**Analytical SQL Lab**

**Lab instructions:**

Login to <https://livesql.oracle.com/apex/f?p=590:1000> and run the following script to create the table and insert the data that you will be working on during the first part of this lab.

**CREATE** **TABLE** cats (

**NAME** varchar2(40),

breed varchar2(40),

weight NUMBER,

color varchar2(40),

age NUMBER

)**;**

**INSERT** **INTO** cats **VALUES**('Ashes', 'Persian', 4.5, 'Black', 5)**;**

**INSERT** **INTO** cats **VALUES**('Molly', 'Persian', 4.2, 'Black', 1)**;**

**INSERT** **INTO** cats **VALUES**('Felix', 'Persian', 5.0, 'Tortoiseshell', 2)**;**

**INSERT** **INTO** cats **VALUES**('Smudge', 'British Shorthair', 4.9, 'Black', 4)**;**

**INSERT** **INTO** cats **VALUES**('Tigger', 'British Shorthair', 3.8, 'Tortoiseshell', 2)**;**

**INSERT** **INTO** cats **VALUES**('Alfie', 'Siamese', 5.5, 'Brown', 5 )**;**

**INSERT** **INTO** cats **VALUES**('Oscar', 'Siamese', 6.1, 'Black', 1)**;**

**INSERT** **INTO** cats **VALUES**('Millie', 'Maine Coon', 5.4, 'Tortoiseshell', 5)**;**

**INSERT** **INTO** cats **VALUES**('Misty', 'Maine Coon', 5.7, 'Brown', 2)**;**

**INSERT** **INTO** cats **VALUES**('Puss', 'Maine Coon', 5.1, 'Tortoiseshell', 2)**;**

**INSERT** **INTO** cats **VALUES**('Smokey', 'Maine Coon', 6.1, 'Brown', 4)**;**

**INSERT** **INTO** cats **VALUES**('Charlie', 'British Shorthair', 4.8, 'Black', 4)**;**

1. Cats are vain. Each cat would like to pretend it has the lowest weight for its color.

Print cat name, color and the minimum weight of cats with that color.

Return: name, color, lowest\_weight\_by\_color  
Order by: color, name

SELECT NAME, WEIGHT, COLOR ,

MIN(WEIGHT) OVER(PARTITION BY COLOR ORDER BY WEIGHT)

FROM CATS

ORDER BY COLOR,NAME;

--another answer

SELECT NAME, WEIGHT, COLOR ,

first\_value(WEIGHT) OVER(PARTITION BY COLOR ORDER BY WEIGHT)

FROM CATS

ORDER BY COLOR,NAME;

1. Cats are fickle. Each cat would like to lose weight according to their breed. Each cat would like to lose weight to be the equivalent weight of the cat in the same breed weighing just less than it.

Print a list of cats, their breeds, weights and the weight difference between them and the nearest lighter cat of the same breed.

Return: name, breed, weight, weight\_to\_lose  
Order by: weight

SELECT NAME, BREED, WEIGHT,

COALESCE( WEIGHT - LAG(WEIGHT ,1) OVER(PARTITION BY BREED ORDER BY WEIGHT), 0 )AS WEIGHT\_TO\_LOSE

FROM CATS

ORDER BY WEIGHT;

--another answer

SELECT name, breed, weight,

COALESCE( cast( WEIGHT - LAG(WEIGHT ,1) OVER(PARTITION BY BREED ORDER BY WEIGHT) as varchar(25)), 'no need to lose weight' )AS WEIGHT\_TO\_LOSE

FROM CATS

ORDER BY WEIGHT;

1. Each cat would like to see the next heaviest cat's weight when grouped by breed. If there is no heavier cat print 'fattest cat'.

Print a list of cats, their weights and either the next heaviest cat's weight or 'fattest cat'.

Return: name, weight, breed, next\_heaviest  
Order by: weight

SELECT NAME, WEIGHT, BREED,

COALESCE(TO\_CHAR (LEAD(WEIGHT, 1) OVER (PARTITION BY BREED ORDER BY WEIGHT )), 'FATTEST\_CAT') AS NXT\_HAVIEST\_CAT

FROM CATS

order by WEIGHT DESC, BREED DESC

1. The cats have decided the correct weight is the same as the 4th lightest cat. All cats shall have this weight. Except in a fit of jealous rage they decide to set the weight of the lightest three to 99.9.

Print a list of cats, their weights and their imagined weight.

Return: name, weight, imagined\_weight  
Order by: weight

SELECT NAME, BREED, WEIGHT ,

COALESCE ( NTH\_VALUE(WEIGHT, 4) FROM FIRST OVER(ORDER BY WEIGHT), 99.9)AS IMAGINED\_WEIGHT

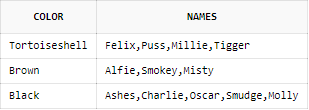
FROM CATS

ORDER BY WEIGHT;

1. We would like to group our cats by color.

Return 3 rows, each row containing a color and a list of cat names.

Return: color, names Order by: color DESC



SELECT DISTINCT COLOR,

LISTAGG(NAME, ',') WITHIN GROUP(ORDER BY NAME) OVER(PARTITION BY COLOR) AS NAMES

FROM CATS

ORDER BY COLOR;

1. Consider cats with the same breed and color to be duplicate rows.

Return one row for cats from the same breed and having the same color.

SELECT NAME, WEIGHT, COLOR, BREED

FROM (SELECT NAME, WEIGHT, COLOR, BREED,

ROW\_NUMBER () OVER (PARTITION BY BREED, COLOR ORDER BY BREED) AS ROW\_NUM

FROM CATS)

WHERE ROW\_NUM = 1;

1. Count the number of unique breeds in two different ways.

SELECT NAME, BREED, WEIGHT, COLOR, AGE,

COUNT(DISTINCT BREED) OVER () AS UNIQUE\_COUNT

FROM CATS

ORDER BY NAME, BREED;

--ANOTHER ANSWER

select NAME, BREED, WEIGHT, COLOR, AGE , Max(rk) over ()  
from(  
SELECT NAME, BREED, WEIGHT, COLOR, AGE,  
  dense\_rank() over(order by BREED) as rk  
    FROM CATS);

Return new column with the count.



Create another detailed table that holds the dates when the weight of the cats was recorded.

**CREATE** **TABLE** cats\_details (

**NAME** varchar2(40),

breed varchar2(40),

weight NUMBER,

color varchar2(40),

age NUMBER ,

weight\_recording\_dt DATE

)**;**

**INSERT** **INTO** cats\_details **VALUES**('Ashes', 'Persian', 4.5, 'Black', 5, To\_Date('2020-03-01', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Molly', 'Persian', 4.2, 'Black', 1, To\_Date('2020-03-02', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Felix', 'Persian', 5.0, 'Tortoiseshell', 2, To\_Date('2020-03-03', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Smudge', 'British Shorthair', 4.9, 'Black', 4, To\_Date('2020-03-04', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Tigger', 'British Shorthair', 3.8, 'Tortoiseshell', 2, To\_Date('2020-03-05', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Alfie', 'Siamese', 5.5, 'Brown', 5, To\_Date('2020-03-06', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Ashes', 'Persian', 5.1, 'Black', 5, To\_Date('2020-03-03', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Molly', 'Persian', 4.4, 'Black', 1, To\_Date('2020-03-04', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Felix', 'Persian', 5.2, 'Tortoiseshell', 2, To\_Date('2020-03-06', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Oscar', 'Siamese', 6.1, 'Black', 1, To\_Date('2020-03-01', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Millie', 'Maine Coon', 5.4, 'Tortoiseshell', 5, To\_Date('2020-03-02', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Misty', 'Maine Coon', 5.7, 'Brown', 2, To\_Date('2020-03-03', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Puss', 'Maine Coon', 5.1, 'Tortoiseshell', 2, To\_Date('2020-03-04', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Smokey', 'Maine Coon', 6.1, 'Brown', 4, To\_Date('2020-03-05', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Charlie', 'British Shorthair', 4.8, 'Black', 4, To\_Date('2020-03-06', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Ashes', 'Persian', 5.0, 'Black', 5, To\_Date('2020-03-02', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Molly', 'Persian', 4.0, 'Black', 1, To\_Date('2020-03-01', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Felix', 'Persian', 5.0, 'Tortoiseshell', 2, To\_Date('2020-03-10', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Smudge', 'British Shorthair', 3.1, 'Black', 4, To\_Date('2020-03-01', 'YYYY-MM-DD'))**;**

**INSERT** **INTO** cats\_details **VALUES**('Tigger', 'British Shorthair', 2.8, 'Tortoiseshell', 2, To\_Date('2020-03-15', 'YYYY-MM-DD'))**;**

1. Transform the above table from transactional to historical table.

SELECT NAME,

BREED, WEIGHT,COLOR,AGE,FIRST\_DATE,LAST\_DATE,

ABS (LAST\_WEIGHT - FIRST\_WEIGHT) AS WEIGHT\_DIFF

FROM (SELECT NAME,

BREED,

WEIGHT,

COLOR,

AGE,

FIRST\_VALUE (WEIGHT\_RECORDING\_DT)

OVER (PARTITION BY name ORDER BY weight\_recording\_dt)

AS FIRST\_DATE,

LAST\_VALUE (

weight\_recording\_dt)

OVER (

PARTITION BY name

ORDER BY weight\_recording\_dt

ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)

AS LAST\_DATE,

FIRST\_VALUE (weight)

OVER (PARTITION BY name ORDER BY weight\_recording\_dt)

AS FIRST\_WEIGHT,

LAST\_VALUE (

weight)

OVER (

PARTITION BY name

ORDER BY weight\_recording\_dt

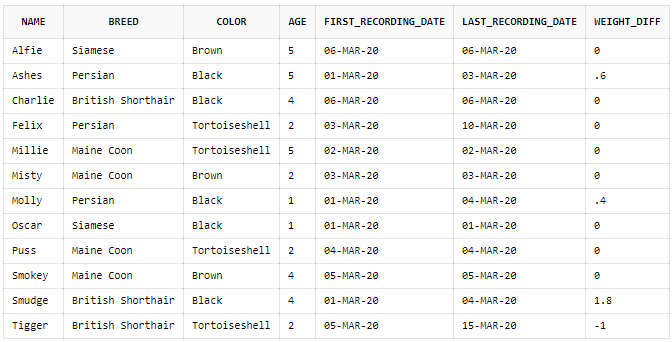
ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)

AS LAST\_WEIGHT,

ROW\_NUMBER () OVER (PARTITION BY name ORDER BY name) ROW\_NUM

FROM Cats\_details)

WHERE ROW\_NUM = 1;



1. Healthy cats keep track of their weight daily.

Return the names of the healthy cats.

SELECT DISTINCT NAME , HEALTHY\_COUNT

FROM (

SELECT NAME,

COUNT(\*) OVER (PARTITION BY NAME ORDER BY weight\_recording\_dt RANGE BETWEEN CURRENT ROW AND INTERVAL '1' DAY FOLLOWING) HEALTHY\_COUNT

FROM Cats\_details)

Where HEALTHY\_COUNT>1;

SELECT DISTINCT NAME,

COALESCE(NEXT\_DATE-WEIGHT\_RECORDING\_DT, 0) AS DAYS\_DIFF FROM

( SELECT NAME, WEIGHT\_RECORDING\_DT, LEAD(WEIGHT\_RECORDING\_DT,1) OVER (PARTITION BY NAME ORDER BY WEIGHT\_RECORDING\_DT) AS NEXT\_DATE FROM CATS\_DETAILS)

WHERE COALESCE(NEXT\_DATE-WEIGHT\_RECORDING\_DT, 0) = 1;