



پاسخ مسئله‌ی ۱.

در این سوال فرض کرده‌ایم که همه ستون‌ها *Atomic* هستند.

الف

$$R = (X, Y, Z, S, T, U, V)$$

$$FDs = \begin{cases} S \rightarrow X \\ S \rightarrow V \\ T \rightarrow Y \\ X \rightarrow Y \\ XY \rightarrow TUZ \end{cases} = \begin{cases} S \rightarrow X \\ S \rightarrow V \\ T \rightarrow Y \\ X \rightarrow Y \\ XY \rightarrow T \\ XY \rightarrow U \\ XY \rightarrow Z \end{cases} = \begin{cases} S \rightarrow X \\ S \rightarrow V \\ T \rightarrow Y \\ X \rightarrow T \\ X \rightarrow U \\ X \rightarrow Z \end{cases}$$

همانطور که مشخص است، ستون S می‌تواند باقی ستون‌ها را تعیین کند و از دیگر ستون‌ها نمی‌توان به S رسید. پس حتماً S داخل کلید کاندید ما موجود است و چون باقی ستون‌ها از آن بدست می‌آیند، پس کلید کاندید ما فقط $\{S\}$ است.

همه ستون‌های دیگر به کلید کاندید وابستگی مستقیم یا غیرمستقیم دارند، چون کلید کاندید ما تک عضوی است، پس این دیتابیس $2NF$ است. اما دیتابیس از نوع $3NF$ نیست زیرا که به عنوان مثال ستون Y به ستون T که کلید کاندید نیست وابستگی دارد.

حال با شکستن روابط تعدی موجود، دیتابیس را به فرم نرمال مرتبه سوم تبدیل می‌کنیم:

$$R_1 = \{\underline{T}, Y\}$$

$$R_2 = \{\underline{S}, X, V\}$$

$$R_3 = \{\underline{X}, T, U, Z\}$$

حال چون فقط یک کلید کاندید داریم و دیتابیس از نوع $3NF$ هم هست، پس قطعاً از نوع $BCNF$ هم هست.

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

$$FDs = \begin{cases} A \rightarrow BC \\ BC \rightarrow AD \\ D \rightarrow E \end{cases} = \begin{cases} A \rightarrow B \\ A \rightarrow C \\ BC \rightarrow D \\ BC \rightarrow A \\ D \rightarrow E \end{cases}$$

کلیدهای کاندید دیتابیس ما $\{A, BC\}$ هستند.

همه ستون‌های غیر کلید به کل کلید اصلی وابستگی تابعی دارد، پس از نوع $1NF$ نرمال شده است. اما چون ستون غیر کلیدی پیدا می‌شود که به ستون غیر کلید دیگری وابستگی تابعی دارد، پس از نوع $3NF$ نرمال شده نیست. به عنوان مثال ستون E به ستون D وابسته است.

پس جدول را به این شکل می‌شکنیم تا در هر رابطه‌ی تابعی، ستون سمت چپ یک کلید کاندید باشد. پس:

$$R_1 = \{\underline{A}, B, C\}$$

$$R_2 = \{\underline{B}, C, D\}$$

$$R_3 = \{\underline{D}, E\}$$

با توجه به این که در تمامی FD ها، ستون سمت چپ یک کلید کاندید است، پس دیتابیس ما از نوع $BCNF$ نرمال شده است.

پاسخ مسئله‌ی ۲.

$$R_1 = (X, Y, Z)$$

$$FDs_1 = \begin{cases} Y \rightarrow X \\ XZ \rightarrow Y \\ X \rightarrow Z \end{cases}$$

ابتدا مجموعه FDs را ساده می‌کنیم:

$$\rightarrow FDs_1 = \begin{cases} Y \rightarrow X \\ X \rightarrow Y \\ X \rightarrow Z \end{cases} \rightarrow FDs_1 = \begin{cases} Y \rightarrow X \\ X \rightarrow YZ \end{cases}$$

الف

این عبارت نادرست است. زیرا Z خودش از X بدست می‌آید و از آن نمی‌توان چیزی را بدست آورد.

ب

این عبارت نادرست است. اگر رابطه $X \rightarrow Z$ را حذف کنیم، از روابط باقی‌مانده نمی‌توان آن را بدست آورد.

ج

این عبارت نادرست است. زیرا خود Z از X بدست می‌آید. پس وجود Z اضافی است و باید حذف شود.

د

این عبارت درست است. با توجه به این که دیتابیس ما $1NF$ است و CK ‌های ما همگی تک‌عضوی هستند، پس $2NF$ هم هست. با توجه به این که X, Y کلیدهای کاندید ما هستند و ستون Z به ستونی به جز کلیدهای کاندید وابسته نیست، پس دیتابیس $3NF$ هم هست. همچنین در سمت چپ تمامی FD ‌های موجود، فقط کلید کاندید وجود دارد، پس دیتابیس ما در نهایت $BCNF$ است.

ه

این عبارت درست است.

$$R_2 = (A, B, C, D, E)$$

$$FDs_2 = \begin{cases} A \rightarrow B \\ AB \rightarrow CD \\ D \rightarrow ABC \end{cases}$$

واضح است که کلیدهای کاندید این دیتابیس، DE, AE هستند. همچنین دیتابیس اول هم دارای کلیدهای کاندید X, Y هستند. پس کلید کاندید دیتابیس حاصل *Cartesian Product* این دو، باید شامل یک کلید کاندید از اولی و یک کلید کاندید از دومی باشد. پس در کل 2×2 حالت داریم:

$$CK = \{YAE, YDE, XAE, XDE\}$$

پاسخ مسئله‌ی ۳.

الف

مراحل زیر را طی می‌کنیم:

۱. هر وابستگی تابعی در F را به طوری تجزیه کنید که در سمت راست فقط یک ستون وجود داشته باشد.
۲. صفات اضافی را با محاسبه بسته closure صفات سمت چپ به جز صفت مورد نظر، حذف کنید.
۳. وابستگی‌های تابعی زائد را حذف کنید.
۴. اطمینان حاصل کنید که سمت راست هر وابستگی تابعی فقط شامل یک صفت باشد.
۵. روابطی که با استفاده از تعدی به وجود آمده‌اند را حذف می‌کنیم.

ب

پس از اجرای الگوریتم فوق، به مجموعه FDs زیر می‌رسیم:

$$FDs = \begin{cases} msgID, wordPosition \rightarrow wordID \\ wordID \rightarrow wordText \\ wordText \rightarrow wordID \\ msgID \rightarrow visibility \\ msgID \rightarrow userID \end{cases}$$

ج

این عبارت درست است. زیرا:

$$\begin{cases} msgID, wordID \rightarrow visibility \\ msgID \rightarrow userID \end{cases} \longrightarrow \{ msgID, wordID, visibility \rightarrow userID \}$$

د

می‌دانیم که کلید کاندید ما $\{msgID, wordPosition\}$ است. در ابتدا *Partial Dependency* موجود در $wordID$ و سپس *Transitive Dependency* موجود بین $wordText$ و کلید کاندید را از بین می‌بریم. پس در کل به ۳ رابطه نیاز داریم:

$$\begin{aligned} R_1 &= \{ \underline{wordID}, wordText \} \\ R_2 &= \{ msgID, \underline{userID}, visibility \} \\ R_3 &= \{ \underline{msgID}, \underline{wordPosition}, wordID \} \end{aligned}$$

با توجه به اینکه که مجموعه کلیدهای کاندید ما تک عضوی است، پس دیتابیس ما *BCNF* هم هست. در نتیجه چون نرمال‌سازی ما در سطح *BCNF* است، پس *LossLess* است و *Dependency*‌ها را حفظ می‌کند.

ه

$$R = R_1 \bowtie_{R_1.wordID=R_3.wordID} R_3 \bowtie_{R_3.msgID=R_2.msgID} R_2$$

پاسخ مسئله‌ی ۴.

پاسخ مسئله‌ی ۵.