Avery Peiffer

aep65

CS 1555/2055

Assignment 6

1.

AB 🡪 E

B 🡪 ED

E 🡪 D

DF 🡪 A

C 🡪 F

DC 🡪 A

**Step 1**: Make RHS singleton

AB 🡪 E

B 🡪 E

B 🡪 D

E 🡪 D

DF 🡪 A

C 🡪 F

DC 🡪 A

**Step 2**: Remove extraneous LHS

B 🡪 E

B 🡪 D

E 🡪 D

DF 🡪 A

C 🡪 F

DC 🡪 A

**Step 3**: Remove redundant FDs

B 🡪 E

E 🡪 D

DF 🡪 A

C 🡪 F

**Step 4**: Find key of relation using closures

A+: A 🡪

B+: B 🡪 BED 🡪

C+: C 🡪 CF 🡪

D+: D 🡪

E+: E 🡪 ED 🡪

F+: F 🡪

AB+: AB 🡪 ABE 🡪 ABED 🡪

AC+: AC 🡪 ACF 🡪

AD+: AD 🡪

AE+: AE 🡪 AED 🡪

AF+: AF 🡪

BC+: BC 🡪 BCF 🡪 BCDEF 🡪 ABCDEF

BD+: BD 🡪 BDE 🡪

BE+: BE 🡪 BDE 🡪

BF+: BF 🡪

CD+: CD 🡪 CDF 🡪

CE+: CE 🡪 CDE 🡪 CDEF 🡪 ACDEF 🡪

CF+: CF 🡪

DE+: DE 🡪

DF+: DF 🡪 ADF 🡪

EF+: EF 🡪 DEF 🡪 ADEF 🡪

Since BC is the only candidate with two elements that covers the entire relation, BC is the minimal key.

**Step 5**: Combine FDs that have the same determinant (none do)

**Step 6**: Construct relation for each FD

R1 (B, E)

R2 (E, D)

R3 (D, F, A)

R4 (C, F)

**Step 7**: Add key (B, C) to relations

**3NF = { (B, E), (E, D), (C, F), (D, F, A), (B, C) }**

2.

A 🡪 B

B 🡪 CD

A 🡪 D

B 🡪 C

AB 🡪 CD

**Step 1**: Find canonical cover (will do this more briefly than in Question 1)

A 🡪 B

B 🡪 C

B 🡪 D

A 🡪 D

AB 🡪 C

AB 🡪 D

AB 🡪 C and AB 🡪 D are extraneous since we already have B 🡪 C and B 🡪 D. A 🡪 D can also be eliminated due to the transitive property. So we are left with: A 🡪 B, B 🡪 C, B 🡪 D

**Step 2**: Find key using closures.

A+: A 🡪 AB 🡪 ABCD

B+: B 🡪 BCD 🡪

C+: C 🡪

D+: D 🡪

Since A is the only candidate of one element that covers the entire set, it is the minimal key.

**Step 3:** Decompose canonical cover using B 🡪 D to get R1 (A, B, C) and R2 (B, D). R2 is in BCNF, but R1 is not because it contains a transitive dependency.

**Step 4:** Decompose R1 using B 🡪 C to get R11(A, B) and R12 (B, C). These are both in BCNF and therefore no further decomposition is necessary.

**BCNF = { (A, B), (B, C), (B, D) }**

3.

R1: (ProductID, Length, Width, Height, Weight, OrderID, OrderDate, CustomerID, TotalPrice)

R2: (CustomerID, Address, City, State, ZipCode, PhoneNumber)

R3: (ProductID, OrderID, ProductQuantity)

FD1: ProductID → Length, Width, Height, Weight

FD2: OrderID → OrderDate, CustomerID, TotalPrice

FD3: CustomerID → Address, City, State, ZipCode, PhoneNumber

FD4: ProductID, OrderID → ProductQuantity

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PID | Len | Wid | Ht | Wt | OID | ODt | CID | Pr | Add | Cit | St | Zip | Ph | Quan |
| R1 | K | K | K | K | K | K | K | K | K | U | U | U | U | U | U |
| R2 | U | U | U | U | U | U | U | K | U | K | K | K | K | K | U |
| R3 | K | U | U | U | U | K | U | U | U | U | U | U | U | U | K |

Steps:

- Using FD1, we know that ProductID implies Length, Width, Height, and Weight. In R3, we have ProductID, so we can make those four parameters known.

- Using FD2, we know that OrderID implies OrderDate, CustomerID, and TotalPrice. In R3, we have OrderID, so we can make those three parameters known.

- Using FD3, we know that CustomerID implies Address, City, State, ZipCode, and PhoneNumber. In R3, we have CustomerID, so we can make these five parameters known.

Since we were able to fill the third row entirely with knowns, the decomposition is lossless. However, the decomposition is not dependency preserving, because FD1 and FD2 are combined into one relation; they need to have their own relations in order for the decomposition to be dependency preserving. Therefore, the decomposition is UGLY.