Group - 3_13
Tiles Industry Database Project
Functional Dependencies (FD), Minimal FD Set, Key Of Relation, Type Of Relation

---

Utpal Busa - 202101193
Fenil Vaghasiya - 202101215
Krushang Bhoraniya - 202101175

1) **Branch (branch_no, branch_name, city, owner_name, branch_contact) :**

FDs :

branch_no → branch_name
branch_no → owner_name
branch_no → branch_contact
branch_no → city
branch_name → branch_no
branch_name → owner_name
branch_name → branch_contact
branch_name → city

→ In this relation two keys possible branch_no and branch_name but, we consider branch_no so, in minimal fds branch_no is the key.

Minimal FDs :

branch_no → branch_name
branch_no → owner_name
branch_no → branch_contact
branch_no → city

Key → branch_no

Type → BCNF
{**Reason :** Every attribute of Branch Relation is dependent only and only on Key of Relation (branch_no)}

2) **Prod_Stock (prod_stc_id, prod_stc_date, prod_stc_qty, branch_no) :**

FDs :

{prod_stc_id, prod_stc_date} → prod_stc_qty
{prod_stc_id, prod_stc_date} → branch_no

Minimal FDs :

{prod_stc_id, prod_stc_date} → prod_stc_qty
{prod_stc_id, prod_stc_date} → branch_no

Key → {prod_stc_id, prod_stc_date}

Type → BCNF
{**Reason :** Every attribute of Prod_Stock Relation is dependent only and only on Key of Relation (prod_stc_id, prod_stc_date)}

3) **Raw_Mat_Stock (rm_stc_id, rm_stc_date, rm_stc_qty, branch_no) :**

FDs :

{rm_stc_id, rm_stc_date} → rm_stc_qty
{rm_stc_id, rm_stc_date} → branch_no

Minimal FDs :

{rm_stc_id, rm_stc_date} → rm_stc_qty
{rm_stc_id, rm_stc_date} → branch_no

Key → {rm_stc_id, rm_stc_date}

Type → BCNF
{**Reason :** Every attribute of Raw_Mat_Stock Relation is dependent only and only on Key of Relation (rm_stc_id, rm_stc_date)}

4) **Product (prod_id, prod_name, design, category, color, size, saleprice, description, branch_no) :**

FDs :

prod_id → prod_name
prod_id → design
prod_id → category
prod_id → color
prod_id → size
prod_id → saleprice
prod_id → description
prod_id → branch_no

Minimal FDs :

prod_id → prod_name
prod_id → design
prod_id → category
prod_id → color
prod_id → size
prod_id → saleprice
prod_id → description
prod_id → branch_no

Key : prod_id

Type : BCNF
{Reason : Every attribute of Product Relation is dependent only and only on Key of Relation (prod_id)}

5) **Department (dep_no, dep_name, mgr_id, branch_no) :**

FDs :

dep_no → dep_name
dep_no → mgr_id
dep_no → branch_no

Minimal FDs :

dep_no → dep_name
dep_no → mgr_id
dep_no → branch_no

Key : dep_no

Type : BCNF
{**Reason :** Every attribute of Department Relation is dependent only and only on Key of Relation (dep_no)}

6) **Employee (emp_id, emp_name, email, city, age, gender, emp_contact, salary, dep_no) :**

FDs :

emp_id → emp_name
emp_id → email
emp_id → city
emp_id → age
emp_id → gender
emp_id → emp_contact

emp_id → salary
emp_id → dep_no

Minimal FDs :

emp_id → emp_name
emp_id → email
emp_id → city
emp_id → age
emp_id → gender
emp_id → emp_contact
emp_id → salary
emp_id → dep_no

Key : emp_id

Type : BCNF
{**Reason :** Every attribute of Employee Relation is dependent only and only on Key of Relation (emp_id)}

7) **Customer (cus_no, cus_name, street, city, pincode, state, country, cus_contact, rating) :**

FDs :

cus_no → cus_name
cus_no → cus_city
cus_no → cus_contact
cus_no → rating
cus_no → street
cus_no → city
cus_no → pincode
cus_no → state
cus_no → country

Minimal FDs :

cus_no → cus_name
cus_no → cus_city
cus_no → cus_contact
cus_no → rating
cus_no → street
cus_no → city
cus_no → pincode
cus_no → state
cus_no → country

Key : cus_no

Type : BCNF
{**Reason :** Every attribute of Customer Relation is dependent only and only on Key of Relation (cus_no)}

8) **Order_Info (ord_no, ord_date, purpose, cus_no, prod_id, prod_qty, prod_rate) :**

FDs/Minimal FDs :

ord_no → ord_date
ord_no → purpose
ord_no → cus_no
{ord_no, prod_id} → prod_qty
{ord_no, prod_id} → prod_rate

Key : {ord_no, prod_id}

Here, first three FDs are violating the BCNF requirement. So, we have to Decompose this relation and bring it to BCNF form.

Now, ord_no^{+} = {ord_no, ord_date, purpose, cus_no}

So, we decompose the Order_Info Relation into two Relations Order and Order_Detail which are in BCNF.

8.a) Order (ord_no, ord_date, purpose, cus_no) :

FDs :

ord_no → ord_date
ord_no → purpose
ord_no → cus_no

Minimal FDs :

ord_no → ord_date
ord_no → purpose
ord_no → cus_no

Key : ord_no

Type : BCNF
{**Reason :** Every attribute of Order Relation is dependent only and only on Key of Relation (ord_no)}

8.b) Order_Detail (ord_no, prod_id, prod_qty, prod_rate) :

FDs :

{ord_no, prod_id} → prod_qty
{ord_no, prod_id} → prod_rate

Minimal FDs :

{ord_no, prod_id} → prod_qty
{ord_no, prod_id} → prod_rate

Key : {ord_no, prod_id}

Type : BCNF
{**Reason :** Every attribute of Order_Detail Relation is dependent only and only on Key of Relation ({ord_no, prod_id})}

9) Order_Bill (bill_no, bill_date, order_no, cus_no, amount) :

FDs :

bill_no → bill_date
bill_no → amount
bill_no → order_no
bill_no → cus_no

Minimal FDs :

bill_no → bill_date
bill_no → amount
bill_no → order_no
bill_no → cus_no

Key : bill_no

Type : BCNF
{**Reason :** Every attribute of Order_Bill Relation is dependent only and only on Key of Relation (bill_no)}

10) Raw_Material (rm_id, rm_name, branch_no) :

FDs :

rm_id → rm_name
rm_id → branch_no

Minimal FDs :

rm_id → rm_name
rm_id → branch_no

Key : rm_id

Type : BCNF
{**Reason :** Every attribute of Raw_Material Relation is dependent only and only on Key of Relation (rm_id)}

11) Raw_Mat_Detail (rm_bill_no, rm_bill_date, sup_no, sup_name, street, city, pincode, state, country, sup_contact) :

FDs/Minimal FDs :

rm_bill_no → rm_bill_date
rm_bill_no → sup_no
rm_bill_no → sup_name
rm_bill_no → sup_contact
rm_bill_no → sup_city
sup_no → sup_name
sup_no → street
sup_no → city
sup_no → pincode
sup_no → state
sup_no → country
sup_no → sup_contact

Key : {rm_bill_no }

Here,last three FDs are violating the BCNF requirement. So, we have to Decompose this relation and bring it to BCNF form.

Now, sup_no^{+} = {sup_no, sup_name, sup_city, sup_contact}

So, we decompose the Raw_Mat_Detail Relation into two Relations Supplier and Raw_Mat_Bill which are in BCNF.

11.a) Supplier (sup_no, sup_name, street, city, pincode, state, country, sup_contact) :

Minimal FDs :

sup_no → sup_name
sup_no → sup_contact
sup_no → street
sup_no → city
sup_no → pincode
sup_no → state
sup_no → country
sup_no → sup_contact

Key : sup_no

Type : BCNF
{**Reason :** Every attribute of Supplier Relation is dependent only and only on Key of Relation (sup_no)}

11.b) Raw_Mat_Bill (rm_bill_no, rm_bill_date, sup_no) :

Minimal FDs :

rm_bill_no → rm_bill_date
rm_bill_no → sup_no

Key : rm_bill_no

Type : BCNF
{**Reason :** Every attribute of Raw_Mat_Bill Relation is dependent only and only on Key of Relation (rm_bill_no)}

12) Purchase_Detail (rm_bill_no, rm_id, rm_volume, rm_rate) :

FDs :

{rm_bill_no, rm_id} → rm_volume
{rm_bill_no, rm_id} → rm_rate

Minimal FDs :

{rm_bill_no, rm_id} → rm_volume
{rm_bill_no, rm_id} → rm_rate

Key : {rm_id, rm_bill_no}

Type : BCNF
{**Reason :** Every attribute of Purchase_Detail Relation is dependent only and only on Key of Relation ({rm_id, rm_bill_no})}

13) Used_Raw_Material (prod_id, rm_id) :

FDs : No FDs present in this relation, because all attributes are combined generate Primary Key. Hence, this relation is also in BCNF.

Key : {prod_id, rm_id}

Type : BCNF

DDL Script :

```
create schema Tiles_Industry_Database_Project;

set search_path to Tiles_Industry_Database_Project;


1)---------- Branch ----------
create table Branch(
    branch_no int,
    branch_name varchar(20),
    owner_name varchar(30),
    branch_contact varchar(10) ,
    city varchar(30),
    primary key(branch_no)
);


2)---------- Department ----------
create table Department(
    dep_no int,
    dep_name varchar(40),
    mgr_id varchar(20),
    branch_no int,
    primary key(dep_no),
    foreign key(branch_no) references Branch(branch_no)
    on update cascade
    on delete cascade
);


3)---------- Employee ----------
create table Employee(
    emp_id varchar(20),
    emp_name varchar(30),
    email varchar(50),
    city varchar(30),
```

```
    age int,
    gender varchar(10),
    emp_contact varchar(10),
    salary int,
    dep_no int,
    primary key(emp_id),
    foreign key(dep_no) references Department(dep_no)
    on update cascade
    on delete cascade
);


4)---------- Product ----------
create table Product(
    prod_id varchar(20),
    prod_name varchar(40),
    design varchar(40),
    category varchar(30),
    color varchar(30),
    size varchar(20),
    saleprice float,
    description varchar(60),
    branch_no int,
    primary key(prod_id),
    foreign key(branch_no) references Branch(branch_no)
    on update cascade
    on delete cascade
);


5)---------- Raw_Material ----------
create table Raw_Material(
    rm_id varchar(20),
    rm_name varchar(40),
    branch_no int,
    primary key(rm_id),
    foreign key(branch_no) references Branch(branch_no)
    on update cascade
```

```
    on delete cascade
);
```

6)---------- Supplier ------------

```
create table Supplier(
    sup_no int,
    sup_name varchar(30),
    street varchar(50),
    city varchar(30),
    pincode varchar(30),
    state varchar(30),
    country varchar(30),
    sup_contact varchar(10),
    primary key(sup_no)
);
```

7)---------- Raw_Mat_Bill ----------

```
create table Raw_Mat_Bill(
    rm_bill_no int primary key,
    rm_bill_date DATE ,
    sup_no int,
    foreign key(sup_no) references Supplier(sup_no)
    on update cascade
    on delete cascade
);
```

8)---------- Purchase_Detail ----------

```
create table Purchase_Detail(
    rm_bill_no int,
    rm_id varchar(20),
    rm_qty int,
    rm_rate float,
    primary key(rm_bill_no,rm_id),
    foreign key(rm_bill_no) references Raw_Mat_Bill(rm_bill_no)
    on update cascade
```

```
    on delete cascade,
    foreign key(rm_id) references Raw_Material(rm_id)
    on update cascade
    on delete cascade
);


9)---------- Customer ----------
create table Customer(
    cus_no int primary key,
    cus_name varchar(30),
    street varchar(50),
    city varchar(30),
    pincode varchar(30),
    state varchar(30),
    country varchar(30),
    cus_contact varchar(10),
    rating float
);


10)---------- Order_Table (Order) ----------
create table Order_Table(
    ord_no int primary key,
    ord_date DATE,
    purpose varchar(50),
    cus_no int,
    foreign key(cus_no) references Customer(cus_no)
    on update cascade
    on delete cascade
);


11)---------- Order_Detail ----------
create table Order_Detail(
    ord_no int,
    prod_id varchar(20),
    prod_qty int ,
```

```
    prod_rate float,
    primary key(ord_no,prod_id),
    foreign key(ord_no) references Order_Table(ord_no)
    on update cascade
    on delete cascade,
    foreign key(prod_id) references Product(prod_id)
    on update cascade
    on delete cascade
);


12)---------- Order_Bill ----------
create table Order_Bill(
    bill_no int primary key,
    bill_date DATE,
    amount float,
    ord_no int,
    cus_no int,
    foreign key(ord_no) references Order_Table(ord_no)
    on update cascade
    on delete cascade,
    foreign key(cus_no) references Customer(cus_no)
    on update cascade
    on delete cascade
);

13)---------- Used_Raw_Material ----------
create table Used_Raw_Material(
    prod_id varchar(20),
    rm_id varchar(20),
    primary key(prod_id, rm_id),
    foreign key(prod_id) references Product(prod_id)
    on update cascade
    on delete cascade,
    foreign key(rm_id) references Raw_Material(rm_id)
    on update cascade
    on delete cascade
);
```

```
14)---------- Prod_Stock ----------
create table Prod_Stock(
    prod_stc_id varchar(20),
    prod_stc_date DATE,
    prod_stc_qty int,
    branch_no int,
    primary key(prod_stc_id, prod_stc_date),
    foreign key(branch_no) references Branch(branch_no)
    on update cascade
    on delete cascade
);


15)---------- Raw_Mat_Stock ----------
create table Raw_Mat_Stock(
    rm_stc_id varchar(20),
    rm_stc_date DATE,
    rm_stc_qty int,
    branch_no int,
    primary key(rm_stc_id, rm_stc_date),
    foreign key(branch_no) references Branch(branch_no)
    on update cascade
    on delete cascade
);
```