

可參閱老師筆記網站:

<https://hackmd.io/@mackliu/HyCjh1oWJI#修改後的-SQL> :

※用學號找同學是什麼班級

```
SELECT `students`.`id`,
       `students`.`school_num`,
       `students`.`name`,
       `classes`.`name`,
       `class_student`.`seat_num`
FROM `students`,`classes`,`class_student`
WHERE `students`.`school_num`=`class_student`.`school_num` &&
      `classes`.`code`=`class_student`.`class_code`;
```

↓ 執行結果

id	school_num 學號	name 姓名	name	seat_num
1	911001	丁于穎	一年一班	1
2	911002	尹慧如	一年一班	2
3	911003	孔琇榆	一年一班	3
4	911004	文勝真	一年一班	4
5	911005	方玉婷	一年一班	5
6	911006	毛家男	一年一班	6
7	911007	王鳳如	一年一班	7
8	911008	史乾君	一年一班	8
9	911009	田泓宣	一年一班	9
10	911010	白金圓	一年一班	10
11	911011	石政華	一年一班	11
12	911012	任佩君	一年一班	12
13	911013	朱怡蓉	一年一班	13
14	911014	江欣欽	一年一班	14
15	911015	何秋桂	一年一班	15

※ 讓他明確只顯示479筆

```
SELECT *  
FROM `students`,`class_student`  
WHERE `students`.`school_num`=`class_student`.`school_num`
```

※ 計算班級人數

```
SELECT `class_student`.`class_code`,COUNT(*) as '人數'  
FROM `class_student` GROUP BY `class_student`.`class_code`;
```

※ 計算班級人數+班級平均成績

| (計算班級平均成績)

```
SELECT `class_student`.  
        `class_code`,  
        COUNT(*) as '人數',  
        AVG(`student_scores`.`score`) AS '班級平均'  
FROM `class_student`,`student_scores`  
WHERE `class_student`.`school_num`=`student_scores`.`school_num`  
GROUP BY `class_student`.`class_code`;
```

```
1 SELECT `class_student`.  
2     `class_code`,  
3     count(*) as '人數',  
4     AVG(`student_scores`.`score`) AS '班級平均'  
5 FROM `class_student`,`student_scores`  
6 WHERE `class_student`.`school_num`=`student_scores`.`school_num`  
7 GROUP BY `class_student`.`class_code`
```

執行結果

class_code	人數	班級平均
101	49	74.0612
102	38	71.8947
103	57	75.8070
104	59	68.9492
105	55	68.7636
106	48	75.3542
107	47	70.8936
108	43	79.3488
109	42	77.9286
110	41	74.9024

※計算班級+人數+班級平均成績

(班級代碼換成班級名稱)

```
SELECT `classes`.`name` as '班級',  
       COUNT(*) as '人數',  
       AVG(`student_scores`.`score`) AS '班級平均'  
FROM `class_student`, `student_scores`, `classes`  
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&  
      `class_student`.`class_code`=`classes`.`code`  
GROUP BY `class_student`.`class_code`;
```

```
1 SELECT `classes`.`name` as '班級',  
2       count(*) as '人數',  
3       AVG(`student_scores`.`score`) AS '班級平均'  
4 FROM `class_student`, `student_scores`, `classes`  
5 WHERE `class_student`.`school_num`=`student_scores`.`school_num` && `class_student`.`class_code`=`classes`.`code`  
6 GROUP BY `class_student`.`class_code` |
```

執行結果

班級	人數	班級平均
一年一班	49	74.0612
一年二班	38	71.8947
一年三班	57	75.8070
一年四班	59	68.9492
一年五班	55	68.7636
一年六班	48	75.3542
一年七班	47	70.8936
一年八班	43	79.3488
一年九班	42	77.9286
一年十班	41	74.9024

※計算班級+人數+班級平均成績+班導師

加入班導師

```
SELECT `classes`.`name` as '班級',
       `classes`.`tutor` as '班導師',
       COUNT(*) as '人數',
       AVG(`student_scores`.`score`) AS '班級平均'
FROM `class_student`, `student_scores`, `classes`
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
      `class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`;
```

執行結果

班級	班導師	人數	班級平均
一年一班	孫承翰	49	74.0612
一年二班	王詠晴	38	71.8947
一年三班	賴冠廷	57	75.8070
一年四班	郭欣妤	59	68.9492
一年五班	謝家豪	55	68.7636
一年六班	黃雅婷	48	75.3542
一年七班	林志偉	47	70.8936
一年八班	趙宜蓁	43	79.3488
一年九班	張建宏	42	77.9286
一年十班	許麗華	41	74.9024

※計算班級+人數+班級平均成績+班導師+班級排名

依班級平均由大到小排序

```
SELECT `classes`.`name` as '班級',
       `classes`.`tutor` as '班導師',
       COUNT(*) as '人數',
       AVG(`student_scores`.`score`) AS '班級平均'
FROM `class_student`, `student_scores`, `classes`
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
      `class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`
ORDER BY `班級平均` DESC;
```

執行結果

班級	班導師	人數	班級平均 ▾ 1
一年八班	趙宜蓁	43	79.3488
一年九班	張建宏	42	77.9286
一年三班	賴冠廷	57	75.8070
一年六班	黃雅婷	48	75.3542
一年十班	許麗華	41	74.9024
一年一班	孫承翰	49	74.0612
一年二班	王詠晴	38	71.8947
一年七班	林志偉	47	70.8936
一年四班	郭欣妤	59	68.9492
一年五班	謝家豪	55	68.7636

※計算班級+人數+班級平均成績+班導師+班級排名+班級最高分/最低分

班級最高分/最低分

```
SELECT `classes`.`name` as '班級',
       `classes`.`tutor` as '班導師',
       COUNT(*) as '人數',
       AVG(`student_scores`.`score`) AS '班級平均',
```

```

        max(`student_scores`.`score`) AS '最高分',
        min(`student_scores`.`score`) AS '最低分'
FROM `class_student`,`student_scores`,`classes`
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
`class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`
ORDER BY `班級平均` DESC;

```

執行結果

班級	班導師	人數	班級平均 ▼ 1	最高分	最低分
一年八班	趙宜蓁	43	79.3488	88	69
一年九班	張建宏	42	77.9286	88	70
一年三班	賴冠延	57	75.8070	89	56
一年六班	黃雅婷	48	75.3542	89	62
一年十班	許麗華	41	74.9024	86	64
一年一班	孫承翰	49	74.0612	88	60
一年二班	王詠晴	38	71.8947	87	60
一年七班	林志偉	47	70.8936	85	60
一年四班	郭欣妤	59	68.9492	88	60
一年五班	謝家豪	55	68.7636	87	60

※計算班級+人數+班級平均成績+班導師+班級排名+班級最高分/最低分+校平均

增加校平均

```

SELECT `classes`.`name` as '班級',
       `classes`.`tutor` as '班導師',
       COUNT(*) as '人數',
       AVG(`student_scores`.`score`) AS '班級平均',
       max(`student_scores`.`score`) AS '最高分',
       min(`student_scores`.`score`) AS '最低分',
       (SELECT AVG (`student_scores`.`score`) FROM
`student_scores`) AS '校平均'
FROM `class_student`,`student_scores`,`classes`

```

```
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
`class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`
ORDER BY `班級平均` DESC
```

執行結果

班級	班導師	人數	班級平均 ▼ 1	最高分	最低分	校平均
一年八班	趙宜蓁	43	79.3488	88	69	73.5637
一年九班	張建宏	42	77.9286	88	70	73.5637
一年三班	賴冠廷	57	75.8070	89	56	73.5637
一年六班	黃雅婷	48	75.3542	89	62	73.5637
一年十班	許麗華	41	74.9024	86	64	73.5637
一年一班	孫承翰	49	74.0612	88	60	73.5637
一年二班	王詠晴	38	71.8947	87	60	73.5637
一年七班	林志偉	47	70.8936	85	60	73.5637
一年四班	郭欣妤	59	68.9492	88	60	73.5637
一年五班	謝家豪	55	68.7636	87	60	73.5637

※計算班級+人數+班級平均成績+班導師+班級排名+班級最高分/最低分+校平均+依照校平均去根據班級平均分數排名(如果班級平均大於校平均=表現優良；小於校平均=有待加強)

依照校平均去根據班級平均分數排名(如果班級平均大於校平均=表現優良；小於校平均=有待加強)

根據校平均做班級判斷

```
SELECT `classes`.`name` as '班級',
       `classes`.`tutor` as '班導師',
       COUNT(*) as '人數',
       AVG(`student_scores`.`score`) AS '班級平均',
       max(`student_scores`.`score`) AS '最高分',
       min(`student_scores`.`score`) AS '最低分',
       (SELECT AVG (`student_scores`.`score`) FROM
`student_scores`) AS '校平均',
```

```

CASE
    WHEN `校平均` <= `班級平均` THEN '表現良好'
    WHEN `校平均` > `班級平均` THEN '有待加強'
END
FROM `class_student`,`student_scores`,`classes`
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
`class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`
ORDER BY `班級平均` DESC

```

上層會找不到校平均的資料，因為FROM在WHERE之前

所以改用子查詢的方式，將校平均/班級平均的公式，帶入CASE...WHEN 的判斷中

WHEN 校平均 <= 班級平均 THEN '表現良好' 改為如下，有待加強的判斷異同

```

WHEN (SELECT AVG (`student_scores`.`score`) FROM `student_scores`)
<= AVG(`student_scores`.`score`) THEN '表現良好'

```

※修改後的完整程式碼

```

SELECT `classes`.`name` as '班級',
       `classes`.`tutor` as '班導師',
       COUNT(*) as '人數',
       AVG(`student_scores`.`score`) AS '班級平均',
       max(`student_scores`.`score`) AS '最高分',
       min(`student_scores`.`score`) AS '最低分',
       (SELECT AVG (`student_scores`.`score`) FROM
`student_scores`) AS '校平均',
CASE
    WHEN (SELECT AVG (`student_scores`.`score`) FROM

```



```

`student_scores`) <= AVG(`student_scores`.`score`) THEN '表現良好'
        WHEN (SELECT AVG (`student_scores`.`score`)
FROM `student_scores`) > AVG(`student_scores`.`score`) THEN '有待加強'

        END AS '評語'

FROM `class_student`,`student_scores`,`classes`
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
`class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`
ORDER BY `班級平均` DESC

```

執行結果

班級	班導師	人數	班級平均 ▼ 1	最高分	最低分	校平均	評語
一年八班	趙宜蓁	43	79.3488	88	69	73.5637	表現良好
一年九班	張建宏	42	77.9286	88	70	73.5637	表現良好
一年三班	賴冠廷	57	75.8070	89	56	73.5637	表現良好
一年六班	黃雅婷	48	75.3542	89	62	73.5637	表現良好
一年十班	許麗華	41	74.9024	86	64	73.5637	表現良好
一年一班	孫承翰	49	74.0612	88	60	73.5637	表現良好
一年二班	王詠晴	38	71.8947	87	60	73.5637	有待加強
一年七班	林志偉	47	70.8936	85	60	73.5637	有待加強
一年四班	郭欣妤	59	68.9492	88	60	73.5637	有待加強
一年五班	謝家豪	55	68.7636	87	60	73.5637	有待加強

※增加排名欄位(先讓他複製兩張表)

```

SELECT * FROM

        (SELECT `classes`.`name` as '班級',
        `classes`.`tutor` as '班導師',
        COUNT(*) as '人數',
        AVG(`student_scores`.`score`) AS '班級平均',
        max(`student_scores`.`score`) AS '最高分',
        min(`student_scores`.`score`) AS '最低分',
        (SELECT AVG (`student_scores`.`score`) FROM
`student_scores`) AS '校平均',

```

```

CASE
    WHEN (SELECT AVG (`student_scores`.`score`) FROM
`student_scores`) <= AVG(`student_scores`.`score`) THEN '表現良好'
    WHEN (SELECT AVG (`student_scores`.`score`)
FROM `student_scores`) > AVG(`student_scores`.`score`) THEN '有待加
強'
END AS '評語'
FROM `class_student`,`student_scores`,`classes`
WHERE `class_student`.`school_num`=`student_scores`.`school_num` &&
`class_student`.`class_code`=`classes`.`code`
GROUP BY `class_student`.`class_code`
ORDER BY `班級平均` DESC)A

```

執行結果

班級	班導師	人數	班級平均	最高分	最低分	校平均	評語
一年八班	趙宜蓁	43	79.3488	88	69	73.5637	表現良好
一年九班	張建宏	42	77.9286	88	70	73.5637	表現良好
一年三班	賴冠廷	57	75.8070	89	56	73.5637	表現良好
一年六班	黃雅婷	48	75.3542	89	62	73.5637	表現良好
一年十班	許麗華	41	74.9024	86	64	73.5637	表現良好
一年一班	孫承翰	49	74.0612	88	60	73.5637	表現良好
一年二班	王詠晴	38	71.8947	87	60	73.5637	有待加強
一年七班	林志偉	47	70.8936	85	60	73.5637	有待加強
一年四班	郭欣妤	59	68.9492	88	60	73.5637	有待加強
一年五班	謝家豪	55	68.7636	87	60	73.5637	有待加強

※增加排名欄位(chatGPT)

```

SELECT * ,
    (SELECT COUNT(*)
    FROM (
        SELECT
            AVG(ss.score) AS class_avg
        FROM
            class_student cs,

```

```

        student_scores ss,
        classes c
    WHERE
        cs.school_num = ss.school_num
        AND cs.class_code = c.code
    GROUP BY
        cs.class_code
    ) AS subquery
    WHERE subquery.class_avg >= A.班級平均
    ) AS '排名'
FROM (
    SELECT
        classes.name AS '班級',
        classes.tutor AS '班導師',
        COUNT(*) AS '人數',
        AVG(student_scores.score) AS '班級平均',
        MAX(student_scores.score) AS '最高分',
        MIN(student_scores.score) AS '最低分',
        (SELECT AVG(score) FROM student_scores) AS '校平均',
        CASE
            WHEN (SELECT AVG(score) FROM student_scores) <=
                AVG(student_scores.score) THEN '表現良好'
            WHEN (SELECT AVG(score) FROM student_scores) >
                AVG(student_scores.score) THEN '有待加強'
        END AS '評語'
    FROM
        class_student, student_scores, classes
    WHERE
        class_student.school_num = student_scores.school_num
        AND class_student.class_code = classes.code
    GROUP BY
        class_student.class_code

```

```
ORDER BY
```

```
'班級平均' DESC
```

```
) A;
```

執行結果

班級	班導師	人數	班級平均	最高分	最低分	校平均	評語	排名
一年一班	孫承翰	49	74.0612	88	60	73.5637	表現良好	6
一年二班	王詠晴	38	71.8947	87	60	73.5637	有待加強	7
一年三班	賴冠廷	57	75.8070	89	56	73.5637	表現良好	3
一年四班	郭欣妤	59	68.9492	88	60	73.5637	有待加強	9
一年五班	謝家豪	55	68.7636	87	60	73.5637	有待加強	10
一年六班	黃雅婷	48	75.3542	89	62	73.5637	表現良好	4
一年七班	林志偉	47	70.8936	85	60	73.5637	有待加強	8
一年八班	趙宜蓁	43	79.3488	88	69	73.5637	表現良好	1
一年九班	張建宏	42	77.9286	88	70	73.5637	表現良好	2
一年十班	許麗華	41	74.9024	86	64	73.5637	表現良好	5

目前比較新的做法

```
WITH ClassAvg AS (
```

```
SELECT
```

```
    c.name AS '班級',
```

```
    c.tutor AS '班導師',
```

```
    COUNT(*) AS '人數',
```

```
    AVG(ss.score) AS '班級平均',
```

```
    MAX(ss.score) AS '最高分',
```

```
    MIN(ss.score) AS '最低分',
```

```
    (SELECT AVG(score) FROM student_scores) AS '校平均',
```

```
    CASE
```

```
        WHEN (SELECT AVG(score) FROM student_scores) <=
             AVG(ss.score) THEN '表現良好'
```

```
        WHEN (SELECT AVG(score) FROM student_scores) >
             AVG(ss.score) THEN '有待加強'
```

```

        END AS '評語'
    FROM
        class_student cs
    JOIN student_scores ss ON cs.school_num = ss.school_num
    JOIN classes c ON cs.class_code = c.code
    GROUP BY cs.class_code
),
RankedClass AS (
    SELECT
        *,
        RANK() OVER (ORDER BY `班級平均` DESC) AS '排名'
    FROM ClassAvg
)
SELECT * FROM RankedClass;

```

←T→		▼	班級	班導師	人數	班級平均	最高分	最低分	校平均	評語	排名	
<input type="checkbox"/>	 編輯	 複製	 刪除	一年八班	趙宜蓁	43	79.3488	88	69	73.5637	表現良好	1
<input type="checkbox"/>	 編輯	 複製	 刪除	一年九班	張建宏	42	77.9286	88	70	73.5637	表現良好	2
<input type="checkbox"/>	 編輯	 複製	 刪除	一年三班	賴冠廷	57	75.8070	89	56	73.5637	表現良好	3
<input type="checkbox"/>	 編輯	 複製	 刪除	一年六班	黃雅婷	48	75.3542	89	62	73.5637	表現良好	4
<input type="checkbox"/>	 編輯	 複製	 刪除	一年十班	許麗華	41	74.9024	86	64	73.5637	表現良好	5
<input type="checkbox"/>	 編輯	 複製	 刪除	一年一班	孫承翰	49	74.0612	88	60	73.5637	表現良好	6
<input type="checkbox"/>	 編輯	 複製	 刪除	一年二班	王詠晴	38	71.8947	87	60	73.5637	有待加強	7
<input type="checkbox"/>	 編輯	 複製	 刪除	一年七班	林志偉	47	70.8936	85	60	73.5637	有待加強	8
<input type="checkbox"/>	 編輯	 複製	 刪除	一年四班	郭欣妤	59	68.9492	88	60	73.5637	有待加強	9
<input type="checkbox"/>	 編輯	 複製	 刪除	一年五班	謝家豪	55	68.7636	87	60	73.5637	有待加強	10

補充:某公司面試SQL題

hackmd.io/gdvLd4fgThaWifXbHsi7gA

DinBenDon 管理員登入 - 學員... 泰山龍訓網頁設計 HTML : 超文本標記... W3Schools Online... Lorem Picsum Laravel - The PHP... Adobe Acrobat 主選項目 | 月光林地...

HackMD 某公司面試SQL題 SQL

變更於 2 個月前

1 0 已訂閱

某公司面試SQL題

全部展開 回到頂部 移至底部

emp_no	dept_no	from_date	to_date
10001	d001	1986-06-26	9999-01-01
10002	d001	1996-08-03	9999-01-01
10003	d002	1996-08-03	9999-01-01

emp_no	salary	from_date	to_date
10001	88958	2002-06-22	9999-01-01
10002	72527	2001-08-02	9999-01-01
10003	92527	2001-08-02	9999-01-01

1. 查詢員工編號10001的薪資

```
1 SELECT `emp_no`,`salary`
2 FROM `salaries`
3 WHERE `emp_no`='10001'
```

2. 刪除部門d001的員工資料 (V)

以下語法可以同時對兩張資料表的內容做刪除

```
1 DELETE `dept_emp`,
2 `salaries`
3 FROM `salaries`
4 LEFT JOIN `dept_emp`
5 ON `dept_emp`.`emp_no`=`salaries`.`emp_no`
6 WHERE `dept_emp`.`dept_no`='d001';
```

3. 修改員工10002的薪資為66666

```
1 UPDATE `salaries`
2 SET `salary`=66666
3 WHERE `emp_no`='10002'
```