Course: BINF-6211, Lab_8
Name: Drashti Mehta, SID: 801262877

Lab 8

The final modification that we will do is eliminating multiple values in the attribute **taxonomy** of the table **mirna_species**.

This case is slightly difference from the **authors** column of **literature_references** in that the taxonomy has the hierarchical structure. For example, class *Mammalia* has several subclasses *Primates*, *Rodentia*, etc. In turn, *Primates* are split into *Hominidae*, *Atelidae*, etc.

Part 1

In order to preserve this hierarchical structure, we'll create a new table **taxonomy** with attributes **auto_id**, **name**, **parent_id**, where **parent_id** contain the ID of the taxonomy superclass.

But before that, we need to get all taxonomy values and their parents/superclasses. We can use Excel for that:

1. Export the data from the table **mirna_species** into a CSV file
2. Open that file in Excel and use functions "Text To Columns" and "Remove Duplicated" to get a list of unique taxonomy names. Put them all in a column called "name".
3. Add column "auto_id" and autogenerate integer numbers: 1, 2, 3, …
4. Add column "parent_id" and put there the ID of the superclass (if any) for each taxonomy value.

This work is quite tedious so I've completed it for you. You can download the resulting file from Canvas.

Part 2 (5pts)

Import data from the resulting CSV file into a new table **taxonomy**. SQLite can automatically generate a table structure for you, based on column names in the CSV file. But make sure that the generated table is correct (check data types, primary key, etc.)

Report the SQL query for creating the table **taxonomy**.

Ans:

```
CREATE TABLE taxonomy (
auto_id INTEGER PRIMARY KEY,
name VARCHAR,
parent_id INTEGER
);
```

After that we import the table values from the taxonomy.csv file.

Course: BINF-6211, Lab_8
Name: Drashti Mehta, SID: 801262877

Part 3 (15pts)

Because the relationship between **mirna_species** and **taxonomy** are many-to-many, we'll need a separate table **species_taxonomy** to implement that relationship. Create that table and fill it out with correct IDs.

Try to come up with an INSERT query based on what you learned in Lab 7.

Report SQL queries for creating **species_taxonomy** and filling it out.

Answer:

As these 2 tables are many to many in relationship hence we need to create another table "species_taxonomy" and insert values into it from the 2 tables.

```
CREATE TABLE species_taxonomy (
species_auto_id INTEGER REFERENCES mirna_species (auto_id),
taxonomy_auto_id INTEGER REFERENCES taxonomy (auto_id),
name VARCHAR,
taxonomy VARCHAR
);

INSERT OR IGNORE INTO species_taxonomy (species_auto_id, taxonomy_auto_id,
name, taxonomy)
SELECT mirna_species.auto_id as species_auto_id, taxonomy.auto_id as
taxonomy_auto_id, taxonomy.name, mirna_species.taxonomy
FROM mirna_species JOIN taxonomy
WHERE INSTR(mirna_species.taxonomy, taxonomy.name) > 0;
```

Part 4 (15pts)

Try to write SQL queries to retrieve all subclasses of *Primates*, or all species that belong to *Primates*.

Think about what advantages and disadvantages of storing taxonomy values in a separate table.

In the report, write down whether you prefer to have a multi-valued column **taxonomy** (original miRbase approach) or store taxonomy values in a separate table with the many-to-many relationship to the table **mirna_species**. Justify your choice. List all advantages and disadvantages of each method that you can think about.

Answer:

```
SELECT mirna_species.name as species_name, taxonomy.name as taxonomy_name
FROM mirna_species JOIN taxonomy
WHERE taxonomy.name = "Primates";
```

Choosing table preference:

Course: BINF-6211, Lab_8
Name: Drashti Mehta, SID: 801262877

Advantages of using the taxonomy table is that if we have any particular values from the taxonomy column in the mirna_species table, say to find about the primates, it becomes way easier using the taxonomy table by searching it through where clause which makes it tedious with the mirna_species table (using taxonomy column from that table).

But when we have to fetch connecting values or any additional information like name of the species from parent table, i.e., mirna_species table, it becomes easier to fetch it directly from mirna_species table. If we use taxonomy table here, we will have to join the table with mirna_species table (parent) which makes it longer. From mirna_species table it becomes easier but with taxonomy table it becomes tedious.

In my opinion, using taxonomy table is easier and preferrable which isn't multivalued.