Andrew Tran (ant111)

CS 1555

Homework 8

1. Consider the following set of functional dependencies:

FD1: ItemId → ItemDescription, ItemPrice, StockQuantity

FD2: PurchaceId → PurchaceDate, VendorCode

FD3: VendorCode → VendorName, VendorAddress

FD4: ItemId, PurchaceId → OrderQuantity

* 1. Using universal relational approach (top-down process), construct a set of 3NF/BCNF relations from the above functional dependencies. Indicate the primary keys for the result relations. Please show all steps clearly as mentioned in the lecture slides.

Key = {ItemId, PurchaseId}

R = (ItemId, ItemDescription, ItemPrice, StockQuantity, PurchaseId, PurchaseDate, VendorCode, VendorName, VendorAddress, OrderQuantity)

R Decompisition 🡪

R1 = (VendorCode, VendorName, VendorAddress) 🡪 FD3

R2 = (ItemId, ItemDescription, ItemPrice, StockQuantity, PurchaseId, PurchaseDate, VendorCode, OrderQuantity)

R2 🡪 Decompisition

R3 = (ItemId, ItemDescription, ItemPrice, StockQuantity) 🡪 FD1

R4 = (ItemId, PurchaseId, PurchaseDate, VendorCode, OrderQuantity)

R4 Decompisition 🡪

R5 = (PurchaseId, PurchaseDate, VendorCode) 🡪 FD2

R6 = (ItemId, PurchaseId, OrderQuantity) 🡪 FD4

**Final Decompisition:**

**R1 = (VendorCode, VendorName, VendorAddress)**

**R3 = (ItemId, ItemDescription, ItemPrice, StockQuantity)**

**R5 = (PurchaseId, PurchaseDate, VendorCode)**

**R6 = (ItemId, PurchaseId, OrderQuantity)**

* 1. Using the table method, check whether the constructed set of relations is lossless or not. Also, state if your decomposition is good, bad or ugly. You must show all steps. Hint: Bad decomposition is a lossy one, while ugly decomposition is lossless but does not preserve some dependencies.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ItemId | ItemDescription | ItemPrice | StockQuantity | PurchaseId | PurchaseDate | VendorCode | VendorName | VendorAddress | OrderQuantity |
| R1 | U | U | U | U | U | U | **K** | K | K | U |
| R3 | **K** | K | K | K | U | U | U | U | U | U |
| R5 | U | U | U | U | **K** | K | **K** | U 🡪 K | U 🡪 K | U |
| R6 | **K** | U 🡪 K | U 🡪 K | U 🡪 K | **K** | U 🡪 K | U 🡪 K | U | U | K |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ItemId | ItemDescription | ItemPrice | StockQuantity | PurchaseId | PurchaseDate | VendorCode | VendorName | VendorAddress | OrderQuantity |
| R1 | U | U | U | U | U | U | K | K | K | U |
| R3 | **K** | **K** | **K** | **K** | U | U | U | U | U | U |
| R5 | U | U | U | U | **K** | **K** | **K** | K | K | U |
| R6 | **K** | **K** | **K** | **K** | **K** | **K** | **K** | U 🡪 K | U 🡪 K | K |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ItemId | ItemDescription | ItemPrice | StockQuantity | PurchaseId | PurchaseDate | VendorCode | VendorName | VendorAddress | OrderQuantity |
| R1 | U | U | U | U | U | U | K | K | K | U |
| R3 | K | K | K | K | U | U | U | U | U | U |
| R5 | U | U | U | U | K | K | K | K | K | U |
| R6 | **K** | **K** | **K** | **K** | **K** | **K** | **K** | **K** | **K** | **K** |

**Good Decomposition**

1. Consider the following set of functional dependencies:

A → B,

B → CD,

A → D,

B → C,

AB → CD,

A → C,

E → F

* 1. Using synthesis approach (bottom-up process), construct a set of 3NF/BCNF relations from the above functional dependencies. Indicate the primary keys for the result relations and whether or not they are in 3NF or BCNF. Please show all steps clearly as mentioned in the lecture slides.

A → D, - inferred from (A 🡪 B, B 🡪 CD) then A 🡪 CD then A 🡪 C and A 🡪 D [Transitive and Decomposition]

A → C, - inferred from (A 🡪 B, B 🡪 CD) then A 🡪 CD then A 🡪 C and A 🡪 D [Transitive and Decomposition]

B → C, - inferred from B 🡪 CD then B🡪 C and B 🡪 D [Decomposition]

AB → CD, - inferred from (A 🡪 B, B 🡪 CD) [Composition]

Canonical Cover:

A → B,

B → CD,

E → F

Key: {A, E}

R1: (A, B)

R2: (B, C, D)

R3: (E, F)

R4: (A, E) 🡪 added because no other relation contains the key

* 1. Using the table method, check whether the constructed set of relations is lossless or not. If not, correct them.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| R1 | **K** | **K** | U 🡪 K | U 🡪 K | U 🡪 K | U |
| R2 | U | **K** | K | K | U | U |
| R3 | U | U | U | U | **K** | K |
| R4 | **K** | U | U | U | **K** | U 🡪 K |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| R1 | K | K | K | K | **K** | U 🡪 K |
| R2 | U | K | K | K | U | U |
| R3 | U | U | U | U | **K** | K |
| R4 | K | U | U | U | **K** | K |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| R1 | **K** | **K** | **K** | **K** | **K** | **K** |
| R2 | U | K | K | K | U | U |
| R3 | U | U | U | U | K | K |
| R4 | K | U | U | U | K | K |

**Lossless**