

September 1, 2018, SQL Special:  
175,595,613,627,584,586,620,610,6  
03,577,607,182,181,183,597,619,596,1  
96,197,176,570,608,612,626,585,60  
2,580,574,578,178,180,184,614,177,57  
1,618,569,601,615,579,185,262

☐ 262,579,571,618,569

## 175

```
SELECT FirstName, LastName, City, State FROM (Person AS p LEFT  
JOIN Address as a ON p.PersonId = a.PersonId)
```

## 595

1. SELECT

```
SELECT name, population, area FROM World WHERE area > 3000000  
OR population > 25000000;
```

2. UNION

```
SELECT name, population, area FROM World WHERE area > 3000000  
UNION  
SELECT name, population, area FROM World WHERE population > 25  
000000;
```

## 613

1. 开始一位是排好序的，所以想用错位，最后还是没搞出来：

```
SELECT MIN(ABS(a.x - b.x)) AS shortest FROM point AS a, point  
AS b WHERE a.x != b.x;
```

2. Use Self Join:

```
SELECT MIN(a.x - b.x) AS shortest FROM point as a JOIN point a  
s b WHERE a.x > b.x;
```

## 627

### 1. CASE WHEN STATEMENT

```
UPDATE salary  
SET sex = (CASE WHEN sex = 'm' THEN 'f' ELSE 'm' END);
```

### 2. 数学方法：

```
UPDATE salary  
SET sex = CHAR(ORD('f') + ORD('m') - ORD(sex));
```

## 584

### 1. 为啥还要加上 referee\_id IS NULL:

```
SELECT name FROM customer WHERE referee_id IS NULL OR referee_id != 2;
```

## 586

### 1. 非常笨的方法：

```
SELECT g.customer_number FROM  
(SELECT COUNT(order_number) AS cnt, customer_number FROM orders  
GROUP BY customer_number) AS g WHERE g.cnt =  
(SELECT MAX(f.cnt) FROM (SELECT COUNT(order_number) AS cnt FROM  
orders GROUP BY customer_number) AS f);
```

### 2. 标准答案还没老子的快：

```
SELECT g.customer_number FROM  
(SELECT COUNT(order_number) AS cnt, customer_number FROM orders  
GROUP BY customer_number ORDER BY cnt DESC LIMIT 1) AS g;
```

## 620

### 1. 我都能做出来的题：

```
SELECT * FROM cinema WHERE id%2 AND description != 'boring' ORDER BY rating DESC;
```

## 610

### 1. IF statement:

```
SELECT x, y, z, IF (x+y>z AND x+z>y AND y+z>x, 'Yes', 'No') as triangle FROM triangle;
```

## 2. CASE statement:

```
SELECT x, y, z, (CASE WHEN x+y>z AND x+z>y AND y+z>x THEN 'Yes' ELSE 'No' END) as triangle FROM triangle;
```

## 603

### 1. 做题简单要花九牛二虎之力：

```
SELECT a.seat_id AS seat_id FROM cinema AS a, cinema AS b WHERE a.seat_id = b.seat_id - 1 AND a.free AND b.free  
  
UNION  
  
SELECT a.seat_id AS seat_id FROM cinema AS a, cinema AS b WHERE a.seat_id = b.seat_id + 1 AND a.free AND b.free  
  
ORDER BY seat_id
```

## 577

### 1. 我都可以做对的题：

```
select name, bonus FROM  
  
(Employee LEFT JOIN Bonus ON Employee.empId = Bonus.empId) WHERE bonus IS NULL OR bonus < 1000;
```

## 607

### 1. 才知道SQL还有 IS IN 语句：

```
SELECT name FROM salesperson WHERE sales_id NOT IN  
  
(SELECT sales_id FROM orders WHERE com_id IN (  
  
SELECT com_id FROM company WHERE name = 'RED'))
```

## 182

### 1. 靠自己的力量做出来的：

```
SELECT r.Email FROM  
  
(SELECT Email, COUNT(*) AS Cnt FROM Person GROUP BY Email) AS r WHERE r.Cnt > 1;
```

## 181

1. 靠自己的力量做出来的：

```
SELECT Name As Employee FROM
Employee as e LEFT JOIN
(SELECT Id AS MId, Salary AS MSal FROM Employee) AS m ON e.Man
agerId = m.MId
WHERE Salary > MSal;
```

2. Only Use SELECT:

```
SELECT Name AS Employee FROM Employee AS e WHERE Salary > (SEL
ECT Salary FROM Employee WHERE Id = e.ManagerId);
```

## 183

1. IS IN 语句，刚学：

```
SELECT Name AS Customers FROM Customers WHERE Id NOT IN
(SELECT CustomerId FROM Orders);
```

2. Use JOIN:

```
SELECT Name AS Customers FROM
(Customers LEFT JOIN (SELECT Id AS OrderId, CustomerId FROM Or
ders) AS o ON Customers.Id = o.CustomerId)
WHERE OrderId IS NULL;
```

## 597

1. 注意看题：没说只统计在friend\_request 中出现的：

```
SELECT (CASE WHEN g.d > 0 THEN ROUND(g.c/g.d, 2) ELSE 0.0 END)
AS accept_rate FROM
(SELECT COUNT(DISTINCT a.requester_id, a.accepter_id) AS c, CO
UNT(DISTINCT b.sender_id, b.send_to_id) AS d FROM
request_accepted AS a, friend_request AS b) AS g;
```

## 619

1. 独立完成！

```
SELECT MAX(b.num) as num FROM
(SELECT a.num FROM
```

```
(SELECT num, COUNT(*) AS cnt FROM number GROUP BY num) AS a W  
HERE a.cnt = 1) AS b;
```

## 596

1. 独立完成：

```
SELECT a.class FROM  
(SELECT class, COUNT(DISTINCT student) AS cnt FROM courses GRO  
UP BY class) AS a WHERE a.cnt >= 5;
```

2. 学到了一个 `HAVING` statement:

```
SELECT class FROM courses  
GROUP BY class  
HAVING COUNT(DISTINCT(student))>=5
```

## 196

1. 居然意外得做对了：

```
DELETE FROM Person WHERE (Id, Email) NOT IN(  
    SELECT a.Id, a.Email FROM(  
        SELECT MIN(Id) AS Id, Email FROM Person GROUP BY Emai  
l) AS a);
```

## 197

1. 如果不知道 `datediff` 函数，这题完全没法做：

```
SELECT DISTINCT a.Id FROM Weather AS a, Weather AS b WHERE dat  
ediff(a.RecordDate, b.RecordDate) = 1 AND a.Temperature > b.Te  
mperature;
```

2. USE `SELECT` and `TO_DAYS` 函数：

```
SELECT t1.Id FROM Weather AS t1 INNER JOIN Weather AS t2 ON TO  
_DAYS(t1.RecordDate) = TO_DAYS(t2.RecordDate) + 1  
WHERE t1.Temperature > t2.Temperature
```

## 176

1. 独立完成：

```
SELECT (CASE WHEN (SELECT COUNT(DISTINCT Salary) FROM Employee) >=2 THEN a.Salary ELSE NULL END) AS SecondHighestSalary FROM  
M  
(SELECT DISTINCT Salary FROM Employee ORDER BY Salary DESC LIMIT 2) AS a ORDER BY a.Salary LIMIT 1;
```

## 2. 高手解法：

```
SELECT MAX(Salary) AS SecondHighestSalary FROM Employee WHERE  
Salary != (SELECT MAX(Salary) FROM Employee)
```

## 3. 下面这个不懂：

☐ 到底是 Which Union Which???

```
SELECT Salary AS SecondHighestSalary FROM Employee  
UNION  
SELECT NULL  
ORDER BY SecondHighestSalary DESC LIMIT 1,1
```

# 570

## 1. 独立完成的第一个Medium：

```
SELECT a.Name FROM  
Employee AS a INNER JOIN  
(SELECT ManagerId, COUNT(*) AS cnt FROM Employee GROUP BY ManagerId) AS b ON a.Id = b.ManagerId  
Where b.cnt >= 5;
```

## 2. Use HAVING COUNT:

```
SELECT Name FROM Employee WHERE Id IN  
(SELECT ManagerId FROM Employee GROUP BY ManagerId HAVING COUNT(*) >= 5);
```

# 608

## 1. 独立完成：

```
SELECT id,  
(CASE WHEN p_id IS NULL THEN 'Root' WHEN id IN (SELECT p_id FROM tree) THEN 'Inner' ELSE 'Leaf' END)  
AS type FROM tree ORDER BY id;
```

☐ 为什么反过来用 NOT IN 不work :

```
SELECT id,  
(CASE WHEN p_id IS NULL THEN 'Root' ELSE (CASE WHEN id NOT IN  
(SELECT p_id FROM tree) THEN 'Leaf' ELSE 'Inner' END) END)  
AS type FROM tree ORDER BY id;
```

2. 这个可以 :

```
SELECT id,  
IF(p_id IS NULL, 'Root', IF(id NOT IN (SELECT p_id FROM tree W  
HERE p_id IS NOT NULL), 'Leaf', 'Inner')) AS type FROM tree OR  
DER BY id;
```

3. 用 IF:

```
# Write your MySQL query statement below  
SELECT id,  
IF(p_id IS NULL, 'Root', IF(id IN (SELECT p_id FROM tree), 'In  
ner', 'Leaf')) AS type FROM tree ORDER BY id;
```

## 612

1. 独立完成 :

```
SELECT ROUND(MIN(c.r), 2) AS shortest FROM  
(SELECT SQRT((a.x-b.x)*(a.x-b.x) + (a.y-b.y)*(a.y-b.y)) AS r F  
ROM point_2d AS a, point_2d AS b WHERE a.x != b.x OR a.y != b.  
y)  
AS c;
```

2. 用 JOIN:

```
SELECT MIN( ROUND ( SQRT( POWER(L.X-R.X,2) + POWER(L.Y-R.Y,  
2) ), 2) ) AS shortest FROM POINT_2D L  
JOIN POINT_2D R  
ON L.X != R.X OR L.Y != R.Y;
```

## 626

1. 独立完成 , Ugly Solution:

```
SELECT x.id, IF(x.id%2=0 || x.id<(SELECT COUNT(*) FROM seat),
```

```
        (SELECT student FROM seat WHERE id=x.id - 1 +
2*(x.id%2)), x.student)
AS student FROM seat AS x ORDER BY x.id;
```

## 2. 自己研发的NB方法：

```
SELECT a.id, b.student FROM seat AS a,
(SELECT * FROM seat
UNION ALL
(SELECT (id+1) as id, student FROM seat ORDER BY id DESC LIMIT
1)) as b
WHERE b.id = a.id - 1 + 2*(a.id%2);
```

## 585

### 1. 用了九牛二虎之力终于做出来了：

```
SELECT ROUND(SUM(TIV_2016), 2) AS TIV_2016 FROM insurance WHERE
TIV_2015 IN (SELECT TIV_2015 FROM insurance GROUP BY TIV_2015
HAVING COUNT(*) > 1) AND
(LAT, LON) IN (SELECT LAT, LON FROM insurance GROUP BY LAT, LO
N HAVING COUNT(*) = 1);
```

## 602

### 1. 这题真是莫名其妙：非要把COUNT DISTINCT去掉才过

```
SELECT y.id, y.num FROM
(SELECT x.id, COUNT(x.frd) AS num FROM
(SELECT requester_id AS id, acceptor_id AS frd FROM request_ac
cepted
UNION ALL
SELECT acceptor_id AS id, requester_id AS frd FROM request_acc
epted) AS x GROUP BY x.id)
AS y ORDER BY y.num DESC LIMIT 1;
```

☐ 这个反而错了：

```
SELECT y.id, y.num FROM
```



```
(SELECT x.id, COUNT(DISTINCT x.frd) AS num FROM
(SELECT requester_id AS id, acceptor_id AS frd FROM request_ac
cepted
UNION ALL
SELECT acceptor_id AS id, requester_id AS frd FROM request_acc
epted) AS x GROUP BY x.id)
AS y ORDER BY y.num DESC LIMIT 1;
```

## 580

1. 注意排序即可：

```
SELECT z.dept_name, IF(z.cnt IS NULL, 0, z.cnt) AS student_num
ber FROM
(SELECT dept_name, cnt FROM (SELECT dept_id, COUNT(DISTINCT st
udent_id) AS cnt FROM student GROUP BY dept_id) as x
RIGHT JOIN department ON x.dept_id = department.dept_id) AS z
ORDER BY z.cnt DESC, z.dept_name;
```

2. 更简洁：

```
SELECT dept_name, COUNT(DISTINCT student_id) AS student_number
FROM
student RIGHT JOIN department ON student.dept_id = department.
dept_id
GROUP BY dept_name ORDER BY student_number DESC, dept_name;
```

## 574

1. 题意很模糊，不仅要票数最多，二且要再Candidate Table 里：

```
SELECT g.Name FROM
(SELECT x.Name, COUNT(DISTINCT x.id) AS cnt FROM
(SELECT Vote.id AS id, Candidate.Name as Name FROM Vote LEFT
JOIN Candidate ON Vote.CandidateId = Candidate.id)
AS x GROUP BY x.Name ORDER BY cnt DESC LIMIT 1) AS g WHERE g.
Name in (SELECT Name FROM Candidate);
```

## 578

1. 题都没读懂，莫名其妙就过了。。。

```
SELECT z.question_id AS survey_log FROM
(SELECT x.question_id, (x.ans / y.shw) AS rate FROM
((SELECT a.question_id, COUNT(*) as ans FROM (SELECT question_
id FROM survey_log WHERE action = 'answer')
AS a GROUP BY a.question_id) AS x
INNER JOIN
(SELECT b.question_id, COUNT(*) as shw FROM (SELECT question_i
d FROM survey_log)
AS b GROUP BY b.question_id) AS y
ON x.question_id = y.question_id) ORDER BY rate DESC LIMIT 1)
AS z;
```

2. 原来还能这么些：

```
select question_id as survey_log
from survey_log
group by question_id
order by count(answer_id)/count(*) desc
limit 1
```

## 178

1. 很简单：

```
SELECT y.Score AS Score, (SELECT COUNT(DISTINCT x.Score) FROM
Scores AS x WHERE x.Score>y.Score) + 1 AS Rank
FROM Scores AS y ORDER BY Rank;
```

2. 这个完全没懂

☐ [@Zebo L](https://leetcode.com/problems/rank-scores/discuss/165000/One-line-solution-and-fast)

## 180

1. 太Easy：

```
SELECT DISTINCT x.Num AS ConsecutiveNums FROM Logs AS x, Logs
AS y, Logs As z
```

```
WHERE x.Id = y.Id-1 AND x.Id = z.Id-2 AND x.Num = y.Num AND x.  
Num = z.Num;
```

## 2. Use INNER JOIN:

```
SELECT DISTINCT x.Num AS ConsecutiveNums FROM  
Logs AS x INNER JOIN  
Logs AS y ON x.Id = y.Id + 1 AND x.Num = y.Num INNER JOIN  
Logs AS z ON x.Id = z.Id + 2 AND x.Num = z.Num;
```

## 184

### 1. 非常笨的方法，用Salary的value做JOIN:

```
SELECT x.Name AS Department, y.Name AS Employee, y.Salary AS S  
alary FROM  
(SELECT Department.Name AS Name, Department.Id AS Id, MAX(Sala  
ry) AS val FROM  
(Employee INNER JOIN Department ON Employee.DepartmentId = Dep  
artment.Id) GROUP BY Department.Name)  
AS x LEFT JOIN Employee AS y ON x.val = y.Salary AND x.Id = y.  
DepartmentId;
```

### 2. 高手解法，利用比最大 Salary大的value的个数为0:

```
select a.Name as Department,b. Name as Employee ,b.Salary from  
Department as a join  
(select * from Employee e1 where(select count(*) from Employee  
e2 where e2.Salary>e1.Salary and e1.DepartmentId=e2.Department  
Id)<1) as b  
on b.DepartmentId=a.Id
```

## 614

### 1. 简单的 GROUP BY 语句：

```
SELECT x.follower AS follower, COUNT(DISTINCT y.follower) AS n  
um FROM  
follow AS x, follow AS y WHERE x.follower = y.followee GROUP B  
Y x.follower;
```

## 177

### 1. 居然tm 过了：第一次遇到定义函数

```
CREATE FUNCTION getNthHighestSalary(N INT) RETURNS INT
BEGIN
  DECLARE M INT;
  SET M = N-1;
  RETURN (
    # Write your MySQL query statement below.
    SELECT DISTINCT Salary FROM Employee ORDER BY Salary DESC
    LIMIT M,1
  );
END
```

- CREATE FUNCTION FUNCTION\_NAME(VAR TYPE) RETURN TYPE
- BEGIN
- DECLARE VARS TYPES;...
- SET VARS = ? (Initialization)
- RETURN(SELECT 语句)
- END

## 571

### 1. 超时的解法：

```
SELECT AVG(u.target) AS median FROM(
SELECT MIN(w.num) AS target FROM
(SELECT z.num, SUM(z.cnt) AS accu FROM
(SELECT x.Number, y.Number AS num, (SELECT @total := SUM(Frequency) FROM Numbers), x.Frequency AS cnt FROM Numbers x, Numbers y WHERE x.Number <= y.Number)
AS z GROUP BY z.num) AS w WHERE w.accu >= FLOOR((@total + 1)/
2)
UNION
SELECT MIN(w.num) AS target FROM
(SELECT z.num, SUM(z.cnt) AS accu FROM
```

```
(SELECT x.Number, y.Number AS num, x.Frequency AS cnt FROM Numbers x, Numbers y WHERE x.Number <= y.Number)
AS z GROUP BY z.num) AS w WHERE w.accu >= FLOOR((@total + 2)/2)) AS u;
```

## 2. 别人的解法：

```
select avg(Number) median from(
    select Number, @prev := @count as prevCount, (@count := @count + Frequency) as curCount
    from Numbers, (select @prev:=0, @count:=0, @total:=(select sum(Frequency) from Numbers)) temp order by Number
) n1
where n1.curCount >= floor((@total+1)/2)
and @total-n1.prevCount >= floor((@total+1)/2)
```

# 618

## SQL Hard, 九牛二虎

```
SELECT a.name as America, b.name as Asia, c.name as Europe FROM
M
(SELECT @arank := @arank + 1 AS Id, s.name AS name FROM
(SELECT name FROM student where continent = 'America' ORDER BY
name) AS s, (SELECT @arank := 0) AS r) AS a
LEFT JOIN
(SELECT @brank := @brank + 1 AS Id, s.name AS name FROM
(SELECT name FROM student where continent = 'Asia' ORDER BY name) AS s, (SELECT @brank := 0) AS r) AS b ON a.Id = b.Id
LEFT JOIN
(SELECT @crank := @crank + 1 AS Id, s.name AS name FROM
(SELECT name FROM student where continent = 'Europe' ORDER BY
name) AS s, (SELECT @crank := 0) AS r) AS c ON a.Id = c.Id
```

## GET INDEX 用：

### 1. SELECT @RANK

```
SELECT @rank := @rank + 1 AS Id, col FROM tablename, (SELECT @rank := 0) AS r
```

## 2. ROW\_NUMBER():

```
SELECT ROW_NUMBER() OVER(ORDER BY YourColumn) AS Rank FROM tablename
```

**569**

**601**

### 1. 靠自己的力量做出的第一道 Hard，虽然方法很笨：

```
SELECT w.id, w.date, w.people FROM
(
    SELECT x.id, x.date, x.people FROM stadium x, stadium y, stadium z WHERE
        x.id = y.id-1 AND x.id = z.id-2 AND x.people >= 100 AND y.people >= 100 AND z.people >= 100
    UNION
    SELECT y.id, y.date, y.people FROM stadium x, stadium y, stadium z WHERE
        x.id = y.id-1 AND x.id = z.id-2 AND x.people >= 100 AND y.people >= 100 AND z.people >= 100
    UNION
    SELECT z.id, z.date, z.people FROM stadium x, stadium y, stadium z WHERE
        x.id = y.id-1 AND x.id = z.id-2 AND x.people >= 100 AND y.people >= 100 AND z.people >= 100
) AS w ORDER BY w.id;
```

### 2. 这个好像更直接一些：

```
select * from stadium s1
where s1.people>=100 and
((select count(*) from stadium s2 where s2.id in (s1.id+1,s1.id+2) and s2.people>=100)>=2
or (select count(*) from stadium s2 where s2.id in (s1.id+1,s1.id-1) and s2.people>=100)>=2
```

```
or (select count(*) from stadium s2 where s2.id in (s1.id-2,s1.id-1) and s2.people>=100)>=2)
```

### 3. VARIABLE usage?

☐ Ask and investigate.

```
set @a := @b := 0;
select id, date, people
from (
    select *, @b := (@b + 1) * (people >= 100) occmb
    from (
        select *, @a := (@a + 1) * (people >= 100) occma
        from stadium
        order by id desc
    ) t
    order by id
) s
where occma + occmb > 3
```

## 615

### 1. 靠自己做出来的第二道 Hard :

```
SELECT x.pay_date AS pay_month, x.department_id, IF(x.avg0>y.avg1, 'higher', if(x.avg0<y.avg1, 'lower', 'same')) AS comparison FROM
((SELECT e.department_id AS department_id, (SUM(s.amount)/COUNT(s.amount)) AS avg0, date_format(s.pay_date, "%Y-%m") AS pay_date FROM
salary s LEFT JOIN employee e ON s.employee_id = e.employee_id
GROUP BY e.department_id, date_format(s.pay_date, "%Y-%m")) AS x
LEFT JOIN
(SELECT (SUM(amount)/COUNT(amount)) AS avg1, date_format(pay_date, "%Y-%m") AS pay_date FROM salary GROUP BY date_format(pay_date, "%Y-%m")) AS y ON x.pay_date = y.pay_date)
```

## 2. Leetcode discussion:

- ☐ <https://leetcode.com/problems/average-salary-departments-vs-company/discuss/118586/Simple-solution-using-one-subqueries-With-Explanation>
- ☐ <https://leetcode.com/problems/average-salary-departments-vs-company/discuss/104245/AC-solution-using-join>

## 579

### 1. Use two tables to get Cumulative Sum:

```
SELECT z.Id, MAX(z.Month) AS Month, SUM(z.tar) AS Salary FROM
(SELECT x.Id, x.Month, x.Salary AS tar, y.Month AS Mon FROM Em
ployee x, Employee y
WHERE x.Id = y.Id AND x.Month < y.Month AND x.Month >= y.Month
-3) AS z GROUP BY z.Id, z.Mon
ORDER BY z.Id, Month DESC;
```

### 2. Another 高手的Solution :

```
select E1.id, E1.month, (ifnull(E1.salary,0) +ifnull(E2.salar
y,0) + ifnull(E3.salary,0)) as Salary from
(Select id,max(month) as month from Employee group by id havin
g count(*) > 1) as maxmonth
left Join Employee E1 on (maxmonth.id = E1.id and maxmonth.mon
th > E1.month)
left Join Employee E2 on (E1.id = E2.id and E1.month = E2.mont
h + 1)
left Join Employee E3 on (E1.id = E3.id and E1.month = E3.mont
h + 2)
Order by id ASC, month DESC
```

## 185

### 1. 靠自己做出来的第三道Hard :

```
SELECT w.Name AS Department, z.Employee, z.Salary FROM
(
SELECT x.DepartmentId, x.Name AS Employee, x.Salary FROM Emplo
yee x, Employee y
```



```

WHERE x.DepartmentId = y.DepartmentId AND x.Salary <= y.Salary
GROUP BY x.Id HAVING COUNT(DISTINCT y.Salary) <= 3
) AS z
JOIN Department AS w ON z.DepartmentId = w.Id
ORDER BY Department, z.Salary DESC

```

## 2. Leetcode discussion:

☐ <https://leetcode.com/problems/department-top-three-salaries/discuss/>

# 262

## 1. Ugly solution:

```

SELECT x.Day as Day, ROUND(y.can/x.tot, 2) AS 'Cancellation Rate' FROM
(SELECT Request_at AS Day, COUNT(*) AS tot FROM Trips WHERE
Client_Id IN (SELECT Users_Id FROM Users WHERE Banned = 'No')
AND
Driver_Id IN (SELECT Users_Id FROM Users WHERE Banned = 'No')
AND Request_at BETWEEN '2013-10-01' AND '2013-10-03'
GROUP BY Request_at) x
LEFT JOIN
(SELECT Request_at AS Day, SUM(Client_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Yes'))
AND
Driver_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Yes'))
AND
Status != 'completed') AS can FROM Trips
WHERE Request_at BETWEEN '2013-10-01' AND '2013-10-03'
GROUP BY Request_at) y ON x.Day = y.Day

```

- 学习了 BETWEEN AND clause:
- WHERE col BETWEEN start\_date AND end\_date

## 2. Without using JOIN:

```
SELECT Request_at As Day, ROUND(SUM(
Client_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Y
es')
AND
Driver_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Y
es')
AND
Status != 'completed') / SUM(
Client_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Y
es')
AND
Driver_Id NOT IN (SELECT Users_Id FROM Users WHERE Banned = 'Y
es')
), 2) AS 'Cancellation Rate' FROM Trips WHERE Request_at BETWE
EN '2013-10-01' AND '2013-10-03'
GROUP BY Request_at
```