Advanced Databases - Exercise Sheet No. 1 Solutions

Exercise I

1) BG → DH ∈ F+

- Compute closure of BG: BG+ = {B, G}
- B → D ⇒ add D → {B, G, D}
- G → A ⇒ add A → {A, B, G, D}
- AB → C ⇒ add C → {A, B, C, D, G}
- CD → E ⇒ add E → {A, B, C, D, E, G}
- CE → GH ⇒ add H → {A, B, C, D, E, G, H} = U
- So, BG → DH ∈ F+

2) CD → B ∉ F+

- Compute closure of CD: CD+ = {C, D}
- CD → E ⇒ add E → {C, D, E}
- CE → GH ⇒ add G, H → {C, D, E, G, H}
- G → A ⇒ add A → {A, C, D, E, G, H}
- No dependency adds B
- So, CD → B ∉ F+

3) ABC → DEG ∈ F+

- Compute closure of ABC: {A, B, C}
- B → D ⇒ add D → {A, B, C, D}
- CD → E ⇒ add E → {A, B, C, D, E}
- G → A ⇒ G not in closure yet
- CE → GH ⇒ C,E in closure → add G,H → {A,B,C,D,E,G,H} = U
- So, ABC → DEG ∈ F+

Exercise II

1) CH → AE

- Compute closure of CH: {C,H}
- C → E ⇒ {C,H,E}
- H → B ⇒ {C,H,E,B}

- BC → D ⇒ B,C in closure → add D → {B,C,D,E,H}
- DE → G ⇒ D,E in closure → add G → {B,C,D,E,H,G}
- DG → A ⇒ D,G in closure → add A → {A,B,C,D,E,G,H} = U
- So, CH → AE holds

2) BC → DEG
   - BC+ = {B,C}
   - BC → D ⇒ add D → {B,C,D}
   - C → E ⇒ add E → {B,C,D,E}
   - DE → G ⇒ D,E → add G → {B,C,D,E,G}
   - DG → A ⇒ D,G → add A → {A,B,C,D,E,G}
   - So, BC → DEG holds
3) ADEG → BC
   - ADEG+ = {A,D,E,G}
   - DE → G ⇒ already have G
   - DG → A ⇒ already have A
   - The closure {A,D,E,G} doesn't contain B,C
   - So, ADEG → BC does not hold

Exercise III

1) Attribute C must be in every key

   - No FD in F has C on the right-hand side, so C is not derivable from any other attributes. Thus, any key K (with K+=U) must contain C.

2) Keys of R(U)
   - A key (candidate key) is a minimal set of attributes that can determine all other attributes in the relation.
   - We can use the information from question 1 to narrow down our search for candidate keys: since C must be in every key, any candidate key must contain C.
   - Let's check the possible combinations (always start with the minimal sets, which are singleton attributes):

- {C}

Since C → D so, C+ = {CD} ≠ U

Thus, C is not a super key.

- {A}, {B}, {D}, {E}, {G}, {H} are not super keys because, from Question 1, every key of R must contain C.
- {A,C}

Check $\{A, C\}$

Compute $(AC)^+$:

- Start: $\{A, C\}$.
- $C \to D \Rightarrow$ add D.
- $AC \to B \Rightarrow$ add B.
- $BD \to E$ (B and D present) $\Rightarrow$ add E.
- $BE \to GH$ (B and E present) $\Rightarrow$ add G, H.
- $G \to A$ already have A.

Result: $(AC)^+ = \{A, B, C, D, E, G, H\} = U$. So $\{A, C\}$ is a superkey.

{A,C} is then a key because neither {A} nor {C} is a super key.

{C}+={C,D} ≠ U and {A}+={A} ≠ U

- {B,C}

Compute $(BC)^+$:

- Start: $\{B, C\}$.
- $C \to D \Rightarrow$ add D.
- $BD \to E \Rightarrow$ add E.
- $BE \to GH \Rightarrow$ add G, H.
- $G \to A \Rightarrow$ add A.

Thus {B,C}+ = U, so {B,C} is a super key

{C,B} is then a key since neither {C}, nor {B} is a super key.

- {C,D}

C → D only

Thus {C,D}+ = {C,D} ≠ U, so {C,D} is not a super key.

- {C,E}

C → D

CE → CED (since C → D)

Thus {C,E}+ = {C,E,D} ≠ U, so {C,E} is not a super key.

- {C,G}

Compute $(CG)^+$:

- Start: $\{C, G\}$.
- $C \to D \Rightarrow$ add D.
- $G \to A \Rightarrow$ add A.
- $AC \to B$ (A and C present) $\Rightarrow$ add B.
- $BD \to E \Rightarrow$ add E.
- $BE \to GH \Rightarrow$ add H (G already present).

Thus {C,G}+ = U, so {C,G} is a super key.

{C,G} is then a key since neither {C}, nor {G} is a super key.

- {C,H}

CH → CHD (since C → D)

Thus {C,H}+ = {C,H,D} ≠ U, so {C,H} is not a super key.

    - Since {A, C}, {B, C}, and {C, G} are keys, any other key must include C but exclude A, B, and G; otherwise it would not be minimal.
    - Therefore, any other key must be sought among the attributes {C, D, E, H}.
    - Nevertheless, {C,E,D,H}+={C,E,D,H} ≠ U. Therefore, no other key exists.
    - Thus, the only keys are {A,C}, {C,B} and {C,G}.

Exercise IV

We need to check whether F+ ⊆ G+ and G+ ⊆ F+

- **Check if F+ ⊆ G+**
- AB → C is in both F and G
- B → A is in both F and G
- AD → E is in both F and G (since in G, we have AD → E)
- BD → I is in both F and G (since in G, we have B → A and AD → EI, thus, using pseudo-transitivity, BD → EI, so, using decomposition rule, BD → I)
- Therefore, F+ ⊆ G+

- **Check if G+ ⊆ F+**
  - AB → C is in both F and G
  - B → A is in both F and G
  - AD → EI (AD → ADE since AD → E and we can't add other attributes
  - Thus, {A,D}+={A,D,E}
  - So, G+ ⊈ F+

Therefore, F+ ≠ G+

Exercise V

ClientNum → ClientName

ProductNum → ProductName, VAT, UnitPrice

(ClientNum, ProductNum, Date) → Number

1- A candidate key is a minimal attribute set whose closure contains all attributes. Single attributes obviously don't determine everything. Therfore, let's Compute closure of {ClientNum, ProductNum, Date}

Compute closure of {ClientNum, ProductNum, Date}:

- It trivially contains ClientNum, ProductNum, Date.
- From ClientNum → ClientName we get ClientName.
- From ProductNum → ProductName, VAT, UnitPrice we get ProductName, VAT, UnitPrice.
- From (ClientNum, ProductNum, Date) → Number we get Number.
  So {ClientNum, ProductNum, Date}$^+$ contains every attribute → it is a super key.

Check minimality:

- {ClientNum, ProductNum} does **not** determine Number or Date.
- {ClientNum, Date} does **not** determine ProductNum (hence not product attributes or Number).
- {ProductNum, Date} does **not** determine ClientNum (hence not ClientName or Number).

Therefore, the only candidate key (under these FDs) is:

(ClientNum, ProductNum, Date).

2- The table COMMANDE is in 1NF because all attributes are atomic, and rows are uniquely identifiable.
The table though is not in 2NF because there are some non-key attributes that depend just on part of the primary key, e.g., ClientNum → ClientName.

3- Decomposed table:
CLIENT(**ClientNum**, ClientName)
PRODUCT(**ProductNum**, ProductName, VAT, UnitPrice)
COMMANDE(**ClientNum, ProductNum, Date**, Number)

Exercise VI

1- To check all the keys, we always start with the minimal sets, which are singleton attributes.

Compute closures:

- A^+: from $A \to BC$ get B, C. From $C \to AD$ get D (and A already). From $CD \to BEF$ get E, F. So $A^+ = \{A, B, C, D, E, F\} = U$. ⇒ **A** is a key.
- C^+: $C \to AD$ gives A, D. From $A \to BC$ get B. From $CD \to BEF$ get E, F. So $C^+ = U$. ⇒ **C** is a key.
- E^+: $E \to ABC$ gives A, B, C. From $C \to AD$ get D. From $CD \to BEF$ get E, F. So $E^+ = U$. ⇒ **E** is a key.
- F^+: $F \to CD$ gives C, D. From $C \to AD$ get A. From $A \to BC$ get B. From $CD \to BEF$ get E. So $F^+ = U$. ⇒ **F** is a key.

No smaller subset than a single attribute can be a key, and we found these four single-attribute keys.

B+ ≠ U and D + ≠ U so, neither B or D is super key.

Thus, all candidate keys are: A , C , E , F.

1- R is in 3NF because:
   - R is in 2NF: the keys of R are single attributes, and the non-key attributes B and D therefore cannot depend on any proper subset of a key.
   - R is in 3NF: About D and B: D (respectively B) can only be determined by a set of attributes that contains B (respectively D) and is not a super key, but such a set would have to be exactly {B} (respectively {D}).
   - In other words, there is no attribute set K such that K is not a super key and K → D (respectively K → B).