Northeastern University
CS5200 – DBMS
Spring 2025, Derbinsky

Exam 2

Name: Erdun E

Problem	Points
NORMALIZATION	18.5/20
CONCEPTUAL DESIGN	12.5/15
LOGICAL DESIGN	19.5/20
PHYSICAL DESIGN	9/10
BONUS: FUNCTIONAL DEPENDENCIES	4/5
Total	/65

Instructions

- You will have 90 minutes to complete this exam; do **NOT** begin until instructed to do so.
- You are allowed to use one sheet of 8.5 × 11" paper for reference, as well as the provided ERD reference, but no other resources.
- No electronic devices may be used, including calculators, cell phones, cameras, and computers.
- Please write legibly: what I cannot read, I cannot award credit!

(20 pts.) NORMALIZATION

Consider a database with the following two relations, T1 and T2, and functional dependencies. Your task is to ensure that the database is in 3NF.

T1(A, B, C)
T2(M, N, O, P, Q, R)

$N \rightarrow PQ$
$R \rightarrow O$

1. On this page, show and justify all steps of your analysis, including the current NF of all relations.
2. On the next page, draw your final schema.

T1 is Atomic so it fit 1NF

T1's non prime Attribute fully Functional depenelency on primary key AB → C, so it fit 2NF

T1's Nonprime Attribut is ont Transitve functional depenelency on primay key so it fit 3NF

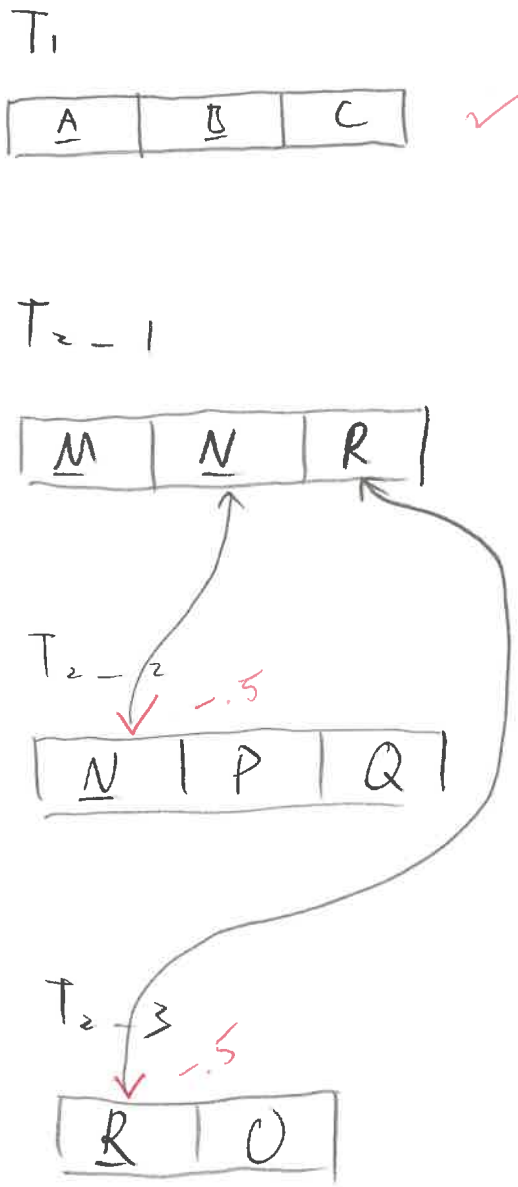
T2 is Antomic so it fit 1NF

T2's non prime attribute is not fully functional dependency on primary key. so it not fit 2NF.

✓ $N \rightarrow PQ$
$R \rightarrow O$
AND $MN \rightarrow R$ ✓

∴ $MN \rightarrow R$ T2-1 (M, N, R) 1NF, 2NF, 3NF
$N \rightarrow PQ$ T2-2 (N, P, Q) 1NF, 2NF, 3NF
$R \rightarrow O$ T2-3 (R, O) 1NF, 2NF, 3NF

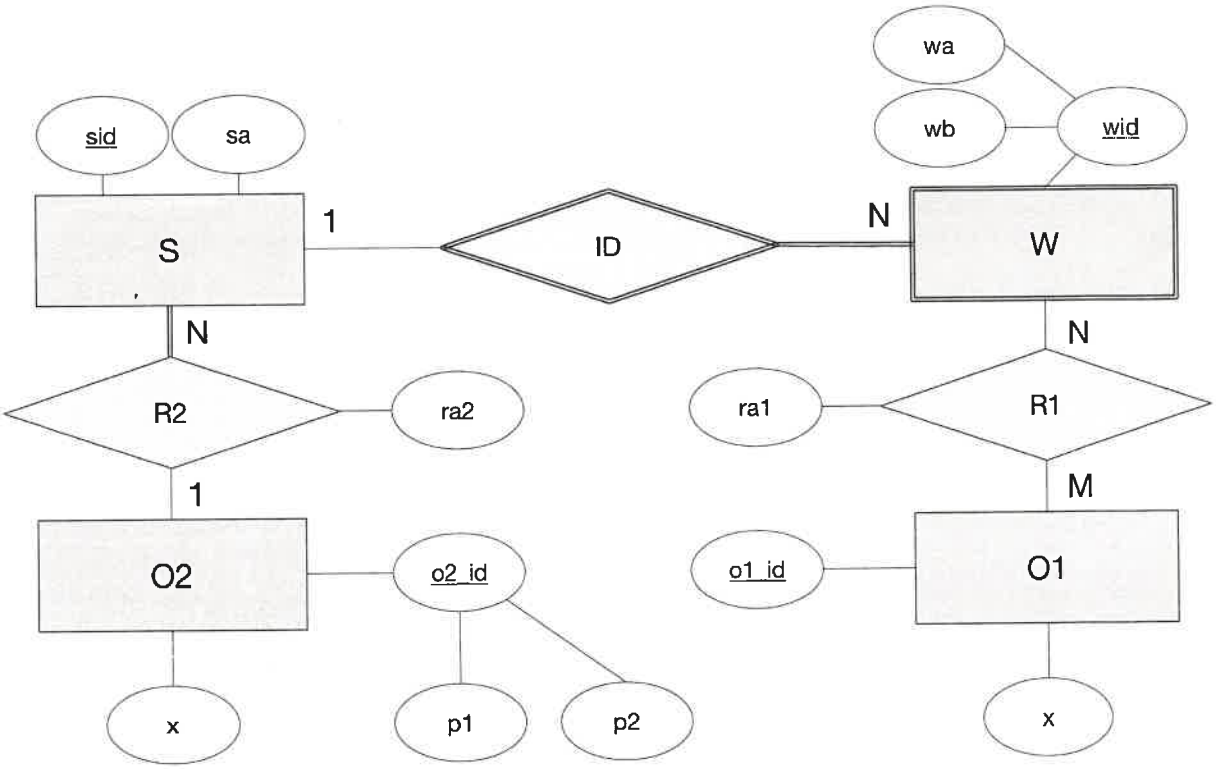
Purposefully left blank for your logical schema – include in your diagram all PKs and FKs.



(15 pts.) Conceptual Design

Respond to the questions below.

a) Consider the following (familiar) ER Diagram:



Indicate the validity of each of the following statements by writing the complete word `true` or `false`.

false — A `W` can be uniquely identified by combining its `wa` and `wb`.
need PK strong entity's PK together

false — An `O2` can be uniquely identified by either its `p1` or `p2`.
and

-.5 ~~false~~ true — Instead of on `R2`, `ra2` could have been equivalently modeled on `S`.
1:N → N

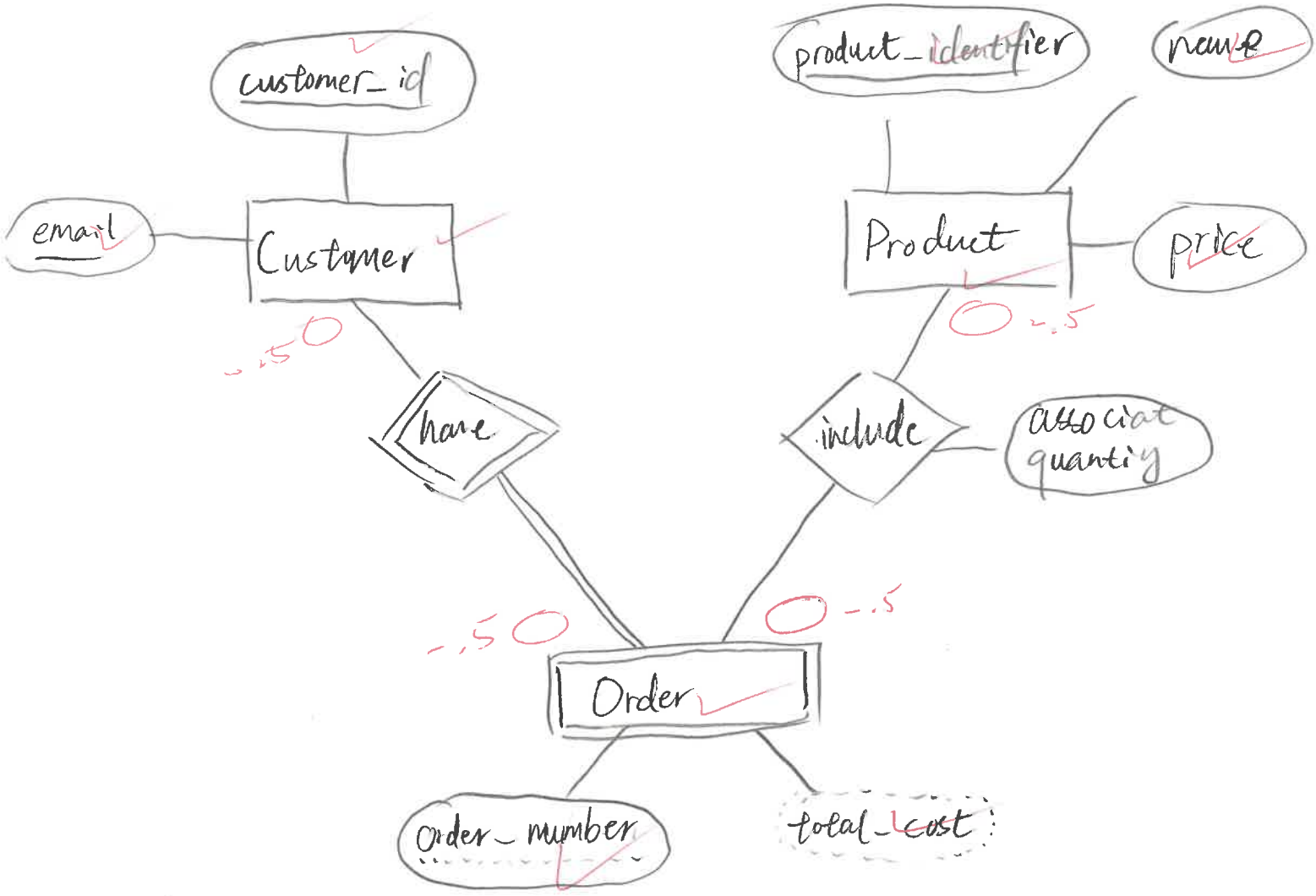
false — Instead of on `R1`, `ra1` could have been equivalently modeled on `O1`.
N:M need bright table

false — It is possible for an `S` to not be associated with any `O2`.
S at least have 1 O2

true — It is possible for an `S` to not be associated with any `W`.
but w at least have 1 S.

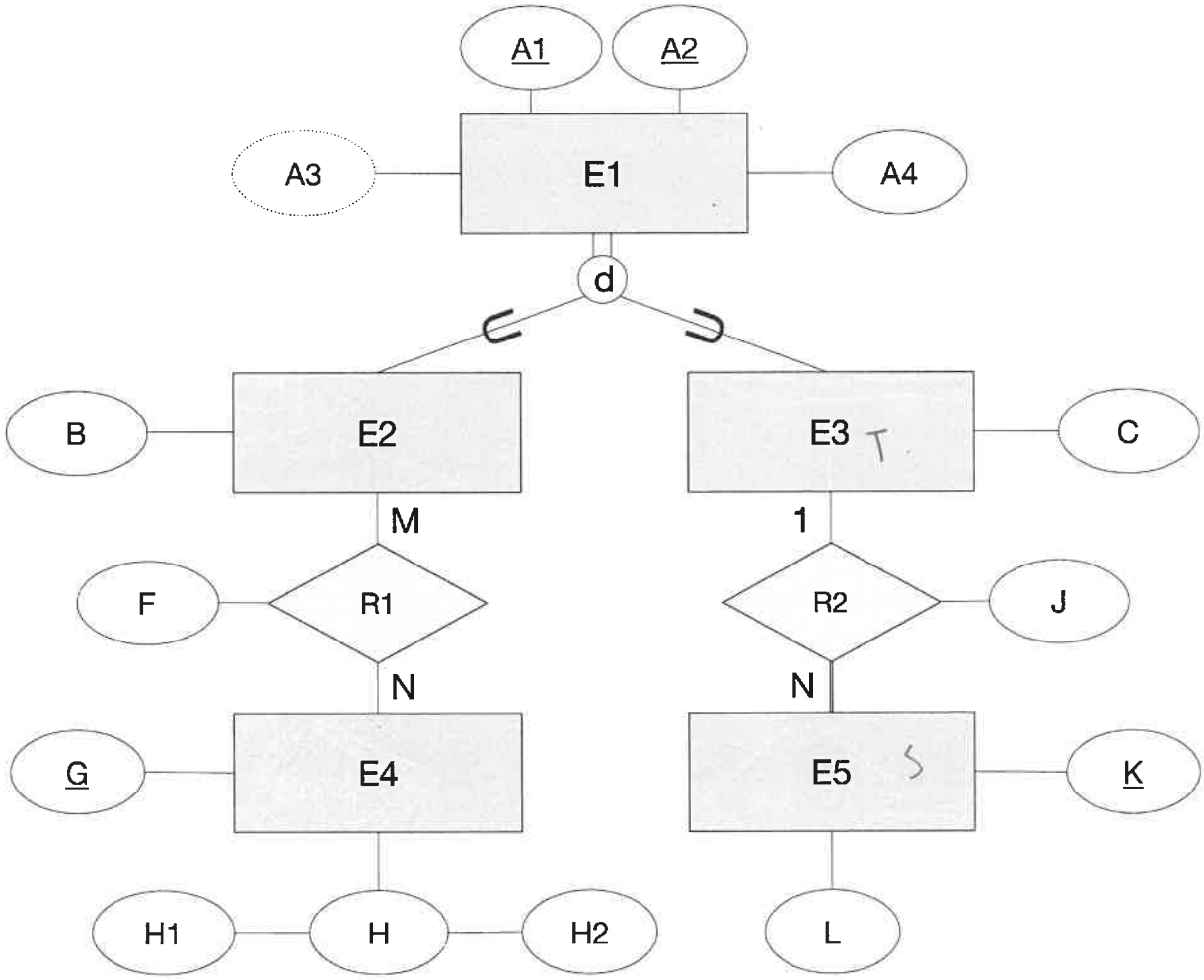
b) Produce an ER Diagram for the following narrative.

- All products on the site have a unique product identifier (PID), a name, and a price.
- All customers have a unique e-mail address and a unique customer id (CID).
- All orders are assigned an order number, unique to each customer. For example, `user1@gmail.com` (CID: 13) may have orders #1, #2, and #3, while `user2@yahoo.com` (CID: 121) may also have orders #1 and #2; however, neither can have a second order #1.
- Each order includes any number of products (including 0 for special promotions), and each product in an order has an associated quantity. For example, an order might include...
  - 2 of "Red Swingline Stapler" (PID: 17, $20 each)
  - 1 of "Multi-Function Printer" (PID: 51, $200 each)
  - 1 of "Baseball Bat" (PID: 3, $10 each)
- It is important to know the total cost of the order (e.g., $250 for the example order above).

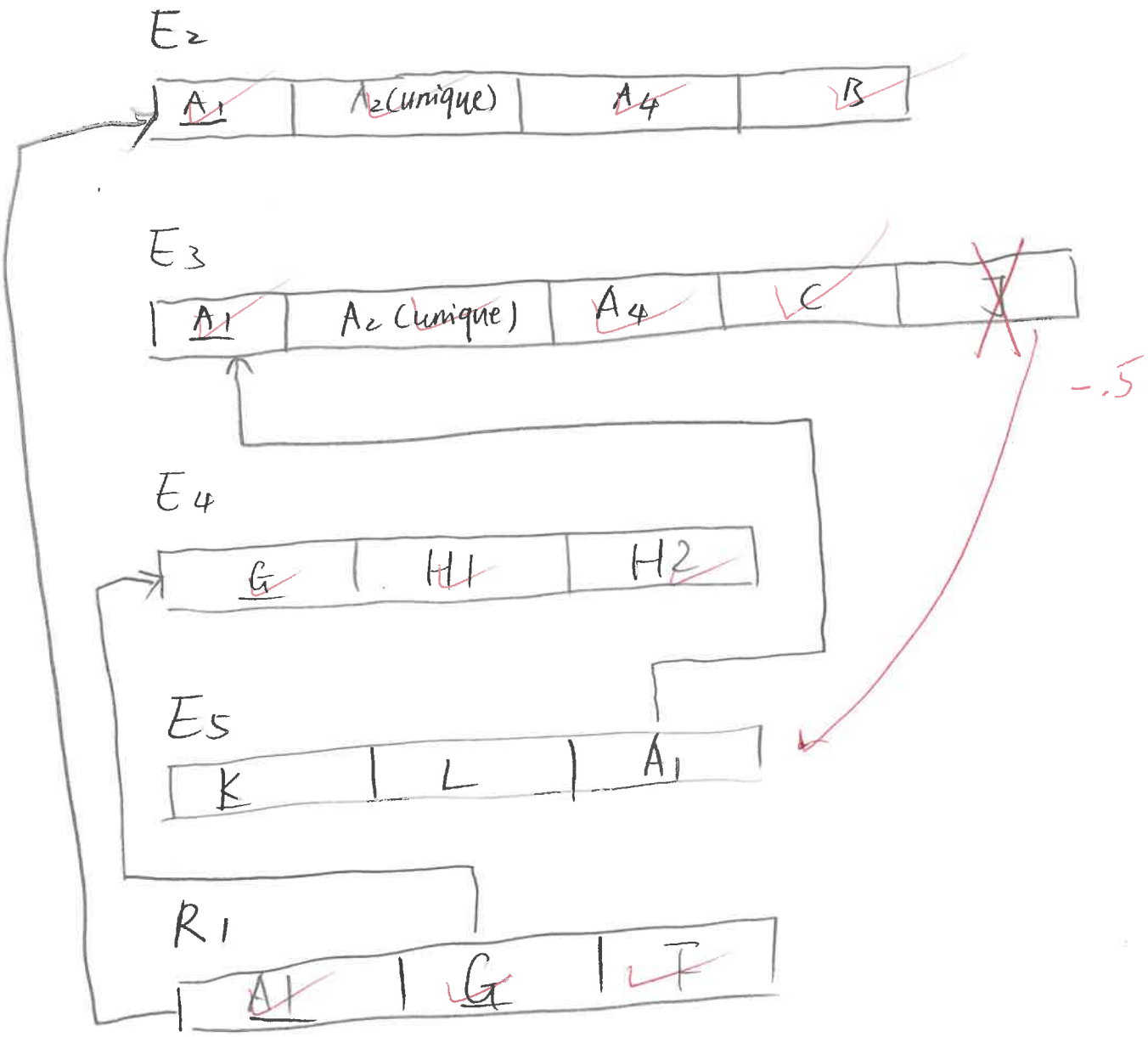


(20 pts.) Logical Design

On the next page, map the following ER Diagram to a relational schema.



Purposefully left blank for your logical schema – include in your diagram all PKs and FKs.



(10 pts.) Physical Design

You are working to improve the performance of a query load on relation `T3(ID,A,B,C)`. After some initial analysis of the state of the relation, you make the following observations:

- Relation `T3` has billions of rows
- Attribute `A` has 366 distinct values
- Attribute `B` has 500 distinct values
- Attribute `C` has 3 distinct values

You then look to the log of past queries involving relation `T3` and it turns out there are only four that have run more than twice, each listed below in parameterized form with associated frequency of execution (`high`=run quite often, `low`=run relatively rarely):

- `low` - `SELECT COUNT(*) FROM T3 WHERE A=?`
- `high` - `SELECT COUNT(*) FROM T3 WHERE B=?`
- `high` - `UPDATE T3 SET A=? WHERE B=?`
- `high` - `SELECT ID FROM T3 WHERE C=?`

Given only this information, you must now justify whether or not to add single-attribute indexes for each attribute below. First, place an × in the appropriate blank indicating whether or not to make the index, then used the supplied space to justify your decision.

i. Attribute `A`: ___ **DO** make index X Do **NOT** make index

Attribute A run query ratherly rarely, and need to be updated frequency.

ii. Attribute `B`: X **DO** make index ___ Do **NOT** make index

Attribut B run query quite often and most distinct value.

iii. Attribute `C`: ___ **DO** make index X Do **NOT** make index

Attribut C only have 3 distinct values

So in the case, whetever C has index or not

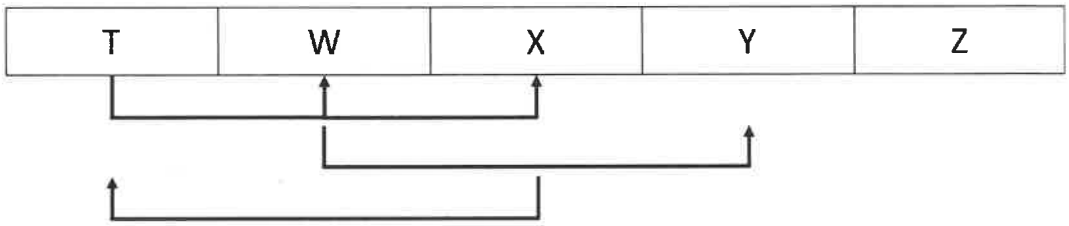
there is no ~~satisfiant~~ performance improvement of it

last query? -1

(5 pts.) Bonus: Functional Dependencies

Respond to the questions below.

a) Consider the following visual depiction of the functional dependencies of a relational schema:



Identify the minimal key(s) of of this relation: TZ

+1

T → WX
W → Y
X → T

T* {T}

T → WX {TWX}
W → Y {TWXY}
X → T {TWXY}

Z {TWXYZ}

b) Consider the current state of the following relation:

	$\mathbf{A_1}$	$\mathbf{A_2}$	$\mathbf{A_3}$
t_1	a	1	x
t_2	b	1	y
t_3	c	2	z
t_4	d	2	z

Your task: given the current state of the relation, for each FD listed below, if you believe it...

- does **NOT** hold: list ALL pairs of rows that invalidate this FD; ex: (t_1, t_4) and (t_2, t_3)
- **DOES** hold: write the full word `valid`

$\{A_1\} \rightarrow \{A_2\}$ valid.

$\{A_2\} \rightarrow \{A_3\}$ (t_1, t_2)

$\{A_2\} \rightarrow \{A_1\}$ $(t_1, t_2)(t_3, t_4)$

+3

$\{A_3\} \rightarrow \{A_2\}$ valid

$\{A_2, A_3\} \rightarrow \{A_1\}$ (t_3, t_4)