1. Harry Potter and his friends are at Ollivander's with Ron, finally replacing Charlie's old broken wand. Hermione decides the best way to choose is by determining the minimum number of gold galleons needed to buy each non-evil wand of high power and age. Write a query to print the id, age, coins_needed, and power of the wands that Ron's interested in, sorted in order of descending power. If more than one wand has same power, sort the result in order of descending age.

```
SELECT W.ID, WP.AGE, W.COINS_NEEDED, W.POWER
FROM WANDS AS W
INNER JOIN WANDS_PROPERTY AS WP
USING(CODE)
WHERE W.COINS_NEEDED = (
            SELECT MIN(W1.COINS_NEEDED)
            FROM WANDS AS W1
            INNER JOIN WANDS_PROPERTY AS WP1
            USING(CODE)
            WHERE WP1.IS_EVIL = '0'
            AND W1.POWER = W.POWER
            AND WP1.AGE = WP.AGE)
ORDER BY W.POWER DESC, WP.AGE DESC;
```

2. Given the CITY and COUNTRY tables, query the names of all the continents (COUNTRY.Continent) and their respective average city populations (CITY.Population) rounded down to the nearest integer.

```
SELECT CO.CONTINENT, FLOOR(AVG(C.POPULATION)) AS AVG_POPULATION
FROM CITY AS C
JOIN COUNTRY AS CO
ON C.COUNTRYCODE = CO.CODE
GROUP BY CO.CONTINENT
```

3. Ketty gives Eve a task to generate a report containing three columns: Name, Grade and Mark. Ketty doesn't want the NAMES of those students who received a grade lower than 8. The report must be in descending order by grade -- i.e. higher grades are entered first. If there is more than one student with the same grade (8-10) assigned to them, order those particular students by their name alphabetically. Finally, if the grade is lower than 8, use "NULL" as their name and list them by their grades in descending order. If there is more than one student with the same grade (1-7) assigned to them, order those particular students by their marks in ascending order.

```
SELECT
  CASE WHEN
  G.GRADE < 8 THEN NULL
  ELSE S.NAME
  END AS NAME,
  G.GRADE,
    S.MARKS
FROM STUDENTS AS S
INNER JOIN GRADES AS G
ON S.MARKS BETWEEN G.MIN_MARK AND G.MAX_MARK
ORDER BY G.GRADE DESC, S.NAME ASC
```

4. Write a query to print the respective hacker_id and name of hackers who achieved full scores for more than one challenge. Order your output in descending order by the total number of challenges in which the hacker earned a full score. If more than one hacker received full scores in same number of challenges, then sort them by ascending hacker_id.

```
SELECT H.HACKER_ID, H.NAME
 FROM HACKERS AS H
            INNER JOIN SUBMISSIONS AS S
            USING(HACKER_ID)
            INNER JOIN CHALLENGES AS C
            USING(CHALLENGE_ID)
            INNER JOIN DIFFICULTY AS D
            USING(DIFFICULTY_LEVEL)
WHERE S.SCORE = D.SCORE
GROUP BY H.HACKER_ID, H.NAME
HAVING COUNT(*) > 1
ORDER BY COUNT(*) DESC, H.HACKER_ID ASC
```

5. Samantha was tasked with calculating the average monthly salaries for all employees in the EMPLOYEES table, but did not realize her keyboard's 0 key was broken until after completing the calculation. She wants your help finding the difference between her miscalculation (using salaries with any zeros removed), and the actual average salary. Write a query calculating the amount of error (i.e.: average monthly salaries), and round it up to the next integer

```
SELECT CEIL(AVG(SALARY) - AVG(REPLACE(SALARY, '0', '')))
FROM EMPLOYEES
```

6. find the maximum total earnings for all employees as well as the total number of employees who have maximum total earnings

```
SELECT MAX(SALARY*MONTHS), COUNT(SALARY*MONTHS)
FROM EMPLOYEE
WHERE SALARY*MONTHS =
   (SELECT MAX(SALARY*MONTHS) FROM EMPLOYEE)
```

7. Query the Western Longitude (LONG_W) for the largest Northern Latitude (LAT_N) in STATION that is less than 137.2345. Round your answer to 4 decimal places.

```
SELECT ROUND(LONG_W, 4)
FROM STATION
WHERE LAT_N = (SELECT MAX(LAT_N)
        FROM STATION
        WHERE LAT_N < 137.3245)
```

8. Query the smallest Northern Latitude (LAT_N) from STATION that is greater than 38.7780. Round your answer to 4 decimal places.

```
SELECT ROUND(LAT_N, 4)
FROM STATION
WHERE LAT_N > 38.7780
ORDER BY LAT_N ASC
LIMIT 1

SELECT ROUND(LONG_W,4)
FROM STATION
WHERE LAT_N = ( SELECT MIN(LAT_N) FROM STATION WHERE LAT_N >
38.7780)
```

9. Write a query to print the company_code, founder name, total number of lead managers, total number of senior managers, total number of managers, and total number of employees. Order your output by ascending company_code.

```
select c.company_code, c.founder,
    count(distinct l.lead_manager_code),
    count(distinct s.senior_manager_code),
    count(distinct m.manager_code),
    count(distinct e.employee_code)
from Company as c
join Lead_Manager as l
on c.company_code = l.company_code
join Senior_Manager as s
on l.lead_manager_code = s.lead_manager_code
join Manager as m
on m.senior_manager_code = s.senior_manager_code
join Employee as e
on e.manager_code = m.manager_code
group by c.company_code, c.founder
order by c.company_code;
```