 2-phase commit protocol

Durability If a transactions completes successfully, then its effects persist

- Logging and crash recovery

Lecture 05: Functional Dependencies

Outline

- Update anomalies
- Functional Dependencies (FDs)
- Normal Forms, 1NF, 2NF, 3NF, and BCNF

Transactions from an application perspective

Main issues

- Avoid update anomalies
- Minimize locking to improve transaction throughput.
- Maintain integrity constraints.

These issues are related.

Update anomalies

Big Table

sid	name	college	course	part	term_name
yy88	Yoni	New Hall	Algorithms I	IA	Easter
uu99	Uri	King's	Algorithms I	IA	Easter
bb44	Bin	New Hall	Databases	IB	Lent
bb44	Bin	New Hall	Algorithms II	IB	Michaelmas
zz70	Zip	Trinity	Databases	IB	Lent
zz70	Zip	Trinity	Algorithms II	IB	Michaelmas

- How can we tell if an insert record is consistent with current records?
- Can we record data about a course before students enroll?
- Will we wipe out information about a college when last student associated with the college is deleted?

Redundancy implies more locking ...

... at least for correct transactions!

Big Table

sid	name	college	course	part	term_name
yy88	Yoni	New Hall	Algorithms I	IA	Easter
uu99	Uri	King's	Algorithms I	IA	Easter
bb44	Bin	New Hall	Databases	IB	Lent
bb44	Bin	New Hall	Algorithms II	IB	Michaelmas
zz70	Zip	Trinity	Databases	IB	Lent
zz70	Zip	Trinity	Algorithms II	IB	Michaelmas

- Change New Hall to Murray Edwards College
  - Conceptually simple update
  - May require locking entire table.

Redundancy is the root of (almost) all database evils

- It may not be obvious, but redundancy is also the cause of update anomalies.
- By redundancy we do not mean that some values occur many times in the database!
  - A foreign key value may be have millions of copies!
- But then, what do we mean?

Functional Dependency

Functional Dependency (FD)

Let $R(\mathbf{X})$ be a relational schema and $\mathbf{Y} \subseteq \mathbf{X}$, $\mathbf{Z} \subseteq \mathbf{X}$ be two attribute sets. We say $\mathbf{Y}$ functionally determines $\mathbf{Z}$, written $\mathbf{Y} \rightarrow \mathbf{Z}$, if for any two tuples u and v in an instance of $R(\mathbf{X})$ we have

$$u.\mathbf{Y} = v.\mathbf{Y} \rightarrow u.\mathbf{Z} = v.\mathbf{Z}.$$

We call $\mathbf{Y} \rightarrow \mathbf{Z}$ a functional dependency.

A functional dependency is a <u>semantic</u> assertion. It represents a rule that should always hold in any instance of schema $R(\mathbf{X})$.

Example FDs

Big Table

sid	name	college	course	part	term_name
yy88	Yoni	New Hall	Algorithms I	IA	Easter
uu99	Uri	King's	Algorithms I	IA	Easter
bb44	Bin	New Hall	Databases	IB	Lent
bb44	Bin	New Hall	Algorithms II	IB	Michaelmas
zz70	Zip	Trinity	Databases	IB	Lent
zz70	Zip	Trinity	Algorithms II	IB	Michaelmas

- **sid** → **name**
- **sid** → **college**
- **course** → **part**
- **course** → **term_name**

Keys, revisited

Candidate Key

Let $R(\mathbf{X})$ be a relational schema and $\mathbf{Y} \subseteq \mathbf{X}$. $\mathbf{Y}$ is a candidate key if

1. The FD $\mathbf{Y} \rightarrow \mathbf{X}$ holds, and
2. for no proper subset $\mathbf{Z} \subset \mathbf{Y}$ does $\mathbf{Z} \rightarrow \mathbf{X}$ hold.

Prime and Non-prime attributes

An attribute A is prime for $R(\mathbf{X})$ if it is a member of some candidate key for R. Otherwise, A is non-prime.

Database redundancy roughly means the existence of non-key functional dependencies!

First Normal Form (1NF)

We will assume every schema is in 1NF.

1NF

A schema $R(A_1 : S_1,\ A_2 : S_2,\ \cdots\ ,\ A_n : S_n)$ is in First Normal Form (1NF) if the domains S_1 are elementary — their values are atomic.

name
Timothy George Griffin

$\Longrightarrow$

first_name	middle_name	last_name
Timothy	George	Griffin

Second Normal Form (2NF)

Second Normal Form (2CNF)

A relational schema R is in 2NF if for every functional dependency $\mathbf{X} \rightarrow A$ either

- $A \in \mathbf{X}$, or
- $\mathbf{X}$ is a superkey for R, or
- A is a member of some key, or
- $\mathbf{X}$ is not a proper subset of any key.

3NF and BCNF

Third Normal Form (3CNF)

A relational schema R is in 3NF if for every functional dependency $\mathbf{X} \rightarrow A$ either

- $A \in \mathbf{X}$, or
- $\mathbf{X}$ is a superkey for R, or
- A is a member of some key.

Boyce-Codd Normal Form (BCNF)

A relational schema R is in BCNF if for every functional dependency $\mathbf{X} \rightarrow A$ either

- $A \in \mathbf{X}$, or
- $\mathbf{X}$ is a superkey for R.

Inclusions

Clearly BCNF $\subseteq$ 3NF $\subseteq$ $2NF$. These are proper inclusions:

In 2NF, but not 3NF

$R(A,\ B,\ C)$, with $F = \{A \rightarrow B,\ B \rightarrow C\}$.

In 3NF, but not BCNF

$R(A,\ B,\ C)$, with $F = \{A, B \rightarrow C,\ C \rightarrow B\}$.

- This is in 3NF since AB and AC are keys, so there are no non-prime attributes
- But not in BCNF since C is not a key and we have $C \rightarrow B$.

The Plan

Given a relational schema $R(\mathbf{X})$ with FDs F :

- Reason about FDs
  - Is F missing FDs that are logically implied by those in F?
- Decompose each $R(\mathbf{X})$ into smaller $R_1(\mathbf{X}_1)$, $R_2(\mathbf{X}_2)$, $\cdots R_k(\mathbf{X}_k)$, where each $R_i(\mathbf{X}_i)$ is in the desired Normal Form.

Are some decompositions better than others?

Desired properties of any decomposition

Lossless-join decomposition

A decomposition of schema $R(\mathbf{X})$ to $S(\mathbf{Y} \cup \mathbf{Z})$ and $T(\mathbf{Y} \cup (\mathbf{X} - \mathbf{Z}))$ is a lossless-join decomposition if for every database instances we have $R = S \bowtie T$.

Dependency preserving decomposition

A decomposition of schema $R(\mathbf{X})$ to $S(\mathbf{Y} \cup \mathbf{Z})$ and $T(\mathbf{Y} \cup (\mathbf{X} - \mathbf{Z}))$ is dependency preserving, if enforcing FDs on S and T individually has the same effect as enforcing all FDs on $S \bowtie T$.

We will see that it is not always possible to achieve both of these goals.

Lecture 06: Reasoning about FDs

Outline

- Implied dependencies (closure)
- Armstrong's Axioms

Semantic Closure

Notation

$$F \models \mathbf{Y} \rightarrow \mathbf{Z}$$

means that any database instance that that satisfies every FD of F, must also satisfy $\mathbf{Y} \rightarrow \mathbf{Z}$.

The semantic closure of F, denoted F^+, is defined to be

$$F^+ = \{\mathbf{Y} \rightarrow \mathbf{Z} \mid \mathbf{Y} \cup \mathbf{Z} \subseteq \text{atts}(F) and \wedge F \models \mathbf{Y} \rightarrow \mathbf{Z}\}.$$

The membership problem is to determine if $\mathbf{Y} \rightarrow \mathbf{Z} \in F^+$.

Reasoning about Functional Dependencies

We write $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$ when $\mathbf{Y} \rightarrow \mathbf{Z}$ can be derived from F via the following rules.

Armstrong's Axioms

Reflexivity If $\mathbf{Z} \subseteq \mathbf{Y}$, then $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$.

Augmentation If $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$ then $F \vdash \mathbf{Y}, \mathbf{W} \rightarrow \mathbf{Z}, \mathbf{W}$.

Transitivity If $F \vdash \mathbf{Y} \rightarrow \mathbf{Z}$ and $F \models \mathbf{Z} \rightarrow \mathbf{W}$, then $F \vdash \mathbf{Y} \rightarrow \mathbf{W}$.

Logical Closure (of a set of attributes)

Notation

$$\text{closure}(F, \mathbf{X}) = \{A \mid F \vdash \mathbf{X} \to A\}$$

Claim 1

If $\mathbf{Y} \to \mathbf{W} \in F$ and $\mathbf{Y} \subseteq \text{closure}(F, \mathbf{X})$, then $\mathbf{W} \subseteq \text{closure}(F, \mathbf{X})$.

Claim 2

$\mathbf{Y} \to \mathbf{W} \in F^+$ if and only if $\mathbf{W} \subseteq \text{closure}(F, \mathbf{Y})$.

Soundness and Completeness

Soundness

$$F \vdash f \implies f \in F^+$$

Completeness

$$f \in F^+ \implies F \vdash f$$

Proof of Completeness (soundness left as an exercise)

Show $\neg(F \vdash f) \implies \neg(F \models f)$:

- Suppose $\neg(F \vdash \mathbf{Y} \rightarrow \mathbf{Z})$ for $R(\mathbf{X})$.
- Let $\mathbf{Y}^{+} = \text{closure}(F,\ \mathbf{Y})$.
- $\exists B \in \mathbf{Z}$, with $B \notin \mathbf{Y}^{+}$.
- Construct an instance of R with just two records, u and v, that agree on $\mathbf{Y}^{+}$ but not on $\mathbf{X} - \mathbf{Y}^{+}$.
- By construction, this instance does not satisfy $\mathbf{Y} \rightarrow \mathbf{Z}$.
- But it does satisfy F! Why?
  - let $\mathbf{S} \rightarrow \mathbf{T}$ be any FD in F, with $u.[\mathbf{S}] = v.[\mathbf{S}]$.
  - So $\mathbf{S} \subseteq \mathbf{Y}+$. and so $\mathbf{T} \subseteq \mathbf{Y}+$ by claim 1,
  - and so $u.[T] = v.[T]$

Consequences of Armstrong's Axioms

Union If $F \models \mathbf{Y} \rightarrow \mathbf{Z}$ and $F \models \mathbf{Y} \rightarrow \mathbf{W}$, then $F \models \mathbf{Y} \rightarrow \mathbf{W}, \mathbf{Z}$.

Pseudo-transitivity If $F \models \mathbf{Y} \rightarrow \mathbf{Z}$ and $F \models \mathbf{U}, \mathbf{Z} \rightarrow \mathbf{W}$, then $F \models \mathbf{Y}, \mathbf{U} \rightarrow \mathbf{W}$.

Decomposition If $F \models \mathbf{Y} \rightarrow \mathbf{Z}$ and $\mathbf{W} \subseteq \mathbf{Z}$, then $F \models \mathbf{Y} \rightarrow \mathbf{W}$.

Exercise : Prove these using Armstrong's axioms!

Proof of the Union Rule

Suppose we have

$$F \models \mathbf{Y} \rightarrow \mathbf{Z},$$
$$F \models \mathbf{Y} \rightarrow \mathbf{W}.$$

By augmentation we have

$$F \models \mathbf{Y}, \mathbf{Y} \rightarrow \mathbf{Y}, \mathbf{Z},$$

that is,

$$F \models \mathbf{Y} \rightarrow \mathbf{Y}, \mathbf{Z}.$$

Also using augmentation we obtain

$$F \models \mathbf{Y}, \mathbf{Z} \rightarrow \mathbf{W}, \mathbf{Z}.$$

Therefore, by transitivity we obtain

$$F \models \mathbf{Y} \rightarrow \mathbf{W}, \mathbf{Z}.$$

Example application of functional reasoning.

Heath's Rule

Suppose $R(A,\ B,\ C)$ is a relational schema with functional dependency $A \rightarrow B$, then

$$R = \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R).$$

Proof of Heath's Rule

We first show that $R \subseteq \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R)$.

- If $u = (a,\ b,\ c) \in R$, then $u_1 = (a,\ b) \in \pi_{A,B}(R)$ and $u_2 = (a,\ c) \in \pi_{A,C}(R)$.
- Since $\{(a,\ b)\} \bowtie_A \{(a,\ c)\} = \{(a,\ b,\ c)\}$ we know $u \in \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R)$.

In the other direction we must show $R' = \pi_{A,B}(R) \bowtie_A \pi_{A,C}(R) \subseteq R$.

- If $u = (a,\ b,\ c) \in R'$, then there must exist tuples $u_1 = (a,\ b) \in \pi_{A,B}(R)$ and $u_2 = (a,\ c) \in \pi_{A,C}(R)$.
- This means that there must exist a $u' = (a,\ b',\ c) \in R$ such that $u_2 = \pi_{A,C}(\{(a,\ b',\ c)\})$.
- However, the functional dependency tells us that $b = b'$, so $u = (a,\ b,\ c) \in R$.

Closure Example

$R(A, B, C, D, D, F)$ with

$$A, B \rightarrow C$$
$$B, C \rightarrow D$$
$$D \rightarrow E$$
$$C, F \rightarrow B$$

What is the closure of $\{A, B\}$?

$$\{A, B\} \stackrel{A,B \rightarrow C}{\Longrightarrow} \{A, B, C\}$$
$$\stackrel{B,C \rightarrow D}{\Longrightarrow} \{A, B, C, D\}$$
$$\stackrel{D \rightarrow E}{\Longrightarrow} \{A, B, C, D, E\}$$

So $\{A, B\}^+ = \{A, B, C, D, E\}$ and $A, B \rightarrow C, D, E$.