Part 2: Functional Dependencies, Decompositions, Normal Forms

Question 1

A relation *R* with attributes *ABCDEFGH* and functional dependencies

$S = \{BH \rightarrow AD, D \rightarrow BH, BCE \rightarrow F, F \rightarrow C, A \rightarrow GEF\}$

(a)

- $BH+ = BHADGEFC$, so *BH* is a superkey, $BH \rightarrow AD$ does not violate BCNF.
- $D+ = DBHAGEFC$, so *D* is a superkey, $D \rightarrow BH$ does not violate BCNF.
- $BCE+ = BCEF$, so *BCE* is not a superkey, $BCE \rightarrow F$ violates BCNF.
- $F+ = FC$, so *F* is not a superkey, $F \rightarrow C$ violates BCNF.
- $A+ = AGEFC$, so *A* is not a superkey, $A \rightarrow GEF$ violates BCNF.

So the last three functional dependencies violate BCNF.

(b)

- Decompose *R* using FD $BCE \rightarrow F$. $BCE+ = BCEF$, so this yields two relations: $R1 = BCEF$ and $R2 = ADGHBCE$
- Project the FDs onto $R1 = BCEF$

B	C	E	F	closure	FDs
√				$B+ = B$	nothing
	√			$C+ = C$	nothing
		√		$E+ = E$	nothing
			√	$F+ = FC$	$F \rightarrow C$: violates BCNF; abort

- We must decompose *R1* further.

- Decompose *R1* using FD $F \rightarrow C$. This yields two relations: $R3 = FC$ and $R4 = BEF$.

- Project the FDs onto *R3 = FC*.

F	C	closure	FDs
√		*F+ = FC*	*F* → *C*; *F* is a superkey of *R3*
	√	*C+ = C*	nothing
√	√	irrelevant	weaker FD than what we already have

- This relation satisfies BCNF.

- Project the FDs onto *R4 = BEF*.

B	E	F	closure	FDs
√			*B+ = B*	nothing
	√		*E+ = E*	nothing
		√	*F+ = FC*	nothing
√	√		*BE+ = BE*	nothing
√		√	*BF+ = BFC*	nothing
	√	√	*EF+ = EFC*	nothing

- This relation satisfies BCNF.

- Return to *R2 = ADGHBCE* and project the FDs onto it.

A	D	G	H	B	C	E	closure	FDs
√							*A+ = AGEFC*	*A* → *GEC*: violates BCNF; abort

- We must decompose *R2* further.

- Decompose *R2* using FD *A* → *GEF*. This yields two relations: *R5 = AGCE* and *R6 = ADHB*.

- Project the FDs onto *R5 = AGCE*.

A	G	C	E	closure	FDs
√				$A+ = AGEFC$	$A \to GCE$
	√			$G+ = G$	nothing
		√		$C+ = C$	nothing
			√	$E+ = E$	nothing

- This relation satisfies BCNF.
- Project the FDs onto $R6 = ADHB$.

A	D	H	B	closure	FDs
√				$A+ = AGEFC$	nothing
	√			$D+ = DBHAGEFC$	$D \to AHB$
		√		$H+ = H$	nothing
			√	$B+ = B$	nothing

- This relation satisfies BCNF.
- So the final decomposition is:
    - $R3 = FC$ with FD $F \to C$,
    - $R4 = BEF$ with no FDs,
    - $R5 = AGCE$ with $A \to GCE$,
    - $R6 = ADHB$ with $D \to AHB$.

Question 2

A relation R with attributes $ABCDEFG$ and functional dependencies

$S = \{DBE \to FC, CD \to AF, D \to AB, D \to G, BADE \to C, ABD \to E, D \to F, EF \to B\}$

(a)

- By observation, $D+ = ABCDEFG$, which means D is a key and no superset of D can be a key.
- Since every FD in S has D on left hand side, except $EF \to B$ (and $EF+ = EFB$ at most), we don't have a

key anymore.

- So the only key is *D*.

(b)

- Simplify to singleton right-hand sides at the beginning, say the following is *S1*:

order	FD
1	$ABD \rightarrow E$
2	$ABDE \rightarrow C$
3	$BDE \rightarrow C$
4	$BDE \rightarrow F$
5	$CD \rightarrow A$
6	$CD \rightarrow F$
7	$D \rightarrow A$
8	$D \rightarrow B$
9	$D \rightarrow F$
10	$D \rightarrow G$
11	$EF \rightarrow B$

- Look for FDs to eliminate. Each row in the table below indicates which of the 11 FDs we still have on hand as we consider removing the next one. Of course, as we do the closure test to see whether we can remove $X \rightarrow Y$, we can't use $X \rightarrow Y$, so an FD is never included in its own row.

FD	Exclude these from S1 when computing closure	Closure	Decision
1	1	*ABD+ = ABDFG*, no way to get *E* without this	keep
2	2	*ABDE+ = ABDECFG* , can still get *C*	discard
3	2, 3	*BDE+ = ABDEFG*, no way to get *C*	keep
4	2, 4	*BDE+ = ABCDEFG*, still get *F*	discard
5	2, 4, 5	*CD+ = ABCDFG*, still get *A*	discard
6	2, 4, 5, 6	*CD+ = ABCDEFG*, still get *F*	discard
7	2, 4, 5, 6, 7	*D+ = BDFG*, no way to get *A*	keep
8	2, 4, 5, 6, 8	*D+ = ADFG*, no way to get *B*	keep
9	2, 4, 5, 6, 9	*D+ = ABCDEG*, no way to get *F*	keep
10	2, 4, 5, 6, 10	*D+ = ABDCEF*, no way to get *G*	keep
11	2, 4, 5, 6, 11	*EF+ = EF*, no way to get *B*	keep

- So the remaining FDs *S2*:

order	FD
1	$ABD \rightarrow E$
3	$BDE \rightarrow C$
7	$D \rightarrow A$
8	$D \rightarrow B$
9	$D \rightarrow F$
10	$D \rightarrow G$
11	$EF \rightarrow B$

- Try reducing the LHS of any FDs with multiple attributes on the LHS. For these closures, we will close over the full set S2, including even FD being considered for simplification; remember that we are not considering removing FD, just strengthening it.

- 1 $ABD \to E$
  - $A+ = A$ so we can't reduce the LHS to A.
  - $B+ = B$ so we can't reduce the LHS to B.
  - $D+ = ABCDEFG$ so we can reduce the LHS to D.
- 3 $BDE \to C$
  - $B+ = B$ so we can't reduce the LHS to B.
  - $D+ = ABCDEFG$ so we can reduce the LHS to D.
- 11 $EF \to B$
  - $E+ = E$ so we can't reduce the LHS to E.
  - $F+ = F$ so we can't reduce the LHS to F.
  - so this FD remains as it is.

- Call the newly simplified FDs *S3*:

order	**FD**
1	$D \to E$
3	$D \to C$
7	$D \to A$
8	$D \to B$
9	$D \to F$
10	$D \to G$
11	$EF \to B$

- Do process similar to what we did to *S1*, just in case.

FD	Exclude these from S1 when computing closure	Closure	Decision
1	1	*D+ = ABCDFG*	keep
3	3	*D+ = ABDEFG*	keep
7	7	*D+ = BCDEFG*	keep
8	8	*D+ = ABCDEFG*, still get *B* from 11!	discard!
9	8, 9	*D+ = ACDEG*	keep
10	8, 10	*D+ = ABCDEF*	keep
11	8, 11	*D+ = ACDEFG*	keep

- The following FD *S4* is a minimal basis:

order	FD
1	$D \rightarrow E$
3	$D \rightarrow C$
7	$D \rightarrow A$
9	$D \rightarrow F$
10	$D \rightarrow G$
11	$EF \rightarrow B$

(c)

- Using *S4*, merge RHS, call this *S5*:
  - $D \rightarrow ACEFG$
  - $EF \rightarrow B$
- The set of relations that would be:
  - *R1(A, C, D, E, F, G)*, *R2(B, E, F)*.

(d)

- As we formed each relation from an FD, the LHS of those FDs are indeed superkeys for their relations. But we still need to check that other FDs are not violating BCNF (which allows redundancy). What we need to do is to project the FDs onto each relation.
- Let's look at $EF \rightarrow B$ projecting on the relation *R2*, $E+ = E$, so E is not a superkey of *R2*, contradicting BCNF.
- Hence, the schema allows redundancy.