Homework 1

1. 以不列舉欄位的方式新增一筆食物資料

```
INSERT INTO food
VALUES ('DO001', '甜甜圈', '2019/09/18', 'TW', 30, '點心');
```

2. 以列舉欄位的方式新增一筆食物資料

```
INSERT INTO food (id,name,expiredate,placeid,price,catalog)
VALUES ('BO001', '珍珠奶茶', '2019/12/12', 'TW', 60, '飲料');
```

3. 以不列舉欄位的方式新增多產地資料

```
INSERT INTO place
VALUES ('AU', '澳洲'),('KR', '韓國');
```

4. 修改一筆食物資料的價格

```
UPDATE food
   set price=70
WHERE id='BO001';
```

按價格分 250 以下、251~500 和 501 以上三種分別增加 8%,5%和 3%且價格無條件捨去成整數

```
SET SQL_SAFE_UPDATES = 0 ; -- safe mode 關掉
UPDATE food
SET  price =
    case
        WHEN price <=250 THEN FLOOR(price*1.08)
        WHEN price BETWEEN 250 AND 500 THEN FLOOR(price*1.05)
        WHEN price >500 THEN FLOOR(price*1.03)
    END;
SET SQL_SAFE_UPDATES = 1 ; -- safe mode 開啟
```

5. 刪除一筆食物資料

```
DELETE FROM food WHERE id='BO001';
```

Homework 2

1. 查詢所有食物表格中所有欄位的資料

```
SELECT *
FROM food;
```

2. 查詢所有食物名稱、到期日和價格

```
SELECT  name, expiredate,price
FROM food;
```

3. 查詢所有食物名稱、到期日和價格，並將表頭重新命為'名稱'、'到期日'和'價格'

```
SELECT  name AS '名稱',expiredate '到期日',price '價格'
from food;
```

4. 查詢所有食物的種類有哪些？(重覆的資料只顯示一次)

```
SELECT DISTINCT catalog -- 方法1
FROM food;
SELECT catalog    -- 方法2
FROM food
GROUP BY catalog;
```

5. 查詢所有食物名稱和種類，並串接成一個字串，中間以空白隔開，並將表頭重新命為'Food name & catalog'

```
SELECT name,catalog,concat(food.name , ' ' ,catalog)  'Food name & catalog'
FROM food;
```

--- WHERE 子句

6. 查詢所有食物價格超過 400 的食物名稱和價格

```
SELECT name,price
FROM food
where price>400;
```

7. 查詢所有食物價格介於 250~530 之間的食物名稱和價格

```
SELECT name,price
FROM food
where price>=250 and price<=530;
```

8. 查詢所有食物價格不介於 250~530 之間的食物名稱和價格

```
SELECT name,price
FROM food
where price<250 or price>530;
```

9. 查詢所有食物種類為'點心'的食物名稱和價格

```
SELECT name,price
FROM food
where catalog like '點心';
```

10. 查詢所有食物種類為'點心'和'飲料'的食物名稱、價格和種類

```
SELECT name,price,catalog
FROM food
where catalog in('點心', '飲料');
```

11. 查詢所有食物產地為'TW'和'JP'的食物名稱和價格

```
SELECT name,price
FROM food
WHERE placeid in( 'TW' , 'JP');
```

12. 查詢所有食物名稱有'油'字的食物名稱、到期日和價格

```
SELECT name,expiredate,price
FROM food
WHERE name LIKE '%油%';
```

13. 查詢所有食物到期日在今年底以前到期的食物名稱和價格

```
SELECT name,price
FROM food
WHERE DATEDIFF(expiredate,'2019/12/31')<0;
```

14. 查詢所有食物到期日在明年 6 月底以前到期的食物名稱和價格

```
SELECT name,price
FROM food
WHERE DATEDIFF(expiredate,'2020/06/30')<0;
```

15. 查詢所有食物 6 個月內到期的食物名稱和價格

```
SELECT name,price
FROM food
WHERE DATEDIFF(expiredate,NOW()+INTERVAL 6 MONTH) <0
 AND expiredate>NOW();
```

--- ORDER BY 子句

16. 查詢所有食物名稱、到期日和價格，並以價格做降冪排序

```
SELECT f.name,f.expiredate,f.price
FROM food f
ORDER BY price DESC;
```

17. 查詢前三個價格最高的食物名稱、到期日和價格，並以價格做降冪排序

```
SELECT f.name,f.expiredate,f.price
FROM food f
ORDER BY price DESC
LIMIT 3;
```

18. 查詢種類為'點心'且價格低於等於 250 的食物名稱和價格，並以價格做升冪排序

```
SELECT f.name,f.price
FROM food f
WHERE catalog like '點心' AND price<=250
ORDER BY price ASC;
```

19. 顯示所有食物名稱、價格和增加 5%且四捨五入為整數後的價格，新價格並將表頭命名為'New Price'

```
SELECT f.name,f.price,ROUND(price*1.05) 'New Price'
FROM food f;
```

20. 接續上題，再增加一個表頭命名為'Increase'，顯示 New price 減去 price 的值

```
SELECT f.name,f.price,ROUND(price*1.05) 'New Price', (ROUND(price*1.05)-price) 'Increase'
FROM food f;
```

21. 顯示所有食物名稱、價格和整數後的價格，新價格並將表頭命名為'New Price'；按價格分 250 以下、251~500 和 501 以上三種分別增加 8%,5%和 3%且價格無條件捨去成整數

```
SELECT f.name,f.price,ROUND(price*1.05) 'Round Price',
    case
        WHEN ROUND(price*1.05) <=250 THEN FLOOR(ROUND(price*1.05)*1.08)
        WHEN ROUND(price*1.05) BETWEEN 250 AND 500 THEN FLOOR(ROUND(price*1.05)*1.05)
        WHEN ROUND(price*1.05) >500 THEN FLOOR(ROUND(price*1.05)*1.03)
    END 'New Price'
FROM food f;
```

22. 查詢所有食物名稱、種類、距離今天尚有幾天到期(正數表示)或已過期幾天(負數表示)和註記(有'已過期'或'未過期'兩種)，並將後兩者表頭分別命名為'Days of expired'和'expired or not'

```
SELECT f.name,f.catalog,DateDIFF(expiredate,NOW()) 'Days of expired',IF(DateDIFF(expiredate,NOW())>0,'未過期','已過期') 'expired or not'
FROM food f;  -- 方法1

SELECT f.name, f.catalog, IF(DATEDIFF(f.expiredate, NOW()) > 0, DATEDIFF(f.expiredate, NOW()), ' ') AS 'Non_Over_Days',
                          IF(DATEDIFF(f.expiredate, NOW()) < 0, DATEDIFF(f.expiredate, NOW()), ' ') AS 'Over_Days',
                          IF(DATEDIFF(f.expiredate, NOW()) > 0, '未過期', '已過期') AS 'expired_Or_Not'
FROM food AS f; -- 方法2
```

23. 接續上題，並以過期天數做升冪排序

```
SELECT f.name,f.catalog,DateDIFF(expiredate,NOW()) 'Days of expired',IF(DateDIFF(expiredate,NOW())>0,'未過期','已過期') 'expired or not'
FROM food f
ORDER BY DateDIFF(expiredate,NOW()) ASC;
```

--- GROUP BY & HAVING 子句

24. 查詢所有食物最高、最低、加總和平均價格，表頭分別命名為'Max'、'Min'、'Sum'和'Avg'，結果皆以四捨五入的整數來顯示

```
SELECT ROUND(Max(price)) 'Max',ROUND(Min(price)) 'Min',ROUND(Sum(price)) 'Sum',ROUND(Avg(price)) 'Avg'
FROM food;
```

25. 接續上題，查詢每個種類

```
SELECT catalog,ROUND(Max(price)) 'Max',ROUND(Min(price)) 'Min',ROUND(Sum(price)) 'Sum',ROUND(Avg(price)) 'Avg'
FROM food
GROUP BY catalog;
```

26. 接續上題，查詢每個種類且平均價格超過 300，並以平均價格做降冪排序

```
SELECT catalog,ROUND(Max(price)) 'Max',ROUND(Min(price)) 'Min',ROUND(Sum(price)) 'Sum',ROUND(Avg(price)) 'Avg'
FROM food
GROUP BY catalog
HAVING ROUND(Avg(price))>300
ORDER BY Avg DESC;
```

27. 顯示查詢每個種類的食物數量

```
SELECT catalog,count(name)
FROM food
GROUP BY catalog;
```

28. 查詢不同產地和每個種類的食物數量

```
SELECT f.catalog,count(f.name),placeid
FROM food f
GROUP BY catalog,placeid;
```

作業更改數值後:

```
INSERT INTO food VALUES ('CK001', '曲奇餅乾', '2018/01/10', 'TL', 250, '點心');
INSERT INTO food VALUES ('CK002', '蘇打餅乾', '2019/10/12', 'TW', 80, '點心');
INSERT INTO food VALUES ('DK001', '高山茶', '2018/05/23', 'TW', 780, '飲料');
INSERT INTO food VALUES ('DK002', '綠茶', '2019/06/11', 'JP', 530, '飲料');
INSERT INTO food VALUES ('OL001', '苦茶油', '2020/03/16', 'TW', 360, '調味品');
INSERT INTO food VALUES ('OL002', '橄欖油', '2018/07/25', 'TL', 420, '調味品');
INSERT INTO food VALUES ('CK003', '仙貝', '2020/11/01', 'JP', 270, '點心');
INSERT INTO food VALUES ('SG001', '醬油', '2019/05/05', 'JP', 260, '調味品');
INSERT INTO food VALUES ('OL003', '葡萄子油', '2019/05/05', 'JP', 550, '調味品');
INSERT INTO food VALUES ('CK004', '鳳梨酥', '2020/10/12', 'TW', 340, '點心');
INSERT INTO food VALUES ('CK005', '太陽餅', '2017/08/27', 'TW', 150, '點心');
INSERT INTO food VALUES ('DK003', '紅茶', '2019/11/12', 'TL', 260, '飲料');
INSERT INTO food VALUES ('SG002', '醋', '2019/09/18', 'TW', 60, '調味品');
```

Homework3

1. 查詢所有食物名稱、產地編號、產地名稱和價格

```
SELECT f.name,f.placeid,p.name,f.price
FROM food f JOIN place p
ON f.placeid = p.id;
```

2. 查詢所有食物名稱和產地名稱，並串接成一個字串，中間以空白隔開，並將表頭重新命為'Food name & place'

```
SELECT f.name,p.name,concat(f.name,' ',p.name) 'Food name & place'
FROM food f JOIN place p
ON f.placeid = p.id;
```

3. 查詢所有'台灣'生產的食物名稱和價格

```
SELECT f.name,f.price,p.name
FROM food f JOIN place p
ON f.placeid = p.id
WHERE p.name like '台灣';
```

4. 查詢所有'台灣'和'日本'生產的食物名稱和價格，並以價格做降冪排序

```
SELECT f.name,f.price, p.name
FROM food f JOIN place p
ON f.placeid = p.id
WHERE p.name like '台灣' OR p.name like'日本'
ORDER BY f.price DESC;

-- WHERE f.placeid = p.id AND p.name IN ('台灣', '日本');   用in
```

5. 查詢前三個價格最高且'台灣'生產的食物名稱、到期日和價格，並以價格做降冪排序

```
SELECT f.name,f.expiredate,f.price
FROM food f JOIN place p
ON f.placeid = p.id
WHERE p.name like '台灣'
ORDER BY f.price DESC
LIMIT 3;
```

6. 查詢每個產地(顯示產地名稱)最高、最低、加總和平均價格，表頭分別命名為'Max'、'Min'、'Sum'和'Avg'，結果皆以四捨五入的整數來顯示

```
SELECT ROUND(MAX(f.price)) 'Max', ROUND(MIN(f.price)) 'Min',ROUND(SUM(f.price)) 'Sum',ROUND(AVG(f.price)) 'Avg',p.name
FROM food f JOIN place p
ON f.placeid = p.id
GROUP BY p.name;
```

7. 查詢不同產地(顯示產地名稱)和每個種類的食物數量

```
SELECT p.name,f.catalog,count(f.name)
FROM food f JOIN place p
ON f.placeid = p.id
GROUP BY p.name,catalog;
```

Homework4

1. 查詢所有比'鳳梨酥'貴的食物名稱、到期日和價格

```
SELECT f.name,f.expiredate,f.price
FROM food f
WHERE price>(SELECT price
             FROM food
             WHERE name ='鳳梨酥');
```

2. 查詢所有比'曲奇餅乾'便宜且種類是'點心'的食物名稱、到期日和價格

```
SELECT f.name,f.expiredate,f.price
FROM food f
WHERE price<(SELECT price
             FROM food
             WHERE name ='曲奇餅乾')
AND   catalog='點心';
```

3. 查詢所有和'鳳梨酥'同一年到期的食物名稱、到期日和價格

```
SELECT f.name,f.expiredate,f.price
FROM food f
WHERE Year(expiredate)=(SELECT Year(expiredate)
                        FROM food
                        WHERE name ='鳳梨酥');  -- 去除自己!??!
```

4. 查詢所有比平均價格高的食物名稱、到期日和價格

```
SELECT f.name,f.expiredate,f.price
FROM food f
WHERE price>(SELECT AVG(price)
             FROM food);
```

5. 查詢所有比平均價格低的'台灣'食物名稱、到期日和價格

```
SELECT f.name,f.price,p.name
FROM food f JOIN place p
ON f.placeid = p.id
WHERE p.name like '台灣'
AND price<(SELECT AVG(price)
           FROM food);
```

6. 查詢所有種類和'仙貝'相同且價格比'仙貝'便宜的食物名稱、到期日和價格

```
SELECT f.name,f.expiredate,f.price
FROM food f
WHERE price<(SELECT price
             from food
             WHERE name='仙貝')
AND   catalog=(SELECT catalog
               from food
               WHERE name='仙貝');
```

7. 查詢所有產地和'仙貝'相同且過期超過 6 個月以上的食物名稱、到期日和價格

```
SELECT f.name,f.expiredate,f.price,f.placeid
FROM food f
WHERE DATEDIFF(expiredate,NOW()-INTERVAL 6 MONTH) <0
AND   placeid = (SELECT placeid
                 FROM food f
                 WHERE name like '仙貝');
```

8. 查詢每個產地價格最低的食物名稱、到期日和價格

```
SELECT name,expiredate,price
FROM food f
WHERE price in (SELECT MIN(price)
                FROM food
                GROUP BY placeid
                HAVING placeid=f.placeid);  -- 方法1

SELECT name,expiredate,price
FROM food f
WHERE price in (SELECT MIN(price)
                FROM food
                WHERE placeid=f.placeid);    -- 方法2
```

9. 查詢每個種類的食物價格最高者的食物名稱和價格

```
SELECT name,expiredate,price
FROM food f
WHERE price in (SELECT Max(price)
                FROM food
                GROUP BY catalog
                HAVING catalog=f.catalog);  -- 方法1

SELECT name,expiredate,price
FROM food f
WHERE price = (SELECT Max(price)
                FROM food
                WHERE catalog=f.catalog);    -- 方法2
```

10. 查詢所有種類不是'點心'但比種類是'點心'貴的食物名稱、種類和價格，並以價格做降冪排序

```
SELECT name,catalog,price     -- 我的
FROM food f
WHERE catalog not like '點心'
AND price>(SELECT price
           FROM food
           GROUP BY catalog
           HAVING catalog='點心')
ORDER BY price desc;


SELECT name,catalog,price    -- 家禎
FROM food f
WHERE catalog not like '點心'
AND price>any(SELECT price
           FROM food
           WHERE catalog='點心')
ORDER BY price desc;
```

11. 查詢每個產地(顯示產地名稱)的食物價格最高者的食物名稱和價格

```
SELECT f.name,f.price,p.name
FROM food f join place p
on f.placeid=p.id
WHERE price in
          (SELECT MAX(price )
           FROM food
           GROUP BY placeid
           HAVING placeid=f.placeid);
```

Homework5

1. 以不列舉欄位的方式新增一筆食物資料

```
INSERT INTO food
VALUES ('DO001', '甜甜圈', '2019/09/18', 'TW', 30, '點心');
```

2. 以列舉欄位的方式新增一筆食物資料

```
INSERT INTO food (id,name,expiredate,placeid,price,catalog)
VALUES ('BO001', '珍珠奶茶', '2019/12/12', 'TW', 60, '飲料');
```

3. 以不列舉欄位的方式新增多產地資料

```
INSERT INTO place
VALUES ('AU', '澳洲'),('KR', '韓國');
```

4. 修改一筆食物資料的價格

```
UPDATE food
   set price=70
WHERE id='BO001';
```

5. 按價格分 250 以下、251~500 和 501 以上三種分別增加 8%,5%和 3%且價格無條件捨去成整數

```
SET SQL_SAFE_UPDATES = 0 ; -- safe mode 關掉
UPDATE food
SET  price =
    case
        WHEN price <=250 THEN FLOOR(price*1.08)
        WHEN price BETWEEN 250 AND 500 THEN FLOOR(price*1.05)
        WHEN price >500 THEN FLOOR(price*1.03)
    END;
SET SQL_SAFE_UPDATES = 1 ; -- safe mode 開啟
```

6. 刪除一筆食物資料

```
DELETE FROM food WHERE id='BO001';
```