

**Washington State University**  
**School of Electrical Engineering and Computer Science**  
**CptS 451 – Introduction to Database Systems**  
**Online**

Dr. Sakire Arslan Ay

**Homework-5**  
**Design Theory and Normal Forms**

Name: Nam Jun Lee

Student Number: 11606459

## Question 1: Identifying Functional Dependencies

a)

```
CREATE TABLE MySales (  
    pname VARCHAR (10),  
    discount VARCHAR (4),  
    month CHAR (3),  
    price INTEGER);
```

b)

### Simple functional dependencies:

pname -> discount

pname -> month

pname -> price

discount -> pname

discount -> month

discount -> price

month -> pname

month -> discount

month -> price

price -> pname

price -> discount

price -> month

### SQL query:

```
SELECT *
```

```
FROM mysales as sales1, mysales as sales2
```

```
WHERE sales1.pname = sales2.pname AND  
sales1.discount <> sales2.discount;
```

```
SELECT *
```

```
FROM mysales as sales1, mysales as sales2
```

```
WHERE sales1.pname = sales2.pname AND
```

sales1.month <> sales2.month;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.pname = sales2.pname AND

sales1.price <> sales2.price;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND

sales1.pname <> sales2.pname;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND

sales1.month <> sales2.month;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND

sales1.price <> sales2.price;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.month = sales2.month AND

sales1.pname <> sales2.pname;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.month = sales2.month AND

sales1.discount <> sales2.discount;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.month = sales2.month AND

sales1.price <> sales2.price;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.price = sales2.price AND

sales1.pname <> sales2.pname;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.price = sales2.price AND

sales1.discount <> sales2.discount;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.price = sales2.price AND

sales1.month <> sales2.month;

### **FD holds:**

pname -> price

month -> discount

### **Two attributes functional dependencies:**

pname, discount -> month

pname, discount -> price (Not check (pname -> price))

pname, month -> discount (Not check (month -> discount))

pname, month -> price (Not check (pname -> price))

pname, price -> discount

pname, price -> month

discount, month -> pname

discount, month -> price

discount, price -> pname

discount, price -> month

month, price -> pname

month, price -> discount (Not check (month -> discount))

**SQL query:**

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.pname = sales2.pname AND sales1.discount = sales2.discount AND  
sales1.month <> sales2.month;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.pname = sales2.pname AND sales1.price = sales2.price AND  
sales1.discount <> sales2.discount;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.pname = sales2.pname AND sales1.price = sales2.price AND  
sales1.month <> sales2.month;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND sales1.month = sales2.month AND  
sales1.pname <> sales2.pname;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND sales1.month = sales2.month AND  
sales1.pname <> sales2.pname;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND sales1.month = sales2.month AND  
sales1.pname <> sales2.pname;

SELECT \*

FROM mysales as sales1, mysales as sales2

WHERE sales1.discount = sales2.discount AND sales1.month = sales2.month AND  
sales1.price <> sales2.price;

SELECT \*

```

FROM mysales as sales1, mysales as sales2
WHERE sales1.discount = sales2.discount AND sales1.price = sales2.price AND
sales1.pname <> sales2.pname;
SELECT *
FROM mysales as sales1, mysales as sales2
WHERE sales1.discount = sales2.discount AND sales1.price = sales2.price AND
sales1.month <> sales2.month;
SELECT *
FROM mysales as sales1, mysales as sales2
WHERE sales1.month = sales2.month AND sales1.price = sales2.price AND
sales1.pname <> sales2.pname;

```

### **FD holds:**

No holds

### **Three attributes functional dependencies:**

pname, discount, month -> price (Not check (pname -> price))  
 pname, discount, price -> month  
 pname, month, price -> discount (Not check (month -> discount))  
 discount, month, price -> pname

### **SQL query:**

```

SELECT *
FROM mysales as sales1, mysales as sales2
WHERE sales1.pname = sales2.pname AND sales1.discount = sales2.discount AND
sales1.price = sales2.price AND sales1.month <> sales2.month;
SELECT *
FROM mysales as sales1, mysales as sales2
WHERE sales1.discount = sales2.discount AND sales1.month = sales2.month AND
sales1.price = sales2.price AND sales1.pname <> sales2.pname;

```

FD holds:

No holds

**ALL Functional Dependencies:**

pname -> price

month -> discount

**c)**

```
CREATE TABLE price (
```

```
    pname VARCHAR(10) PRIMARY KEY,
```

```
    price INTEGER);
```

```
CREATE TABLE discount (
```

```
    month CHAR(3) PRIMARY KEY,
```

```
    discount VARCHAR(4));
```

```
CREATE TABLE mysales1 (
```

```
    pname VARCHAR(10),
```

```
    month CHAR(3),
```

```
    PRIMARY KEY (pname, month),
```

```
    FOREIGN KEY (pname) REFERENCES price(pname),
```

```
    FOREIGN KEY (month) REFERENCES discount(month));
```

**d)**

```
INSERT INTO price
```

```
SELECT pname, price
```

```
FROM mysales
```

```
GROUP BY pname, price;
```

```
INSERT INTO discount
```

```
SELECT month, discount
```

```
FROM mysales
```

GROUP BY month, discount;

INSERT INTO mysales1

SELECT pname, month

FROM mysales

GROUP BY pname, month;

SELECT \* FROM price;

36 tuples in price table

SELECT \* FROM discount;

12 tuples in discount table

SELECT \* FROM mysales1;

426 tuples in mysales1 table

## Question 2: BCNF Decomposition

1. a) minimal key for the relation:

Isolated: E; Left: D; Both: A; Right: B,C,F. Hence, DE.

**b) relation is not in BCNF** because FD  $A \rightarrow BC$  and  $D \rightarrow AF$  is not trivial and does not have a superkey on its left-hand sides. Therefore,  $R(A,B,C,D,E,F)$  decompose into relations  $R1(A,B,C)$ ,  $R3(A,D,F)$ , and  $R4(D,E)$ .

**c)  $A \rightarrow BC$  is preserved in  $R1$ ,  $D \rightarrow AF$  is preserved in  $R3$ .**

2. a) minimal keys for the relation:

Isolated: None; Left: B,C; Both: A,D; Right: None. Hence, ABC & BCD.

**b) relation is not in BCNF** because FD  $ADC \rightarrow D$  has a superkey on its left-hand side but FD  $D \rightarrow A$  is not trivial and does not have a superkey on its left-hand side. Therefore,  $S(A,B,C,D)$  decompose into relations  $S1(A,D)$  and  $S2(B,C,D)$ .

**c)  $D \rightarrow A$  is preserved in  $S1$ ,  $ABC \rightarrow D$  is not preserved because no dependencies in decompose  $S2$ .**