Zoo

Rextester URL

The Zoo database has been uploaded here, for you to practise your SQL statements:

https://rextester.com/TDOIT91316

Important

Tables names in SQL have to be typed exactly. When we imported the tables to Rextester, the tableNames all became lowercase.

Original TableNames :

```
 mysql> show tables;
    +---------------+
    | Tables_in_Zoo |
    +---------------+
    | tblCarnivores |
    | tblVetVisits  |
    +---------------+
```

Have chaned to :

```
rextester> show tables;
    +---------------+
    | Tables_in_Zoo |
    +---------------+
    | tblcarnivores |
    | tblvetvisits  |
    +---------------+
```

The document will follow, how Rextester accepts the queries. But note when we put the queries back into Delphi, we have to use the original tablenames. Delphi always follows what the Tables look like Microsoft Access when you open the .mdb file, so use that if you get confused.

Database Information

There are two tables in Zoo.mdb

```
show tables;
+---------------+
| Tables_in_Zoo |
+---------------+
```

```
| tblCarnivores |
| tblVetVisits  |
+---------------+
```

tblcarnivores

```
  describe tblcarnivores;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| EnclosureNo    | varchar(8)  | YES  |     | NULL    |       |
| NumAdults      | int(11)     | YES  |     | NULL    |       |
| NumYoung       | int(11)     | YES  |     | NULL    |       |
| Endangered     | varchar(4)  | YES  |     | NULL    |       |
| FamilyName     | varchar(30) | YES  |     | NULL    |       |
| ScientificName | varchar(60) | YES  |     | NULL    |       |
| GeneralName    | varchar(60) | YES  |     | NULL    |       |
| EnclosureSize  | int(11)     | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+

  select * from tblcarnivores limit 5;
+-------------+-----------+----------+------------+------------+-----------
-------+----------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName |
ScientificName   | GeneralName          | EnclosureSize |
+-------------+-----------+----------+------------+------------+-----------
-------+----------------------+---------------+
| ZA1         |         2 |        1 | VU         | Felidae    | Acinonyx
jubatus | Cheetah              |            50 |
| ZA9         |         3 |        3 | LE         | Felidae    | Felis
silvestris | Wildcat              |            36 |
| ZB6         |         4 |        2 | VU         | Felidae    | Panthera
leo     | Lion                 |           800 |
| ZB5         |         3 |        3 | LE         | Felidae    | Panthera
pardus  | Leopard              |           700 |
| ZC5         |         3 |        2 | LE         | Viverridae | Genetta
maculata | Rusty-spotted genet |            50 |
+-------------+-----------+----------+------------+------------+-----------
-------+----------------------+---------------+
5 rows in set (0.04 sec)
```

tblvetvisits

```
  describe tblvetvisits;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| VisitID          | int(11)     | YES  |     | NULL    |       |
| VisitDate        | datetime    | YES  |     | NULL    |       |
```

```
| EnclosureNo      | varchar(8)  | YES  |     | NULL    |       |
| FollowUp         | char(1)     | NO   |     | NULL    |       |
| Animal_ID        | varchar(20) | YES  |     | NULL    |       |
| Reason For Visit | varchar(60) | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

  select * from tblvetvisits limit 5;
+---------+---------------------+-------------+----------+-----------+-----
-------------+
| VisitID | VisitDate           | EnclosureNo | FollowUp | Animal_ID |
Reason For Visit |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
|       1 | 2012-09-20 00:00:00 | ZA1         | 1        | ZA1_9     | Skin
problem     |
|       2 | 2012-09-20 00:00:00 | ZC2         | 0        | ZC2_3     |
Routine check-up |
|       3 | 2012-09-20 00:00:00 | ZC3         | 1        | ZC3_8     |
Injured eye      |
|       4 | 2012-09-20 00:00:00 | ZD4         | 0        | ZD4_2     |
Routine check-up |
|       5 | 2012-09-20 00:00:00 | ZA5         | 0        | ZA5_2     |
Routine check-up |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
5 rows in set (0.03 sec)
```

1.1

Menu Option A Display all the details of the animals stored in the tblcarnivores table, sorted firstly by the FamilyName field and secondly by the ScientificName field in alphabetical order.

1. First look at the output of what they expect your answer to look like. From this we can see that they want results from tblcarnivores, however note that the **column** strucutre is not the same.

```
  SELECT
EnclosureNo,FamilyName,ScientificName,GeneralName,NumAdults,NumYoung,Enclos
ureSize,Endangered from tblcarnivores;
```

2. Now that we have the correct Column structure, we look at sorting the FamilyName column.

```
  SELECT
EnclosureNo,FamilyName,ScientificName,GeneralName,NumAdults,NumYoung,Enclos
ureSize,Endangered from tblcarnivores ORDER BY FamilyName,ScientificName
ASC;
```

3. And Finally, sorting the ScientificName column as well.

```
  SELECT
EnclosureNo,FamilyName,ScientificName,GeneralName,NumAdults,NumYoung,Enclos
ureSize,Endangered from tblcarnivores ORDER BY FamilyName,ScientificName
ASC Limit 5
```

Note That ORDER BY sorts columns by ascending order automatically, but we just included the ASC to be more thorough.

1.2

The user enters the family name of an animal via the keyboard. Display the scientific name, general name, enclosure number and enclosure size for animals belonging to the family name entered by the user, and housed in the ZE area of the zoo. The EnclosureNo field indicates the specific enclosure within an area where the animals are housed, for example in enclosure number ZE5. (ZE represents the area of the zoo and 5 the specific enclosure in that area).

Note The question needs to be read very carefully, to construct the SQL statement. By reading it carefully it says:

- User must enter in FamilyName
- "Household in ZE area" - which we know is the EnclosureNo column by reading the question completely.

1. Again we start by looking at the columns the answers expects, and constructing an apporpirate SQL statement.

```
  SELECT ScientificName, GeneralName, EnclosureNo, EnclosureSize from
tblcarnivores;
+--------------------------+-----------------------+-------------+--------
-------+
| ScientificName           | GeneralName           | EnclosureNo |
EnclosureSize |
+--------------------------+-----------------------+-------------+--------
-------+
| Acinonyx jubatus         | Cheetah               | ZA1         |
50 |
| Felis silvestris         | Wildcat               | ZA9         |
36 |
| Panthera leo             | Lion                  | ZB6         |
800 |
| Panthera pardus          | Leopard               | ZB5         |
700 |
| Genetta maculata         | Rusty-spotted genet   | ZC5         |
50 |
```

2. The Question states the User must enter in "FAMILY NAME" and HouseHold Must be "in the ZE area of the ZOO". We start with the "ZE area part first, by filtering our results using a LIKE statement. In

the EnclosureNO Column we can see that the "ZE" has values after it like ZE3, and ZE4. Therefore, we need to use a ***WILDCARD*** to cater for these values. ***% Represents zero or more characters***

```
  SELECT ScientificName, GeneralName, EnclosureNo,EnclosureSize FROM
tblcarnivores WHERE EnclosureNo LIKE "ZE%";
+--------------------+---------------------+-------------+---------------+
| ScientificName     | GeneralName         | EnclosureNo | EnclosureSize |
+--------------------+---------------------+-------------+---------------+
| Crocuta crocuta    | Spotted hyena       | ZE3         |            66 |
| Hyaena brunnea     | Brown hyena         | ZE4         |            66 |
| Vulpes chama       | Cape fox            | ZE6         |            46 |
| Canis mesomelas    | Black-backed jackal | ZE8         |            52 |
| Proteles cristatus | Aardwolf            | ZE5         |            66 |
```

3. Our final part of the question is to include code, so the our results are filtered by FamilyName based on User Input. First lets look at some sample data from the tblcarnivores to get an idea what the "FamilyName" column looks like.

```
  SELECT * from tblcarnivores limit 5;
+-------------+-----------+----------+------------+------------+-----------
-------+---------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName |
ScientificName   | GeneralName         | EnclosureSize |
+-------------+-----------+----------+------------+------------+-----------
-------+---------------------+---------------+
| ZA1         |         2 |        1 | VU         | Felidae    | Acinonyx
jubatus | Cheetah             |            50 |
| ZA9         |         3 |        3 | LE         | Felidae    | Felis
silvestris | Wildcat             |            36 |
| ZB6         |         4 |        2 | VU         | Felidae    | Panthera
leo     | Lion                |           800 |
```

We can include a test value in our SQL statement to make sure its the right statement, by looking at the ***SAMPLE ANSWER*** listed in the question paper - where they used a value of "Canidae".

```
  SELECT ScientificName, GeneralName, EnclosureNo,EnclosureSize FROM
tblcarnivores WHERE EnclosureNo LIKE "ZE%" AND FamilyName = "Canidae";
+------------------+---------------------+-------------+---------------+
| ScientificName   | GeneralName         | EnclosureNo | EnclosureSize |
+------------------+---------------------+-------------+---------------+
| Vulpes chama     | Cape fox            | ZE6         |            46 |
| Canis mesomelas  | Black-backed jackal | ZE8         |            52 |
| Otocyon megalotis | Bat-eared fox      | ZE9         |            52 |
| Canis adustus    | Side-striped jackal | ZE7         |            46 |
+------------------+---------------------+-------------+---------------+
4 rows in set (0.00 sec)
```

We now update our SQL statement to include userinput via an inputbox: ***Note** Now its important here to look at how this "" are enclosed inside of '' If you look at the output above, the string value "Canidae" is inclosed inside of "". That is SQL. However! Delphi expects the SQL statemetn to enclosed inside of '' so we get:

```
procedure TfrmRec.mnuOptionBClick(Sender: TObject);
var
   sX : String;
begin
  sX := INPUTBOX('Question 1', 'Question 1', 'Please Enter in a Family
Name');
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT ScientificName, GeneralName,'
  +'EnclosureNo,EnclosureSize FROM tblcarnivores WHERE '+
  'EnclosureNo LIKE "ZE%" AND FamilyName = "'+ sX + '"';
  qryRec.Open;
end;
```

Now we used + to break the question into smaller parts, but it can be also written continously on one line.

```
procedure TfrmRec.mnuOptionBClick(Sender: TObject);
var
   sX : String;
begin
  sX := INPUTBOX('Question 1', 'Question 1', 'Please Enter in a Family
Name');
  qryRec.Close;
  qryRec.SQL.Text :='SELECT ScientificName, GeneralName,
EnclosureNo,EnclosureSize FROM tblcarnivores WHERE EnclosureNo LIKE "ZE%"
AND FamilyName = "'+sX+'"';
  qryRec.Open;
end;
```

References

AND Statement : https://www.w3schools.com/sql/sql_and_or.asp

LIKE Statement : https://www.w3schools.com/sql/sql_like.asp

Wildcard : https://www.w3schools.com/sql/sql_wildcards.asp

1.3

Display the categories of endangered species and the total number of animal species in each category housed at the zoo. Use a calculated field with the heading CountAnimals for the calculation.

1. First we identify where is the Endangered column present. For that we need to look at our tables.

```
  SELECT * from tblcarnivores limit 5;
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName |
ScientificName   | GeneralName         | EnclosureSize |
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
| ZA1         |         2 |        1 | VU         | Felidae    | Acinonyx
jubatus | Cheetah             |            50 |
| ZA9         |         3 |        3 | LE         | Felidae    | Felis
silvestris | Wildcat             |            36 |
| ZB6         |         4 |        2 | VU         | Felidae    | Panthera
leo     | Lion                |           800 |
| ZB5         |         3 |        3 | LE         | Felidae    | Panthera
pardus  | Leopard             |           700 |
| ZC5         |         3 |        2 | LE         | Viverridae | Genetta
maculata | Rusty-spotted genet |            50 |
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
```

2. We can see that Endangered is a Column in tblcarnivores. Also, based on the sample answer, we can that the question is counting the number of "Endangered" animals. There we will use to additional SQL statements in our query:
    - AS to rename a column
    - COUNT to count the number of Endangered Animals

SELECT Endangered,Count(*) from tblcarnivores; ERROR 1140 (42000): In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'Zoo.tblcarnivores.Endangered'; this is incompatible with sql_mode=only_full_group_by

3. Unfortunately as we can see above, using the Count(*) requires that we use it conjuntion with Group BY.

When the Count statemet is used by itself, then there is no problem.

```
  SELECT Count(Endangered) from tblcarnivores;
+-------------------+
| Count(Endangered) |
+-------------------+
|                42 |
+-------------------+
```

However, when we use it with another Column, in this case Endangered (since the output requires us to print two columns!) we have to use Group By. Therefore we ammend our statement taking this into consideration.

```
  SELECT Endangered,Count(Endangered) from tblcarnivores Group By
Endangered;
+------------+------------------+
| Endangered | Count(Endangered) |
+------------+------------------+
| EN         |                1 |
| LE         |               37 |
| NE         |                1 |
| VU         |                3 |
```

The above statement could have also been written as :

```
  SELECT Endangered,Count(*) from tblcarnivores Group By Endangered;
+------------+----------+
| Endangered | Count(*) |
+------------+----------+
| EN         |        1 |
| LE         |       37 |
| NE         |        1 |
| VU         |        3 |
+------------+----------+
4 rows in set (0.00 sec)
```

4. We have one final ammendment to make to our question so that it looks like the output the question wants, therefore we will rename our "Count Column"

```
  SELECT Endangered,Count(*) AS CountAnimals from tblcarnivores Group By
Endangered;
+------------+--------------+
| Endangered | CountAnimals |
+------------+--------------+
| EN         |            1 |
| LE         |           37 |
| NE         |            1 |
| VU         |            3 |
+------------+--------------+
4 rows in set (0.02 sec)
```

References

AS Statement : https://www.w3schools.com/sql/sql_ref_as.asp

Count Statement : https://www.w3schools.com/sql/sql_count_avg_sum.asp

Group By Statement : https://www.w3schools.com/sql/sql_groupby.asp

1.4

Display the enclosure number and a calculated field showing the space available for each animal of the different mongoose species indicated in the GeneralName field. Calculate the space available per animal by creating a formula which divides the enclosure size by the total number of animals housed in the enclosure. The calculated values should be displayed with a maximum of two decimal digits. Display the calculated field with the heading SpacePerAnimal. HINT: Use the NumAdults, NumYoung and EnclosureSize fields as part of the formula.

1. The first thing we do is always look at our tables:

```
  SELECT * from tblcarnivores limit 5;
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName |
ScientificName   | GeneralName         | EnclosureSize |
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
| ZA1         |         2 |        1 | VU         | Felidae    | Acinonyx
jubatus | Cheetah             |            50 |
| ZA9         |         3 |        3 | LE         | Felidae    | Felis
silvestris | Wildcat             |            36 |
| ZB6         |         4 |        2 | VU         | Felidae    | Panthera
leo     | Lion                |           800 |
| ZB5         |         3 |        3 | LE         | Felidae    | Panthera
pardus  | Leopard             |           700 |
| ZC5         |         3 |        2 | LE         | Viverridae | Genetta
maculata | Rusty-spotted genet |            50 |
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
5 rows in set (0.00 sec)
```

The first sentence the questions is divided into two parts:

```
- Display the enclosure number
- and a calculated field showing the space available for each animal of the
different mongoose species indicated in the GeneralName field
```

```
  SELECT EnclosureNo,SpacePerAnimal from tblcarnivores;
ERROR 1054 (42S22): Unknown column 'SpacePerAnimal' in 'field list'
```

Unfortunately, SpacePerAnimal Column does not exist as yet, we will have to create it using our "special formulae". But for now we just add one more piece of data from the first sentence :

```
  SELECT EnclosureNo from tblcarnivores WHERE GeneralName="Mongoose";
Empty set (0.17 sec)
```

This is not the exptected result, did we spell Mongoose correctly, what is the issue why is there no results ? For this we need to look at the complete table:

```
  select * from tblcarnivores;
```

Based on seeing the whole table, we can see values like "Egyptian mongoose " and "Banded mongoose" under General Name. Therefore we will use wildcards!

```
  SELECT EnclosureNo from tblcarnivores WHERE GeneralName LIKE "%Mongoose";
+-------------+
| EnclosureNo |
+-------------+
| ZD5         |
| ZD6         |
| ZD10        |
| ZD9         |
| ZD4         |
```

Now we are winning! **Note** Even though we spelt "Mongoose" with an uppercase "M", sql matched it again "Egyptian mongoose " and "Banded mongoose"

2. Now we look at the second part of our question :

Display the enclosure number and a calculated field showing the space available for each animal of the different mongoose species indicated in the GeneralName field. Calculate the space available per animal by creating a formula which divides the enclosure size by the total number of animals housed in the enclosure. The calculated values should be displayed with a maximum of two decimal digits. Display the calculated field with the heading SpacePerAnimal. HINT: Use the NumAdults, NumYoung and EnclosureSize fields as part of the formula.

Lets look at our table data again :

```
  SELECT * from tblcarnivores WHERE GeneralName LIKE "%Mongoose";
+-------------+-----------+----------+------------+-------------+----------
-------------+-----------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName  |
ScientificName         | GeneralName           | EnclosureSize |
+-------------+-----------+----------+------------+-------------+----------
-------------+-----------------------+---------------+
| ZD5         |         4 |        4 | LE         | Herpestidae | Helogale
parvula        | Common dwarf mongoose |            26 |
| ZD6         |         4 |        0 | LE         | Herpestidae | Herpestes
ichneumon     | Egyptian mongoose     |            44 |
| ZD10        |         2 |        3 | LE         | Herpestidae | Mungos
mungo           | Banded mongoose       |            40 |
```

```
| ZD9         |         2 |        1 | LE          | Herpestidae |
Rhynchogale melleri    | Meller's mongoose     |            48 |
| ZD4         |         3 |        1 | LE          | Herpestidae | Galerella
sanguinea    | Slender mongoose      |            42 |
| ZD7         |         4 |        4 | LE          | Herpestidae | Ichneumia
albicauda    | White-tailed mongoose |            44 |
| ZD8         |         5 |        5 | LE          | Herpestidae |
Paracynictis selousi   | Selous' mongoose      |            44 |
| ZD1         |         2 |        3 | LE          | Herpestidae | Atilax
paludinosus     | Marsh mongoose        |            34 |
| ZD2         |         2 |        6 | LE          | Herpestidae | Cynictis
penicillata   | Yellow mongoose       |            44 |
| ZD3         |         6 |        5 | LE          | Herpestidae | Galerella
pulverulenta | Cape grey mongoose    |            34 |
+-------------+-----------+----------+-------------+-------------+----------
-------------+----------------------+---------------+|
```

They are telling us to Use NumAdults,NumYoung and EnclosureSize in the Calculation. Based on Maths, (NumAdults+NumYoung)/EnclosureSize seems like a reasonable calculation. Lets look at the **SAMPLE ANSWER** in the question paper and see how our calucation compares against what is listed.

For the first calculation (ie : ZD1) we get (2+3)/34 = 0.147058824 which when compared does not look correct ! Lets swap the calculation around and say (EnclosureSize)/(NumAdults+NumYoung) = Now 34/(2+3) = 6.8 Yes that is correct!

Lets Modify Our query to include our new Discovery:

```
  SELECT EnclosureNo,EnclosureSize/(NumYoung+NumAdults) as SpacePerAnimal
from tblcarnivores WHERE GeneralName LIKE "%Mongoose";
+-------------+----------------+
| EnclosureNo | SpacePerAnimal |
+-------------+----------------+
| ZD5         |         3.2500 |
| ZD6         |        11.0000 |
| ZD10        |         8.0000 |
| ZD9         |        16.0000 |
| ZD4         |        10.5000 |
| ZD7         |         5.5000 |
| ZD8         |         4.4000 |
| ZD1         |         6.8000 |
| ZD2         |         5.5000 |
| ZD3         |         3.0909 |
+-------------+----------------+
```

3. Unfortunaly, even though we know that we got the right answer we have to make it look **EXACTLY** as the output listed. Always remember that.
    - Therefore to narrow down our decimal we will use the FORMAT sql keyword
    - ORDER BY to sort the EnclosureNo

First we FORMAT to two decimal points:

```
  SELECT EnclosureNo,Format(EnclosureSize/(NumYoung+NumAdults),".00") as
SpacePerAnimal  from tblcarnivores WHERE GeneralName LIKE "%Mongoose";
+-------------+----------------+
| EnclosureNo | SpacePerAnimal |
+-------------+----------------+
| ZD5         | 3.25           |
| ZD6         | 11.00          |
| ZD10        | 8.00           |
| ZD9         | 16.00          |
| ZD4         | 10.50          |
| ZD7         | 5.50           |
| ZD8         | 4.40           |
| ZD1         | 6.80           |
| ZD2         | 5.50           |
| ZD3         | 3.09           |
+-------------+-----------
```

Then we ORDER BY to sort :

```
  SELECT EnclosureNo,Format(EnclosureSize/(NumYoung+NumAdults),".00") as
SpacePerAnimal  from tblcarnivores WHERE GeneralName LIKE "%Mongoose" ORDER
BY EnclosureNo;
+-------------+----------------+
| EnclosureNo | SpacePerAnimal |
+-------------+----------------+
| ZD1         | 6.80           |
| ZD10        | 8.00           |
| ZD2         | 5.50           |
| ZD3         | 3.09           |
| ZD4         | 10.50          |
| ZD5         | 3.25           |
| ZD6         | 11.00          |
| ZD7         | 5.50           |
| ZD8         | 4.40           |
| ZD9         | 16.00          |
+-------------+----------------+
```

References

FORMAT Statement : https://www.w3schools.com/sql/func_sqlserver_format.asp

1.4

Increase the number of young animals in the ZF1 enclosure by 3. If the updating of the record was done successfully, an output message, stating that the record was successfully processed, will be displayed.

When making changes to our tables in SQL, we will always either use INSERT/UPDATE or delete. In this case, its a Update.

Before we update, lets take a look at our table again :

```
  SELECT * FROM tblcarnivores limit 5;
+-------------+-----------+----------+------------+------------+-----------
-------+---------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName |
ScientificName   | GeneralName         | EnclosureSize |
+-------------+-----------+----------+------------+------------+-----------
-------+---------------------+---------------+
| ZA1         |         2 |        1 | VU         | Felidae    | Acinonyx
jubatus | Cheetah             |            50 |
| ZA9         |         3 |        3 | LE         | Felidae    | Felis
silvestris | Wildcat             |            36 |
| ZB6         |         4 |        2 | VU         | Felidae    | Panthera
leo     | Lion                |           800 |
| ZB5         |         3 |        3 | LE         | Felidae    | Panthera
pardus  | Leopard             |           700 |
| ZC5         |         3 |        2 | LE         | Viverridae | Genetta
maculata | Rusty-spotted genet |            50 |
+-------------+-----------+----------+------------+------------+-----------
-------+---------------------+---------------+
```

The question states that we specifically UPDATE ZF1. Therefore lets print our table again with a LIKE statement :

```
  SELECT EnclosureNo, NumAdults, NumYoung from tblcarnivores WHERE
EnclosureNo LIKE "ZF1";
+-------------+-----------+----------+
| EnclosureNo | NumAdults | NumYoung |
+-------------+-----------+----------+
| ZF1         |         2 |        1 |
+-------------+-----------+----------+
1 row in set (0.01 sec)
```

Before we update, how do we increase a columns value ? If we ran the same query above using a plus symbol :

```
  SELECT EnclosureNo, NumAdults, NumYoung+3 from tblcarnivores WHERE
EnclosureNo LIKE "ZF1";
+-------------+-----------+------------+
| EnclosureNo | NumAdults | NumYoung+3 |
+-------------+-----------+------------+
| ZF1         |         2 |          4 |
```

```
+-------------+-----------+------------+
1 row in set (0.00 sec)
```

From this we can see that NumYoung+3 works, thefore :

```
  UPDATE tblcarnivores SET NumYoung = NumYoung + 3 WHERE EnclosureNo LIKE
"ZF1";
Query OK, 1 row affected (0.28 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Confirm your Output by running :

```
  SELECT EnclosureNo, NumAdults, NumYoung+3 from tblcarnivores WHERE
EnclosureNo LIKE "ZF1";
+-------------+-----------+------------+
| EnclosureNo | NumAdults | NumYoung+3 |
+-------------+-----------+------------+
| ZF1         |         2 |          7 |
+-------------+-----------+------------+
1 row in set (0.00 sec)
```

Reference

Update Statement : https://www.w3schools.com/sql/sql_update.asp

1.6

The user is requested to enter the day of the month when the veterinarian (vet) visited the animals, for example 23. Display the enclosure number, the general name, the date of the visit, the IDs of the particular animals visited and the reason why each of the animals was visited on the specific day of the month entered by the user. HINT: Use the DAY() SQL function in the SQL statement. Example of the output of the first five records for 23 September 2012:

1. We break the query down first :
    - The user is requested to enter the day of the month when the veterinarian (vet) visited the animals, for example 23.
    - Display the enclosure number, the general name, the date of the visit, the IDs of the particular animals visited and the reason why each of the animals was visited on the specific day of the month entered by the user.

For now, we will manually enter in the "23" to simluate user input. Lets get our expected Columns first :

```
  SELECT EnclosureNo,GeneralName,VisitDate,Animal_ID,ReasonForVisit from
tblcarnivores;
ERROR 1054 (42S22): Unknown column 'VisitDate' in 'field list'
```

We are receiving an error "Unknown Column". Perhaps these new columns reside in the other table ? Lets check :

```
  SELECT * from tblvetvisits LIMIT 5;
+---------+---------------------+-------------+----------+-----------+-----
-------------+
| VisitID | VisitDate           | EnclosureNo | FollowUp | Animal_ID |
Reason For Visit |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
|       1 | 2012-09-20 00:00:00 | ZA1         | 1        | ZA1_9     | Skin
problem     |
|       2 | 2012-09-20 00:00:00 | ZC2         | 0        | ZC2_3     |
Routine check-up |
|       3 | 2012-09-20 00:00:00 | ZC3         | 1        | ZC3_8     |
Injured eye      |
|       4 | 2012-09-20 00:00:00 | ZD4         | 0        | ZD4_2     |
Routine check-up |
|       5 | 2012-09-20 00:00:00 | ZA5         | 0        | ZA5_2     |
Routine check-up |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
5 rows in set (0.03 sec)
```

From this information we can see that VisitDate, Animal_ID and ReasonForVisit are three of the columns the questions asks for, which reside in tblVetVists. Therefore: - EnclosureNo, GeneralName : tblcarnivores - VisitDate, Animal_ID and ReasonForVisit : tblvetvisits

Now we need to identify what these tables have in **common**. That will the link, that will join the two tables:

```
  SELECT * from tblvetvisits LIMIT 5;
+---------+---------------------+-------------+----------+-----------+-----
-------------+
| VisitID | VisitDate           | EnclosureNo | FollowUp | Animal_ID |
Reason For Visit |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
|       1 | 2012-09-20 00:00:00 | ZA1         | 1        | ZA1_9     | Skin
problem     |
|       2 | 2012-09-20 00:00:00 | ZC2         | 0        | ZC2_3     |
Routine check-up |
|       3 | 2012-09-20 00:00:00 | ZC3         | 1        | ZC3_8     |
Injured eye      |
|       4 | 2012-09-20 00:00:00 | ZD4         | 0        | ZD4_2     |
Routine check-up |
|       5 | 2012-09-20 00:00:00 | ZA5         | 0        | ZA5_2     |
Routine check-up |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
```

```
5 rows in set (0.03 sec)

  SELECT * from tblcarnivores LIMIT 5;
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
| EnclosureNo | NumAdults | NumYoung | Endangered | FamilyName |
ScientificName   | GeneralName         | EnclosureSize |
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
| ZA1         |         2 |        1 | VU         | Felidae    | Acinonyx
jubatus | Cheetah             |            50 |
| ZA9         |         3 |        3 | LE         | Felidae    | Felis
silvestris | Wildcat             |            36 |
| ZB6         |         4 |        2 | VU         | Felidae    | Panthera
leo     | Lion                |           800 |
| ZB5         |         3 |        3 | LE         | Felidae    | Panthera
pardus  | Leopard             |           700 |
| ZC5         |         3 |        2 | LE         | Viverridae | Genetta
maculata | Rusty-spotted genet |            50 |
+-------------+-----------+----------+------------+-----------+-----------
-------+---------------------+---------------+
5 rows in set (0.00 sec)
```

From this information we can see that "EnclosureNo" is common to both tables. We will use this column to link both tables. Lets update our select statement :

```
SELECT tblcarnivores.EnclosureNo, GeneralName, VisitDate,ReasonForVisit,
Animal_ID FROM tblcarnivores,tblvetvisits WHERE
tblcarnivores.EnclosureNo = tblvetvisits.EnclosureNo;
```

Notice : - The two table names after the FROM statement (becuase we selecting columns from two tables) - tblcarnivores.EnclosureNo = tblvetvisits.EnclosureNo; (because this is common column that links both tables)

2. The question is not finished, as we have to accomodate the user entering a "DAY Value". Now if we look at our results, we do see there is Date Under "VisitDate" column.

```
  select * from tblvetvisits limit 5;
+---------+---------------------+-------------+----------+-----------+-----
-------------+
| VisitID | VisitDate           | EnclosureNo | FollowUp | Animal_ID |
Reason For Visit |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
|       1 | 2012-09-20 00:00:00 | ZA1         | 1        | ZA1_9     | Skin
problem     |
|       2 | 2012-09-20 00:00:00 | ZC2         | 0        | ZC2_3     |
```

```
Routine check-up |
|       3 | 2012-09-20 00:00:00 | ZC3         | 1        | ZC3_8     |
Injured eye      |
```

To extract the "Day" from The VisitDate Column we can do use the DAY() SQL function :

```
SELECT VisitID,Day(VisitDate),EnclosureNo from tblvetvisits;

+---------+----------------+-------------+
| VisitID | Day(VisitDate) | EnclosureNo |
+---------+----------------+-------------+
|       1 |             20 | ZA1         |
|       2 |             20 | ZC2         |
|       3 |             20 | ZC3         |
|       4 |             20 | ZD4         |
|       5 |             20 | ZA5         |
```

We no now update our full SQL query, and To accomodate our user input, we can put in a test value of 23.

```
SELECT tblcarnivores.EnclosureNo, GeneralName, VisitDate,ReasonForVisit,
Animal_ID FROM tblcarnivores,tblvetvisits WHERE
tblcarnivores.EnclosureNo = tblvetvisits.EnclosureNo AND Day(VisitDate) =
23;
```

SELECT tblcarnivores.EnclosureNo, GeneralName, VisitDate,ReasonForVisit, Animal_ID FROM tblcarnivores,tblvetvisits WHERE tblcarnivores.EnclosureNo = tblvetvisits.EnclosureNo AND Day(VisitDate) = 23;

3. Now that we have the correct SQL statement we can transfer the statement into DELPHI, to accomodate User Input.

```
procedure TfrmRec.mnuOptionFClick(Sender: TObject);
var
   sX : String;
begin
  sX := INPUTBOX('Question 1', 'Question 1', '23');
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT tblcarnivores.EnclosureNo,'
  +'GeneralName, VisitDate,ReasonForVisit, Animal_ID FROM '
  +'tblcarnivores,tblvetvisits WHERE tblcarnivores.EnclosureNo'
  +'= tblvetvisits.EnclosureNo AND Day(VisitDate) ='+sX;
  qryRec.Open;
end;
```

Note Notice that sX is not enclosed in "" as the previous user input question in this document. That is becuase this sX is an integer, and integeres in SQL are take as is. On a side note to see what your sql statement looks like do the following : - Add in a ShowMessage like below - the showmessage has to come after the qryRec.SQL.Text

```
procedure TfrmRec.mnuOptionFClick(Sender: TObject);
var
   sX,str : String;
begin
  sX := INPUTBOX('Question 1', 'Question 1', '23');
  qryRec.Close;
  qryRec.SQL.Text :=  'SELECT tblcarnivores.EnclosureNo,'
  +'GeneralName, VisitDate,ReasonForVisit, Animal_ID FROM '
  +'tblcarnivores,tblvetvisits WHERE tblcarnivores.EnclosureNo'
  +'= tblvetvisits.EnclosureNo AND Day(VisitDate) ='+sX;
  Showmessage(qryRec.SQL.Text);
  qryRec.Open;
end;
```

References

Day Statement : https://www.w3schools.com/sql/func_sqlserver_day.asp

1.7

On 25 September 2012 the veterinarian examined the animal with the ID ZD5_3 in enclosure ZD5 for an ear infection. He indicated that a follow-up visit would be required. Add this data as a new record into the tblvetvisits table. If the record was added to the table successfully, an output message stating that the record was processed successfully will be displayed.

Hint : Use option F to verify the adding of the record to the database. Use 25 September 2012 as input data.

1. Lets take a look at tblvetvists first :

```
select * from tblvetvisits limit 5;
+---------+---------------------+-------------+----------+-----------+-----
-------------+
| VisitID | VisitDate           | EnclosureNo | FollowUp | Animal_ID |
ReasonForVisit |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
|       1 | 2012-09-20 00:00:00 | ZA1         | 1        | ZA1_9     | Skin
problem     |
|       2 | 2012-09-20 00:00:00 | ZC2         | 0        | ZC2_3     |
Routine check-up |
|       3 | 2012-09-20 00:00:00 | ZC3         | 1        | ZC3_8     |
Injured eye      |
|       4 | 2012-09-20 00:00:00 | ZD4         | 0        | ZD4_2     |
```

```
Routine check-up |
|       5 | 2012-09-20 00:00:00 | ZA5         | 0        | ZA5_2     |
Routine check-up |
+---------+---------------------+-------------+----------+-----------+-----
-------------+
```

The question states that we need to "add a new record"

Before inserting a new record, we have to have values for all the fields.

- VisitID: If we look at tblvetvisits in Microsoft Access, and Click On VisitID, and then FIELDS at the Top you see the DataType is set to autoNumber However, this information should be in the Question aswell, or at the back of exam paper. Therefore, when we insert we do not have to put in a value for this column because it is the primary key.
- VisitDate : A date is inserted into SQL by enclosing the date inside ##. For the 26 September 2012 #26/09/12#
- EnclosureNo : Given in the question as ZD5
- FollowUp : 1 for True
- Animal_ID : ZD5_3
- ReasonForVisit : Ear Infection

Lets take a look how tblvetvists is "describes" rextester first :

```
describe tblvetvisits;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| VisitID          | tinyint(4)  | YES  |     | NULL    |       |
| VisitDate        | varchar(19) | YES  |     | NULL    |       |
| EnclosureNo      | varchar(8)  | YES  |     | NULL    |       |
| FollowUp         | varchar(5)  | YES  |     | NULL    |       |
| Animal_ID        | varchar(20) | YES  |     | NULL    |       |
| Reason For Visit | varchar(60) | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
```

- Note here that VisitID is not a Primary Key, therefore you would have to include a value for it.
- That all fields excep the first are "varchars" or in other words strings with different lengths
- Also note that Delphi through it SQL driver does not support the DESCRIBE statement.

Therefore the following will be accepted by Rextester :

```
INSERT INTO `tblvetvisits`
(VisitID,VisitDate,EnclosureNo,ReasonForVisit,FollowUp,Animal_ID) VALUES
```

```
  (46,'26-September-2012','ZA1','Skin problem','True','ZA1_9');
```

Lets head over to Microsoft Access, and view the MDB file.

```
1. On the left hand side panel, right click on tblvetvisits
2. Then select Design View
```

Notice the data types are different here. We now construct an equivalent statement for Delphi/Access:

```
procedure TfrmRec.mnuOptionGClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text :=  'INSERT INTO tblvetvisits (VisitDate,'
  +'EnclosureNo,ReasonForVisit, FollowUp, Animal_ID) VALUES'
  +'(#2012/10/25#, "ZD5", "Ear infection", True, "ZD5_3")';
  qryRec.ExecSQL;
  MessageDlg('Record Processed Successfully',mtInformation,[mbok],0);
end;
```

Reference

Insert Statement : https://www.w3schools.com/sql/sql_insert.asp