**LAB – 2:**

1. **File “lab2.db” after completing parts 1-3.**

Submission attached.

1. **CREATE TABLE statements for creating the two tables in Part 1 and the table in Part**

**3.**

Code:

> CREATE TABLE Gene (gene\_id INTEGER PRIMARY KEY, name, gene\_type, organism, summary);

> CREATE TABLE GeneSynonym (gene\_synonym\_id INTEGER PRIMARY KEY, gene\_id

REFERENCES Gene (Gene\_id), name);

> CREATE TABLE RefSeq (gene\_id INTEGER PRIMARY KEY, refseq\_id, precursor);

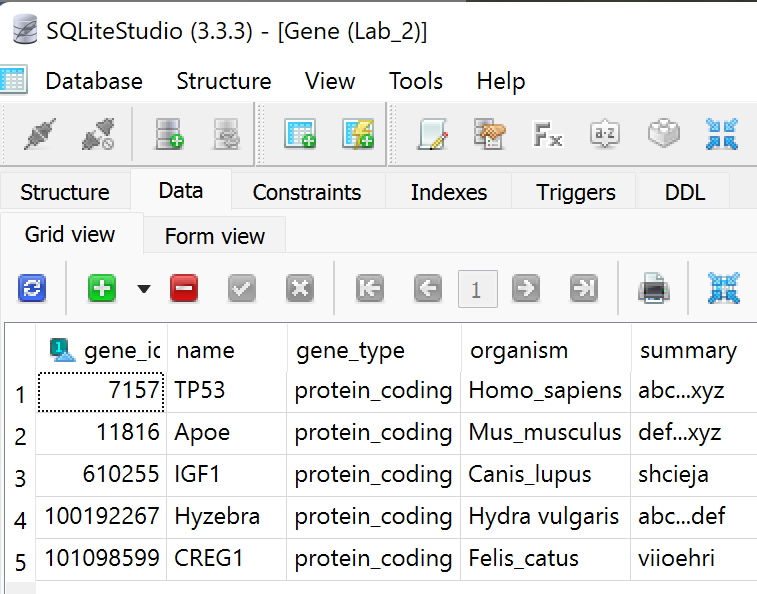


Fig. 1: Gene table

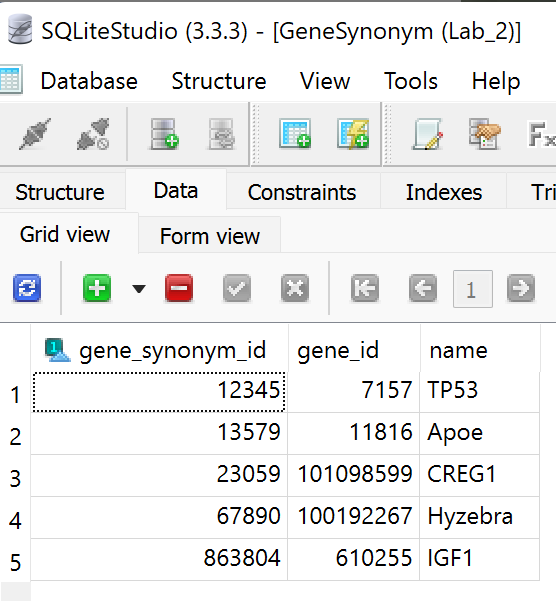


Fig 2: GeneSynonym table

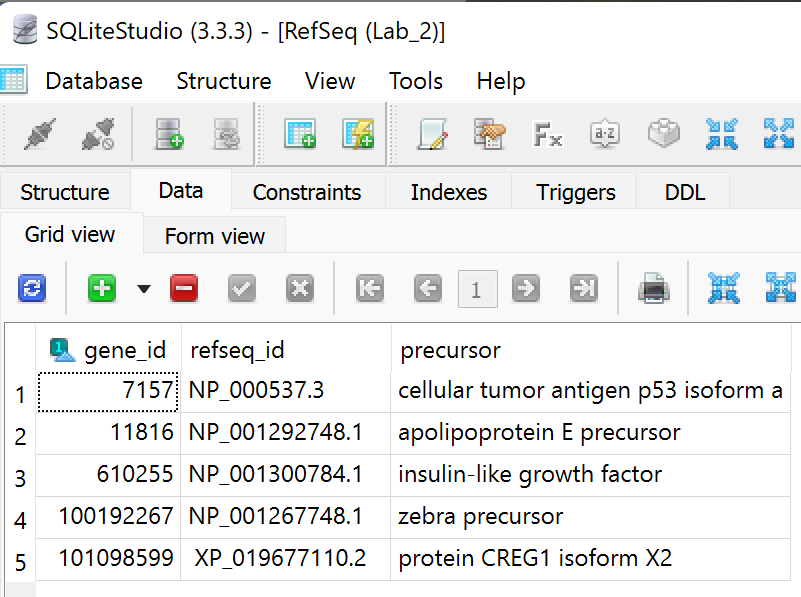


Fig 3: RefSeq table

**Linking tables:**

SELECT

name,

refseq\_id

FROM

Gene

INNER JOIN Refseq

ON Gene.gene\_id = Refseq.gene\_id;

* So, when we join or link tables Gene and RefSeq by the gene\_id, we can see that it is a **1:1 sort of relationship.**

1. **ALTER TABLE statement for changing the Gene table in Part 2.**

When we are using the SQLite command line, we write the codes for all the tasks we perform like Alter or Update… etc. When we are using the SQLiteStudio, we can do it by the features/icons shown on the bar.

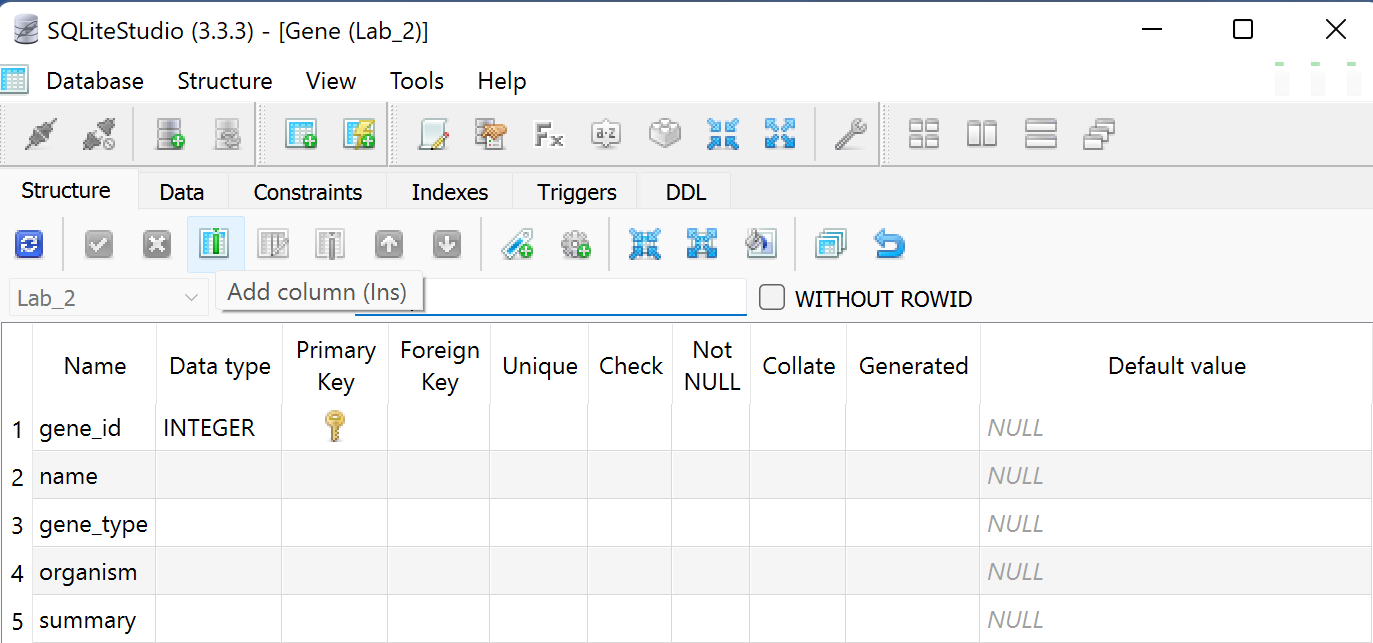


Fig 4: The add column icon (Location)

Code:

> ALTER TABLE Gene

> ADD COLUMN Location;

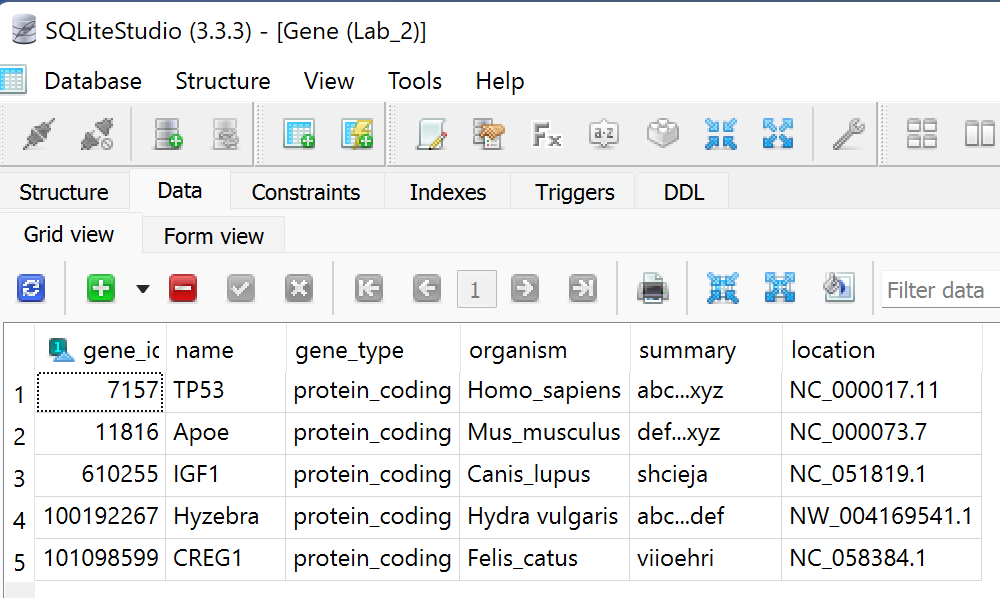


Fig 5: Alter table Gene with new column called Location

1. **A single example of the INSERT table.**

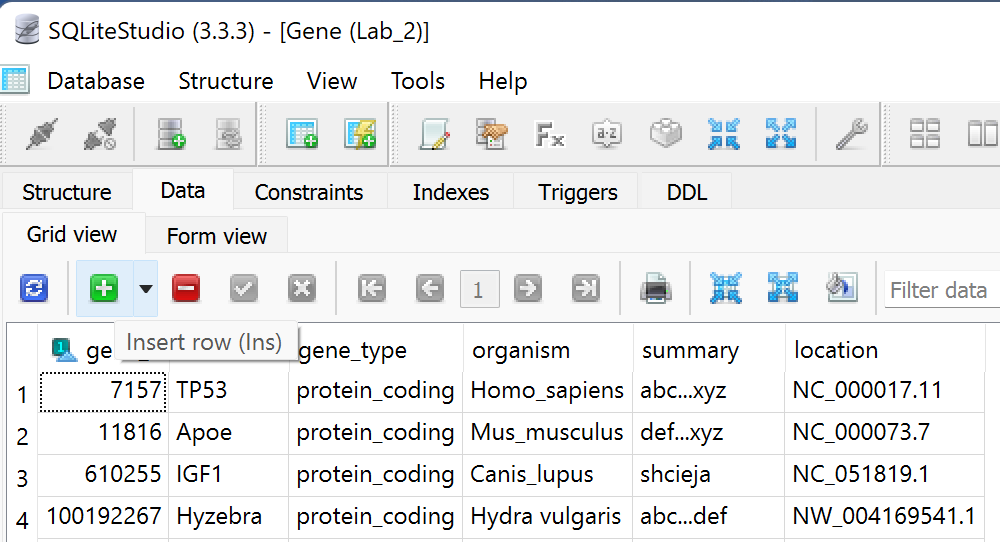
****

Fig 6: Insert row icon

INSERT INTO Gene (Name)

VALUES

("TP53"),

("Apoe"),

(“IGF1”),

("Hyzebra"),

(“CREG1”);

1. **A single example of the UPDATE table.**

As mentioned in the Alter table answer, we can do the similar way here. When we add a column with the help of alter table, and refresh in the data section, we can simply add the values. But on command line we need to code.

Code:

UPDATE Gene

SET Location = NC\_000017.11

WHERE Gene\_id = 7157;