

Sometimes a good explanation that shows a deeper understanding or reasonable reinterpretation can mitigate or eliminate the point loss of an incorrect answer

I will not use a source other than my brain on this exam: _____ (please sign)

1. (4 pts) Consider the relation $R(A, B, C, D, E, F)$. You are given the following functional dependencies:

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

List all the minimal keys for R. **D, F does not appear on RHS of any FD; must be part of any key, but $\{D, F\} \rightarrow \{D, F\}$**

D, F, C and D, F, A, B

**$\{D, F, C\} \rightarrow \{D, F, C, E, A, B\}$ i.e., D, F, C is a key
 $\{D, F, A, B\} \rightarrow \{D, F, A, B, C, E\}$ i.e., D, F, A, B is a key**

2. (3 pts) Consider the relation $R(A, B, C, D, E, F, G)$. You are given the following functional dependencies:

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

Can LHS of any FD be simplified? Yes

A, B, C CAN BE simplified; A can be inferred from B, C ($B, C \rightarrow G$; $G \rightarrow A$)
 so replace, $A, B, C \rightarrow D$ with $B, C \rightarrow D$

**$B, C \rightarrow D$ or $B, C \rightarrow D, E, G$
 $B, C \rightarrow E, G$ $G \rightarrow A$
 $G \rightarrow A$**

Give a minimal set of these FDs. If the set is already minimal, say so.

3. (3 pts) Consider the relation $R(A, B, C, D, E, F)$. You are given the following functional dependencies:

$A \rightarrow B$

$B \rightarrow A$

$C \rightarrow D$

$C, F \rightarrow E$

$D \rightarrow E$

$E \rightarrow F$

Can LHS of any FD be simplified? Yes

C, F CAN BE simplified; F can be inferred from C ($C \rightarrow D$; $D \rightarrow E$; $E \rightarrow F$)

so, replace $C, F \rightarrow E$ with $C \rightarrow E$

But $C \rightarrow E$ is redundant, because $C \rightarrow D \rightarrow E$

$A \rightarrow B$

$B \rightarrow A$

$C \rightarrow D$

~~$C, F \rightarrow E$~~

$D \rightarrow E$

$E \rightarrow F$

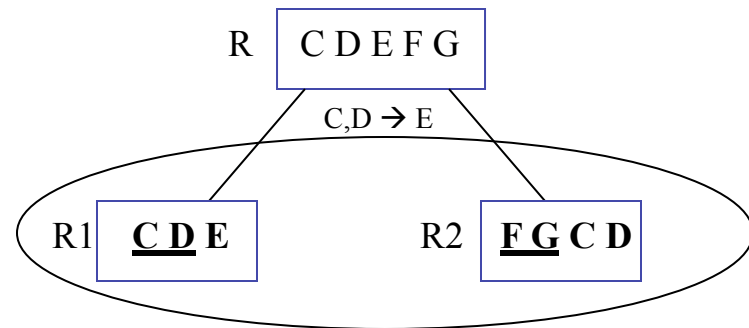
Same as given
 but without
 $CF \rightarrow E$

Give a minimal set of FDs equivalent to the given FDs. If the given FDs are already minimal, then say so.

Consider the relation R(C D E F G), with the following functional dependencies:

$C,D \rightarrow E$
 $F,G \rightarrow C,D$

F, G is the only minimal key of R. Consider this dependency-preserving decomposition into BCNF relations inside the ellipse to the right.



4. (4 pts) Give SQL CREATE TABLE statements that implement R1 and R2 in a way that is consistent with the constraints on the decomposition into R1 and R2 relations above. List the attributes C, D, E, F, G as appropriate in the table definition(s), but don't worry about specifying attribute types. Define PRIMARY KEYs, FOREIGN KEYs, UNIQUEs, and NOT NULLs, as appropriate. If there is more than one way to define the tables that is consistent with the decomposition above, then pick one of these ways (and give a brief explanation if you like).

CREATE TABLE R1 (
 C, D, E,
 PRIMARY KEY (C, D))

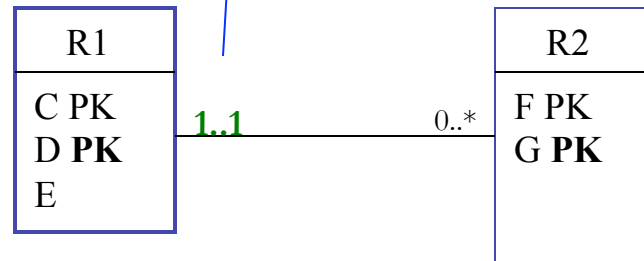
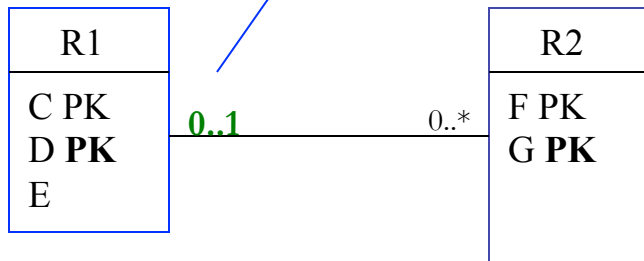
*Some put FK(C,D) REFS R2, but
 C,D isn't the PK of R2*

CREATE TABLE R2 (
 C, D, F, G,
 PRIMARY KEY (F, G),
 FOREIGN KEY (C, D)
 REFERENCES R1 (C,D))

or

CREATE TABLE R2 (
 C NOT NULL, D NOT NULL, F, G,
 PRIMARY KEY (F, G),
 FOREIGN KEY (C, D)
 REFERENCES R1 (C,D))

5. (4 pts) Give a UML snippet that represents R1 and R2 in a way that is consistent with the constraints on the decomposition into R1 and R2 relations above, and is also consistent with your implementation into tables for question 4. List the attributes C, D, E, F, G as appropriate in the class and association definition(s). Include cardinalities on associations. If there is more than one way to define the classes and associations that is consistent with the decomposition above, then pick the representation that is also consistent with your answer to question 4.



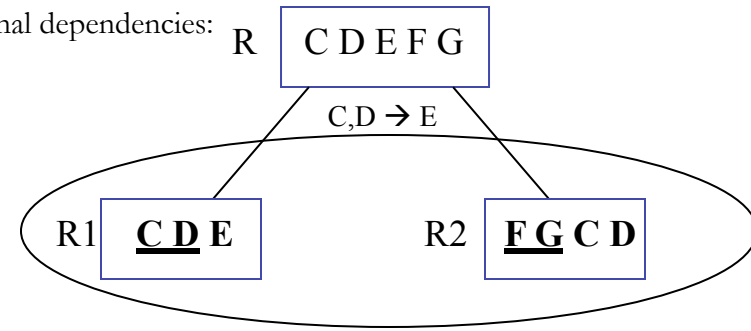
Notice that C,D not shown explicitly in the R2 class – that would be an implicit association; but C,D would be in R2 TABLE translation because of the 0..1 and 1..1 cardinality

Consider the same setup as the previous page. The relation $R(C\ D\ E\ F\ G)$, with functional dependencies:

$C,D \rightarrow E$

$F,G \rightarrow C,D$

F, G is the only minimal key of R . Consider this dependency-preserving decomposition into BCNF relations inside the ellipse to the right.



6. (1 pt) Explain why this decomposition is dependency preserving

All FDs can be assigned to a relation in the decomposition; $CD \rightarrow E$ can be assigned to R_1 ; $F, G \rightarrow C, D$ can be assigned to R_2

7. (1 pt) Explain why R_1 and R_2 are each in BCNF

The lefthand side of every FD assigned to a relation is a key for that relation; CD (of $CD \rightarrow E$) is a key of R_1 ;
 F, G (of $F, G \rightarrow C, D$) is a key of R_2

8. (5 pts) Circle all correct options.

- (a) Hash indexes are ideal for speeding up queries that involve range conditions of the form $T.a < x$ or $T.B \geq 25$ *Hash indexes really only help with equality conditions*
- (b) Lookup using balanced search tree indexes, like B+ trees, is logarithmic in the number of leaves in the tree (or tuples in the indexed table)
- (c) In strict 2 phase locking, locks made by a transaction are only released when the transaction commits
- (d) Indexes come with no cost to database response time performance *If you picked this one, you didn't lose points, because you might have very reasonably interpreted this as a "query-time only" response time, but response times during insertions, deletions, and updates generally increase with indexes.*
- (e) The ACID property that is enabled by the SQL ROLLBACK command is Atomicity
- (f) Serializability implements the ACID property of Isolation
- (g) Confidence is the ACID property most concerned with data security *Consistency is the 'C' in ACID*

9. (5 pts) Answer the following about Assignments 1-3

(a) **Assignment 1's** analysis of the paper “*Does Providing Dormitory Residents with Feedback on Energy and Water Usage Lead to Reduced Consumption*”, asked you to consider *the environmental and operational implications of residential technologies that require energy and/or water* (e.g., appliances, computers, lights, water heating and pressure), through the lens of which tenet of the IEEE Code of Conduct?

_____ **Tenet 5** _____ (see Rubric for Assignment 1) See extra credit question at back for list of tenets

(b) **Assignment 1's** analysis of the paper “*Does Providing Dormitory Residents with Feedback on Energy and Water Usage Lead to Reduced Consumption*”, asked you to consider *if and how human behavior is changed through information technology*, through the lens of which tenet of the IEEE Code of Conduct?

_____ **Tenet 5** _____ (see Rubric for Assignment 1)

(c) **Assignment 2's** reading by Don Norman, “*Human Centered Design Considered Harmful*”, says

“The individual is a moving target. Design for the individual of today, and the design will be wrong tomorrow. Indeed, the more successful the product, the more that it will no longer be appropriate.”

Circle the possible database design scenarios in which we tried to anticipate changes, which “clients” might not anticipate

- (i) Electrical-plug-level granularity of measurement in building energy monitoring
- (ii) Recursive definitions of sub-regions to indeterminate depth to support a Rosling-like visualization

(d) In **Assignment 3**, we read “*Sustainable Interaction Design*” by Blevis, and we speculated that in the Internet of Things, data stored in the cloud about individual “things” could imbue these “things” with (circle one) – haughty | heirloom | cool | dominant – status, which would encourage retention of these things (rather than shipping them to the junkyard).

10. (5 pts) Consider the following two transactions, T1 and T2:

T1: Read(A), $\text{Op}_{11}(\text{A})$, Write(A), Read(B), $\text{Op}_{12}(\text{B})$, Write(B), Commit

T2: Read(A), $\text{Op}_{21}(\text{A})$, Write(A), Read(B), $\text{Op}_{22}(\text{B})$, Write(B), Commit

(a) Give a schedule of these two transactions, just showing disk reads (i.e., $R(A)$ and $R(B)$) and writes (i.e., $W(A)$ and $W(B)$) that is guaranteed of satisfying the *serializability* constraint (i.e., that the execution of the schedule is equivalent to some sequential (serial) order of T_1 and T_2 , even without knowing details of the operations performed on A and B are performed). (b) Also show a schedule that does not satisfy the serializability constraint. In each case below, the schedule has been started for you. Complete each schedule, being very clear about the temporal order in which the various reads and writes across transactions are executed.

Serializable Schedule

T1	T2
	R(A) W(A)
See next page	

See next page

Non-Serializable Schedule

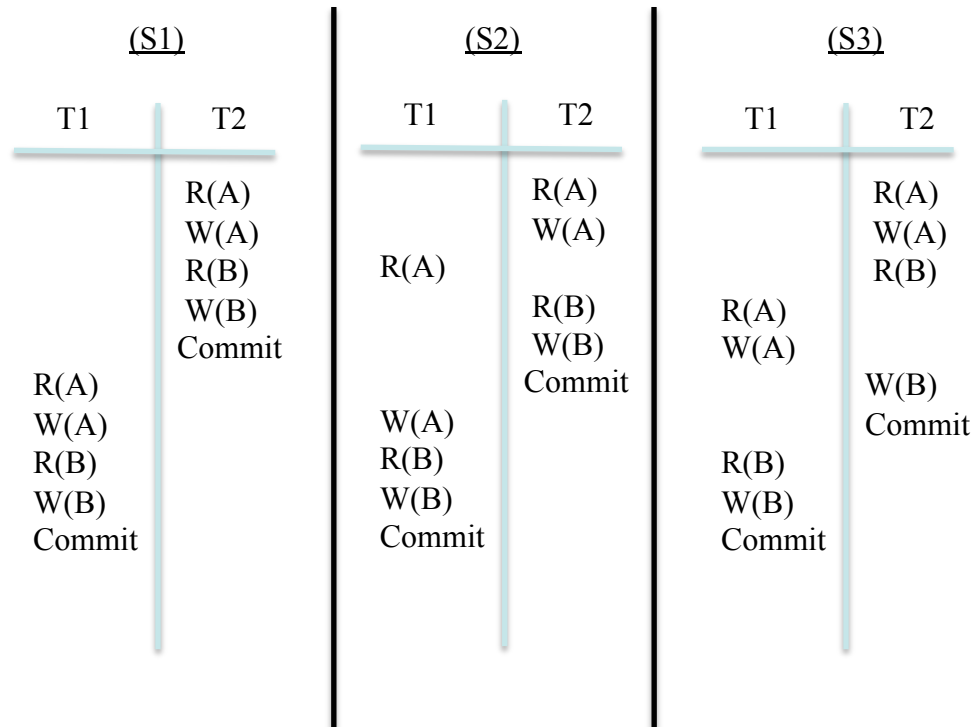
T1	T2
R(A)	
W(A)	

10. (5 pts) Consider the following two transactions, T1 and T2:

T1: Read(A), Op₁₁(A), Write(A), Read(B), Op₁₂(B), Write(B), Commit

T2: Read(A), Op₂₁(A), Write(A), Read(B), Op₂₂(B), Write(B), Commit

Examples of schedules exhibiting serializability and beginning with the initially given sequence in T2; each would be correct for the first answer



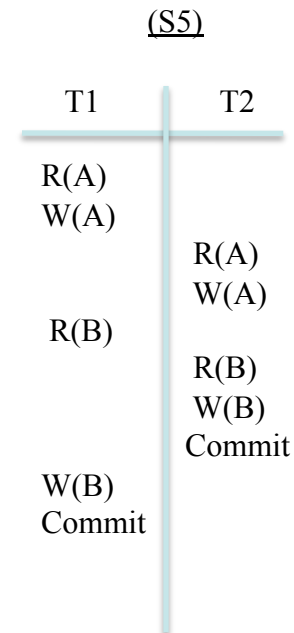
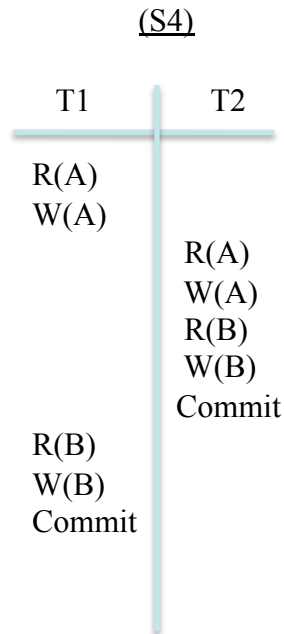
I was forgiving of missing Commits (since the question did not assume 2PL, but more flexible, albeit unnamed, locking and unlocking protocols, their placement wasn't as important as they would have under strict 2PL)

10. (5 pts) Consider the following two transactions, T1 and T2:

T1: Read(A), Op₁₁(A), Write(A), Read(B), Op₁₂(B), Write(B), Commit

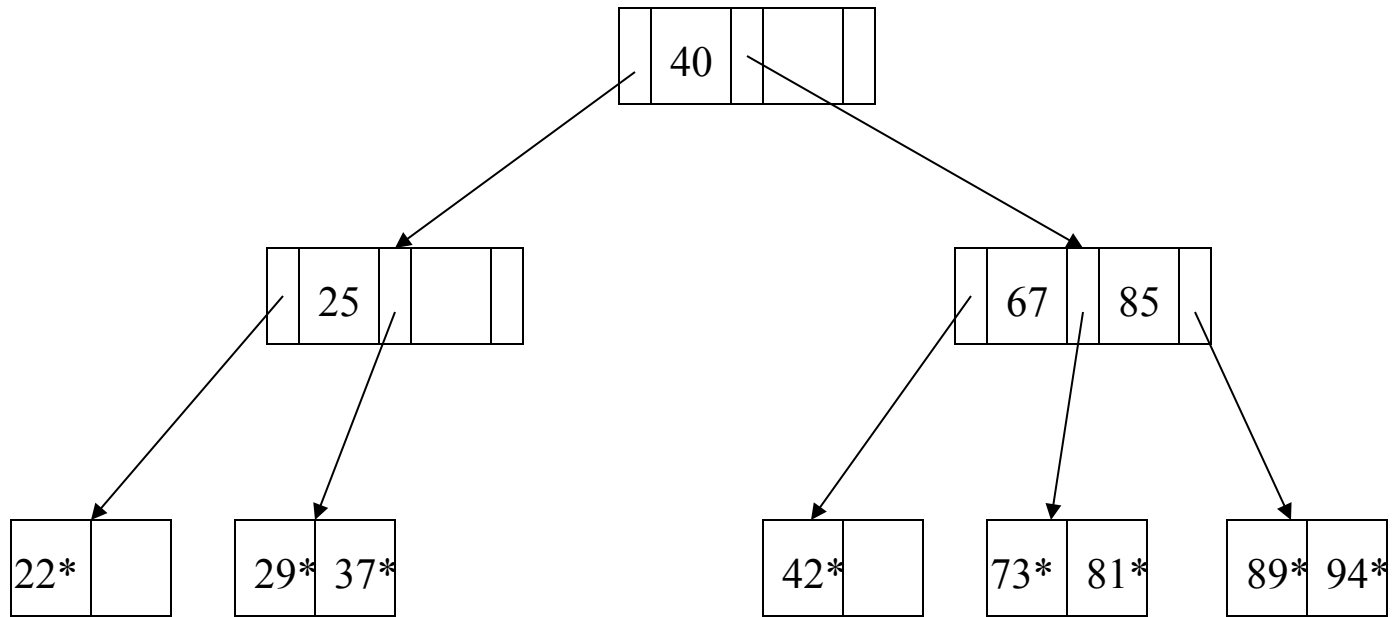
T2: Read(A), Op₂₁(A), Write(A), Read(B), Op₂₂(B), Write(B), Commit

A NON-serializable that begins with the initially given sequence in T1;
not many of them that would not violate locking protocols



Reordering the operations in a given transaction also only received partial credit, because
This amounts to creating a different transaction than the one you were given

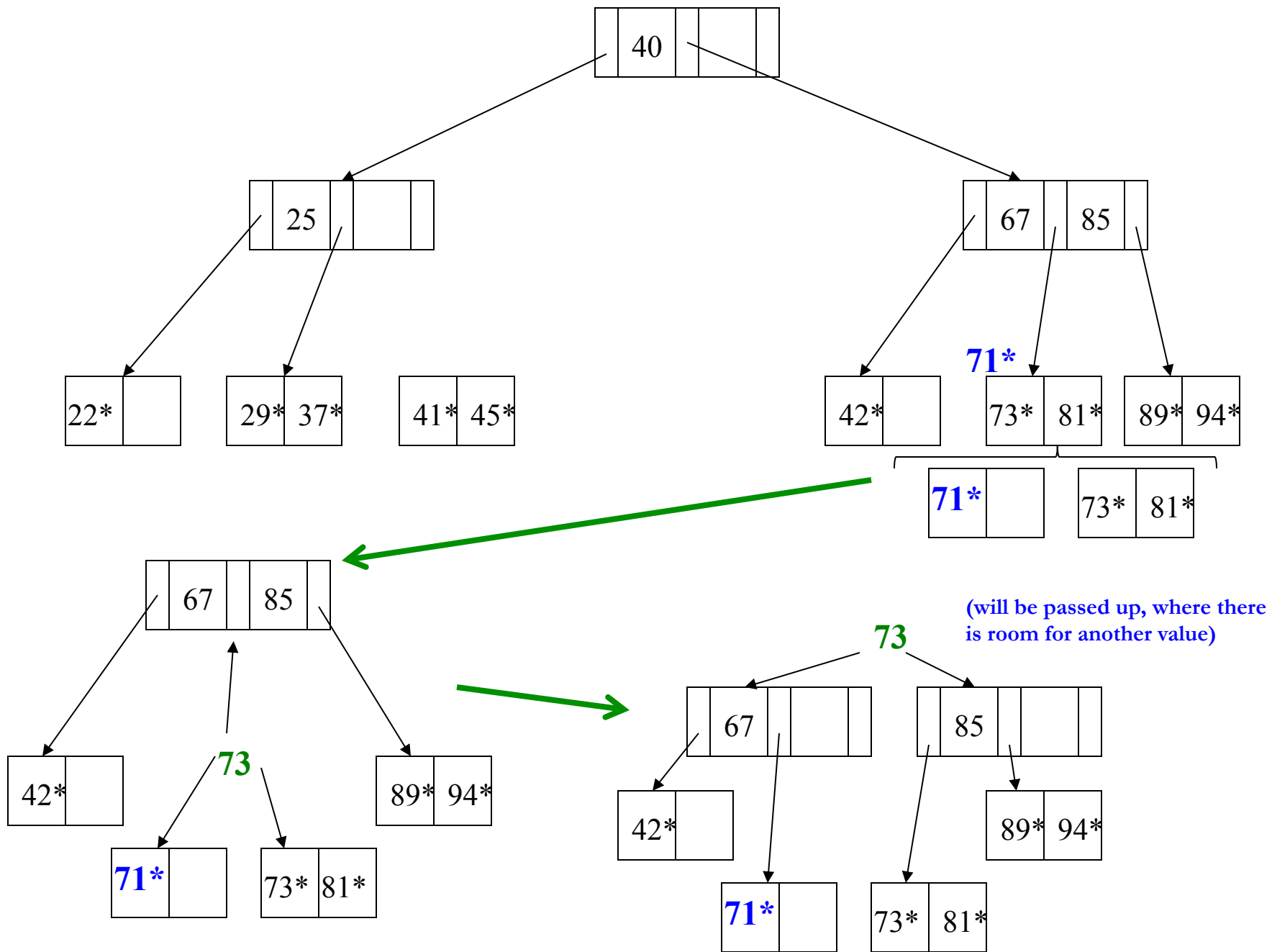
11. (5 pts) Consider the B+ tree below.

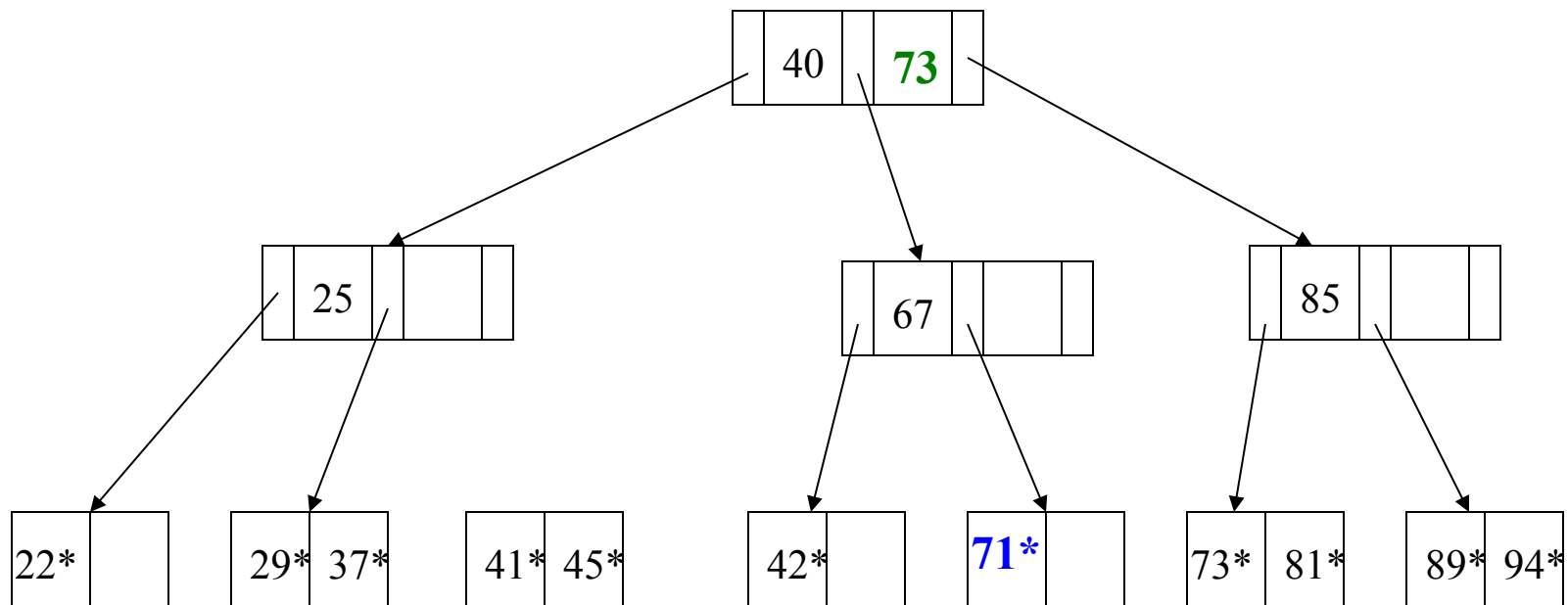


Note that this tree does not show data nodes, and you do not need to see the data nodes to answer this question. At each leaf, N^* is an index of the form $\langle N, \langle \text{page id, slot \#} \rangle \rangle$, where N is the value of the search key.

Show the tree that results from inserting a record with search key **71**, using the insertion procedure described on video, in class, and practice exercises. If you can do so clearly and unambiguously, then you can circle and label sub-trees in this diagram that do not change and use those labels in your answer on the next page.

answer on next page



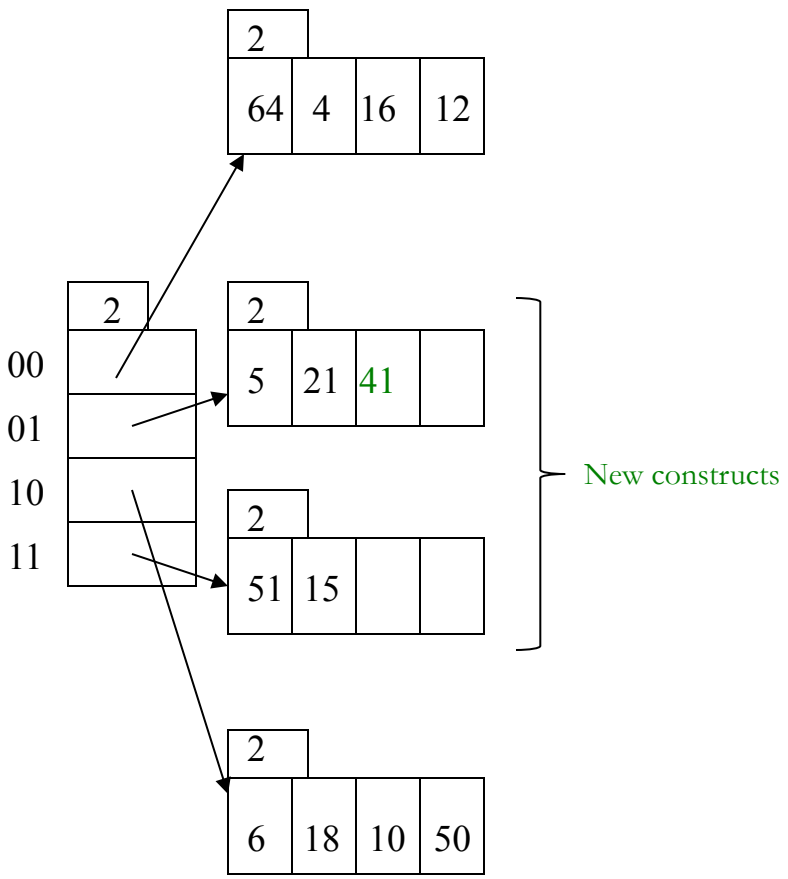
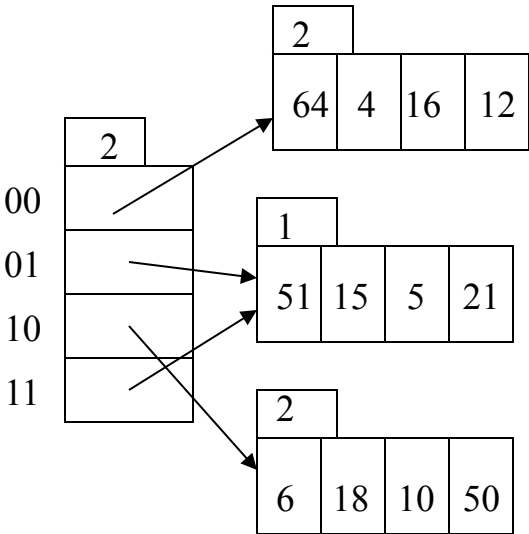


Only correct tree

-2 if 71 or 73 does not appear at leaf

-2 for any tree that isn't 3 levels. Use discretion on partial credit.

12. (5 pts) Consider the extendible hash table to the left. Assume $\text{Hash}(x) = x$. Show the result of inserting **41**



Answer here

EXTRA CREDIT (3 pts) Consider the following table definitions:

CREATE TABLE RelA (Aid integer, a1 integer, a2 integer, PRIMARY KEY (Aid))

CREATE TABLE RelB (Aid integer, Cid integer, b1 integer,
PRIMARY KEY (Aid, Cid, b1),
FOREIGN KEY (Aid) REFERENCES RelA,
FOREIGN KEY (Cid) REFERENCES RelC)

CREATE TABLE RelC (Cid integer, c1 integer, c2 integer, c3 integer, PRIMARY KEY (Cid))

A Cid may participate with many Aids, through RelB, Temp1 finds the Cids that participate with q, Temp2 finds the Cids that participate with r. The c1 values of Cids that participate with both q and r are returned.

Circle all queries below that are equivalent to: $\pi_{c1} ((\text{Temp1} \cap \text{Temp2}) \bowtie \text{RelC})$
where $\text{Temp1} = \pi_{\text{Cid}} ((\sigma_{a2=q} \text{RelA}) \bowtie \text{RelB})$ and
 $\text{Temp2} = \pi_{\text{Cid}} ((\sigma_{a2=r} \text{RelA}) \bowtie \text{RelB})$

1 pt per correct pick,
-1 pt per incorrect

By equivalent, we mean produces the same output given the same input (and we are not referring to efficiency or elegance)

(a) SELECT DISTINCT C.c1
FROM RelC C, RelB B1, RelA A1, RelB B2, RelA A2
WHERE C.Cid = B1.Cid AND B1.Aid = A1.Aid AND
C.Cid = B2.Cid AND B2.Aid = A2.Aid AND
A1.a2 = q AND A2.a2 = r

(b)) SELECT DISTINCT C.c1
FROM RelC C, RelB B, RelA A
WHERE C.Cid = B.Cid AND B.Aid = A.Aid AND A.a2 = q
INTERSECT
SELECT DISTINCT C2.c1
FROM RelC C2, RelB B2, RelA A2
WHERE C2.Cid = B2.Cid AND B2.Aid = A2.Aid AND A2.a2 = r

Intersecting on c1 is not
The same as intersecting on
Cid, and getting the c1 values
of the result

(c) SELECT DISTINCT C.c1
FROM RelA A, RelB B, RelC C
WHERE C.Cid = B.Cid AND B.Aid = A.Aid AND A.a2 = q AND
C.Cid IN (SELECT C2.Cid
FROM RelC C2, RelA A2, RelB B2
WHERE C2.Cid = B2.Cid AND
B2.Aid = A2.Aid AND A2.a2 = r)

(d) SELECT DISTINCT C.c1
FROM RelC C
WHERE C.Cid IN ((SELECT B.Cid
FROM RelA A, RelB B
WHERE B.Aid = A.Aid AND A.a2 = q)
INTERSECT
((SELECT B2.Cid
FROM RelA A2, RelB B2
WHERE B2.Aid = A2.Aid AND A2.a2 = r)))

didn't mean to test you
on imbalanced parens
because of imbalanced parens

(e) $\pi_{c1} (\text{Temp1} \bowtie \text{RelB} \bowtie \text{RelC})$
where $\text{Temp1} = (\sigma_{a2=q} \text{RelA}) \cap (\sigma_{a2=r} \text{RelA})$

The intersection is of entire tuples is
will return the empty set when any attributes,
including a2 (q, r) disagree

(f) None of the above

EXTRA CREDIT (2 pts) University IT has contracted with **an outside company** to build a database that will support your university's administrative applications, to include library applications like checkout and lost-book billing. In contrast, your **local university IT group** builds the application software for all administrative applications. In the lost-book billing application, for example, the application software (that is built locally) inserts a record into a view called the ChargesView with a lost book fee of \$100 for each book (or more realistically, the price of the book) that is more than 30 days overdue. It makes these insertions daily, with the expectation that if a book/person lost fee has been entered once, subsequent attempts to insert the same book/person pair will be rejected because (they believe) the insertion would violate a DB constraint that a person can only be charged a lost-book fee for a given book once. They are operating under assumptions that are faulty.

```
CREATE VIEW LostBookView (PersonId, BookId) /* application programmers see this header */  
AS SELECT CO.PersonId, CO.BookId)  
FROM CheckedOut CO  
WHERE DiffDate('NOW', CO.DueDate) > 30 /* Diff Date computes the difference between two dates in terms of days */  
);
```

As noted above, there is a second view called **ChargesView** that is maintained, with inserts into it occurring daily, as books become (and remain) sufficiently overdue.

A monthly bill is sent out that sums the total charges to each patron (like this example, for 'Doug' only – the real application uses GROUP BY, to sum up charges for everyone in the table)

```
SELECT SUM(C.LostBookFee) FROM Charges C WHERE C.PersonId = 'Doug';
```

Doug calls within a couple of days of the notification, complaining that though he has only one book out (i.e., 'I Robot'), he has been charged almost \$3000!

Looking more deeply, you find that whenever an insert into ChargesView is made, an INSTEAD OF TRIGGER is inserting a record into a Charges base table, which is just like the view, except that it has an additional AUTOCOUNTER PRIMARY KEY attribute, a value for which is “tacked” on just before insertion into the Charges base table. There are no constraints in this base table on PersonId or BookId or their pairing! Ugh!!!

How would you fix the database implementation (NOT the application software) in the simplest possible way to guard against the overcharging that can and is occurring?

Full credit for any of following

- **Make PRIMARY KEY (PersonId, BookId) or**
- **Make UNIQUE(PersonId, BookId), or**
- **Add a WHEN clause to the INSTEAD OF trigger that checks to see if (PersonId, BookId) pair already in Charges**

The first two require changes to the table schema, which can be clunky (e.g., see https://www.sqlite.org/lang_altertable.html#otheralter), but may be best. Because of the difficulty of making changes after a DB “goes live”, it's much better to think it through in advance!

EXTRA CREDIT (2 pts) Circle the three tenets that you think are most relevant to the vignette of question 5, both from the perspective of the DB developers and the Application programmers. Briefly describe why they are most relevant to you. You may address more than three. While the particular scenario may not seem that compelling, even something that simple can be, if people believe a (poorly designed) technology over a complainant. Even if you think the previous example is contrived, analogues are not uncommon. **Below I simply list numbers of answers that listed each tenet in the three (or on a few cases, four) most relevant tenets. I don't give the reasons that students gave, but leave it as an exercise to see the relevance of each. Most all (but not all) students received 5 points, and a few received a bonus of 0.5 – 1 point for an outstanding answer (5.5- 6.0 pts total)**

1. to accept responsibility in making engineering decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment; **20 chose this as among the three (or four) most relevant.**
2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist; **5 chose this as among the three (or four) most relevant.**
3. to be honest and realistic in stating claims or estimates based on available data; **11 chose this as among the three (or four) most relevant.**
4. to reject bribery in all its forms; **0 chose this as among the three (or four) most relevant.**
5. to improve the understanding of technology, its appropriate application, and potential consequences; **19 chose this as among the three (or four) most relevant.**
6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations; **17 chose this as among the three (or four) most relevant.**
7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others; **20 chose this as among the three (or four) most relevant.**
8. to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin; **2 chose this as among the three (or four) most relevant.**
9. to avoid injuring others, their property, reputation, or employment by false or malicious action; **10 chose this as among the three (or four) most relevant.**
10. to assist colleagues and co-workers in their professional development and support them in following this code of ethics. **16 chose this as among the three (or four) most relevant.**