# Assignment 2

name: Yu Feng ID:z5094935

Q1.

1) $\pi_{\text{movieID}}(\sigma_{name=\text{"Leonardo DiCaprio"}}(Actors \bowtie AppearsIn))$
2) $\pi_{name}((Actors \bowtie AppearsIn) \div \left(\sigma_{name=\text{"Leonardo DiCaprio"}}(Actors \bowtie AppearsIn)\right))$
3) Let: $\mathrm{A} = \pi_{actorID}(\text{AppearsIn})$ ,

$\mathrm{B} = \pi_{movieID}(\sigma_{name=\text{"Leonardo DiCaprio"}}(\text{Actors} \bowtie \text{AppearsIn}))$

Select all actors who did not appear in **ALL** the movie that "Leonardo DiCaprio" acted:

$\pi_{actorID}(\mathrm{A} \times \mathrm{B} - \text{AppearsIn})$

So let the people who attend all such movies be S:

$\mathrm{S} = \mathrm{A} - \pi_{actorID}(\mathrm{A} \times \mathrm{B} - \text{AppearsIn})$

Specifically, the name of them is:

$\pi_{name}(S \bowtie Actors)$

As result, get together A, B and S, the answer is:

$$\pi_{name}(\pi_{actorID}(\text{AppearsIn}) - \pi_{actorID}(\pi_{actorID}(\text{AppearsIn}) \times \pi_{movieID}(\sigma_{name=\text{"Leonardo DiCaprio"}}(\text{Actors} \bowtie \text{AppearsIn})) - \text{AppearsIn}) \bowtie Actors)$$

Q2.

1) $\mathrm{F}_m = \{A \to H, G \to A, E \to D, D \to G, E \to I, AB \to C, AB \to E, CD \to K\}$
2) All candidate keys are: **{ABJ, GBJ, EBJ, DBJ }**
3)

| | A | B | C | D | E | K | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| ABC | $a_1$ | $a_2$ | $a_3$ | $b_{14}$ | $b_{15}$ | $b_{16}$ | $b_{17}$ | $b_{18}$ | $b_{19}$ | $b_{20}$ |
| DEKG | $b_{21}$ | $b_{22}$ | $b_{23}$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $b_{28}$ | $b_{29}$ | $b_{210}$ |
| HIJ | $b_{31}$ | $b_{32}$ | $b_{33}$ | $b_{34}$ | $b_{35}$ | $b_{36}$ | $b_{37}$ | $a_8$ | $a_9$ | $a_{10}$ |

Because they do not have the same attributes in the same column. It means different tuples can not nature join each other and has the original relations, so it is **NOT lossless-join.**

4) F obey **2NF**, all non-prime attribute is fully functionally dependent on the relation keys. And it do not obey 3NF. i.e. $AB \to C$ and $AB \to E$, it can be shorted in $C \to E$.

5) According minimal cover, we can get 3NF decomposition.$\mathrm{A} \to \mathrm{H}$ means A is prime key.

$R1 = (A\ H)$ $R4 = (D\ G)$
$R2 = (ED\ I)$ $R5 = (AB\ CE)$
$R3 = (GA)$ $R6 = (CDK)$

Lossless join property test: J does not have any relation ship between others, so I ignore it.

| | A | B | C | D | E | K | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| AH | a | b | b | b | b | b | b | a | b |
| EDI | b | b | b | a | a | b | b | b | a |
| GA | a | b | b | b | b | b | a | b | b |
| DG | b | b | b | a | b | b | a | b | b |
| ABCE | a | a | a | b | a | b | b | b | b |
| CDK | b | b | a | a | b | a | b | b | b |

| | A | B | C | D | E | K | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| AH | a | | | | | | | a | |
| EDI | | | | a | a | | | | a |
| GA | a | | | | | | a | | |
| DG | | | | a | | | a | | |
| ABCE | a | a | a | a | a | a | a | a | a |
| CDK | | | a | a | | a | | | |

It's lossless join.

functional dependency: $A^{+}$:={A,H}, $G^{+}$:={G, A, H}, $E^{+}$:={D, I, G, A, H, E}, $D^{+}$={D, G, A, H}
$AB^{+}$:={C, E, A, H, D, I, G}, $ABC^{=}$:={ C, E, A, H, D, I, G, K},
$CD^{+}$={K, D, G, A, H }

it's contain all the original function, so it's dependency-preserving

6) In question 5, It already had decomposition in 3NF:
$R1 = (A\ H)$ $R4 = (D\ G)$
$R2 = (E\ D\ I)$ $R5 = (AB\ CE)$
$R3 = (GA)$ $R6 = (CDK)$
$F = \{A \rightarrow H, D \rightarrow G, E \rightarrow D\ I, AB \rightarrow CE, G \rightarrow A, CD \rightarrow K\}$
in this relation, just change R2 into R21 and R22, R5 into R51 and R52:
$R1 = (A\ H)$ $R4 = (D\ G)$
$R21 = (ED)$ $R22 = (EI)$ $R51 = (ABC)$ $R52 = (ABE)$
$R3 = (GA)$ $R6 = (CDK)$
it's **lossless-join** and **dependency-preserving**. Because the nature joins of R21 and R22 become R2, and R51 join R52 become R5, which is same as decomposition in Q2(5). It has already justified that is lossless-join and dependency-preserving.

Q3

1)

Data pages: P1, P2, P3

Q1: read P1; Q2: read P2; Q3: read P3; Q4: read P2; Q5:read P3; Q6:read P2

Buffer of first two steps:

| | |
|---|---|
| P1 | P2 |

Loop

For LRU, to the Q3, it replace the first buffer (saved P1), write and read P3. And in Q4 it read P2 directly. But for MRU, it replaces the second buffer and read, for Q4 it replaced the second buffer, and did it the entire loops, which cost a lot more time.

2)

Data pages: P1, P2, P3

Q1: read P1; Q2: read P2; Q3: read P3; Q4: read P1; Q5:read P2;

Buffer of first two steps:

| | |
|---|---|
| P1 | P2 |

Loop

For MRU every loop it needs change twice. For FIFO every loop it needs change three times.