PRICE-HISTORY

| SName | PName | Start-date | End-date | Price |  |
| --- | --- | --- | --- | --- | --- |
|  | Note 5 | 3 | 4 | $100 |  |
|  | Note 5 | 4 | 4 | $120 |  |
|  | Note 5 | 5 | 5 | $100 |  |
|  | Note 5 | 7 | 7 | $120 |  |
|  |  |  |  |  |  |

Functional Dependencies:

1. PName,SName,Start-date -> Price, End-date
2. PName, SName, End-date-> Price, Start-date

Minimal Basis:

Step 1 (Transform so that RHS only one attribute):

PName,SName,Start-date -> Price

PName,SName,Start-date -> End-date

PName,SName,End-date -> Price

PName,SName,End-date -> Start-date

Step 2 (Remove redundant FDs, assume FD doesn’t exist, does LHS closure contain RHS?):

Remove PName,SName,Start-date->Price?

{**PName,SName,Start-date**}+ = {PName,SName,Start-date,End-date,**Price**}

Step 3 (Composite LHS, remove each one and check if closure still contains RHS):

NA

1. PName,SName,Start-date -> End-date
2. PName,SName,End-date -> Price
3. PName,SName,End-date -> Start-date

Computing Closures:  
 {PName,SName,End-date}+ = {PName,SName,End-date,Price,Start-date}

Keys: {OID} {PName,SName,End-date}

Primary Key: OID {PName,SName,End-date}

LHS of non-trivial FD-1 does not contain the key. It is not in BCNF.

LHS of all other non-trivial FDs contains the key.

RHS of FD-1 is part of the key. Therefore PRICE-HISTORY is in 3NF.

The left-hand side of functional dependency 1 contains a key. ORDERS is in the third normal form.

PName, SName, start-date -> price, end-date

Assumptions:

1. No price change within the same day

add sname? or use SPID

<https://stackoverflow.com/questions/58996506/does-always-weak-entity-need-a-partial-key>

Deriving FDs

* Can there be same PName with different price?
  + Yes
  + PName !-> Price
* Can there be same PName with different Price, Start-date, end-date?
  + Yes
  + PName !-> Price, Start-date, End-date
* Can there be same Start-date with different End-date?
  + No
  + Start-date -> End-date
* Can there be same End-date with different Start-date?
  + No
  + End-date -> Start-date
* FDs:
  + PName, SName, Start-date, End-date -> Price
  + PName, SName, Start-date -> End-date
  + PName, SName, End-date -> Start-date

3NF

* Find Key
  + PName
* Not in 3NF
  + Start Date -> End-date violates 3NF
* Decomposition
  + Find Minimal basis
    - Transform the FDs, so that each right hand side contains only one attribute
      * Resulting FDs:
        + PName -> Price
        + PName -> Start-date
        + PName -> End-date
        + Start-date -> End-date
        + End-date -> Start-date
    - No FD on the minimal basis is redundant.
      * Without PName -> Price
        + {PName}+ = {Pname, Start-date, End-date}
      * Without PName -> Start-date
        + {PName}+ = {Pname, Price, Start-date, End-date}
        + Remove PName -> Start-date
      * Without PName -> End-date
        + Assuming, we removed Pname -> Start-Date
        + {PName}+ = {Pname, Price}
      * Without …
      * Resulting FDs:
        + PName -> Price
        + PName -> End-date
        + Start-date -> End-date
        + End-date -> Start-date
    - For each FD in the minimal basis, none of the attributes on the left hand side is redundant
      * Resulting FDs:
        + PName -> Price
        + PName -> End-date
        + Start-date -> End-date
        + End-date -> Start-date
  + In the minimal basis, combine the FDs whose left hand sides are the same
    - Resulting FDs:
      * PName -> Price, End-date
      * Start-date -> End-date
      * End-date -> Start-date
  + Create a table for each FD remained
    - Resulting tables:
      * R1(PName, Price, End-date)
      * R2(Start-date, End-date)
      * R3(End-date,Start-date)
  + If none of the tables contain a key of the original table R, create a table that contains a key of R
    - Resulting tables:
      * R1(PName, Price, End-date)
      * R2(Start-date, End-date)
      * R3(End-date,Start-date)
  + Remove redundant tables
    - Resulting tables:
      * R1(PName, Price, End-date)
      * R2(Start-date, End-date)
      * R3(End-date,Start-date)

PRODUCTS-IN-SHOPS

| PName | SPID | SPrice | SQuantity | SName |
| --- | --- | --- | --- | --- |
| Note5 | C\_Note5 |  |  | C |
| Note6 | C\_Note6 |  |  | C |
| Note5 | I\_Note5 |  |  | I |
| Note6 | I\_Note6 |  |  | I |

Take Note : SID may be prefix from Shop Name and suffix from Product Name

Assumption: Whole SName is taken as a prefix and whole PName is taken as a suffix

Deriving FDs:

* Can there be ….
* FDs:
  + SPID -> PName
  + SName,PName -> SPID,SPrice,SQuantity
  + ~~SName -> SPID (if we assume that SPID is unique across shops)~~
  + ~~PName -> SPID~~
  + ~~SPID -> SName~~
  + ~~Pname, SPID -> SPrice~~
  + SPID -> SPrice, SQuantity
  + SName, PName -> SPID, SPrice, SQuantity
  + ~~SPID -> PName, SName~~
  + ~~SPID -> PName, SName, SPrice, SQuantity~~

3NF:

* Find key
  + Key: PName,SName
* Find minimal basis
  + Split to single on RHS
    - FDs:
      * PName -> SPID
      * SPID -> PName
      * SName -> SPID (if we assume that SPID is unique across shops)
      * Pname, SPID -> SPrice
  + Remove redundant FDs
    - Without SPID -> Pname
      * {SPID}+ = {SPID}
      * Need SPID -> Pname
    - Without Pname,SPID -> SPrice
      * {Pname,SPID}+ = {Pname,SPID}
      * Need this
  + Remove attributes on LHS
    - Remove SPID from PName,SPID
      * {PName}+ = {PName,SPID,SPrice}
      * Remove PName,SPID -> SPrice and replace with PName -> SPrice
  + FDs:
    - PName -> SPID
    - SPID -> PName
    - SName -> SPID (if we assume that SPID is unique across shops)
    - Pname-> SPrice
* Combine FDs
  + FDs:
    - PName -> SPID, SPrice
    - SPID -> PName
    - SName -> SPID
* Key?
  + Add table
    - PName,SName
* Remove redundant tables
  + No change
* Tables:
  + PName,SName
  + PName -> SPID, SPrice
  + SPID -> PName
  + SName -> SPID