

A. 함정 정보 관리

- 테이블 전체 출력 캡쳐본

<classes Table>

```
MariaDB [db2017320233]> desc classes;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| class | varchar(20) | NO | PRI | NULL | 
| type | varchar(10) | YES | | NULL | 
| numGuns | int(11) | YES | | NULL | 
| displacement | int(11) | YES | | NULL | 
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select * from classes;
+-----+-----+-----+-----+
| class | type | numGuns | displacement |
+-----+-----+-----+-----+
| 광개토대왕급 | DD | 15 | 5000 |
| 금강산급 | FF | 5 | 500 |
| 독도급 | CC | 20 | 10000 |
| 세종대왕급 | BB | 25 | 100000 |
| 이순신급 | EE | 10 | 1000 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

<ships Table>

```
MariaDB [db2017320233]> desc ships;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| s_name | varchar(20) | NO | PRI | NULL | 
| class | varchar(20) | NO | | NULL | 
| country | varchar(30) | YES | | NULL | 
| launched | int(11) | YES | | NULL | 
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select * from ships;
+-----+-----+-----+-----+
| s_name | class | country | launched |
+-----+-----+-----+-----+
| 거북선함 | 이순신급 | 미국 | 1952 |
| 고구려함 | 광개토대왕급 | 미국 | 1949 |
| 구름함 | 금강산급 | 일본 | 1958 |
| 대왕비함 | 광개토대왕급 | 일본 | 1951 |
| 백의종군함 | 이순신급 | 미국 | 1953 |
| 비로봉함 | 금강산급 | 미국 | 1956 |
| 선녀함 | 금강산급 | 대한민국 | 1957 |
| 연평도함 | 독도급 | 캐나다 | 1948 |
| 울릉도함 | 독도급 | 대한민국 | 1947 |
| 인조함 | 이순신급 | 대한민국 | 1954 |
| 임진함 | 이순신급 | 캐나다 | 1955 |
| 제주도함 | 독도급 | 미국 | 1946 |
| 천마총함 | 광개토대왕급 | 대한민국 | 1950 |
| 한글함 | 세종대왕급 | 미국 | 1945 |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

<battles Table>

```
MariaDB [db2017320233]> desc battles;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| b_name | varchar(20) | NO | PRI | NULL | |
| year | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select * from battles;
+-----+-----+
| b_name | year |
+-----+-----+
| 노량 | 1950 |
| 명량 | 1953 |
| 연평 | 2000 |
| 진주만 | 1945 |
| 한산도 | 1957 |
+-----+-----+
5 rows in set (0.00 sec)
```

<Outcomes Table>

```
MariaDB [db2017320233]> desc outcomes;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| s_name | varchar(20) | NO | PRI | NULL | |
| b_name | varchar(20) | NO | PRI | NULL | |
| result | varchar(10) | YES | | NULL | |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select * from outcomes;
+-----+-----+-----+
| s_name | b_name | result |
+-----+-----+-----+
| 거북선함 | 한산도 | 손상 |
| 고구려함 | 노량 | 손상 |
| 대왕비함 | 노량 | 침몰 |
| 백의종군함 | 명량 | 손상 |
| 연평도함 | 노량 | OK |
| 출동도함 | 연평 | OK |
| 인조함 | 노량 | 침몰 |
| 임진항 | 한산도 | OK |
| 제주도함 | 명량 | OK |
| 천마총함 | 명량 | 손상 |
| 한글함 | 노량 | OK |
| 한글함 | 명량 | 침몰 |
| 한글함 | 연평 | OK |
| 한글함 | 진주만 | 손상 |
| 한글함 | 한산도 | 손상 |
+-----+-----+-----+
15 rows in set (0.00 sec)
```

- 각 질문 SQL 결과 캡쳐본

A.(1) “노량” 해전에서 “침몰” 된 군함의 보유 국가명을 찾아라.

```
MariaDB [db2017320233]> select country
  -> from ships natural join outcomes
  -> where b_name = '노량'
  -> and result = '침몰';
+-----+
| country |
+-----+
| 일본 |
| 대한민국 |
+-----+
2 rows in set (0.00 sec)
```

A.(2) 취역한 년도에 해전에 참전하여 손상을 당한 군함의 이름을 찾아라.

```
MariaDB [db2017320233]> select s_name
  -> from ships natural join battles natural join outcomes
  -> where launched=year
  -> and result='손상';
+-----+
| s_name |
+-----+
| 백의종군함 |
| 한글함 |
+-----+
2 rows in set (0.00 sec)
```

A.(3) 가장 많은 함포를 보유하고 있는 군함의 이름(class)을 찾아라. – 군함급명칭 찾기

```
MariaDB [db2017320233]> select class
  -> from classes
  -> where numGuns = (select max(numGuns) from classes);
+-----+
| class |
+-----+
| 세종대왕급 |
+-----+
1 row in set (0.00 sec)
```

A.(4) 배수량이 5000톤보다 큰 모든 급(class)의 군함을 보유한 국가의 이름을 찾아라.

```
MariaDB [db2017320233]> select country
  -> from ships natural join (select distinct class
  ->                               from classes
  ->                               where displacement>5000) as A
  -> group by country
  -> having count(distinct class) = (select count(distinct class)
  ->                               from (select distinct class
  ->                               from classes
  ->                               where displacement>5000) as B);
+-----+
| country |
+-----+
| 미국    |
+-----+
1 row in set (0.00 sec)
```

A.(5) “대한민국” 이 보유하고 있는 군함들의 이름들을 찾되 해당 군함들이 참전한 기록이 있다면 참전한 해전명을 함께 찾아라.

```
MariaDB [db2017320233]> select s_name, b_name
  -> from ships natural left outer join outcomes
  -> where country = '대한민국';
+-----+-----+
| s_name | b_name |
+-----+-----+
| 선녀함 | NULL   |
| 율류도함 | 연평   |
| 인조함 | 노량   |
| 천마총함 | 명량  |
+-----+-----+
4 rows in set (0.00 sec)
```

A.(6) 국가별 보유 군함의 수와 총배수량을 구하여라(aggregate function을 이용).

```
MariaDB [db2017320233]> select country, count(distinct s_name), sum(displacement)
-> from ships natural join classes
-> group by country;
+-----+-----+-----+
| country | count(distinct s_name) | sum(displacement) |
+-----+-----+-----+
| 대한민국 | 4 | 16500 |
| 미국 | 6 | 117500 |
| 일본 | 2 | 5500 |
| 캐나다 | 2 | 11000 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

A.(7) 다섯 번 이상 해전에 참전한 군함을 보유한 국가명을 찾아라.

```
MariaDB [db2017320233]> select country
-> from (select s_name, count(distinct b_name) as cnt
->         from outcomes
->         group by s_name
->         having cnt >= 5) as T natural join ships;
+-----+
| country |
+-----+
| 미국 |
+-----+
1 row in set (0.00 sec)
```

A.(8) “1945”에 발발한 해전에 참전한 군함의 이름과 취역 년도를 찾아라.

```
MariaDB [db2017320233]> select s_name, launched
-> from ships natural join battles natural join outcomes
-> where year = 1945;
+-----+-----+
| s_name | launched |
+-----+-----+
| 한글함 | 1945 |
+-----+-----+
1 row in set (0.01 sec)
```

B. 호텔 관리

- 테이블 전체 출력 캡쳐본

<hotel Table>

```
MariaDB [db2017320233]> desc hotel;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| hotel_id   | int(11)    | NO   | PRI | 0        |
| hotel_name | varchar(20) | YES  |     | NULL    |
| city       | varchar(20) | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
MariaDB [db2017320233]> select * from hotel;
+-----+-----+-----+
| hotel_id | hotel_name | city   |
+-----+-----+-----+
| 1        | 한국호텔   | Jeju  |
| 2        | 페어몬트   | Banff |
| 3        | 스위쏘텔   | London |
| 4        | 힐튼       | London |
| 5        | 신라       | Seoul  |
| 6        | 그랜드     | New York |
| 7        | 라플스     | Singapore |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

<room Table>

```
MariaDB [db2017320233]> desc room;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| hotel_id   | int(11)    | NO   | PRI | 0        |
| room_id    | int(11)    | NO   | PRI | 0        |
| type       | varchar(20) | YES  |     | NULL    |
| price      | int(11)    | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
MariaDB [db2017320233]> select * from room;
+-----+-----+-----+-----+
| hotel_id | room_id | type      | price   |
+-----+-----+-----+-----+
| 1        | 101     | single_room | 100    |
| 1        | 102     | double_room | 150    |
| 1        | 103     | premium_room | 200    |
| 2        | 201     | single_room | 400    |
| 2        | 202     | double_room | 600    |
| 2        | 203     | premium_room | 800    |
| 3        | 301     | single_room | 600    |
| 3        | 302     | double_room | 800    |
| 3        | 303     | premium_room | 1000   |
| 4        | 401     | single_room | 300    |
| 4        | 402     | double_room | 400    |
| 4        | 403     | premium_room | 500    |
| 5        | 501     | single_room | 150    |
| 5        | 502     | double_room | 250    |
| 5        | 503     | premium_room | 350    |
| 6        | 601     | single_room | 200    |
| 6        | 602     | double_room | 300    |
| 6        | 603     | premium_room | 400    |
| 7        | 701     | single_room | 300    |
| 7        | 702     | double_room | 450    |
| 7        | 703     | premium_room | 600    |
+-----+-----+-----+-----+
21 rows in set (0.00 sec)
```

<booking Table>

```
MariaDB [db2017320233]> desc booking;
+-----+-----+-----+-----+-----+
| Field    | Type     | Null | Key | Default | Extra   |
+-----+-----+-----+-----+-----+
| guest_id | int(11) | NO  | PRI | 0       |          |
| hotel_id | int(11) | NO  | PRI | 0       |          |
| room_id  | int(11) | NO  | PRI | 0       |          |
| date_from | date    | NO  | PRI | NULL    |          |
| date_to   | date    | NO  | PRI | NULL    |          |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select * from booking;
+-----+-----+-----+-----+-----+
| guest_id | hotel_id | room_id | date_from | date_to   |
+-----+-----+-----+-----+-----+
| 1        | 1        | 101     | 2018-03-01 | 2018-03-02 |
| 1        | 5        | 502     | 2018-03-22 | 2018-03-23 |
| 2        | 6        | 602     | 2018-03-11 | 2018-03-12 |
| 3        | 2        | 202     | 2018-03-03 | 2018-03-04 |
| 4        | 3        | 303     | 2018-03-04 | 2018-03-06 |
| 5        | 4        | 402     | 2018-03-06 | 2018-03-09 |
| 6        | 7        | 703     | 2018-03-21 | 2018-04-02 |
| 8        | 5        | 501     | 2018-03-08 | 2018-03-12 |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

<guest Table>

```
MariaDB [db2017320233]> desc guest;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| guest_id | int(11) | NO  | PRI | 0       |       |
| guest_name | varchar(10) | YES |     | NULL    |       |
| age | int(11) | YES |     | NULL    |       |
| guest_city | varchar(20) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select * from guest;
+-----+-----+-----+-----+
| guest_id | guest_name | age | guest_city |
+-----+-----+-----+-----+
| 1 | 흄길동 | 70 | Jeju |
| 2 | 케이틀린 | 26 | Banff |
| 3 | 케이티 | 23 | Montreal |
| 4 | 카일라 | 22 | Toronto |
| 5 | 킴 | 22 | London |
| 6 | 애슬리 | 55 | New York |
| 7 | 제인 | 29 | Singapore |
| 8 | 아이리스 | 26 | Taiwan |
+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

- 각 질의문 SQL 결과 캡쳐본

B.(1) “한국호텔” 의 “single_room” 을 예약한 고객명(guest_name)을 찾아라.

```
MariaDB [db2017320233]> select guest_name
      -> from guest natural join booking natural join hotel natural join room
      -> where hotel_name='한국호텔' and type='single_room';
+-----+
| guest_name |
+-----+
| 흄길동 |
+-----+
1 row in set (0.00 sec)
```

B.(2) 보유하고 있는 방들의 가격(price) 평균이 가장 큰 호텔명(hotel_name)을 찾아라.

```
MariaDB [db2017320233]> select hotel_name
    ->           from ((select hotel_id, avg(price) as avg_price
    ->                         from room
    ->                         group by hotel_id) as B, (select max(avg_price) as max_avg
    ->                           from (select avg(price) as avg_price
    ->                                 from room
    ->                                 group by hotel_id) as A) as C)
    ->           natural join hotel
    ->           where max_avg = avg_price;
+-----+
| hotel_name |
+-----+
| 스위 쏘텔 |
+-----+
1 row in set (0.00 sec)
```

B.(3) 나이(age)가 50세 이상의 모든 고객의 이름(guest_name)을 찾되 해당 고객들 중
호텔 방을 예약한 이력이 있는 경우에는 예약한 호텔명(hotel_name)도 함께 찾아라.

```
MariaDB [db2017320233]> select guest_name, hotel_name
    ->           from (guest natural left outer join booking) natural join hotel
    ->           where age>=50;
+-----+-----+
| guest_name | hotel_name |
+-----+-----+
| 흥길동     | 한국호텔     |
| 흥길동     | 신라         |
| 애슬리     | 라플스       |
+-----+-----+
3 rows in set (0.00 sec)
```

B.(4) 호텔 방의 가격 중 가장 싼 가격(price)를 찾아라.

```
MariaDB [db2017320233]> select min(price)
    ->   from room;
+-----+
| min(price) |
+-----+
|      100   |
+-----+
1 row in set (0.00 sec)
```

B.(5) 고객이 거주하는 도시에 위치한 호텔을 예약한 고객 이름(guest_name)을 찾아라.

```
MariaDB [db2017320233]> select guest_name
    -> from guest natural join booking natural join hotel
    -> where guest_city = city;
+-----+
| guest_name |
+-----+
| 흥길동   |
| 킴        |
+-----+
2 rows in set (0.00 sec)
```

B.(6) “흥길동” 고객에 의해 예약된 호텔명(hotel_name)들을 찾아라.

```
MariaDB [db2017320233]> select hotel_name
    -> from booking natural join hotel natural join guest
    -> where guest_name='흥길동';
+-----+
| hotel_name |
+-----+
| 한국호텔   |
| 신라        |
+-----+
2 rows in set (0.00 sec)
```

B.(7) 호텔별 가장 비싼 방의 가격들 중 가장 비싼 방의 가격을 구하여라.

```
MariaDB [db2017320233]> select max(expensive_price)
    ->           from (select hotel_id, max(price) as expensive_price
    ->                     from room
    ->                   group by hotel_id) as A;
+-----+
| max(expensive_price) |
+-----+
|          1000         |
+-----+
1 row in set (0.00 sec)
```

B.(8) London에 위치하는 모든 호텔에 방을 예약한 고객이름(guest_name)을 찾아라.

```
MariaDB [db2017320233]> select guest_name
    -> from (select distinct hotel_id, guest_name
    ->           from guest natural join booking) as T1 natural join(select hotel_id
    ->                           from hotel
    ->                         where city='London') as T2
    -> group by guest_name
    -> having count(*) = (select count(*)
    ->                       from (select hotel_id, city
    ->                           from hotel
    ->                         where city='London') as T2);
Empty set (0.01 sec)
```

```
MariaDB [db2017320233]> select guest_name
-> from (guest natural join booking natural join (select distinct hotel_id
->                                         from hotel
->                                         where city='London') as A)
-> group by guest_name
-> having count(distinct hotel_id) = (select count(distinct hotel_id)
->                                         from (select hotel_id
->                                         from hotel
->                                         where city='London') as B);
Empty set (0.01 sec)
```

→ Table에 정보를 추가해서 다시 확인해보았습니다! (이 문제에 대해서만 시도함)

```
MariaDB [db2017320233]> INSERT INTO `booking` VALUES (4, 4, 402, '2018-03-12', '2018-03-15');
Query OK, 1 row affected (0.00 sec)

MariaDB [db2017320233]> select * from booking;
+-----+-----+-----+-----+-----+
| guest_id | hotel_id | room_id | date_from | date_to   |
+-----+-----+-----+-----+-----+
|       1   |      1   |    101  | 2018-03-01 | 2018-03-02 |
|       1   |      5   |    502  | 2018-03-22 | 2018-03-23 |
|       2   |      6   |    602  | 2018-03-11 | 2018-03-12 |
|       3   |      2   |    202  | 2018-03-03 | 2018-03-04 |
|       4   |      3   |    303  | 2018-03-04 | 2018-03-06 |
|       4   |      4   |    402  | 2018-03-12 | 2018-03-15 |
|       5   |      4   |    402  | 2018-03-06 | 2018-03-09 |
|       6   |      7   |    703  | 2018-03-21 | 2018-04-02 |
|       8   |      5   |    501  | 2018-03-08 | 2018-03-12 |
+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

```
MariaDB [db2017320233]> select guest_name
-> from (guest natural join booking natural join (select distinct hotel_id
->                                         from hotel
->                                         where city = 'London') as A)
-> group by guest_name
-> having count(distinct hotel_id) = (select count(distinct hotel_id)
->                                         from (select hotel_id
->                                         from hotel
->                                         where city = 'London') as B);
+-----+
| guest_name |
+-----+
| 카일라    |
+-----+
1 row in set (0.00 sec)
```

답이 도출됩니다!

(Booking table은 추가한 행을 delete해서 원래대로 empty set이 B.(8)의 답이 되도록 되돌려놓았습니다.)

고맙습니다😊😊😊