

Schema Description

* Primary keys are underlined, Foreign keys are bold

Transaction_DealWith_Pay (Transaction#, dateTime, Payment Method, **email**, **e_id**, total)

- Represents: a combined set merged with Transaction entity set, DealWith relationship set and Pay relationship set
- Primary key: Transaction#
- Foreign key: email, e_id
- Domain:
 - Transaction#: must be in the form of digits
 - dateTime: a string in the form of yyyy/mm/dd
 - Payment Method: one of {"Credit Card", "Debit Card", "Cash" }
 - Total: must be an float not equal to 0. Positive value denotes purchase; negative value denotes refund
- Constraints: Each transaction must connect to one customer, one employee, and at least one product for a valid transaction.

Include(**Transaction#**, **p_id**)

- Represents: relationship set *Include*
- Primary key: Transaction#, p_id
- Foreign key: Transaction#, p_id
- Constraints: every Transaction# must be related to at least one p_id

Customer (**email**, c_name, address,password)

- Represents: entity set *Customer*
- Primary key: email
- Domain:
 - email: a string with characters and numbers
 - c_name: a string which is customer's name
 - Address: a string which is customer's address
 - Password: a string that corresponds to customer's email

Member (**email**, points, reward rate)

- Represents: entity set Member
- Primary key: email
- Foreign key: email
- Domain:
 - Points: must be an integer ≥ 0
 - Reward Rate: an integer between 1 to 50
(Every \$RewardRate spent, one point is earned.)

Product (**p_id**, category, sale Price, Inventory, Unit, p_name)

- Represents: entity set Product
- Primary key: p_id
- Domain:
 - p_id: has to be in the form of digits and size of char[10]
 - Category: a string
 - sale Price: must be a float > 0
 - Inventory: must be an integer > 0
 - Unit: a quantity unit in the form of string
 - P_name: has to be a size of char[10]
- Constraints: must have a corresponding manufacturer

Manufacturer(m_id, m_name, overallRating)

- Represents: entity set manufacturer
- Primary key: m_id
- Domain:
 - overallRating: an integer from 1 to 10
 - M_id: has to be in the form of digits and size of char[8]
 - M_name: has to be a size of char[10]

Produce(p_id, m_id, purchase Price)

- Represents: relationship set Produce
- Primary key: p_id, m_id
- Foreign key: p_id, m_id
- Domain:
 - purchase Price: must be an integer > 0
- Constraint: every p_id must be related to exactly one m_id

Employee(e_id, e_name, salary, startDate, type, password)

- Represents: a Employee entity set
- Primary key: e_id
- Domain:
 - e_id: has to be in the form of digits and size of char[8]
 - e_name : has to be a size of char[10]
 - salary: an integer > 0
 - startDate: a string in the form of yyyy/mm/dd
 - Type: one of {"Cashier", "Manager"}
 - Password: a string that corresponds to employee's e_id

SQL DDL

CREATE TABLE Transaction_DealWith_Pay {

```

Transaction# char[10],
dateTime char[10],
Payment Method char[10],
email char[20] NOT NULL,
e_id char[8] NOT NULL ,
Total float,
PRIMARY KEY(Transaction#),
FOREIGN KEY(email) REFERENCES Customer
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
FOREIGN KEY(e_id) REFERENCES Employee
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
}
INSERT
INTO Transaction_DealWith_Pay(Transaction#, dateTime, Payment Method, email, e_id, total)
VALUES ('0000000001', '2018/02/11', Credit Card, 'adam@gmail.com', '00000001',100.00)

INSERT
INTO Transaction_DealWith_Pay(Transaction#, dateTime, Payment Method, email, e_id, total)
VALUES ('0000000002', '2018/02/11', 'Cash', 'xxx@gmail.com', '00000001',100)

INSERT
INTO Transaction_DealWith_Pay(Transaction#, dateTime, Payment Method, email, e_id, total)
VALUES ('0000000003', '2018/02/12', 'Debit Card', 'aaa@foxmail.com', '00000001',100.00)

INSERT
INTO Transaction_DealWith_Pay(Transaction#, dateTime, Payment Method, email, e_id, total)
VALUES('0000000004', '2018/02/13', 'Cash', 'bbb@gmail.com', '00000003',-100.00)

INSERT
INTO Transaction_DealWith_Pay(Transaction#, dateTime, Payment Method, email, e_id, total)
VALUES ('0000000005', '2018/02/13', 'Cash', 'ccc@gmail.com', '00000003',-50.00)

```

| Transaction# | dateTime | PaymentMethod | email | e_id | total |
|--------------|------------|---------------|--------------------|----------|--------|
| 0000000001 | 2018/02/11 | Credit Card | aaa111@gmail.com | 00000001 | 100.00 |
| 0000000002 | 2018/02/11 | Cash | bbb222@gmail.com | 00000001 | 100.00 |
| 0000000003 | 2018/02/12 | Debit Card | ccc333@foxmail.com | 00000001 | 100.00 |

| | | | | | |
|------------|------------|------|------------------|----------|---------|
| 0000000004 | 2018/02/13 | Cash | ddd444@gmail.com | 00000003 | -100.00 |
| 0000000005 | 2018/02/13 | Cash | eee555@gmail.com | 00000003 | -50.00 |

```

CREATE TABLE Include{
    Transaction# char[10]
    P_id Char[10]
    PRIMARY KEY (Transaction#, P_id)
    FOREIGN KEY (Transaction#) REFERENCES Transaction_DealWith_Pay
    FOREIGN KEY (P_id) REFERENCES Product
}

```

```

INSERT
INTO Include(Transaction#, p_id)
VALUES ('0000000001', '2947563856')

```

```

INSERT
INTO Include(Transaction#, p_id)
VALUES ('0000000001', '2873648592')

```

```

INSERT
INTO Include(Transaction#, p_id)
VALUES ('0000000001', '2748364834')

```

```

INSERT
INTO Include(Transaction#, p_id)
VALUES ('0000000002', '1234778374')

```

```

INSERT
INTO Include(Transaction#, p_id)
VALUES ('0000000003', '1237283848')

```

| Transaction# | P_id |
|--------------|------------|
| 0000000001 | 2947563856 |
| 0000000001 | 2873648592 |
| 0000000001 | 2748364834 |
| 0000000002 | 1234778374 |

| | |
|------------|------------|
| 0000000003 | 1237283848 |
|------------|------------|

```
CREATE TABLE Customer {
    Email Char[20],
    Name Char[20],
    Address Char[20],
    Password Char[20],
    PRIMARY KEY(Email)
}
```

```
INSERT INTO Customer (Email, Name, Address, Password) VALUES ('aaa111@gmail.com',
'Sarah King', '1 W16 St','111111')
```

```
INSERT INTO Customer (Email, Name, Address, Password) VALUES ('bbb222@gmail.com',
'Mike Queen', '2 W16 St','222222')
```

```
INSERT INTO Customer (Email, Name, Address, Password) VALUES ('ccc333@gmail.com',
'Jane Knight', '3 W16 St','333333')
```

```
INSERT INTO Customer (Email, Name, Address, Password) VALUES ('ddd444@gmail.com',
'Zed Castle', '4 W16 St','444444')
```

```
INSERT INTO Customer (Email, Name, Address, Password) VALUES ('eee555@gmail.com',
'Andy Liu', '5 W16 St','555555')
```

| Email | Name | Address | Password |
|------------------|-------------|----------|----------|
| aaa111@gmail.com | Sarah King | 1 W16 St | 111111 |
| bbb222@gmail.com | Mike Queen | 2 W16 St | 222222 |
| ccc333@gmail.com | Jane Knight | 3 W16 St | 333333 |
| ddd444@gmail.com | Zed Castle | 4 W16 St | 444444 |
| eee555@gmail.com | Andy Liu | 5 W16 St | 555555 |

```
CREATE TABLE Member {
    Email Char[20]
    Points integer,
    rewardRate integer,
    PRIMARY KEY(Email),
    FOREIGN KEY(Email) REFERENCES Customer,
    ON DELETE CASCADE
    ON UPDATE CASCADE
}
```

```
INSERT INTO Member (Email, Points, rewardRate) VALUES (aaa111@gmail.com, 100, 20)
```

```

INSERT INTO Member (Email, Points, rewardRate) VALUES (bbb222@gmail.com, 500, 10)
INSERT INTO Member (Email, Points, rewardRate) VALUES (ccc333@gmail.com, 24, 2)
INSERT INTO Member (Email, Points, rewardRate) VALUES (ddd444@hotmail.com, 35, 5)
INSERT INTO Member (Email, Points, rewardRate) VALUES (eee555@gmail.com, 890, 10)

```

| Email | Points | rewardRate |
|------------------|--------|------------|
| aaa111@gmail.com | 100 | 20 |
| bbb222@gmail.com | 500 | 10 |
| ccc333@gmail.com | 24 | 2 |
| ddd444@gmail.com | 35 | 5 |
| eee555@gmail.com | 890 | 10 |

```

CREATE TABLE Product {
    P_id Char[10]
    Category Char[10]
    salePrice Float,
    Inventory Float,
    Unit Char[4]
    P_name Char[10]
    PRIMARY KEY(P_id)
}

```

```

INSERT
INTO
Product (P_id,Category,salePrice,Inventory, Unit,P_name) VALUES ('3456782910', 'Fruit',
19.98,10000.00, 'kg','Europe's Best Caribbean Treasuse 600g')

```

```

INSERT
INTO
Product (P_id,Category,salePrice,Inventory, Unit,P_name) VALUES ('2345625142', 'Vegetable',
25.50,4000.00, 'kg','President's Choice Green Beans')

```

```

INSERT
INTO
Product (P_id,Category,salePrice,Inventory, Unit,P_name) VALUES ('3482716482', 'Drinks',
20.99,2000.00, 'bottle','Coca-Cola')

```

```

INSERT
INTO

```

Product (P_id,Category,salePrice,Inventory, Unit,P_name) VALUES ('9028364715',
'Cosmetics', 2.99,100.00, 'bottle','1 Million Eau de Toilette Spray')

INSERT

INTO

Product (P_id,Category,salePrice,Inventory, Unit,P_name) VALUES ('2736451801', 'Fruit',
5.90,3000.00, 'kg', 'PC Organics Canadian Wild Blueberries')

| P_id | Category | salePrice | Inventory | Unit | P_name |
|------------|-----------|-----------|-----------|--------|---------------------------------------|
| 3456782910 | Fruit | 19.98 | 10000.00 | kg | Europe's Best Caribbean Treasuse 600g |
| 2345625142 | Vegetable | 25.50 | 4000.00 | kg | President's Choice Green Beans |
| 3482716482 | Drinks | 20.99 | 2000.00 | bottle | Coca-Cola |
| 9028364715 | Cosmetics | 2.99 | 100.00 | bottle | 1 Million Eau de Toilette Spray |
| 2736451801 | Fruit | 5.90 | 3000.00 | kg | PC Organics Canadian Wild Blueberries |

```
CREATE TABLE Manufacturer{
    M_id Char[8],
    Name Char[10],
    overallRating integer,
    PRIMARY KEY(M_id)
}
```

INSERT INTO Manufacturer (M_id,Name,overallRating) VALUES ('00000001', 'Nestle', 8)

INSERT INTO Manufacturer (M_id,Name,overallRating) VALUES ('00000002', 'Johnson & Johnson', 7)

INSERT INTO Manufacturer (M_id,Name,overallRating) VALUES ('00000003', 'Tyson Foods', 3)

INSERT INTO Manufacturer (M_id,Name,overallRating) VALUES ('00000004', 'Coca Cola', 9)

INSERT INTO Manufacturer (M_id,Name,overallRating) VALUES ('00000005', 'Pepsi', 1)

| M_id | Name | overallRating |
|----------|-------------------|---------------|
| 00000001 | Nestle | 8 |
| 00000002 | Johnson & Johnson | 7 |

| | | |
|----------|-------------|---|
| 00000003 | Tyson Foods | 3 |
| 00000004 | Coca Cola | 9 |
| 00000005 | Pepsi | 1 |

```

CREATE TABLE Produce{
    P_id Char[10]
    M_id Char[8]
    purchasePrice Float
    PRIMARY KEY(P_id,M_id)
    FOREIGN KEY(P_id) REFERENCES Product
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
    FOREIGN KEY (M_id) REFERENCES Manufacturer
        ON DELETE NO ACTION
        ON UPDATE NO ACTION
}

```

```

INSERT INTO Produce (P_id,M_id,purchasePrice) VALUES ('0000000001', '00000001', 1.20)
INSERT INTO Produce (P_id,M_id,purchasePrice) VALUES ('0000000002', '00000002', 0.80)
INSERT INTO Produce (P_id,M_id,purchasePrice) VALUES ('0000000003', '00000001', 0.99)
INSERT INTO Produce (P_id,M_id,purchasePrice) VALUES ('0000000004', '00000003', 12.00)
INSERT INTO Produce (P_id,M_id,purchasePrice) VALUES ('0000000005', '00000003', 11.99)

```

| P_id | M_id | purchasePrice |
|------------|----------|---------------|
| 0000000001 | 00000001 | 1.20 |
| 0000000002 | 00000002 | 0.80 |
| 0000000003 | 00000001 | 0.99 |
| 0000000004 | 00000003 | 12.00 |
| 0000000005 | 00000003 | 11.99 |

```

CREATE TABLE Employee {
    e_id Char[8],
    e_name Char[10],
    Salary float,

```



```

    startDate Char[20],
    Type Char[7],
    Password Char[20],
    PRIMARY KEY (e_id)
}

```

```

INSERT INTO Employee (e_id, e_name, Salary, startDate, type, Password) VALUES
('00000001', 'Amy Acker', 2000.00, '2016/12/01','manager','111111')
INSERT INTO Employee (e_id, e_name, Salary, startDate, type, Password) VALUES
('00000002', 'Bob Zhang', 2000.00, '2016/12/01','cashier','222222')
INSERT INTO Employee (e_id, e_name, Salary, startDate, type, Password) VALUES
('00000003', 'Carrie Walter', 1800.00, '2017/01/22','cashier','333333')
INSERT INTO Employee (e_id, e_name, Salary, startDate, type, Password) VALUES
('00000004', 'David Zhang', 1700.00, '2017/03/13','cashier','444444')
INSERT INTO Employee (e_id, e_name, Salary, startDate, type, Password) VALUES
('00000005', 'Eva Zhang',1600.00, '2017/12/01','cashier','555555')

```

| e_id | e_name | Salary | startDate | Type | Password |
|----------|---------------|---------|------------|---------|----------|
| 00000001 | Amy Acker | 4000.00 | 2016/12/01 | manager | 111111 |
| 00000002 | Bob Zhang | 2000.00 | 2016/12/01 | cashier | 222222 |
| 00000003 | Carrie Walter | 1800.00 | 2017/01/22 | cashier | 333333 |
| 00000004 | David Zhang | 1700.00 | 2017/03/13 | cashier | 444444 |
| 00000005 | Eva Zhang | 1600.00 | 2017/12/01 | cashier | 555555 |

Functional Dependency

Transaction_DealWith_Pay (Transaction#, dateTime, Payment Method, **email**, **e_id**, total)

Transaction# -> dateTime, payment Method, total, e_id, email, p_id

// A transaction# can identify a transaction, and implies other attributes of a transaction

// such as transaction dateTime, the payment method.

// Lastly, this transaction will associate with a certain employee, a customer identified by

// his/her email, and the products purchased in this transaction.

Include(Transaction#, p_id)

Null

Customer (email, c_name, address,password)

Email -> address, c_name,password

// An email can identify a customer, and implies other attributes of a customer including

// customer name and address. password that corresponds to the employee's e_id

Member (email, points, rewardRate)

Email -> points, rewardRate

// An email can identify a member, and implies other attributes of a member including
// customer points and rewardRate.

Product (p_id, category, sale Price, Inventory, Unit, p_name)

P_id -> category, sale Price, Inventory, Unit, p_name

// A p_id can identify a product, and implies other attributes of a product including
// its category, inventory, salePrice, unit, and its name.

Manufacturer(m_id, m_name, overallRating)

M_id -> m_name, overallRating

// A m_id can identify a manufacturer, and implies other attributes of a manufacturer
// including manufacturer's name and its overall rating.

Produce(p_id,m_id,purchase Price)

P_id,m_id -> purchase Price

// the p_id and m_id can identify the unique contract with a product and a manufacturer,
// and imply the purchase price on the contract.

Employee(e_id, e_name, salary, startDate,type,password)

e_id -> salary, startDate, type, e_name,password

// the e_id is the employee id, which can identify an employee, and it can imply
// the name, his/her salary, the startDate of this employee, and the job type, password
// corresponds to the employee's e_id

Note for BCNF:

For each entity, there is one and only one primary key set, and no other candidate keys exist in the diagram. The functional dependency is written in such a way that each primary key can determine the rest of attributes in the table, with no exceptions. Therefore, we have confirmed that everything is fully normalized to be in BCNF.