Да се напише заявка, която извежда име на клас, годината на първата битка, в която кораб на този клас е участвал, годината на последната битка, в която кораб на този клас е участвал, и броя на всички различни битки, в които кораби на този клас са участвали, само за тези класове, започващи с буквата N. Ако за даден клас няма кораб, който да е участвал в битка, за съответните години да се върне стойност null.

```
SELECT classes.class, MIN(battles.date),
                MAX(battles.date), COUNT(battles.name)
FROM classes
LEFT JOIN ships ON ships.class = classes.class
JOIN outcomes ON outcomes.ship = ships.name
JOIN battles ON outcomes.battle = battles.name
WHERE classes.class LIKE "N%"
GROUP BY classes.class;
```

Да се напише заявка, която да изведе имената на тези битки, за които броят на корабите от тип 'bb', участвали в тази битка, е по-голям от броя на корабите от тип 'bc', участвали в същата битка. Битки, в които не е участвал нито един кораб, да не се извеждат в резултата

```
SELECT bb_battles.battle_name
FROM
        (SELECT battles.name AS battle_name, COUNT(outcomes.ship) AS bb_ships
        FROM battles
        JOIN outcomes ON battles.name = outcomes.battle
        JOIN ships ON ships.name = outcomes.ship
        JOIN classes ON classes.class = ships.class
        WHERE classes.type = "bb"
        GROUP BY battles.name) AS bb_battles
JOIN
        (SELECT battles.name AS battle_name, COUNT(outcomes.ship) AS bc_ships
        FROM battles
        JOIN outcomes ON battles.name = outcomes.battle
        JOIN ships ON ships.name = outcomes.ship
        JOIN classes ON classes.class = ships.class
        WHERE classes.type = "bc"
        GROUP BY battles.name) AS bc_battles
ON bb_battles.battle_name = bc_battles.battle_name
WHERE bb_battles.bb_ships > bc_battles.bc_ships;
```

Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име и държава на корабите, които никога не са потъвали в битка (може и да не са участвали).

```
SELECT ships.name, classes.country
FROM ships
LEFT JOIN outcomes ON ships.name = outcomes.ship
JOIN classes ON classes.class = ships.class
WHERE outcomes.result <> "sunk";
```

Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име, водоизместимост и брой оръдия на най-леките кораби с най-много оръдия.

```
SELECT s.name, c.displacement,
                c.numguns
FROM classes c
JOIN ships s ON c.class = s.class
WHERE displacement = (          SELECT MIN(displacement)
                                                FROM classes    )
                        AND
          numguns = (   SELECT MAX(numguns)
                                FROM classes c1
                                WHERE c1.displacement = c.displacement        );
```

Попълнете липсващите части, обозначени с _____ така, че заявката да изведе име на битките, в които е участвал един кораб

```
/*
SELECT battle
FROM outcomes
GROUP BY battle
HAVING COUNT(ship) = 1;
*/
SELECT battle
FROM outcomes o1
WHERE EXISTS (          SELECT *
                                FROM outcomes o2
                                WHERE o1.battle = o2.battle HAVING COUNT(ship) = 1);
```

Да се посочи заявката, която извежда име на класа и брой на потъналите в битка кораби за съответния клас, за тези класове с повече от 5 кораба

```
/*
SELECT t1.class, t1.sunks
FROM
        (SELECT ships.class, COUNT(outcomes.ship) AS sunks
        FROM ships
        JOIN outcomes ON ships.name = outcomes.ship
        WHERE outcomes.result = "sunk"
        GROUP BY ships.class) AS t1
JOIN
        (SELECT ships.class, COUNT(*) AS cnt
        FROM ships
        GROUP BY ships.class
        HAVING cnt > 5) AS t2
ON t1.class = t2.class;
*/
```

А) Не извежда брой потънали кораби

Б) Вярно

В) COUNT в WHERE не може

Г) Във GROUP BY на заявката във FROM има погрешно добавено групиране с name

Да се посочи заявката, която извежда всички държави, които имат поне един кораб, участвал в битка, както и броя на потъналите кораби за всяка от държавите.

```
/*
SELECT t1.country, IF(t2.cnt IS NULL, 0, t2.cnt)
FROM (
        SELECT DISTINCT classes.country
        FROM classes
        JOIN ships ON classes.class = ships.class
        JOIN outcomes ON ships.name = outcomes.ship
        ) AS t1
LEFT JOIN (
        SELECT classes.country, COUNT(*) AS cnt
        FROM classes
        JOIN ships ON classes.class = ships.class
        JOIN outcomes ON ships.name = outcomes.ship
        WHERE outcomes.result = "sunk"
        ) AS t2 ON t1.country = t2.country;
*/
```

А) OR result IS NOT NULL дава всички

Б) Ще пропусне държавите, в които няма потънали кораби

В) HAVING е преди ORDER BY + в HAVING се използва колона, която не присъства в SELECT

Г) Вярно - вложената заявка прави реално JOIN подобен на онзи, който аз правя горе

Да се посочи заявката, която извежда имената на битките, които са по-мащабни (с кораби от повече държави) от битката в Коралово море (Coral Sea).

```
/*
SELECT t1.battle
FROM(
        SELECT outcomes.battle, COUNT(DISTINCT classes.country) AS countries
        FROM classes
        JOIN ships ON ships.class = classes.class
        JOIN outcomes ON ships.name = outcomes.ship
        GROUP BY outcomes.battle
    ) AS t1
WHERE t1.countries > (
        SELECT COUNT(DISTINCT classes.country)
        FROM classes
        JOIN ships ON ships.class = classes.class
        JOIN outcomes ON ships.name = outcomes.ship
        WHERE outcomes.battle="Coral Sea"
    );                                                    */
```

А) Вярно

Б) Прави се декартово произведение, което не изглежда правилно. После в HAVING се прави някакво сравнение, което по никакъв начин не е обвързано с таблиците от FROM

В) Вярно ✗ липсва Distinct

Г) Няма коректно направени JOIN - не се работи с адекватни множества

Оградете буквата на заявката, която извежда имената на всички кораби, пуснати на вода в година, в която е имало битка (не е задължително корабът да е участвал в нея)

```
/*
SELECT ships.name
FROM ships
WHERE launched IN (
        SELECT YEAR(battles.date)
        FROM battles
    );                                                    */
```

А) COUNT без да има групиране

Б) Вярно - комбинира всеки кораб с всяка битка, премахва тези, които не съвпадат по година и накрая премахва дубликатите

В) Връща имена на битки, а не на кораби

Г) Почти същото като Б, но не е ясно name от ships или от battles трябва да се вземе

Оградете буквата на заявката, която за всички държави, които имат най-много 3 (евентуално 0) кораба, извежда името на държавата и броя потънали кораби (който също може да бъде 0).

```
/*
SELECT t1.country, t2.sunks
FROM (
        SELECT classes.country, COUNT(ships.name) AS nships
        FROM classes
        LEFT JOIN ships ON classes.class = ships.class
        GROUP BY classes.country
        HAVING nships<3
    ) AS t1
JOIN (
        SELECT classes.country, SUM(IF(outcomes.result="sunk", 1, 0)) AS sunks
        FROM classes
        LEFT JOIN ships ON classes.class = ships.class
        LEFT JOIN outcomes ON ships.name = outcomes.ship
        GROUP BY classes.country
    ) AS t2 ON t1.country = t2.country;

ВИЖДАМ, ЧЕ РАБОТЯТ ПО ЕДНО И СЪЩО МНОЖЕСТВО
=> оптимизация:

SELECT classes.country, SUM(IF(outcomes.result="sunk", 1, 0)) AS sunks
FROM classes
LEFT JOIN ships ON classes.class = ships.class
LEFT JOIN outcomes ON ships.name = outcomes.ship
GROUP BY classes.country
HAVING COUNT(ships.name) < 3;

АЛТЕРНАТИВНО РЕШЕНИЕ:

SELECT DISTINCT classes.country, IF(t1.sunks IS NULL, 0, t1.sunks) AS sunks
FROM classes
LEFT JOIN(
    SELECT classes.country, COUNT(*) AS sunks
    FROM classes
    LEFT JOIN ships ON classes.class = ships.class
    LEFT JOIN outcomes ON ships.name = outcomes.ship
    WHERE outcomes.result="sunk"
    GROUP BY classes.country
    ) AS t1 ON classes.country = t1.country;
*/
```

А) Не връща тези с 0 - where o.result = 'sunk' лимитира броенето само до потънали

Б) "result is 'sunk'" е невалиден оператор, агрегатна в where, crossjoin, т.н.

В) Where "sunk_cnt<3" не е по условие - там общия брой кораби трябва да е <3

Г) Има count в where,

Д) Вярно

Посочете заявката, която извежда всички държави, които имат както класове с по-малко от 9 оръдия (numguns), така и класове с над 12 оръдия:

```
/*
SELECT t1.country
FROM(
        SELECT country
        FROM classes
        WHERE numguns<9
        ) AS t1
JOIN(
        SELECT country
        FROM classes
        WHERE numguns>12
        ) AS t2 ON t1.country = t2.country;

ВИЖДАМ, че РАБОТЯТ ПО ЕДНО И СЪЩО МНОЖЕСТВО
=> оптимизация

SELECT t1.country
FROM classes AS t1
JOIN classes AS t2 ON t1.country = t2.country
WHERE t1.numguns<9 AND t2.numguns>12

*/
```

А) Невъзможен WHERE

Б) Вярно - направен е self join, т.е. същото, което и аз горе

В) С Union ги подреждаме просто едно под друго, а не правим логическо "И"

Г) Прави същото като А - невъзможен WHERE

Посочете заявката, която за всяка държава, участвала в не повече от 4 битки, извежда името ѝ и броя битки, в които е участвала. Ако дадена държава няма нито един кораб или не е участвала в нито една битка, за нея да извежда 0.

```
/*
SELECT t1.country, COUNT(t1.battle)
FROM(
        SELECT DISTINCT classes.country,  outcomes.battle
        FROM classes
        LEFT JOIN ships ON classes.class = ships.class
        LEFT JOIN outcomes ON ships.name = outcomes.ship
        ) AS t1
GROUP BY t1.country;
*/
```

А) пропуска 0 заради inner join (осъществен в WHERE)

Б) Смята грешно, защото брои броя кораби на държава в битка, а не броя участия на държава в битка. Ако се сложи DISTINCT в COUNT горе ще е вярно

В) Вярно. Същото е като Б, но със сложен DISTINCT

Г) COUNT в WHERE

Посочете заявката, която извежда за всеки клас годината на най-рано и най-късно пуснатия на вода кораб

```
/*
SELECT classes.class, MIN(ships.launched), MAX(ships.launched)
FROM classes
JOIN ships ON classes.class = ships.class
GROUP BY classes.class;
*/
```

a) Няма групиране

b) Вярно

c) Невалидно HAVING условие

d) UNION подрежда едно под друго. Освен това не прави групиране.

Посочете заявката, която извежда държавата/държавите с най-много класове

```
/*
Връща име на държава и броя класове
SELECT classes.country, COUNT(*) AS classes_count
FROM classes
GROUP BY country

Взимаме максималния брой класове за държава от всичките
SELECT MAX(classes_count)
FROM ( SELECT classes.country, COUNT(*) AS classes_count
                FROM classes
                GROUP BY country
        ) AS t

=> Намираме държавИТЕ с най-много класове по следния начин:

SELECT classes.country, COUNT(*) AS classes_count
FROM classes
GROUP BY country
HAVING classes_count = (
                SELECT MAX(classes_count)
                FROM ( SELECT classes.country, COUNT(*) AS classes_count
                                FROM classes
                                GROUP BY country
                        ) AS t
        );

Забележка:
Ако се търси само една държава може да се прави
        SELECT classes.country, COUNT(*) AS classes_count
        FROM classes
        GROUP BY country
        ORDER BY classes_count DESC
        LIMIT 1;
*/
```

a) COUNT в WHERE

b) Невалидно условие в HAVING

c) COUNT в WHERE

d) Вярно - оператор ALL връща true ако ВСИЧКИ редове от подадената таблица отговарят на условието

Посочете заявката, която извежда имената на битките, в които няма оцелели кораби (всички участвали кораби са потънали)

```
/*
SELECT t1.battle
FROM (
        SELECT outcomes.battle, COUNT(*) AS all_ships
        FROM outcomes
        GROUP BY outcomes.battle
    ) AS t1
JOIN (
        SELECT outcomes.battle, COUNT(*) AS sinked_ships
        FROM outcomes
        WHERE result="sunk"
        GROUP BY outcomes.battle
    ) AS t2 ON t1.battle = t2.battle
WHERE all_ships = sinked_ships;

Алтернативно (по-добро)

SELECT outcomes.battle, SUM(IF(result = 'sunk',0,1)) AS non_sunks
FROM outcomes
GROUP BY battle
HAVING non_sunks = 0;

*/
```

а) Безсмислен join + връща всички с поне един потънал

б) Безсмислено групиране + връща всички с поне един потънал

в) Вярно. Взима в о множеството на всички битки и за него преброява броя участвали кораби. Взима в о1 множеството на всички битки с потънали кораби и преброява колко са потъналите кораби. Накрая сравнява дали потъналите кораби са равни на общия брой кораби

г) Където има потънал ще дава Y, където няма ще има NULL. Върху това множество прави COUNT, т.е. брои колко потънали хора има в битка. Накрая брои битките, в които НЯМА потънали кораби (ние искаме точно обратното)