

練習題詳解

01. 從 `twElection2020` 資料庫的 `admin_regions` 資料表選擇所有變數，並且使用 `LIMIT 5` 顯示前五列資料，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(5, 4) 的查詢結果。

```
id county town village
0 1 南投縣 中寮鄉 中寮村
1 2 南投縣 中寮鄉 內城村
2 3 南投縣 中寮鄉 八仙村
3 4 南投縣 中寮鄉 和興村
4 5 南投縣 中寮鄉 崁頂村
```

```
SELECT *
FROM admin_regions
LIMIT 5;
```

02. 從 `nba` 資料庫的球隊資料表 `teams` 中選擇 `confName`、`divName`、`fullName` 三個變數，並且使用 `LIMIT 10` 顯示前十列資料，參考下列預期的查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 3) 的查詢結果。

```
confName divName fullName
0 East Southeast Atlanta Hawks
1 East Atlantic Boston Celtics
2 East Central Cleveland Cavaliers
3 West Southwest New Orleans Pelicans
4 East Central Chicago Bulls
5 West Southwest Dallas Mavericks
6 West Northwest Denver Nuggets
7 West Pacific Golden State Warriors
8 West Southwest Houston Rockets
9 West Pacific LA Clippers
```

```
SELECT confName,
divName,
fullName
FROM teams
LIMIT 10;
```

03. 從 `nba` 資料庫的球員資料表 `players` 中選擇 `firstName`、`lastName` 兩個變數，並依序取別名為 `first_name`、`last_name`，使用 `LIMIT 5` 顯示前五列資料，參考下列預期的查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(5, 2) 的查詢結果。

```
first_name last_name
0 LeBron James
1 Carmelo Anthony
2 Udonis Haslem
3 Dwight Howard
4 Andre Iguodala
```

```
SELECT firstName AS first_name,
lastName AS last_name
FROM players
LIMIT 5;
```

04. 從 `twElection2020` 資料庫的 `admin_regions` 資料表選擇「不重複」的縣市（`county`），參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(22, 1) 的查詢結果。

```
distinct_counties
0      南投縣
1      嘉義市
2      嘉義縣
3      基隆市
4      宜蘭縣
5      屏東縣
6      彰化縣
7      新北市
8      新竹市
9      新竹縣
10     桃園市
11     澎湖縣
12     臺中市
13     臺北市
14     臺南市
15     臺東縣
16     花蓮縣
17     苗栗縣
18     連江縣
19     金門縣
20     雲林縣
21     高雄市
```

```
SELECT DISTINCT county AS distinct_counties
FROM admin_regions;
```

05. 從 `nba` 資料庫的 `teams` 資料表選擇「不重複」的分組（`divName`），參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(6, 1) 的查詢結果。

```
distinct_divisions
0      Southeast
1      Atlantic
2      Central
3      Southwest
4      Northwest
5      Pacific
```

```
SELECT DISTINCT divName AS distinct_divisions
FROM teams;
```

06. 從 `covid19` 資料庫的 `daily_report` 資料表根據 `Confirmed`、`Recovered`、`Deaths` 欄位以及下列公式衍生計算欄位 `Active`，參考下列的預期查詢結果。

$$\text{Active} = \text{Confirmed} - \text{Recovered} - \text{Deaths} \quad (1)$$

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3981, 4) 的查詢結果。

```
Confirmed Recovered Deaths Active
0      56454   51550.0   2484   2420.0
1     125157   91271.0   2235  31651.0
```

2	117192	81538.0	3093	32561.0
3	12010	11315.0	115	580.0
4	22311	20493.0	537	1281.0
...
3976	2603	2359.0	35	209.0
3977	242353	215429.0	2627	24297.0
3978	4357	1676.0	888	1793.0
3979	88418	84592.0	1208	2618.0
3980	36882	34686.0	1523	673.0

[3981 rows x 4 columns]

```
SELECT Confirmed,
       Recovered,
       Deaths,
       Confirmed - Recovered - Deaths AS Active
FROM daily_report;
```

07. 從 `nba` 資料庫的 `players` 資料表依據 `heightMeters`、`weightKilograms` 欄位以及下列公式衍生計算欄位 `bmi`，參考下列的預期查詢結果。

$$BMI = \frac{weight_{kg}}{height_m^2} \quad (2)$$

- 預期輸入：SQL 查詢語法。
- 預期輸出：(484, 3) 的查詢結果。

	heightMeters	weightKilograms	bmi
0	2.06	113.4	26.722594
1	2.01	108.0	26.732012
2	2.03	106.6	25.868136
3	2.08	120.2	27.782914
4	1.98	97.5	24.869911
..
479	2.01	104.3	25.816193
480	2.08	106.1	24.523854
481	1.78	88.5	27.932079
482	1.98	90.7	23.135394
483	1.96	83.9	21.839858

[484 rows x 3 columns]

```
SELECT heightMeters,
       weightKilograms,
       weightKilograms / (heightMeters*heightMeters) AS bmi
FROM players;
```

08. 從 `nba` 資料庫的 `teams` 資料表連接 `confName`、`divName` 欄位後使用 `DISTINCT` 去除重複值，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(6, 1) 的查詢結果。

	conf_div
0	East, Southeast
1	East, Atlantic
2	East, Central
3	West, Southwest
4	West, Northwest
5	West, Pacific

```
SELECT DISTINCT confName || ', ' || divName AS conf_div
FROM teams;
```

09. 從 `nba` 資料庫的 `players` 資料表依據 `heightMeters`、`weightKilograms` 以及下列公式衍生計算欄位 `bmi`，並使用 `ROUND` 函數將 `bmi` 的小數點位數調整為 2 位，參考下列的預期查詢結果。

$$BMI = \frac{weight_{kg}}{height_m^2} \quad (1)$$

- 預期輸入：SQL 查詢語法。
- 預期輸出：(484, 3) 的查詢結果。

	heightMeters	weightKilograms	bmi
0	2.06	113.4	26.72
1	2.01	108.0	26.73
2	2.03	106.6	25.87
3	2.08	120.2	27.78
4	1.98	97.5	24.87
..
479	2.01	104.3	25.82
480	2.08	106.1	24.52
481	1.78	88.5	27.93
482	1.98	90.7	23.14
483	1.96	83.9	21.84

[484 rows x 3 columns]

```
SELECT heightMeters,
       weightKilograms,
       ROUND(weightKilograms / (heightMeters*heightMeters), 2) AS bmi
FROM players;
```

10. 從 `nba` 資料庫的 `career_summaries` 資料表中依據 `assists`、`turnovers` 欄位以及下列公式衍生計算助攻失誤比，使用 `CAST` 函數讓衍生計算欄位的資料類型為浮點數 `REAL`，參考下列的預期查詢結果。

$$\text{Assist Turnover Ratio} = \frac{\text{Assists}}{\text{Turnovers}} \quad (2)$$

PS 在練習題預期的查詢結果看到 `NaN` 或者 `None` 代表的就是遺漏值 `NULL`。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(484, 3) 的查詢結果。

	assists	turnovers	assist_turnover_ratio
0	9669.0	4576.0	2.112981
1	3327.0	2981.0	1.116068
2	729.0	804.0	0.906716
3	1615.0	3225.0	0.500775
4	4965.0	2180.0	2.277523
..
479	4.0	3.0	1.333333
480	0.0	1.0	0.000000
481	112.0	39.0	2.871795
482	1.0	0.0	NaN
483	12.0	7.0	1.714286

[484 rows x 3 columns]

```
SELECT assists,
       turnovers,
       CAST(assists AS REAL) / turnovers AS assist_turnover_ratio
FROM career_summaries;
```

11. 從 `covid19` 資料庫的 `time_series` 資料表依據 `Date` 變數，使用 `STRFTIME` 函數查詢時間序列資料有哪些不重複的月份，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(15, 1) 的查詢結果。

```
distinct_year_month
0      2020-01
1      2020-02
2      2020-03
3      2020-04
4      2020-05
5      2020-06
6      2020-07
7      2020-08
8      2020-09
9      2020-10
10     2020-11
11     2020-12
12     2021-01
13     2021-02
14     2021-03
```

```
SELECT DISTINCT STRFTIME('%Y', Date) || '-' || STRFTIME('%m', Date) AS distinct_year_month
FROM time_series;
```

12. 從 `twElection2020` 資料庫的 `presidential` 資料表利用聚合函數彙總有多少人參與了總統副總統的投票選舉，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(1, 1) 的查詢結果。

```
total_presidential_votes
0      14300940
```

```
SELECT SUM(votes) AS total_presidential_votes
FROM presidential;
```

13. 從 `covid19` 資料庫的 `daily_report` 資料表利用聚合函數彙總截至 2021-03-31 全世界總確診數、總痊癒數以及總死亡數，參考下列的預期查詢結果。

註：本題不需考慮 `daily_report` 內的 `Last_Update` 時間戳記，`daily_report` 的數據有效期間就是 2021-03-31。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(1, 3) 的查詢結果。

```
total_confirmed total_recovered total_deaths
0      128822735      73070921      2815166
```

```
SELECT SUM(Confirmed) AS total_confirmed,
       SUM(Recovered) AS total_recovered,
       SUM(Deaths) AS total_deaths
FROM daily_report;
```

14. 從 `nba` 資料庫的 `career_summaries` 資料表中依據 `ppg` (Points per game, 場均得分) 找出場均得分最高的 10 名球員，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 2) 的查詢結果。

	personId	ppg
0	201142	27.1
1	2544	27.0
2	1629029	25.6
3	201935	25.2
4	1629627	25.0
5	203954	24.7
6	203081	24.6
7	1629027	24.0
8	203076	23.9
9	201939	23.8

```
SELECT personId,
       ppg
FROM career_summaries
ORDER BY ppg DESC
LIMIT 10;
```

15. 從 `covid19` 資料庫的 `time_series` 資料表中依據 `Daily_Cases` 找出前十個單日新增確診數最多的日期，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 3) 的查詢結果。

	Date	Country_Region	Daily_Cases
0	2020-12-10	Turkey	823225
1	2021-01-02	US	300308
2	2021-01-08	US	293268
3	2021-01-07	US	278261
4	2021-01-09	US	262622
5	2021-01-06	US	255424
6	2020-12-18	US	251961
7	2020-12-16	US	246702
8	2021-01-15	US	242734
9	2020-12-11	US	239976

```
SELECT Date,
       Country_Region,
       Daily_Cases
FROM time_series
ORDER BY Daily_Cases DESC
LIMIT 10;
```

16. 從 `nba` 資料庫的 `career_summaries` 資料表中依據 `assists`、`turnovers` 欄位以及下列公式衍生計算助攻失誤比，使用 `CAST` 函數讓衍生計算欄位的資料類型為浮點數 `REAL`，找出助攻失誤比最高的前 10 名球員，參考下列的預期查詢結果。

$$\text{Assists Turnover Ratio} = \frac{\text{Assists}}{\text{Turnovers}} \quad (1)$$

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 3) 的查詢結果。

	personId	assists	turnovers
0	1630216	10	1
1	1630184	69	13
2	1628420	682	131
3	1629657	20	4
4	1626145	1292	269
5	1629162	252	62
6	1629603	4	1
7	101108	10051	2572
8	1630201	30	8
9	1627853	441	120

```
SELECT personId,
       assists,
       turnovers
FROM career_summaries
ORDER BY CAST(assists AS REAL) / turnovers DESC
LIMIT 10;
```

17. 從 `covid19` 資料庫的 `time_series` 資料表將台灣的觀測值篩選出來，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(435, 4) 的查詢結果。

	Country_Region	Date	Confirmed	Daily_Cases
0	Taiwan	2020-01-22	1	1
1	Taiwan	2020-01-23	1	0
2	Taiwan	2020-01-24	3	2
3	Taiwan	2020-01-25	3	0
4	Taiwan	2020-01-26	4	1
..
430	Taiwan	2021-03-27	1020	7
431	Taiwan	2021-03-28	1022	2
432	Taiwan	2021-03-29	1023	1
433	Taiwan	2021-03-30	1024	1
434	Taiwan	2021-03-31	1030	6

[435 rows x 4 columns]

```
SELECT Country_Region,
       Date,
       Confirmed,
       Daily_Cases
FROM time_series
WHERE Country_Region = 'Taiwan';
```

18. 從 `imdb` 資料庫的 `movies` 資料表將上映年份為 1994 的電影篩選出來，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(5, 4) 的查詢結果。

	title	rating	director	runtime
0	The Shawshank Redemption	9.3	Frank Darabont	142
1	Pulp Fiction	8.9	Quentin Tarantino	154
2	Forrest Gump	8.8	Robert Zemeckis	142
3	Léon: The Professional	8.5	Luc Besson	110
4	The Lion King	8.5	Roger Allers	88

```
SELECT title,
       rating,
       director,
       runtime
FROM movies
WHERE release_year = 1994;
```

19. 從 `imdb` 資料庫的 `actors` 資料表將 Tom Hanks、Christian Bale、Leonardo DiCaprio 篩選出來，參考下列的預期查詢結果。

PS Tom Hanks 是一位著名的美國男演員及電視製片人，以演技精湛而著稱。他是歷史上第2位連續兩屆獲得奧斯卡金像獎最佳男主角獎的演員，亦是最年輕獲得美國電影學會終身成就獎的演員。Christian Bale 是一名英國男演員和電影製片人，在蝙蝠俠三部曲中飾演 Bruce Wayne 獲得了廣泛讚揚及商業認可。Leonardo DiCaprio 是一位美國著名男演員、電影製片人兼環保概念的推動者，出演了由史詩愛情片鐵達尼號知名度大開。 Source: Wikipedia

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3, 2) 的查詢結果。

	id	name
0	502	Christian Bale
1	1773	Leonardo DiCaprio
2	2865	Tom Hanks

```
SELECT *
FROM actors
WHERE name IN ('Tom Hanks', 'Christian Bale', 'Leonardo DiCaprio');
```

20. 從 `imdb` 資料庫的 `movies` 資料表篩選出由 Christopher Nolan 或 Peter Jackson 所導演的電影，參考下列的預期查詢結果。

PS Christopher Nolan 是一名英國導演、編劇及監製，他的十部電影在全球獲得超過 47 億美元的票房，執導著名電影包含「黑暗騎士三部曲」、全面啟動、星際效應及敦克爾克大行動；Peter Jackson 是一名紐西蘭導演、編劇及監製，執導最出名的作品是「魔戒電影三部曲」與「哈比人電影系列」。 Source: Wikipedia

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 2) 的查詢結果。

	title	director
0	The Dark Knight	Christopher Nolan
1	Inception	Christopher Nolan
2	Interstellar	Christopher Nolan
3	The Prestige	Christopher Nolan
4	Memento	Christopher Nolan
5	The Dark Knight Rises	Christopher Nolan
6	Batman Begins	Christopher Nolan
7	The Lord of the Rings: The Return of the King	Peter Jackson
8	The Lord of the Rings: The Fellowship of the Ring	Peter Jackson
9	The Lord of the Rings: The Two Towers	Peter Jackson

```
SELECT title,
director
FROM movies
WHERE director IN ('Christopher Nolan', 'Peter Jackson')
ORDER BY director;
```

21. 從 `covid19` 資料庫的 `lookup_table` 資料表篩選出 `Country_Region` 名稱有 land 的國家，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 1) 的查詢結果。

	Country_Region
0	Finland
1	Iceland
2	Ireland
3	Marshall Islands
4	Netherlands
5	New Zealand
6	Poland
7	Solomon Islands
8	Switzerland
9	Thailand

```
SELECT DISTINCT Country_Region
FROM lookup_table
WHERE Country_Region LIKE '%land%';
```


22. 從 `covid19` 資料庫的 `daily_report` 資料表將「美國」與「非美國」的觀測值用衍生計算欄位區分，美國的觀測值給予整數 1、非美國的觀測值給予整數 0，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3981, 2) 的查詢結果。

```

Combined_Key  is_us
0      Abruzzo, Italy      0
1      Acre, Brazil      0
2      Adygea Republic, Russia      0
3      Afghanistan      0
4      Aguascalientes, Mexico      0
...      ...      ...
3976      Yuma, Arizona, US      1
3977      Yuma, Colorado, US      1
3978      Zapata, Texas, US      1
3979      Zavala, Texas, US      1
3980      Ziebach, South Dakota, US      1
[3981 rows x 2 columns]
```

```

SELECT Combined_Key,
       CASE WHEN Combined_Key LIKE '%, US' THEN 1
            ELSE 0 END AS is_us
FROM daily_report
ORDER BY is_us;
```

23. 從 `imdb` 資料庫的 `movies` 資料表將評等超過 8.7 (>8.7) 的電影分類為 'Awesome'、將評等超過 8.4 (>8.4) 的電影分類為 'Terrific'，再將其餘的電影分類為 'Great'，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(250, 3) 的查詢結果。

```

title  rating  rating_category
0      The Shawshank Redemption      9.3      Awesome
1      The Godfather      9.2      Awesome
2      The Godfather: Part II      9.0      Awesome
3      The Dark Knight      9.0      Awesome
4      12 Angry Men      9.0      Awesome
..      ...      ...
245      Neon Genesis Evangelion: The End of Evangelion      8.1      Great
246      7 Kogustaki Mucize      8.2      Great
247      Tangerines      8.2      Great
248      Drishyam      8.2      Great
249      Swades      8.2      Great
[250 rows x 3 columns]
```

```

SELECT title,
       rating,
       CASE WHEN rating > 8.7 THEN 'Awesome'
            WHEN rating > 8.4 THEN 'Terrific'
            ELSE 'Great' END AS rating_category
FROM movies;
```

24. 從 `twElection2020` 資料庫的 `admin_regions` 資料表將 `county` 分類為 '六都' 與 '非六都'，參考下列的預期查詢結果。

PS 六都為臺北市、新北市、桃園市、臺中市、臺南市與高雄市。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(22, 2) 的查詢結果。

	county	county_type
0	新北市	六都
1	桃園市	六都
2	臺中市	六都
3	臺北市	六都
4	臺南市	六都
5	高雄市	六都
6	南投縣	非六都
7	嘉義市	非六都
8	嘉義縣	非六都
9	基隆市	非六都
10	宜蘭縣	非六都
11	屏東縣	非六都
12	彰化縣	非六都
13	新竹市	非六都
14	新竹縣	非六都
15	澎湖縣	非六都
16	臺東縣	非六都
17	花蓮縣	非六都
18	苗栗縣	非六都
19	連江縣	非六都
20	金門縣	非六都
21	雲林縣	非六都

```
SELECT DISTINCT county,
       CASE WHEN county IN ('臺北市', '新北市', '桃園市', '臺中市', '臺南市', '高雄市') THEN '六都'
            ELSE '非六都' END AS county_type
FROM admin_regions
ORDER BY county_type,
       county;
```

25. 從 `imdb` 資料庫的 `movies` 資料表計算每一年有幾部在 IMDb.com 獲得高評等的經典電影，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(85, 2) 的查詢結果。

	release_year	number_of_movies
0	1921	1
1	1924	1
2	1925	1
3	1926	1
4	1927	1
..
80	2017	3
81	2018	6
82	2019	8
83	2020	2
84	2021	1

[85 rows x 2 columns]

```
SELECT release_year,
       COUNT(*) AS number_of_movies
FROM movies
GROUP BY release_year
ORDER BY release_year;
```

26. 從 `imdb` 資料庫的 `movies` 資料表計算每一年有幾部在 IMDb.com 獲得高評等的經典電影，只顯示電影數在 5 部以上 (`>= 5`) 的年份，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(19, 2) 的查詢結果。

	release_year	number_of_movies
0	1957	6
1	1988	5
2	1994	5
3	1995	8

4	1997	6
5	1998	5
6	1999	5
7	2000	6
8	2001	5
9	2003	5
10	2004	7
11	2006	5
12	2009	6
13	2010	5
14	2013	6
15	2014	5
16	2015	5
17	2018	6
18	2019	8

```
SELECT release_year,
       COUNT(*) AS number_of_movies
FROM movies
GROUP BY release_year
HAVING number_of_movies >= 5
ORDER BY release_year;
```

27. 從 **twElection2020** 資料庫的 **presidential** 資料表瞭解台灣 2020 總統副總統的選舉結果，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3, 2) 的查詢結果。

candidate_id	total_votes
0	1 608590
1	2 5522119
2	3 8170231

```
SELECT candidate_id,
       SUM(votes) AS total_votes
FROM presidential
GROUP BY candidate_id;
```

28. 從 **nba** 資料庫的 **players** 資料表根據 **country** 瞭解截至 2021-03-31，NBA 由哪些國家的球員所組成，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(43, 2) 的查詢結果。

	country	number_of_players
0	USA	370
1	Canada	17
2	France	11
3	Australia	9
4	Germany	6
5	Serbia	6
6	Croatia	4
7	Spain	4
8	Turkey	4
9	Greece	3
10	Italy	3
11	Latvia	3
12	Lithuania	3
13	Nigeria	3
14	Senegal	3
15	Slovenia	3
16	Bahamas	2
17	Brazil	2
18	Cameroon	2
19	Japan	2
20	Ukraine	2
21	Angola	1
22	Argentina	1
23	Austria	1
24	Bosnia and Herzegovina	1

25	Czech Republic	1
26	DRC	1
27	Dominican Republic	1
28	Egypt	1
29	Finland	1
30	Gabon	1
31	Georgia	1
32	Guinea	1
33	Israel	1
34	Jamaica	1
35	Montenegro	1
36	New Zealand	1
37	Republic of the Congo	1
38	Saint Lucia	1
39	South Sudan	1
40	Sudan	1
41	Switzerland	1
42	United Kingdom	1

```
SELECT country,
       COUNT(*) AS number_of_players
FROM players
GROUP BY country
ORDER BY number_of_players DESC;
```

29. 從 **nba** 資料庫的 **players** 資料表根據 **country** 瞭解截至 2021-03-31, NBA 由哪些國家的球員所組成, 只顯示球員數在 2 位以上 (≥ 2) 並在 9 位以下 (≤ 9) 的國家, 參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(18, 2) 的查詢結果。

	country	number_of_players
0	Australia	9
1	Germany	6
2	Serbia	6
3	Croatia	4
4	Spain	4
5	Turkey	4
6	Greece	3
7	Italy	3
8	Latvia	3
9	Lithuania	3
10	Nigeria	3
11	Senegal	3
12	Slovenia	3
13	Bahamas	2
14	Brazil	2
15	Cameroon	2
16	Japan	2
17	Ukraine	2

```
SELECT country,
       COUNT(*) AS number_of_players
FROM players
GROUP BY country
HAVING number_of_players BETWEEN 2 and 9
ORDER BY number_of_players DESC;
```

30. 從 **nba** 資料庫的 **players** 資料表運用子查詢找出 NBA 中身高最高與最矮的球員是誰, 參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(5, 3) 的查詢結果。

	firstName	lastName	heightMeters
0	Tacko	Fall	2.26
1	Jared	Harper	1.78
2	Tremont	Waters	1.78

3	Markus	Howard	1.78
4	Facundo	Campazzo	1.78

```
SELECT firstName,
       lastName,
       heightMeters
FROM players
WHERE heightMeters = (SELECT MAX(heightMeters) FROM players) OR
       heightMeters = (SELECT MIN(heightMeters) FROM players);
```

31. 從 **nba** 資料庫的 **players** 資料表運用子查詢計算球員的國籍佔比，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(43, 2) 的查詢結果。

	country	player_percentage
0	USA	0.764463
1	Canada	0.035124
2	France	0.022727
3	Australia	0.018595
4	Germany	0.012397
5	Serbia	0.012397
6	Croatia	0.008264
7	Spain	0.008264
8	Turkey	0.008264
9	Greece	0.006198
10	Italy	0.006198
11	Latvia	0.006198
12	Lithuania	0.006198
13	Nigeria	0.006198
14	Senegal	0.006198
15	Slovenia	0.006198
16	Bahamas	0.004132
17	Brazil	0.004132
18	Cameroon	0.004132
19	Japan	0.004132
20	Ukraine	0.004132
21	Angola	0.002066
22	Argentina	0.002066
23	Austria	0.002066
24	Bosnia and Herzegovina	0.002066
25	Czech Republic	0.002066
26	DRC	0.002066
27	Dominican Republic	0.002066
28	Egypt	0.002066
29	Finland	0.002066
30	Gabon	0.002066
31	Georgia	0.002066
32	Guinea	0.002066
33	Israel	0.002066
34	Jamaica	0.002066
35	Montenegro	0.002066
36	New Zealand	0.002066
37	Republic of the Congo	0.002066
38	Saint Lucia	0.002066
39	South Sudan	0.002066
40	Sudan	0.002066
41	Switzerland	0.002066
42	United Kingdom	0.002066

```
SELECT country,
       CAST(COUNT(*) AS REAL) / (SELECT COUNT(*)
                                FROM players) AS player_percentage
FROM players
GROUP BY country
ORDER BY player_percentage DESC,
         country;
```

32. 從 **nba** 資料庫運用子查詢找出 NBA 的場均得分王 (**ppg**)，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。

- 預期輸出：(1, 2) 的查詢結果。

```

firstName lastName
0      Kevin  Durant

```

```

SELECT firstName,
       lastName
FROM players
WHERE personId = (SELECT personId
                  FROM career_summaries
                  WHERE ppg = (SELECT MAX(ppg)
                              FROM career_summaries));

```

33. 從 **nba** 資料庫運用子查詢找出目前布魯克林籃網隊（Brooklyn Nets）的球員名單，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(17, 2) 的查詢結果。

```

firstName      lastName
0  LaMarcus    Aldridge
1   Kevin      Durant
2   Jeff       Green
3  DeAndre     Jordan
4   Blake      Griffin
5   James      Harden
6   Kyrie      Irving
7  Spencer     Dinwiddie
8    Joe       Harris
9   Tyler      Johnson
10 Timothee    Luwawu-Cabarrot
11 Bruce       Brown
12 Alize       Johnson
13 Landry      Shamet
14 Chris       Chiozza
15 Reggie     Perry
16 Nicolas     Claxton

```

```

SELECT firstName,
       lastName
FROM players
WHERE teamId = (SELECT teamId
                FROM teams
                WHERE nickname = 'Nets');

```

34. 從 **twElection2020** 資料庫的 **presidential** 資料表計算各組候選人的得票率，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3, 2) 的查詢結果。

```

candidate_id votes_percentage
0            1           4.26%
1            2          38.61%
2            3          57.13%

```

```

SELECT candidate_id,
       CAST(ROUND(CAST(SUM(votes) AS REAL) / (SELECT SUM(votes) FROM presidential), 4) * 100 AS TEXT) || '%' AS votes_percentage
FROM presidential
GROUP BY candidate_id;

```

35. 從 `covid19` 資料庫查詢截至 2021-03-31 全球前十大確診人數的國家，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 2) 的查詢結果。

	Country_Region	total_confirmed
0	US	30459802
1	Brazil	12748747
2	India	12149335
3	France	4705068
4	Russia	4494234
5	United Kingdom	4359982
6	Italy	3584899
7	Turkey	3317182
8	Spain	3284353
9	Germany	2843644

```
SELECT lookup_table.Country_Region,
       SUM(daily_report.Confirmed) AS total_confirmed
FROM daily_report
JOIN lookup_table
  ON daily_report.Combined_Key = lookup_table.Combined_Key
GROUP BY lookup_table.Country_Region
ORDER BY total_confirmed DESC
LIMIT 10;
```

36. 從 `twElection2020` 資料庫查詢中國國民黨、民主進步黨與親民黨在不分區立委與區域立委的得票率，參考下列的預期查詢結果。

PS 不分區立委的投票資料記錄於資料表 `legislative_at_large`，區域立委的投票資料記錄於資料表 `legislative_regional`。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(6, 3) 的查詢結果。

	party	election	votes_percentage
0	中國國民黨	不分區立委	0.3336
1	民主進步黨	不分區立委	0.3398
2	親民黨	不分區立委	0.0366
3	中國國民黨	區域立委	0.4071
4	民主進步黨	區域立委	0.4511
5	親民黨	區域立委	0.0043

```
SELECT parties.party,
       '不分區立委' AS election,
       ROUND(CAST(SUM(votes) AS REAL) / (SELECT SUM(votes) FROM legislative_at_large), 4) AS votes_percentage
FROM legislative_at_large
JOIN parties
  ON legislative_at_large.party_id = parties.id
WHERE parties.party IN ('中國國民黨', '民主進步黨', '親民黨')
GROUP BY legislative_at_large.party_id
UNION
SELECT parties.party,
       '區域立委' AS election,
       ROUND(CAST(SUM(votes) AS REAL) / (SELECT SUM(votes) FROM legislative_regional), 4) AS votes_percentage
FROM legislative_regional
JOIN candidates
  ON legislative_regional.candidate_id = candidates.id
JOIN parties
  ON candidates.party_id = parties.id
WHERE parties.party IN ('中國國民黨', '民主進步黨', '親民黨')
GROUP BY parties.id
ORDER BY election;
```

37. 從 **nba** 資料庫查詢截至 2021-03-31 洛杉磯湖人隊（Los Angeles Lakers）球員的生涯場均得分 **ppg**，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(16, 3) 的查詢結果。

	team_name	player_name	ppg
0	Los Angeles Lakers	LeBron James	27.0
1	Los Angeles Lakers	Anthony Davis	23.9
2	Los Angeles Lakers	Kyle Kuzma	15.4
3	Los Angeles Lakers	Andre Drummond	14.6
4	Los Angeles Lakers	Marc Gasol	14.2
5	Los Angeles Lakers	Dennis Schroder	14.2
6	Los Angeles Lakers	Montrezl Harrell	13.0
7	Los Angeles Lakers	Wesley Matthews	12.7
8	Los Angeles Lakers	Kentavious Caldwell-Pope	11.3
9	Los Angeles Lakers	Markieff Morris	11.1
10	Los Angeles Lakers	Talen Horton-Tucker	7.6
11	Los Angeles Lakers	Jared Dudley	7.3
12	Los Angeles Lakers	Alex Caruso	5.6
13	Los Angeles Lakers	Alfonzo McKinnie	4.0
14	Los Angeles Lakers	Devontae Cacok	2.2
15	Los Angeles Lakers	Kostas Antetokounmpo	1.1

```
SELECT teams.fullName AS team_name,
       players.firstName || ' ' || players.lastName AS player_name,
       career_summaries.ppg
FROM players
JOIN teams
  ON players.teamId = teams.teamId
JOIN career_summaries
  ON players.personId = career_summaries.personId
WHERE teams.nickname = 'Lakers'
ORDER BY career_summaries.ppg DESC;
```

38. 從 **nba** 資料庫查詢各個球隊的得分王（生涯場均得分 **ppg** 全隊最高）是誰，將查詢結果依隊伍名排序，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(30, 3) 的查詢結果。

	team	player	ppg
0	Atlanta Hawks	Trae Young	24.0
1	Boston Celtics	Kemba Walker	19.8
2	Brooklyn Nets	Kevin Durant	27.1
3	Charlotte Hornets	LaMelo Ball	15.9
4	Chicago Bulls	Zach LaVine	18.8
5	Cleveland Cavaliers	Collin Sexton	19.7
6	Dallas Mavericks	Luka Doncic	25.6
7	Denver Nuggets	Nikola Jokic	18.0
8	Detroit Pistons	Josh Jackson	12.0
9	Golden State Warriors	Stephen Curry	23.8
10	Houston Rockets	John Wall	19.0
11	Indiana Pacers	T.J. Warren	15.5
12	LA Clippers	Paul George	20.1
13	Los Angeles Lakers	LeBron James	27.0
14	Memphis Grizzlies	Ja Morant	18.4
15	Miami Heat	Victor Oladipo	17.6
16	Milwaukee Bucks	Giannis Antetokounmpo	20.7
17	Minnesota Timberwolves	Karl-Anthony Towns	22.8
18	New Orleans Pelicans	Zion Williamson	25.0
19	New York Knicks	Derrick Rose	18.5
20	Oklahoma City Thunder	Shai Gilgeous-Alexander	16.3
21	Orlando Magic	Gary Harris	12.0
22	Philadelphia 76ers	Joel Embiid	24.7
23	Phoenix Suns	Devin Booker	22.8
24	Portland Trail Blazers	Damian Lillard	24.6
25	Sacramento Kings	De'Aaron Fox	17.8
26	San Antonio Spurs	DeMar DeRozan	20.1
27	Toronto Raptors	Kyle Lowry	14.8
28	Utah Jazz	Donovan Mitchell	23.2
29	Washington Wizards	Russell Westbrook	23.2


```

SELECT teams.fullName AS team,
       players.firstName || ' ' || players.lastName AS player,
       MAX(career_summaries.ppg) AS ppg
FROM players
JOIN teams
  ON players.teamId = teams.teamId
JOIN career_summaries
  ON players.personId = career_summaries.personId
GROUP BY teams.fullName
ORDER BY teams.fullName;

```

39. 從 `imdb` 資料庫中查詢 Tom Hanks 與 Leonardo DiCaprio 在 IMDb.com 最高評價的 250 部電影中演出哪些電影，依據 `casting` 資料表中的 `ord` 衍生計算欄位 `is_lead_actor` 註記是否為第一主角（`ord` 若為 1 表示為第一主角），將查詢結果依 `release_year` 排序，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(12, 4) 的查詢結果。

	release_year	title	name	is_lead_actor
0	1994	Forrest Gump	Tom Hanks	1
1	1995	Toy Story	Tom Hanks	1
2	1998	Saving Private Ryan	Tom Hanks	1
3	1999	The Green Mile	Tom Hanks	1
4	2002	Catch Me If You Can	Leonardo DiCaprio	1
5	2002	Catch Me If You Can	Tom Hanks	0
6	2006	The Departed	Leonardo DiCaprio	1
7	2010	Inception	Leonardo DiCaprio	1
8	2010	Toy Story 3	Tom Hanks	1
9	2010	Shutter Island	Leonardo DiCaprio	1
10	2012	Django Unchained	Leonardo DiCaprio	0
11	2013	The Wolf of Wall Street	Leonardo DiCaprio	1

```

SELECT movies.release_year,
       movies.title,
       actors.name,
       CASE WHEN casting.ord = 1 THEN 1
            ELSE 0 END AS is_lead_actor
FROM actors
JOIN casting
  ON actors.id = casting.actor_id
JOIN movies
  ON casting.movie_id = movies.id
WHERE actors.name IN ('Tom Hanks', 'Leonardo DiCaprio')
ORDER BY movies.release_year;

```

40. 從 `covid19` 資料庫建立一個虛擬資料表名為 `total_confirmed_by_country_region` 記錄截至 2021-03-31 全球各國的確診人數，參考下列的預期輸出。

- 預期輸入：SQL 語法。
- 預期輸出：(192, 2) 的虛擬資料表 `total_confirmed_by_country_region`。

```
SELECT * FROM total_confirmed_by_country_region;
```

	Country_Region	total_confirmed
0	Afghanistan	56454
1	Albania	125157
2	Algeria	117192
3	Andorra	12010
4	Angola	22311
..
187	Vietnam	2603
188	West Bank and Gaza	242353
189	Yemen	4357
190	Zambia	88418

```
191         Zimbabwe        36882

[192 rows x 2 columns]
```

```
CREATE VIEW covid19.total_confirmed_by_country_region (Country_Region, total_confirmed)
AS
SELECT lookup_table.Country_Region,
       SUM(daily_report.Confirmed) AS total_confirmed
FROM daily_report
JOIN lookup_table
  ON daily_report.Combined_Key = lookup_table.Combined_Key
GROUP BY lookup_table.Country_Region;
```

41. 從 `twElection2020` 資料庫建立一個虛擬資料表名為 `presidential_total_votes` 記錄三組候選人的總得票數，參考下列的預期輸出。

- 預期輸入：SQL 語法。
- 預期輸出：(3, 3) 的虛擬資料表 `presidential_total_votes`。

```
SELECT * FROM presidential_total_votes;
```

```
number  candidate  total_votes
0       1  宋楚瑜/余湘    608590
1       2  韓國瑜/張善政    5522119
2       3  蔡英文/賴清德    8170231
```

```
CREATE VIEW twElection2020.presidential_total_votes (number, candidate, total_votes)
AS
SELECT candidates.number,
       candidates.candidate,
       SUM(presidential.votes) AS total_votes
FROM presidential
JOIN candidates
  ON presidential.candidate_id = candidates.id
GROUP BY presidential.candidate_id;
```

42. 從 `nba` 資料庫建立一個虛擬資料表名為 `ppg_leader_by_teams` 紀錄各個球隊的得分王（生涯場均得分 `ppg` 全隊最高）是誰，參考下列的預期輸出。

- 預期輸入：SQL 語法。
- 預期輸出：(30, 4) 的虛擬資料表 `ppg_leader_by_teams`。

```
SELECT * FROM ppg_leader_by_teams;
```

```
team      firstName  lastName  ppg
0  Atlanta Hawks      Trae      Young    24.0
1  Boston Celtics     Kemba     Walker    19.8
2  Brooklyn Nets      Kevin     Durant    27.1
3  Charlotte Hornets   LaMelo    Ball      15.9
4  Chicago Bulls       Zach      LaVine    18.8
5  Cleveland Cavaliers Collin     Sexton    19.7
6  Dallas Mavericks    Luka      Doncic    25.6
7  Denver Nuggets      Nikola    Jokic     18.0
8  Detroit Pistons     Josh      Jackson   12.0
9  Golden State Warriors Stephen    Curry     23.8
10 Houston Rockets     John      Wall      19.0
11 Indiana Pacers      T.J.      Warren    15.5
12 LA Clippers         Paul      George    20.1
13 Los Angeles Lakers  LeBron    James     27.0
14 Memphis Grizzlies   Ja        Morant    18.4
15 Miami Heat          Victor     Oladipo   17.6
16 Milwaukee Bucks     Giannis   Antetokounmpo 20.7
17 Minnesota Timberwolves Karl-Anthony Towns    22.8
18 New Orleans Pelicans Zion       Williamson 25.0
19 New York Knicks      Derrick   Rose      18.5
20 Oklahoma City Thunder Shai      Gilgeous-Alexander 16.3
21 Orlando Magic       Gary      Harris    12.0
```

22	Philadelphia 76ers	Joel	Embiid	24.7
23	Phoenix Suns	Devin	Booker	22.8
24	Portland Trail Blazers	Damian	Lillard	24.6
25	Sacramento Kings	De'Aaron	Fox	17.8
26	San Antonio Spurs	DeMar	DeRozan	20.1
27	Toronto Raptors	Kyle	Lowry	14.8
28	Utah Jazz	Donovan	Mitchell	23.2
29	Washington Wizards	Russell	Westbrook	23.2

```
CREATE VIEW nba.ppg_leader_by_teams (team, firstName, lastName, ppg)
AS
SELECT teams.fullName AS team,
       players.firstName,
       players.lastName,
       MAX(career_summaries.ppg)
FROM players
JOIN teams
  ON players.teamId = teams.teamId
JOIN career_summaries
  ON players.personId = career_summaries.personId
GROUP BY teams.fullName
ORDER BY teams.fullName;
```

43. 在已經建立且連線好的 `test` 資料庫新增一個資料表名為 `favorite_players`，具有三個欄位 `name`、`years_pro`、`ppg`，資料類型分別是文字（`TEXT`）、整數（`INTEGER`）與浮點數（`REAL`），參考下列的預期輸出。

- 預期輸入：SQL 語法。
- 預期輸出：(0, 3) 的資料表 `favorite_players`。

```
SELECT * FROM favorite_players;
```

name	years_pro	ppg

```
CREATE TABLE favorite_players (
  name TEXT,
  years_pro INTEGER,
  ppg REAL
);
```

44. 承接上題，在 `test` 資料庫的 `favorite_players` 資料表中新增五筆觀測值，參考下列的預期輸出。

- 預期輸入：SQL 語法。
- 預期輸出：(5, 3) 的資料表 `favorite_players`。

```
SELECT * FROM favorite_players;
```

	name	years_pro	ppg
0	Steve Nash	19	14.3
1	Michael Jordan	14	30.1
2	Paul Pierce	19	19.7
3	Kevin Garnett	21	17.8
4	Hakeem Olajuwon	18	21.8

```
INSERT INTO favorite_players (name, years_pro, ppg)
VALUES
  ('Steve Nash', 19, 14.3),
  ('Michael Jordan', 14, 30.1),
  ('Paul Pierce', 19, 19.7),
  ('Kevin Garnett', 21, 17.8),
  ('Hakeem Olajuwon', 18, 21.8);
```

45. 承接上題，在 `test` 資料庫的 `favorite_players` 資料表將第五位球員 Hakeem Olajuwon 替換成 Tim Duncan，參考下列的預期輸出。

- 預期輸入：SQL 語法。
- 預期輸出：(5, 3) 的虛擬資料表 `favorite_players`。

```
SELECT * FROM favorite_players;
```

	name	years_pro	ppg
0	Steve Nash	19	14.3
1	Michael Jordan	14	30.1
2	Paul Pierce	19	19.7
3	Kevin Garnett	21	17.8
4	Tim Duncan	19	19.0

```
UPDATE favorite_players
SET name = 'Tim Duncan',
    years_pro = 19,
    ppg = 19.0
WHERE name = 'Hakeem Olajuwon';
```

46. 從 `covid19` 資料庫查詢兩艘郵輪（Grand Princess 與 Diamond Princess）的資訊，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(4, 4) 的查詢結果。

	iso2	Country_Region	Province_State	Confirmed
0	CA	Canada	Diamond Princess	0
1	CA	Canada	Grand Princess	13
2	US	US	Diamond Princess	49
3	US	US	Grand Princess	103

```
SELECT lookup_table.iso2,
       lookup_table.Country_Region,
       lookup_table.Province_State,
       Confirmed
FROM daily_report
JOIN lookup_table
ON daily_report.Combined_Key = lookup_table.Combined_Key
WHERE Province_State IN ('Grand Princess', 'Diamond Princess');
```

47. 從 `covid19` 資料庫查詢截至 2021-03-31 所有國家確診與死亡人數的資訊，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(192, 3) 的查詢結果。

	Country_Region	Confirmed	Deaths
0	Afghanistan	56454	2484
1	Albania	125157	2235
2	Algeria	117192	3093
3	Andorra	12010	115
4	Angola	22311	537
..
187	Vietnam	2603	35
188	West Bank and Gaza	242353	2627
189	Yemen	4357	888
190	Zambia	88418	1208
191	Zimbabwe	36882	1523

[192 rows x 3 columns]

```
SELECT lookup_table.Country_Region,
       SUM(Confirmed) AS Confirmed,
       SUM(Deaths) AS Deaths
FROM daily_report
JOIN lookup_table
  ON daily_report.Combined_Key = lookup_table.Combined_Key
GROUP BY lookup_table.Country_Region;
```

48. 從 **imdb** 資料庫查詢「魔戒三部曲」與「蝙蝠俠三部曲」的電影資訊與演員名單，三部曲電影系列中演員重複出演的情況是正常的，這時顯示獨一值即可，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(66, 2) 的查詢結果。

	trilogy	name
0	Batman Trilogy	Aaron Eckhart
1	Batman Trilogy	Aidan Gillen
2	Batman Trilogy	Alon Aboutboul
3	Batman Trilogy	Anne Hathaway
4	Batman Trilogy	Anthony Michael Hall
..
61	The Lord of the Rings Trilogy	Sadwyn Brophy
62	The Lord of the Rings Trilogy	Sala Baker
63	The Lord of the Rings Trilogy	Sam Comery
64	The Lord of the Rings Trilogy	Sean Astin
65	The Lord of the Rings Trilogy	Sean Bean

[66 rows x 2 columns]

```
SELECT CASE WHEN movies.title LIKE '%Lord of the Rings%' THEN 'The Lord of the Rings Trilogy'
       ELSE 'Batman Trilogy' END AS trilogy,
       actors.name
FROM actors
JOIN casting
  ON actors.id = casting.actor_id
JOIN movies
  ON casting.movie_id = movies.id
WHERE movies.title LIKE '%Lord of the Rings%' OR
       movies.title LIKE '%Batman%' OR
       movies.title LIKE '%The Dark Knight%'
GROUP BY trilogy, actors.name;
```

49. 從 **nba** 資料庫查詢截至 2021-03-31 的得分王（生涯場均得分 **ppg** 最高）、助攻王（生涯場均助攻 **apg** 最高）、籃板王（生涯場均籃板 **rpg** 最高）、抄截王（生涯場均抄截 **spg** 最高）以及阻攻王（生涯場均阻攻 **bpg** 最高），參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(6, 4) 的查詢結果。

	firstName	lastName	category	value
0	Andre	Drummond	rpg	13.8
1	Anthony	Davis	bpg	2.4
2	Chris	Paul	apg	9.4
3	Chris	Paul	spg	2.2
4	Hassan	Whiteside	bpg	2.4
5	Kevin	Durant	ppg	27.1

```
SELECT players.firstName,
       players.lastName,
       'ppg' AS category,
```

```

        max_stats.ppg AS value
    FROM players
    JOIN (SELECT playerId,
        ppg
        FROM career_summaries
        WHERE ppg = (SELECT MAX(ppg)
            FROM career_summaries)) AS max_stats
    ON players.personId = max_stats.personId
    UNION
    SELECT players.firstName,
        players.lastName,
        'apg' AS category,
        max_stats.apg AS value
    FROM players
    JOIN (SELECT playerId,
        apg
        FROM career_summaries
        WHERE apg = (SELECT MAX(apg)
            FROM career_summaries)) AS max_stats
    ON players.personId = max_stats.personId
    UNION
    SELECT players.firstName,
        players.lastName,
        'rpg' AS category,
        max_stats.rpg AS value
    FROM players
    JOIN (SELECT playerId,
        rpg
        FROM career_summaries
        WHERE rpg = (SELECT MAX(rpg)
            FROM career_summaries)) AS max_stats
    ON players.personId = max_stats.personId
    UNION
    SELECT players.firstName,
        players.lastName,
        'spg' AS category,
        max_stats.spg AS value
    FROM players
    JOIN (SELECT playerId,
        spg
        FROM career_summaries
        WHERE spg = (SELECT MAX(spg)
            FROM career_summaries)) AS max_stats
    ON players.personId = max_stats.personId
    UNION
    SELECT players.firstName,
        players.lastName,
        'bpg' AS category,
        max_stats.bpg AS value
    FROM players
    JOIN (SELECT playerId,
        bpg
        FROM career_summaries
        WHERE bpg = (SELECT MAX(bpg)
            FROM career_summaries)) AS max_stats
    ON players.personId = max_stats.personId;

```

50. 從 **twElection2020** 資料庫查詢三組總統候選人在各縣市的得票數，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(22, 4) 的查詢結果。

	county	soong_yu_votes	han_chang_votes	tsai_lai_votes
0	南投縣	13315	133791	152046
1	嘉義市	6204	56269	99265
2	嘉義縣	11138	98810	197342
3	基隆市	11878	99360	114966
4	宜蘭縣	10739	90010	173657
5	屏東縣	14021	179353	317676
6	彰化縣	35060	291835	436336
7	新北市	112620	959631	1393936
8	新竹市	14103	102725	144274
9	新竹縣	18435	154224	152380
10	桃園市	63132	529749	718260
11	澎湖縣	2583	20911	27410
12	臺中市	84800	646366	967304
13	臺北市	70769	685830	875854
14	臺南市	41075	339702	786471

15	臺東縣	4163	67413	44092
16	花蓮縣	6869	111834	66509
17	苗栗縣	15222	164345	147034
18	連江縣	188	4776	1226
19	金門縣	1636	35948	10456
20	雲林縣	15331	138341	246116
21	高雄市	55309	610896	1097621

```

SELECT soong_yu.county,
       soong_yu.soong_yu_votes,
       han_chang.han_chang_votes,
       tsai_lai.tsai_lai_votes
FROM (SELECT admin_regions.county,
             SUM(votes) AS soong_yu_votes
      FROM presidential
      JOIN admin_regions
        ON presidential.admin_region_id = admin_regions.id
     WHERE presidential.candidate_id = 1
     GROUP BY admin_regions.county) AS soong_yu
JOIN (SELECT admin_regions.county,
             SUM(votes) AS han_chang_votes
      FROM presidential
      JOIN admin_regions
        ON presidential.admin_region_id = admin_regions.id
     WHERE presidential.candidate_id = 2
     GROUP BY admin_regions.county) AS han_chang
ON soong_yu.county = han_chang.county
JOIN (SELECT admin_regions.county,
             SUM(votes) AS tsai_lai_votes
      FROM presidential
      JOIN admin_regions
        ON presidential.admin_region_id = admin_regions.id
     WHERE presidential.candidate_id = 3
     GROUP BY admin_regions.county) AS tsai_lai
ON soong_yu.county = tsai_lai.county;

```

51. 從 `covid19` 資料庫查詢截至 2021-03-31 美國前十大確診人數的州別，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(10, 2) 的查詢結果。

	Province_State	Confirmed
0	California	3668277
1	Texas	2791910
2	Florida	2057735
3	New York	1873138
4	Illinois	1244499
5	Georgia	1059548
6	Pennsylvania	1027678
7	Ohio	1017566
8	North Carolina	914132
9	New Jersey	908816

```

SELECT lookup_table.Province_State,
       SUM(daily_report.Confirmed) AS Confirmed
FROM daily_report
JOIN lookup_table
  ON daily_report.Combined_Key = lookup_table.Combined_Key
WHERE lookup_table.Country_Region = 'US'
GROUP BY lookup_table.Province_State
ORDER BY Confirmed DESC
LIMIT 10;

```

52. 從 `covid19` 資料庫查詢截至 2021-03-31 台灣、日本、中國、南韓與新加坡五個國家的確診與死亡人數的資訊，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。

- 預期輸出：(5, 3) 的查詢結果。

	Country_Region	Confirmed	Deaths
0	China	101732	4841
1	Japan	474641	9155
2	Korea, South	103639	1735
3	Singapore	60381	30
4	Taiwan	1030	10

```
SELECT lookup_table.Country_Region,
       SUM(daily_report.Confirmed) AS Confirmed,
       SUM(daily_report.Deaths) AS Deaths
FROM daily_report
JOIN lookup_table
  ON daily_report.Combined_Key = lookup_table.Combined_Key
WHERE lookup_table.Country_Region IN ('Taiwan', 'China', 'Japan', 'Korea, South', 'Singapore')
GROUP BY lookup_table.Country_Region;
```

53. 從 **imdb** 資料庫查詢出現最多次的導演為誰，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3, 2) 的查詢結果。

	director	counts
0	Christopher Nolan	7
1	Martin Scorsese	7
2	Stanley Kubrick	7

```
SELECT director,
       counts
FROM (SELECT director,
             COUNT(*) AS counts
      FROM movies
      GROUP BY director) AS director_counts
WHERE director_counts.counts = (SELECT MAX(counts) AS max_counts
                               FROM (SELECT director,
                                             COUNT(*) AS counts
                                    FROM movies
                                    GROUP BY director) AS director_counts);
```

54. 從 **imdb** 資料庫查詢出現最多次的演員為誰，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(3, 2) 的查詢結果。

	actor_id	name	counts
0	2472	Robert De Niro	9

```
SELECT actor_id,
       actors.name,
       counts
FROM (SELECT actor_id,
             COUNT(*) AS counts
      FROM casting
      GROUP BY actor_id) AS actor_counts
JOIN actors
  ON actor_counts.actor_id = actors.id
WHERE actor_counts.counts = (SELECT MAX(counts) AS max_counts
                           FROM (SELECT actor_id,
                                         COUNT(*) AS counts
                                FROM casting
                                GROUP BY actor_id) AS actor_counts);
```


55. 從 `imdb` 資料庫查詢評等大於等於 8.8 (`rating >= 8.8`) 電影的導演以及第一主角 (`ord = 1`) , 參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(13, 3) 的查詢結果。

	title	director \
0	The Shawshank Redemption	Frank Darabont
1	The Godfather	Francis Ford Coppola
2	The Godfather: Part II	Francis Ford Coppola
3	The Dark Knight	Christopher Nolan
4	12 Angry Men	Sidney Lumet
5	Schindler's List	Steven Spielberg
6	The Lord of the Rings: The Return of the King	Peter Jackson
7	Pulp Fiction	Quentin Tarantino
8	The Good, the Bad and the Ugly	Sergio Leone
9	The Lord of the Rings: The Fellowship of the Ring	Peter Jackson
10	Fight Club	David Fincher
11	Forrest Gump	Robert Zemeckis
12	Inception	Christopher Nolan

	lead_actor
0	Tim Robbins
1	Marlon Brando
2	Al Pacino
3	Christian Bale
4	Martin Balsam
5	Liam Neeson
6	Noel Appleby
7	Tim Roth
8	Eli Wallach
9	Alan Howard
10	Edward Norton
11	Tom Hanks
12	Leonardo DiCaprio

```
SELECT movies.title,
       movies.director,
       actors.name AS lead_actor
FROM movies
JOIN casting
  ON movies.id = casting.movie_id
JOIN actors
  ON casting.actor_id = actors.id
WHERE movies.rating >= 8.8 AND
       casting.ord = 1;
```

56. 從 `nba` 資料庫查詢截至 2021-03-31 的得分王 (生涯總得分 `points` 最高)、助攻王 (生涯總助攻 `assists` 最高)、籃板王 (生涯總籃板 `totReb` 最高)、抄截王 (生涯總抄截 `steals` 最高) 以及阻攻王 (生涯總阻攻 `blocks` 最高) , 參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(5, 4) 的查詢結果。

	firstName	lastName	category	value
0	Chris	Paul	assists	10051
1	Chris	Paul	steals	2296
2	Dwight	Howard	blocks	2173
3	Dwight	Howard	totReb	14071
4	LeBron	James	points	35283

```
SELECT players.firstName,
       players.lastName,
       'points' AS category,
       max_stats.points AS value
```

```

FROM players
JOIN (SELECT playerId,
      points
      FROM career_summaries
      WHERE points = (SELECT MAX(points)
                     FROM career_summaries)) AS max_stats
ON players.personId = max_stats.personId
UNION
SELECT players.firstName,
      players.lastName,
      'assists' AS category,
      max_stats.assists AS value
FROM players
JOIN (SELECT playerId,
      assists
      FROM career_summaries
      WHERE assists = (SELECT MAX(assists)
                      FROM career_summaries)) AS max_stats
ON players.personId = max_stats.personId
UNION
SELECT players.firstName,
      players.lastName,
      'totReb' AS category,
      max_stats.totReb AS value
FROM players
JOIN (SELECT playerId,
      totReb
      FROM career_summaries
      WHERE totReb = (SELECT MAX(totReb)
                      FROM career_summaries)) AS max_stats
ON players.personId = max_stats.personId
UNION
SELECT players.firstName,
      players.lastName,
      'steals' AS category,
      max_stats.steals AS value
FROM players
JOIN (SELECT playerId,
      steals
      FROM career_summaries
      WHERE steals = (SELECT MAX(steals)
                      FROM career_summaries)) AS max_stats
ON players.personId = max_stats.personId
UNION
SELECT players.firstName,
      players.lastName,
      'blocks' AS category,
      max_stats.blocks AS value
FROM players
JOIN (SELECT playerId,
      blocks
      FROM career_summaries
      WHERE blocks = (SELECT MAX(blocks)
                      FROM career_summaries)) AS max_stats
ON players.personId = max_stats.personId;

```

57. 從 **nba** 資料庫查詢截至 2021-03-31 各球隊陣中場均得分大於等於 20 分 (**ppg >= 20**) 的球員人數，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(30, 2) 的查詢結果。

	team_name	number_of_players
0	Brooklyn Nets	4
1	Los Angeles Lakers	2
2	Portland Trail Blazers	2
3	Washington Wizards	2
4	Atlanta Hawks	1
5	Dallas Mavericks	1
6	Golden State Warriors	1
7	LA Clippers	1
8	Milwaukee Bucks	1
9	Minnesota Timberwolves	1
10	New Orleans Pelicans	1
11	Philadelphia 76ers	1
12	Phoenix Suns	1
13	San Antonio Spurs	1
14	Utah Jazz	1
15	Boston Celtics	0

16	Charlotte Hornets	0
17	Chicago Bulls	0
18	Cleveland Cavaliers	0
19	Denver Nuggets	0
20	Detroit Pistons	0
21	Houston Rockets	0
22	Indiana Pacers	0
23	Memphis Grizzlies	0
24	Miami Heat	0
25	New York Knicks	0
26	Oklahoma City Thunder	0
27	Orlando Magic	0
28	Sacramento Kings	0
29	Toronto Raptors	0

```

SELECT teams.fullName AS team_name,
       IFNULL(number_of_top_scorers.number_of_players, 0) AS number_of_players
FROM teams
LEFT JOIN (SELECT teams.fullName,
                  COUNT(*) AS number_of_players
            FROM players
            JOIN teams
              ON players.teamId = teams.teamId
            JOIN career_summaries
              ON players.personId = career_summaries.personId
            WHERE ppg >= 20
            GROUP BY teams.fullName) AS number_of_top_scorers
  ON teams.fullName = number_of_top_scorers.fullName
ORDER BY number_of_players DESC,
       team_name;

```

58. 從 `twElection2020` 資料庫查詢中國國民黨與民主進步黨在 2020 年選舉的得票率，包含總統副總統、不分區立委與區域立委，參考下列的預期查詢結果。

PS 不分區立委的投票資料記錄於資料表 `legislative_at_large`，區域立委的投票資料記錄於資料表 `legislative_regional`。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(2, 4) 的查詢結果。

	party	presidential	legislative_regional	legislative_at_large
0	中國國民黨	38.61%	40.71%	33.36%
1	民主進步黨	57.13%	45.11%	33.98%

```

SELECT presidential_result.party,
       presidential_result.presidential,
       legislative_regional_result.legislative_regional,
       legislative_at_large_result.legislative_at_large
FROM (SELECT CASE WHEN candidate_id = 2 THEN '中國國民黨'
                ELSE '民主進步黨' END as party,
          ROUND(CAST(SUM(votes) AS REAL) / (SELECT SUM(votes) FROM presidential)*100, 2) || '%' AS presidential
      FROM presidential
      WHERE candidate_id IN (2, 3)
      GROUP BY candidate_id) AS presidential_result
JOIN (SELECT parties.party,
          ROUND(CAST(SUM(legislative_regional.votes) AS REAL) / (SELECT SUM(votes) FROM legislative_regional)*100, 2) || '%' AS legislative_regional
      FROM legislative_regional
      JOIN candidates
        ON legislative_regional.candidate_id = candidates.id
      JOIN parties
        ON candidates.party_id = parties.id
      WHERE parties.party IN ('中國國民黨', '民主進步黨')
      GROUP BY parties.party) AS legislative_regional_result
ON presidential_result.party = legislative_regional_result.party
JOIN (SELECT parties.party,
          ROUND(CAST(SUM(legislative_at_large.votes) AS REAL) / (SELECT SUM(votes) FROM legislative_at_large)*100, 2) || '%' AS legislative_at_large
      FROM legislative_at_large
      JOIN parties
        ON legislative_at_large.party_id = parties.id
      WHERE parties.party IN ('中國國民黨', '民主進步黨')
      GROUP BY parties.party) AS legislative_at_large_result
ON presidential_result.party = legislative_at_large_result.party;

```

59. 從 `twElection2020` 資料庫查詢代表中國國民黨參選總統副總統的韓國瑜/張善政組合，在台灣 7,737 個選舉區（以村鄰里為一個選舉區）贏得的選舉區（得票數大於 > 蔡英文/賴清德組合）以及淨贏得票數，參考下列的預期查詢結果。

- 預期輸入：SQL 查詢語法。
- 預期輸出：(1332, 4) 的查詢結果。

	county	town	village	net_winning_votes
0	金門縣	金城鎮	西門里	2190
1	高雄市	左營區	海勝里	2096
2	臺北市	松山區	自強里	1802
3	桃園市	中壢區	自立里	1763
4	桃園市	龜山區	陸光里	1560
...
1327	基隆市	中山區	居仁里	1
1328	彰化縣	田中鎮	碧峰里	1
1329	新竹縣	湖口鄉	仁勢村	1
1330	澎湖縣	馬公市	東衛里	1
1331	臺中市	太平區	勤益里	1

[1332 rows x 4 columns]

```
SELECT admin_regions.county,
       admin_regions.town,
       admin_regions.village,
       han_chang.han_chang_votes - tsai_lai.tsai_lai_votes AS net_winning_votes
FROM (SELECT admin_regions.id,
             SUM(presidential.votes) AS han_chang_votes
      FROM presidential
      JOIN admin_regions
        ON presidential.admin_region_id = admin_regions.id
      WHERE presidential.candidate_id = 2
      GROUP BY admin_regions.id) AS han_chang
JOIN (SELECT admin_regions.id,
             SUM(presidential.votes) AS tsai_lai_votes
      FROM presidential
      JOIN admin_regions
        ON presidential.admin_region_id = admin_regions.id
      WHERE presidential.candidate_id = 3
      GROUP BY admin_regions.id) AS tsai_lai
ON han_chang.id = tsai_lai.id
JOIN admin_regions
  ON han_chang.id = admin_regions.id
WHERE net_winning_votes > 0
ORDER BY net_winning_votes DESC,
         admin_regions.id;
```