***Bio Informatica SQL Questions By*** [***Affinity Answers***](https://github.com/princevani/affinityanswers)

***Source/DB:*** [***https://docs.rfam.org/en/latest/database.html***](https://docs.rfam.org/en/latest/database.html)

1. How many types of tigers can be found in the taxonomy table of the dataset? What is the "ncbi\_id" of the Sumatran Tiger? (Hint: use the biological name of the tiger)

Sol: The biological name for tiger species is: 'Panthera Tigris'

1.1 Types of tiger species ('Panthera Tigris') found in taxonomy table is >>

SELECT DISTINCT species AS 'Types of Tiger Species'  
FROM taxonomy  
WHERE species LIKE 'panthera tigris%';

The given SQL query will retrieve distinct values of the "species" column from the "taxonomy" table where the species name contains the phrase "Panthera Tigris".

1.2 Total types of tiger species ('Panthera Tigris') found in taxonomy table is >>

SELECT COUNT (DISTINCT species) AS 'Count of Tiger Species'  
FROM taxonomy  
WHERE species LIKE 'panthera tigris%';

The given SQL query will return the count of distinct tiger species names from the "taxonomy" table where the species name starts with "Panthera Tigris".

1.3 NCBI ID of Sumatran tiger ('Panthera Tigris Sumatrae') is >>

SELECT ncbi\_id, species as 'Tiger Species'  
FROM taxonomy  
WHERE species LIKE '%panthera tigris sumatrae%';

The given SQL query return the "ncbi\_id" and "species" columns from the "taxonomy" table for the rows where the species name contains the phrase "panthera tigris sumatrae".

1. Find all the columns that can be used to connect the tables in the given database.

Sol: To find all the columns that can be used to connect all the tables in a database, we can examine the **foreign keys** present in the tables. Foreign keys establish relationships between tables by referencing primary keys or unique keys in other tables. Alternately by using **describe <table\_name>** we can find the columns that can be used to connect tables.

SELECT  
 TABLE\_NAME, COLUMN\_NAME, CONSTRAINT\_NAME,  
 REFERENCED\_TABLE\_NAME, REFERENCED\_COLUMN\_NAME  
FROM  
 INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE  
WHERE  
 REFERENCED\_TABLE\_NAME IS NOT NULL  
 AND TABLE\_SCHEMA = 'Rfam';

This query retrieves information from the **INFORMATION\_SCHEMA.KEY\_COLUMN\_USAGE** table, which contains metadata about the keys and columns used in tables.

The columns returned by the query provide the following information:

* **TABLE\_NAME:** The name of the table containing the foreign key column.
* **COLUMN\_NAME:** The name of the foreign key column.
* **CONSTRAINT\_NAME:** The name of the constraint associated with the foreign key.
* **REFERENCED\_TABLE\_NAME:** The name of the referenced table.
* **REFERENCED\_COLUMN\_NAME:** The name of the column in the referenced table that the foreign key references.

1. Which type of rice has the longest DNA sequence? (hint: use the rfamseq and the taxonomy tables)

Sol: The given SQL queries retrieves information about the rice species with the longest length of associated RFam sequences.

SELECT taxonomy.species as 'Rice Species', MAX(rfamseq.length) AS max\_len  
FROM taxonomy  
JOIN rfamseq ON taxonomy.ncbi\_id = rfamseq.ncbi\_id  
WHERE taxonomy.species LIKE '%oryza sativa%'  
GROUP BY taxonomy.species  
ORDER BY max\_len DESC  
LIMIT 1;

* This specifies the columns to be selected in the result. The taxonomy.species column is selected and given the alias 'Rice Species'. The MAX(rfamseq.length) function calculates the maximum value of the 'length' column from the 'rfamseq' table.

SELECT taxonomy.species as 'Rice Species', MAX(rfamseq.length) AS max\_len

* This defines the tables to be used in the query. It performs an inner join between the 'taxonomy' and 'rfamseq' tables based on the matching 'ncbi\_id' columns.

FROM taxonomy JOIN rfamseq ON taxonomy.ncbi\_id = rfamseq.ncbi\_id

* This is a condition that filters the rows. It ensures that only rows with the 'species' column in the 'taxonomy' table containing the phrase 'oryza sativa' anywhere in the string are considered.

WHERE taxonomy.species LIKE '%oryza sativa%'

* This groups the result by the 'species' column in the 'taxonomy' table.

GROUP BY taxonomy.species

* This sorts the result in descending order based on the 'max\_len' column and limit the result to one row only.

ORDER BY max\_len DESC Limit 1;

1. We want to paginate a list of the family names and their longest DNA sequence lengths (in descending order of length) where only families that have DNA sequence lengths greater than 1,000,000 are included. Give a query that will return the 9th page when there are 15 results per page. (hint: we need the family accession ID, family name and the maximum length in the results)

Sol: The given SQL query retrieves information about families and their associated RFam sequences that have a length greater than 1,000,000. It also applies pagination to retrieve a specific subset of results.

SELECT family.rfam\_acc as 'Family Accession Id', family.rfam\_id as 'Family Name', MAX(rfamseq.length) AS 'max\_len'  
FROM family  
JOIN full\_region ON family.rfam\_acc = full\_region.rfam\_acc  
JOIN rfamseq ON full\_region.rfamseq\_acc = rfamseq.rfamseq\_acc  
WHERE rfamseq.length > 1000000  
GROUP BY family.rfam\_acc, family.rfam\_id  
ORDER BY max\_len DESC  
LIMIT 120, 15; -- offset (9-1)\*15=120, limit 15

* This specifies the columns to be selected in the result. The 'family.rfam\_acc' column is selected and given the alias 'Family Accession Id', the 'family.rfam\_id' column is selected and given the alias 'Family Name', and the MAX(rfamseq.length) function calculates the maximum value of the 'length' column from the 'rfamseq' table.

SELECT family.rfam\_acc as 'Family Accession Id', family.rfam\_id as 'Family Name', MAX(rfamseq.length) AS 'max\_len'

* This defines the tables to be used in the query. It performs joins between the 'family', 'full\_region', and 'rfamseq' tables based on the matching accession IDs.

FROM family  
JOIN full\_region ON family.rfam\_acc = full\_region.rfam\_acc  
JOIN rfamseq ON full\_region.rfamseq\_acc = rfamseq.rfamseq\_acc

* This is a condition that filters the rows. It ensures that only rows with the 'length' column in the 'rfamseq' table greater than 1,000,000 are considered.

WHERE rfamseq.length > 1000000

* This groups the result by the 'rfam\_acc' and 'rfam\_id' columns in the 'family' table.

GROUP BY family.rfam\_acc, family.rfam\_id

* This sorts the result in descending order based on the 'max\_len' column.

ORDER BY max\_len DESC

* This applies pagination to the result. It skips the first 120 rows and retrieves the next 15 rows, effectively providing results starting from the 121st row.

LIMIT 120, 15; -- offset (9-1)\*15=120, limit 15