### Question 1

**Skills**: Coalesce, IFNULL, ISNULL

**Write an SQL query to report the distance traveled by each user.**

**DB -** [**https://www.db-fiddle.com/f/uNZryJ1ibvBXQoTCMzKi3h/0**](https://www.db-fiddle.com/f/uNZryJ1ibvBXQoTCMzKi3h/0)

Return the result table ordered by travelled\_distance in descending order, if two or more users traveled the same distance, order them by their name in ascending order.

SELECT a.name,IFNULL(sum(b.distance),0) as 'travelled\_distance'

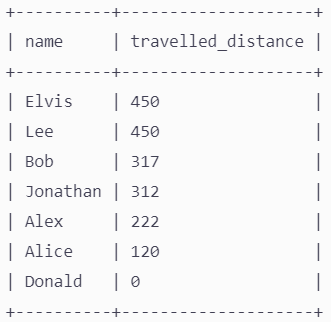
FROM Users a

LEFT JOIN Rides b

ON a.id = b.user\_id

GROUP BY name

ORDER BY travelled\_distance DESC ,name ASC



### Question 2

**Skills**: Subquery in WHERE

As a programmer write a SQL query to display details of all the trainee (**traineeid, traineename, cgpa** and **technology trained**) for the trainees who are eligible for **.NET** project but aren't allocated with any. Display the result in the increasing order of **traineeid**.

DB -<https://www.db-fiddle.com/f/vQepCw3A4RnpZMF3tP6rQm/0>

SELECT traineeid, traineename, cgpa, technology

FROM trainee

WHERE technology = '.NET' AND traineeid NOT IN (

SELECT traineeid

FROM projectallocation a

JOIN projectdetails b

ON a.projectid = b.projectid

WHERE b.technology = '.NET')

ORDER BY traineeid ASC

**(OR)**

SELECT traineeid, traineename, cgpa from trainee

WHERE cgpa >=

ANY(select mincgparequired from projectdetails where technology = ".NET") AND

traineeid <> all(select traineeid from projectallocation)

and technology = ".NET"

### Question 3

**Skills**: WINDOWS FUNCTION (LEAD, LAG), CASE WHEN

**Print the person's name with consecutive years (twice).**

Solve the same question using **WINDOWS FUNCTION**

DB -<https://www.db-fiddle.com/f/8Y61oCnyAkz64v3qFgVV4k/0>

SELECT t1.Name, t1.next\_year, t2.prev\_year

FROM (

SELECT Name, Year, LEAD(Year) OVER (PARTITION BY Name ORDER BY Year) AS next\_year

FROM t

) t1

JOIN (

SELECT Name, Year, LAG(Year) OVER (PARTITION BY Name ORDER BY Year) AS prev\_year

FROM t

) t2 ON t1.Name = t2.Name AND t1.Year = t2.prev\_year

ORDER BY t1.Name,t1.next\_year, t2.prev\_year

**Solve the same question using CASE - WHEN**

SELECT e.Name, e.Year as previous\_year, n.Year as next\_year

FROM t e

JOIN t n

ON e.Name = n.Name

WHERE CASE WHEN e.Year = n.Year - 1 THEN 1 ELSE 0 END = 1

ORDER BY e.Name, e.Year, n.Year

### Question 4

**Skills**: JOIN, String Functions - LOWER, REPLACE, SUBSTR

Use<https://www.db-fiddle.com/f/saATtNXX7L7HBJDDwVmpHW/0>

SELECT name,email,company\_id,company\_name

FROM user\_profile

CROSS JOIN

company\_profile ON (email LIKE CONCAT('%@', company\_name, '.%'))

**(OR)**

SELECT \*,Substr(a.email, Locate('@', a.email) + 1)

FROM user\_profile a

JOIN company\_profile b

ON Lower(b.company\_name) = REPLACE(Substr(Replace(a.email,'fb','facebook'), Locate('@', a.email) + 1), '.com', '');



**(OR)**

WITH CTE AS (

SELECT \*,substring\_index(SUBSTR(email,INSTR(email,'@')+1),'.',1)as domain\_name

FROM user\_profile )

SELECT name,email,company\_id,company\_name FROM company\_profile cmpny

JOIN CTE a

ON cmpny.company\_name = a.domain\_name

### Question 5

**Skills**: Subquery in FROM, String Functions - LEFT, INSTR, SUBSTRING

**Solve the previous problem with alternative string functions.**

WITH CTE AS (

SELECT up.\*,

LEFT(SUBSTRING\_INDEX(SUBSTRING\_INDEX(up.email, '@', -1), '.', 1), 10) AS domain\_name

FROM user\_profile up)

SELECT name,email,company\_id,company\_name

FROM company\_profile cmpny

JOIN CTE a

on cmpny.company\_name = a.domain\_name;



### Question 6

**Skills: SUM OVER PARTITION BY**

**Calculate the cumulative revenue for the year**

SELECT ID,Title,Year,Revenue\_millions,SUM(Revenue\_millions)

OVER (PARTITION BY Year ORDER BY Year ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cumulative\_revenue\_by\_year

FROM imdb\_movies

WHERE Revenue\_millions IS NOT NULL



**The below image shows that for each new year, i.e, 2007 sum of revenue has to start from 0**



SO FOR THE ABOVE IMAGE IT WAS PARTITION BY Year TO GET THE SAME OUTPUT .

### Question 7

Find the top 3 products(‘productLine’) for each month generated the highest profit

**Profit** = **( Price per unit - Buying Price per Unit ) x No. Of Units Ordered**

SELECT \*

FROM (SELECT month\_name,productline,profit,Rank()

OVER (

partition BY month\_name

ORDER BY profit DESC) AS Ranks

FROM (SELECT productline,Monthname(o.orderdate) AS Month\_Name,

Sum(( od.priceeach - p.buyprice ) \* od.quantityordered) AS profit

FROM cr\_orderdetails od

JOIN cr\_orders o

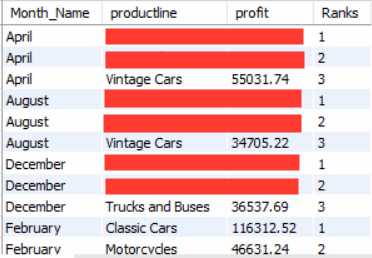
ON od.ordernumber = o.ordernumber

JOIN cr\_products p

ON p.productcode = od.productcode

GROUP BY month\_name,productline) a) b

WHERE ranks <= 3



### Question 8

**Skills**: DENSE RANK OVER PARTITION BY

SELECT Salary AS SecondHighestSalary

FROM (

SELECT Salary, DENSE\_RANK() OVER (ORDER BY Salary DESC) AS DenseRank

FROM Employee

) AS RankedSalary

WHERE DenseRank = 2

**(OR)**

WITH CTE AS(

SELECT Salary ,DENSE\_RANK()

OVER(ORDER BY Salary DESC) AS DenseRank

FROM Employee)

SELECT Salary AS SecondHighestSalary

FROM CTE

WHERE DenseRank = 2

+--------------------------------+

| SecondHighestSalary |

+--------------------------------+

| 200 |

+--------------------------------+

### Question 9

Fix Names in a Table

DB -<https://www.db-fiddle.com/f/v3Qq8ftURBFg1cZL4e7Vd4/0>

**Skills**: STRING Function - UPPER, LOWER, LEFT, SUBSTRING

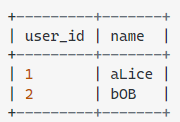
SELECT user\_id, CONCAT(UPPER(SUBSTRING(name, 1, 1)),

LOWER(SUBSTRING(name, 2))) AS name

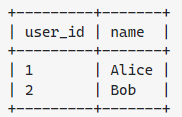
FROM users

ORDER BY user\_id

**Users table:**

****

**Result table:**

****

### Question 10

**Skills**: Coalesce, SUBSTRING

DB -<https://www.db-fiddle.com/f/eRQeiaYJRpaMjW9HKLLg4V/0>

SELECT

COALESCE(id,0) AS id,

COALESCE(p.name, 'No Name') AS name,

COALESCE(phone\_number,

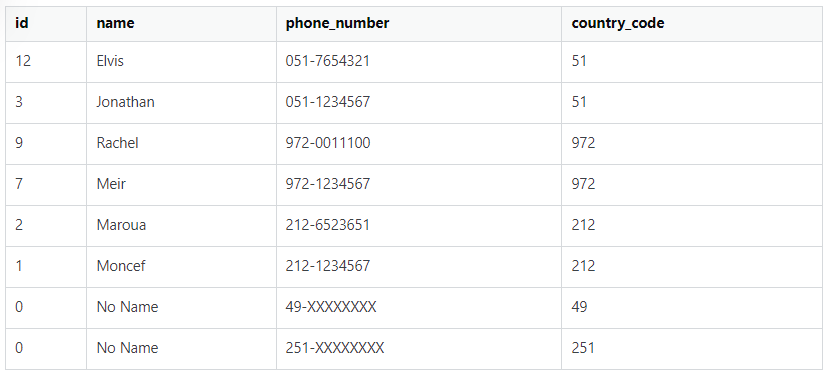
CONCAT(c.country\_code, '-', 'XXXXXXXX')

) AS phone\_number, c.country\_code

FROM person p

RIGHT JOIN country c

ON c.country\_code = SUBSTRING(p.phone\_number FROM 1 FOR 3)



### Question 11

Filter the ‘In Process’ orders and replace the missing comments with ‘Status yet to be updated with the comment’

Use “**Car Retailer**” dataset

**Instructions**: Use COALESCE or IFNULL to solve the problem

**Skills**: COALESCE or IFNULL

SELECT \*,

COALESCE(comments, 'Status yet to be updated with the comment') AS NewComments

FROM cr\_orders

WHERE status = 'In Process'

**AS IN THE TABLE THERE IS NO In Process SO WE WON'T BE GETTING THE RESULT .**

### *Question 12*

*Print the address of the customer as follows,****Format:*** *addressLine1, {state}, city {- postcode}, country*

*addressLine1 followed by the state if present, followed by postal code if present else city, and followed by the country.*

***Instructions****: Use COALESCE or IFNULL to solve the problem*

***Skills****: COALESCE or IFNULL and CONCAT and CONCAT\_WS*

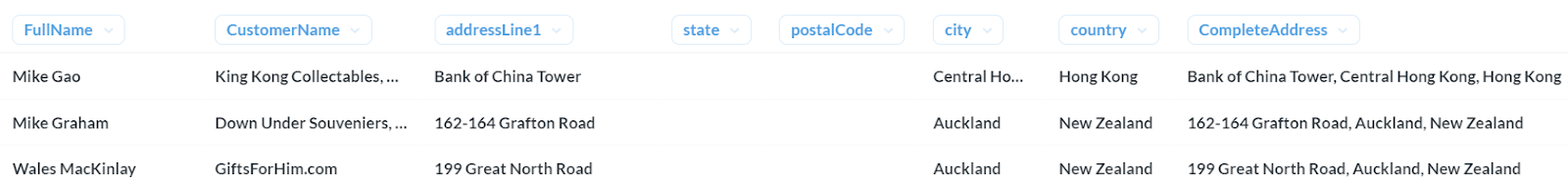
*SELECT CONCAT(contactFirstName,' ',contactLastName) AS FullName,*

*customerName,addressLine1,state,postalCode,city,country,*

*Concat\_ws(', ', addressline1, COALESCE(state, IFNull(postalcode, city)), country) AS CompleteAddress*

*FROM cr\_customers*

*ORDER BY postalCode*

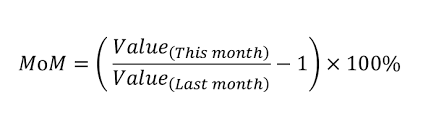
**

**

### *Question 13*

*Find the Month on Month growth in profit for each year.*

*MoM\_growth is calculated as follows.*

**

*For example,*

*MoM (Feb 2003) = ((Annual Revenue of Feb 2003/Annual Revenue of Jan 2003) - 1) \* 100*

***Note****: Please make sure that MoM is calculated for each year. Ideally, the MoM growth of the first month of every year should be NULL, as shown in the sample output below.*

*SELECT Year\_Name, Month\_Name, Profit,*

*CASE*

*WHEN Month\_Num = 1 THEN NULL*

*ELSE ((Profit / LAG(Profit, 1) OVER (PARTITION BY Year\_Name ORDER BY Month\_Num)) - 1) \* 100*

*END AS MoM\_Growth*

*FROM (*

*SELECT*

*Year(o.orderDate) AS Year\_Name,*

*Monthname(o.orderDate) AS Month\_Name,*

*Month(o.orderDate) AS Month\_Num,*

*SUM((od.priceEach - p.buyPrice) \* od.quantityOrdered) AS Profit*

*FROM cr\_orderdetails od*

*JOIN cr\_orders o ON od.orderNumber = o.orderNumber*

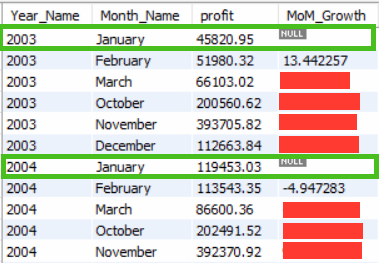
*JOIN cr\_products p ON od.productCode = p.productCode*

*GROUP BY 1, 2, 3*

*) a;*

***Output****:*

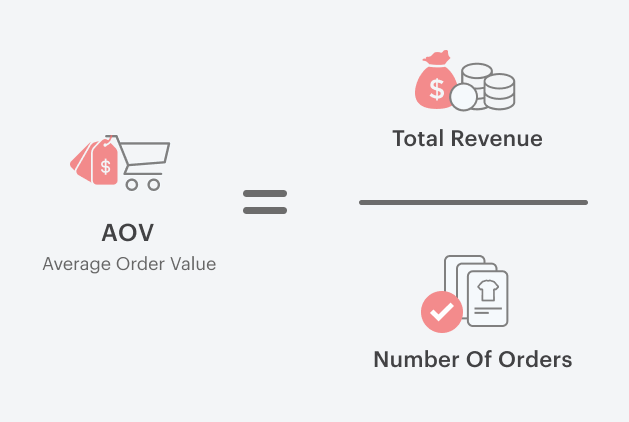
*Only the first three and last months are shown for each year. This is the sample output. You must have all the year's months in the final output.*

**

### *Question 14*

*For each customer i.e (customerNumber) count the number of orders above and below the average order value.*

***“Average Order Value(AOV)”*** *is calculated as below,*

**

***Total Revenue*** *is calculated as the sum of the revenue of each order.*

*Each order revenue is* ***Price per unit \* No. Of Units****.*

***Open Hint****: AVG() OVER() Windows Function can be used here to calculate AOV.*

*Since you need to get the count of orders above and below the AOV, the final output looks like the one below.*

*Conditions to get the following details,*

***Order Above AVG*** *= Actual Order Value* ***>=*** *AVG Order Value*

***Order Below AVG*** *= Actual Order Value* ***<*** *AVG Order Value*

*SELECT customerNumber,contactFirstName,contactLastName,*

*COUNT(CASE WHEN order\_total >= avg\_order\_value THEN 1 END) AS orders\_above\_avg,*

*COUNT(CASE WHEN order\_total < avg\_order\_value THEN 1 END) AS orders\_below\_avg*

*FROM (*

*SELECT c.customerNumber,*

*c.contactFirstName,*

*c.contactLastName,*

*o.orderNumber,*

*od.quantityOrdered,od.priceEach,*

*(priceEach \* quantityOrdered) AS order\_total,*

*ROUND(AVG(priceEach \* quantityOrdered) OVER(),2) avg\_order\_value*

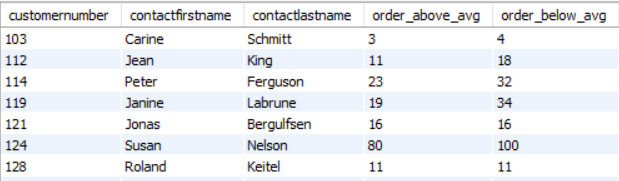
*FROM cr\_customers c*

*JOIN cr\_orders o ON c.customerNumber = o.customerNumber*

*JOIN cr\_orderdetails od ON o.orderNumber = od.orderNumber) AS subquery*

*GROUP BY 1,2,3*

*ORDER BY 1 ASC;*

**

*What is the Average Order Value? Submit the value in the answer.*

***Note****: Round off the value to 2 decimal and use it for the rest of the calculation.*

*SELECT c.customerNumber,*

*c.contactFirstName,*

*c.contactLastName,*

*o.orderNumber,*

*od.quantityOrdered,od.priceEach,*

*(priceEach \* quantityOrdered) AS order\_total,*

*ROUND(AVG(priceEach \* quantityOrdered) OVER(),2) avg\_order\_value*

*FROM cr\_customers c*

*JOIN cr\_orders o ON c.customerNumber = o.customerNumber*

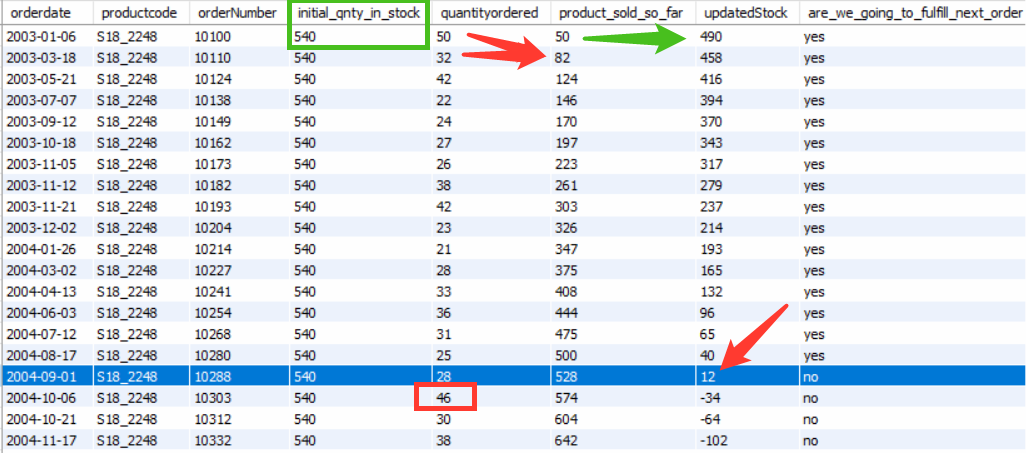
*JOIN cr\_orderdetails od ON o.orderNumber = od.orderNumber*

***Average Order Value - 3205.67***

### *Question 15*

In the business, customers order the product in bulk, and it is the utmost priority of the business to fulfill the order requirement as to when required. Hence, a business must keep checking their stock regularly to deliver uninterrupted.

To make that happen, create a report as shown below to get the updated stock and status to check if we are running out of stock (or are we going to fulfill the next customer order?)

****

Consider the first row as an example where product code “S18\_2248”. Its initial stock quantity was 540. After the first order i.e. ordernumber 10100 from the customer i.e. order quantity of 50, the updated stock value would be (540 - 50) 490. Now, the status needs to be updated based on the next order quantity, for example, the Next order is of quantity 32, and we have 490 products in stock hence we can easily serve the next order request (Status is Yes)

But if you look at the blue highlighted row then, in that case, “updatedStock” value is 12, and next order is of quantity 46 hence the status would No.

SELECT orderDate,

productCode,

orderNumber,

quantityInStock AS initial\_qnty\_in\_stock,

quantityOrdered,

product\_sold\_so\_far,

quantityInStock - product\_sold\_so\_far AS updatedStock,

IF (quantityInStock - product\_sold\_so\_far -

LEAD(quantityOrdered , 1, 0)

OVER(PARTITION BY productCode

ORDER BY orderDate) >= 0, "YES", "NO" )

AS are\_we\_going\_to\_fulfill\_next\_order

FROM (SELECT o.orderDate,

o.orderNumber,

p.productCode,

od.quantityOrdered,

od.priceEach,

p.quantityInStock,

SUM(od.quantityOrdered)

OVER(

PARTITION BY p.productCode

ORDER BY o.orderDate) AS product\_sold\_so\_far

FROM cr\_orderdetails od

JOIN cr\_products p ON od.productCode = p.productCode

JOIN cr\_orders o ON od.orderNumber = o.orderNumber ) a

**Select the list of productCodes that will be getting out of Stock (i.e, “are\_we\_going\_to\_fulfill\_next\_order” value is No)**

SELECT DISTINCT productCode

FROM (

SELECT orderDate,

productCode,

orderNumber,

quantityInStock AS initial\_qnty\_in\_stock,

quantityOrdered,

product\_sold\_so\_far,

quantityInStock - product\_sold\_so\_far AS updatedStock,

IF(

quantityInStock - product\_sold\_so\_far - LEAD(quantityOrdered, 1, 0)

OVER (PARTITION BY productCode ORDER BY orderDate) >= 0,

"YES",

"NO"

) AS are\_we\_going\_to\_fulfill\_next\_order

FROM (

SELECT o.orderDate,

o.orderNumber,

p.productCode,

od.quantityOrdered,

od.priceEach,

p.quantityInStock,

SUM(od.quantityOrdered)

OVER (

PARTITION BY p.productCode

ORDER BY o.orderDate

) AS product\_sold\_so\_far

FROM cr\_orderdetails od

JOIN cr\_products p ON od.productCode = p.productCode

JOIN cr\_orders o ON od.orderNumber = o.orderNumber

) a

) b

WHERE are\_we\_going\_to\_fulfill\_next\_order = "NO";

**productCode**

**S12\_1099**

**S18\_2248**

**S18\_2795**

**S24\_2000**

**S32\_1374**

**S32\_3522**

**S32\_4289**

**S50\_4713**

**S700\_1938**

**S700\_3167**

**S72\_3212**

### Question 16

Find all products whose quantity exceeds the average quantities of all product categories.

**How to get the product category?**

Split the **productCode** by “**\_”** and take the left part of it. For example, **S700\_3962** will result in the **S700** product category.   
**Skills**: ALL, String Functions

SELECT productcode, productName, quantityInStock FROM cr\_products where quantityInStock > all(select avg\_quantityFROM

(SELECT SUBSTR(productcode, 1, locate("\_", productcode)-1) AS Product\_Categories, avg(quantityInStock) AS avg\_quantity FROM cr\_products

GROUP BY Product\_Categories) c )