Hw 6 Solution

Part 1

Given the following functional dependencies:

FD1: {a, b} → {c, d}

FD2: b → {c, e}

FD3: d → f

FD4: c → a

For each of the following, calculate the requested closures. List each step you use to add attributes and note the functional dependency, if any, used to add attributes at each step.

1. (10 points) a^+

STEP 1:	a	add self

no functional dependencies can be used to add more.

1. (10 points) $\{a, d\}^+$

STEP 1:	a, d	add self
STEP 2:	a, d, f	FD3

no functional dependencies can be used to add more.

1. (10 points) b^+

STEP 1:	b	add self
STEP 2:	b, c, e	FD2
STEP 3:	a, b, c, e	FD4
STEP 4:	a, b, c, d, e	FD1
STEP 5:	a, b, c, d, e, f	FD3

no other functional dependencies exist

Part 2

Given R(a, b, c, d, e) with two candidate keys, (a,b) and c and the following functional dependencies:

FD1: {a, b} → {c, d}

FD2: c → {a, b, d, e}

1. (5 points) Is R in 2NF? Justify your answer showing exactly why it is or is not - do not just quote the definition.

   The non-prime attributes are d and e.

   FD1 shows us that d is dependent on {a,b} and FD2 shows us that d is dependent on c. No other functional dependencies exist that could make d only partially dependent on {a, b}

   FD2 shows us that e is dependent on c. The transitive property also gives us e being dependent on {a, b}: ab -> c -> e. No other functional dependencies exist that could make e only partially dependent on {a, b}

   So R is in 2NF.

1. (5 points) If R were in 2NF, would R be in 3NF? Justify your answer showing exactly why it is or is not - do not just quote the definition.

   If R is in 2NF, for each FD of the form X -> A, either X must be a superkey or A must be a prime attribute.

   For FD1, {a,b} is a candidate key, so {a, b} is a superkey.
   For FD2, c is a candidate key, so c is a superkey.

   This means R is in 3NF.

1. (5 points) Is R in BCNF? Justify your answer showing exactly why it is or is not - do not just quote the definition.

   For each FD of the form X -> A, X must be a superkey.

   For FD1, {a,b} is a candidate key, so {a, b} is a superkey.
   For FD2, c is a candidate key, so c is a superkey.

   This means R is in BCNF.

Given S(a, b, c, d, e) with a candidate key (a,b) and given the following set of functional dependencies:

FD1: b → {d, e}

FD2: {a, b} → {c, d, e}

1. (5 points) Is S in 2NF? Justify your answer showing exactly why it is or is not - do not just quote the definition.

   The non-prime attributes are c, d, and e.

   FD2 tells us that c is dependent on {a,b}, and FD1 does not give us a partial dependency.
   FD2 also tells us that d and e are dependent on {a, b}, but FD1 tells us that we have a partial dependency -- d and e can be determined by just b alone.

   S is not in 2NF.

1. (5 points) If S were in 2NF, would S be in 3NF? Justify your answer showing exactly why it is or is not - do not just quote the definition.

   If S is in 2NF, for each FD of the form X -> A, either X must be a superkey or A must be a prime attribute.

   For FD1, b is not a superkey. (It's only part of the candidate key.) The prime attributes are a and b, which do not appear on the right hand side of FD1.

   S in not in 3NF.

1. (5 points) Is S in BCNF? Justify your answer showing exactly why it is or is not - do not just quote the definition.

   For each FD of the form X -> A, X must be a superkey.

   For FD1, b is not a superkey. (It's only part of the candidate key.)

   S is not in BCNF.