Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

Assignment 3

Database Design

Question 1

Consider a relation schema R(A, B, C, D, E), and the set of functional dependencies.

$$F = \{A \rightarrow BC \\ CD \rightarrow E \\ B \rightarrow D \\ E \rightarrow A\}$$

Find all candidate keys (i.e., minimal keys) of relation R. Show all the steps you took to derive each key, and clearly state which of Armstrong's axioms are used in each step.

Attributes on LHS = {A, B, C, D, E}, Find closure on each attribute

A^+:

⇒ by identity

$A \rightarrow A$

⇒ Union with $A \rightarrow BC$

$A \rightarrow ABC$

⇒ $A \rightarrow B$ (by decomposition) and $B \rightarrow D$, then $A \rightarrow D$ by transitivity

⇒ Union with $A \rightarrow ABC$

$A \rightarrow ABCD$

⇒ $A \rightarrow CD$ (by decomposition) and $CD \rightarrow E$, then $A \rightarrow E$ by transitivity

⇒ Union with $A \rightarrow ABCD$

$A \rightarrow ABCDE$

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

B⁺:

⇒ by identity

B → B

⇒ Union with B → D

B → BD

C⁺:

⇒ by identity

C → C

D⁺:

⇒ by identity

D → D

E⁺:

⇒ by identity

E → E

⇒ Union with E → A

E → AE

⇒ E → A (by decomposition) and A → ABCDE (from proof 1), then E → ABCDE by transitivity

⇒ Union with E → AE

E → ABCDE

Ignore any supersets of A or E since alone they are candidate keys, any supersets will result in super keys.

BC⁺:

⇒ by identity

BC → BC

⇒ BC → B (decomposition), B → D, then BC → D by transitivity

⇒ Union with BC → BC

BC → BCD

⇒ BC → CD (by decomposition) and CD → E, then BC → E by transitivity

⇒ Union with BC → BCD

BC → BCDE

⇒ BC → E (by decomposition) and E → A, then BC → A by transitivity

⇒ Union with BC → BCDE

BC → ABCDE

BD^+:

⇒ by identity

BD → BD

CD^+:

⇒ by identity

CD → CD

⇒ Union with CD → E

CD → CDE

⇒ CD → E (decomposition) and E → ABCDE (by proof above), then CD → ABCDE by transitivity

⇒ Union with CD → CDE

CD → ABCDE

Ignore any supersets of CD or BC since alone they are candidate keys, any supersets will result in super keys.

Any other combination will result in a superset of a candidate key; therefore, we can stop here.

{A, E, BC, CD} are the candidate keys for this relation R.

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

Question 2

Consider the relation R(A,B,C,D,E), and the decomposition of R into R1(ABC) and R2(ADE).

(a) Give a set of functional dependencies (FDs) such that the decomposition into R1 and R2 is lossless join and dependency preserving. Show your work and explain why your FDs satisfy the criteria.

R(A,B,C,D,E)

F = (A → BC, D → AE) where,

R1 = (ABC) and R2 = (ADE)

F1 = {A → BC} and F2 = {D → AE}

Prove lossless decomposition:

R1 ∩ R2 = {A} and A^+ = ABC therefore it determines all attributes in R1. A is a candidate key of R1 which makes decomposition lossless.

Prove dependency preserving:

F1 U F2 = F. Therefore, dependency preserving holds.

(b) Give a set of functional dependencies such that the decomposition into R1 and R2 is not lossless join, but dependency preserving. Show your work and explain why your FDs satisfy the criteria.

R(A,B,C,D,E)

F = (C → AB, D → AE) where,

R1 = (ABC) and R2 = (ADE)

F1 = {C → AB} and F2 = {D → AE}

Prove lossless decomposition:

R1 ∩ R2 = {A} and A^+ = A, meaning A is neither a candidate key for R1 or R2 -> decomposition is lossy

Prove dependency preserving:

F1 U F2 = F. Therefore, dependency preserving holds.

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

Question 3

The Department has a database containing information about all lectures during a term. The Schedule relation has the following schema:

Schedule (C, D, T, R, P, A)

where C represents a course, D for day (weekday), T for time, R for room, P for professor, and A for head TA. The set of functional dependencies F defined over Schedule are:

F = {RDT → P, RDT → C, C → A, PDT → R, P DT → C, CDT → P, CDT → R}

a) Is the FD RDT → AP entailed by F ? Explain and show your work with references to Armstrong's axioms.

⇒ by identity

RDT → RDT

⇒ Union with RDT → P and RDT → C

RDT → RDTPC

⇒ RDT → C (by decomposition) and C → A, then RDT → A by transitivity

⇒ Union with E → RDTPC

RDT → RDTPCA

⇒ RDT → AP (by decomposition)

RDT → AP

Therefore RDT → AP is entailed by F.

b) Find all the key(s) of relation Schedule. Show your work (i.e., how each key is derived).

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

Attributes on LHS = {C, RDT, PDT, CDT}, Find closure on each attribute

C^+ = CA

Using (C → A)

RDT^+ = RDTPCA

Using (RDT → P, RDT → C, C → A)

PDT^+ = PDTRCA

Using (PDT → R, PDT → C, C → A)

CDT^+ = CDTPRA

Using (CDT → P, CDT → R, C → A)

Any subset of CDTRPA which is also a subset of CDT, PDT, RDT (i.e., combinations of 1 or 2) will not give a candidate key since we don't have LHS FD's that will help.

Any superset will result in a super key so we can stop here, those are the only candidate keys

{RDT, PDT, CDT} are the candidate keys for this relation R.

c) Is Schedule in BCNF? If not, decompose it into smaller relations that are each in BCNF. Show your work at each step.

Prime attributes: CDTRP

C → A violates 2NF which means it **violates BCNF**

C^+ = CA → R1 = {C, A} and F1 = { C → A } **(BCNF holds)**

Candidate key: C

R2 = {C,D,T,R,P} and

F2 = { RDT → P, RDT → C, PDT → R, P DT → C, CDT → P, CDT → R} **(BCNF holds)**

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

R gets broken down into R1 = {C, A} with F1 = { C → A } and R2 = {C,D,T,R,P} with F2 = { RDT → P, RDT → C, PDT → R, P DT → C, CDT → P, CDT → R}

d) We can create a new relation ProfsSchedule(D, T, P) by projecting some attributes of Schedule. Are there new functional dependencies that hold over ProfsSchedule? If so, state these FDs. If not, state why not. In both cases, show your work at each step to justify your answer.

D^+ = {D}

T^+ = {T}

P^+ = {P}

DT^+ = {DT}

DP^+ = {DP}

TP^+ = {TP}

DTP^+ = {DTPRCA} using (PDT → R, PDT → C, C → A)

Since RCA attributes are not in ProfsSchedule we can ignore, no new FDS hold in ProfsSchedule

e) Find a minimal cover Fmin for F . Show all the steps in your derivation of Fmin

1st Step:

H = { RDT → P, RDT → C, C → A, PDT → R, PDT → C, CDT → P, CDT → R}

2cnd Step:

- RDT → P can be removed, RDT^+ = RDTCAP
- RDT → C cant be removed RDT^+ = RDT
- C → A cant be removed C^+ = C
- PDT → R can be removed PDT^+ = PDTCAR
- PDT → C cant be removed PDT^+ = PDT
- CDT → P cant be removed CDT^+ = CDTRA
- CDT → R cant be removed CDT^+ = CDTPA

H = { RDT → C, C → A, PDT → C, CDT → P, CDT → R}

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

3rd step:

RDT:	PDT:	CDT:
$R^+ = R$	$P^+ = R$	$C^+ = CA$
$D^+ = D$	$D^+ = D$	$D^+ = D$
$T^+ = T$	$T^+ = T$	$T^+ = T$

No redundancy therefore doesn't change.

4th step:

H doesn't change

Min cover = H

H = { RDT → C, C → A, PDT → C, CDT → P, CDT → R}

Mahad Ahmed
Ahmem73
400378176
Dec 1^{st}, 2023

Question 4

Prove the following using Armstrong's axioms (using only the axioms presented in class). Show all the steps of your proof and indicate which of Armstrong's axioms is applied in each step.

a) Consider the schema R(A,B,C,D,E,F), and the following functional dependencies: A → BCD, BC → DE, B → D, D → A. Show that AF is a superkey.

⇒ by identity

AF → AF

⇒ AF → A (decomposition) and A → BCD, then AF → BCD (transitivity)

⇒ Union with AF → AF

AF → ABCDF

⇒ AF → BC (decomposition) and BC → DE, then AF → DE (transitivity)

⇒ Union with AF → ABCDF

AF → ABCDEF

Therefore, AF is a super key

b) Given the relational schema R(A, B, C, D, E, F) and the FDs F1: {AB → C, A → D, CD → EF }. Show that AB → F.

⇒ by identity

AB → AB

⇒ Union with AB → C

AB → ABC

⇒ AB → A (decomposition) and A → D, then AB → D (transitivity)

⇒ Union with AB → ABC

AB → ABCD

⇒ AB → CD (decomposition) and CD → EF, then AB → EF (transitivity)

⇒ Union with AB → ABCD

AB → ABCDEF

⇒ Decomposition

AB → F

Therefore, AB → F

Transactions

Question 5

Consider schedules S1, S2 below. State which of the following properties hold (or not) for each schedule: strict, avoids cascading aborts, recoverability. Provide a brief justification for each answer.

a) S1: R1(X); R2(Z); R1(Z); R3(X); R3(Y); W1(X); C1; W3(Y); C3; R2(Y); W2(Z); W2(Y); C2
   Strict: Yes, When T1 writes it commits and when T3 writes it commits no uncommitted item is read or overwritten
   ACA: Yes, Since Strict
   Recoverable: Yes, Since Strict

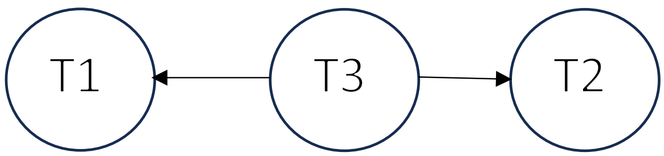
b) S2: R1(X); R2(Z); R3(X); R1(Z); R2(Y); R3(Y); W1(X); W2(Z); W3(Y); W2(Y); C3; C1; C2
   Strict: No, T2 overwrites Y after T3 writes to Y (T3 does not commit before T2 writes)
   ACA: Yes, No Xact reads data when an Xact writes to it – no dependency
   Recoverable: Yes, Since ACA

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

Question 6

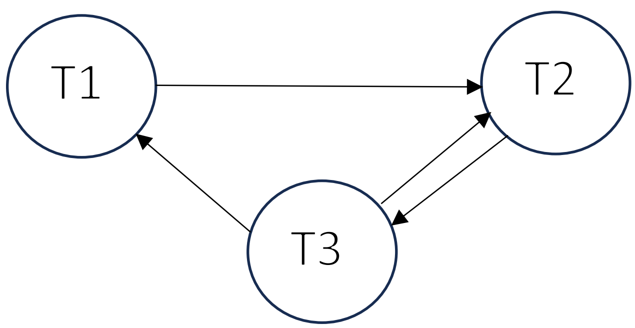
a) Consider the schedules S1 and S2 given below. Draw the serializability (precedence) graphs for S1 and S2 and state whether each schedule is serializable or not. Provide justification to explain your answer. If a schedule is serializable, write down the equivalent serial schedule(s), i.e., (T2, T1, T3), where Ti includes all actions for transaction i.

R1(X), R1(Y), W1(Y), R2(Y), Commit2, W3(Y), W3(Z), W1(Z), Commit1, Commit3



Schedule is serializable since there is no cycle. Equivalent serial schedule: T3T2T1

R1(X); R2(Z); R3(X); R1(Z); R2(Y); R3(Y); W1(X); W2(Z); W3(Y); W2(Y)

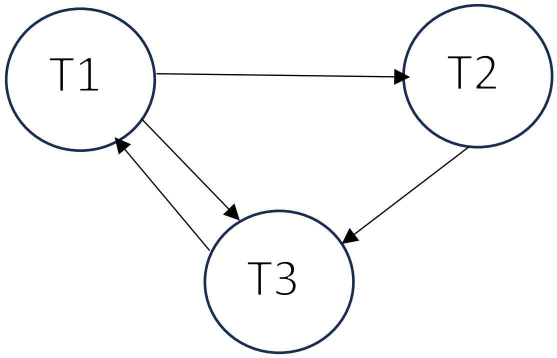


Schedule is not serializable since there is a cycle.

b) Consider schedules S3, S4 below. If a commit or abort is not shown, assume that commit/abort must follow all the listed actions of that transaction (not necessarily immediately). For simplicity, we assume the listed transactions are the only ones active currently in the database. State which of the following properties hold (or not) for each schedule: conflict serializable, avoids cascading aborts, recoverability, 2PL. If you cannot determine whether a schedule satisfies a property, state "Undetermined". Provide a brief justification for each of your answers, for each property.

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

S3: R1(X), R1(Y), W1(Y), R2(Y), Commit2, W3(Y), W3(Z), W1(Z), Commit1, Commit3



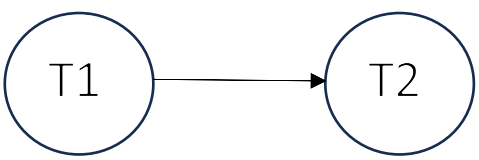
Conflict serializable: No, cycle in precedence graph

ACA: No, T2 reads Y before T1 commits after writing to Y

Recoverability: No, T2 depends on T1 but T2 commits first (ie. T2 reads from T1's writes)

2PL: No, T1 gets XLock on Z, so T3 gets denied writing to Z

S4: W1(Y), R2(Y), R1(X), R2(X)



Conflict serializable: Yes, Acyclic Precedence Graph

ACA: No, T2 reads Y before T1 commits after writing to Y

Recoverability: Undetermined, don't know the order of commits

2PL: Yes, T1 releases its Xlock on Y after writing to it, which than T2 can then acquire

Mahad Ahmed
Ahmem73
400378176
Dec 1st, 2023

Question 7

Consider the following locking protocol: Before a transaction T writes a data object A, T has to obtain an exclusive lock on A. For a transaction T, we hold these exclusive locks until the end of the transaction. If a transaction T reads a data object A, no lock on A is obtained. State which of the following properties are ensured by this locking protocol: serializability, conflict-serializability, recoverability, avoids cascading aborts, avoids deadlock. Justify your answer for each property.

Serializability: No, it does not require an object to be locked when attempting to read. So another Xact can read and then write to the object which will make it non serial

Conflict serializability: No we can come up with an example that follows the locking protocol:

R1(A); R2(A); X1(A); W1(A); R2(A)

This example follows the locking protocol above and has a cycle in its precedence graph showing its not conflict serializability.

Recoverability: Since the protocol isn't conflict serializable, we can conclude its also not recoverable

ACA: since the protocol isn't recoverable, we can conclude its also not ACA

Avoids deadlock: No, because of the following counter example:

X1(A) W1(A) X2(B) W2(B) X1(B) W1(B) X2(A) W2(A)

This falls into a deadlock.