CptS 451 – Class Examples
Functional Dependencies and BCNF

Question1: Identifying Functional Dependencies and BCNF

a)
```
CREATE TABLE MySales (
    pname VARCHAR (15),
    category VARCHAR(6),
    price INTEGER,
    discount INTEGER,
    month CHAR(3)
)
```

b) `pname, category -> price`

The following query finds all pairs of tuples from `MySales` that have the same `(pname,category)` but have different `price`. Since this query returns an empty result, we verify that the functional dependency "`pname,category -> price`" holds on `MySales`.

```
SELECT * FROM MySales as S1, MySales as S2
WHERE S1.pname = S2.pname AND S1.category = S2.category  AND
      S1.price != S2.price
```

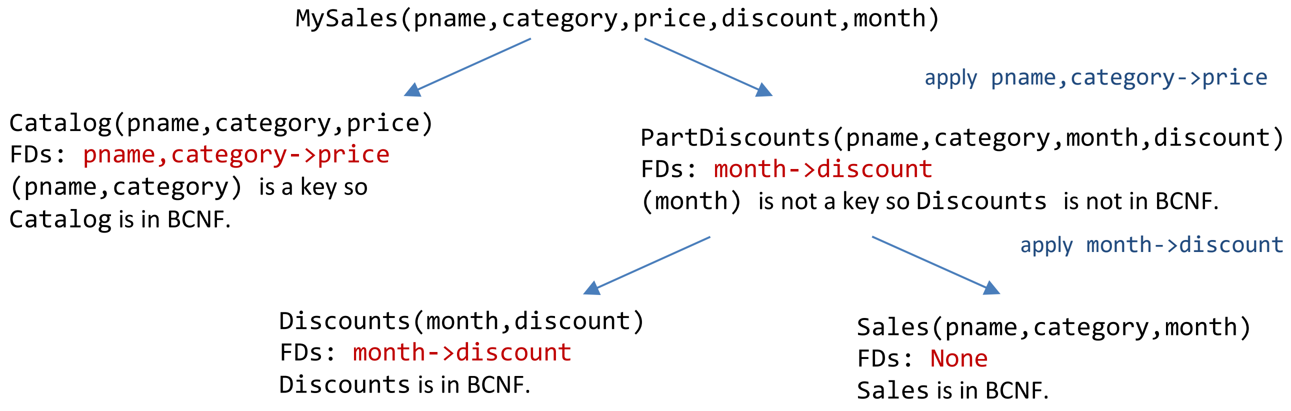
`month -> discount`

The following query finds all pairs of tuples from `MySales` that have the same `month` but have different `discount`. Since this query returns an empty result, we verify that the functional dependency "`month -> discount`" holds on `MySales`.

```
SELECT * FROM MySales S1, MySales S2
WHERE S1.month = S2.month  AND S1.discount != S2.discount
```

c)

Key for MySales: `(pname,category,month)`.
The FD `pname,category->price` violates BCNF since `(pname,category)` is not a key.
Also, the FD `month->discount` violates BCNF since (month) is not a key.



MySales(pname,category,price,discount,month)

apply pname,category->price

Catalog(pname,category,price)
FDs: pname,category->price
(pname,category) is a key so
Catalog is in BCNF.

PartDiscounts(pname,category,month,discount)
FDs: month->discount
(month) is not a key so Discounts is not in BCNF.

apply month->discount

Discounts(month,discount)
FDs: month->discount
Discounts is in BCNF.

Sales(pname,category,month)
FDs: None
Sales is in BCNF.

FDs for Catalog:

Since price doesn’t appear on the left-hand-side of any FD, we don’t need to check the attribute subsets that include price.

{pname}+ = {pname}

{category}+ = {category}

{pname,category}+ = {pname,category,price} so pname,category->price holds

FDs for PartDiscounts:

Since discount doesn’t appear on the left-hand-side of any FD, we don’t need to check the attribute subsets that include discount.

{pname}+ = {pname}

{category}+ = {category}

{month}+ = {month,discount} so month->discount holds

{pname,category}+ = {pname,category,~~price~~}

{pname,month}+ = {pname,month,discount} pname,month->discount is redundant.

{category,month}+ = {category,month}

FDs for Discounts:

Since discount doesn’t appear on the left-hand-side of any FD, we don’t need to check the attribute subsets that include discount.

{month}+ = {month,discount} so month->discount holds

FDs for Sales:

{pname}+ = {pname}

{category}+ = {category}

{month}+ = {month,~~discount~~}

{pname,category}+ = {pname,category,~~price~~}

{pname,month}+ = {pname,month,~~discount~~}

{category,month}+ = {category,month}

No FDs hold.

```
CREATE TABLE Catalog (
    pname VARCHAR (15),
    category VARCHAR(6),
    price INTEGER,
    PRIMARY KEY(pname,category))

 CREATE TABLE Discounts (
    month CHAR(3),
    discount CHAR(4),
    PRIMARY KEY(month))

CREATE TABLE Sales (
    pname VARCHAR (15),
    category VARCHAR(6),
    month CHAR(3),
    PRIMARY KEY(pname,category,month),
    FOREIGN KEY (pname,category) REFERENCES Price(pname,category),
    FOREIGN KEY (month) REFERENCES Discount(month))
```

d) Populate your BCNF tables from the original data. For this you need to write SQL INSERT statements that insert the data into the tables you created at part (c) from the original MySales table. **Provide the INSERT statements and give the number of tuples in each decomposed table** (obtained by running SELECT count(*) FROM Table).

```
INSERT INTO Catalog
( SELECT pname,category,price FROM MySales
  GROUP BY (pname,category,price)
  ORDER BY (pname,category,price))

INSERT INTO Discounts
( SELECT month,discount FROM MySales
  GROUP BY (month,discount)
  ORDER BY (month,discount))

INSERT INTO Sales
( SELECT pname,category,month FROM MySales
  GROUP BY (pname,category,month)
  ORDER BY (pname,category,month))
```

# of tuples in Catalog: 27
# of tuples in Discounts: 12
# of tuples in Sales: 319