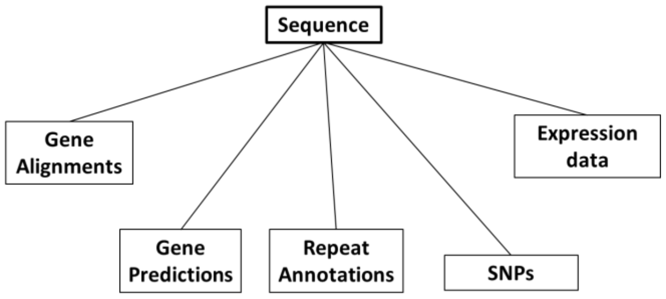
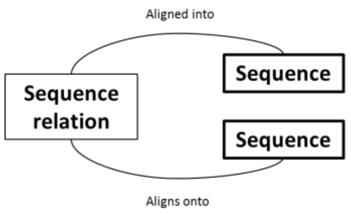
SQL Database example exercise

For a research project looking at the plant species Hieracium, a large amount of sequence of various types (DNA, RNA, predicted proteins) had been accumulated, from a large number of varieties/samples. Further information had been built up in relation to these sequences, e.g. gene predictions, alignments to annotation databases, SNPs, expression data, etc.. A system for storing and querying this data resource was developed, which made use of an SQL backend.

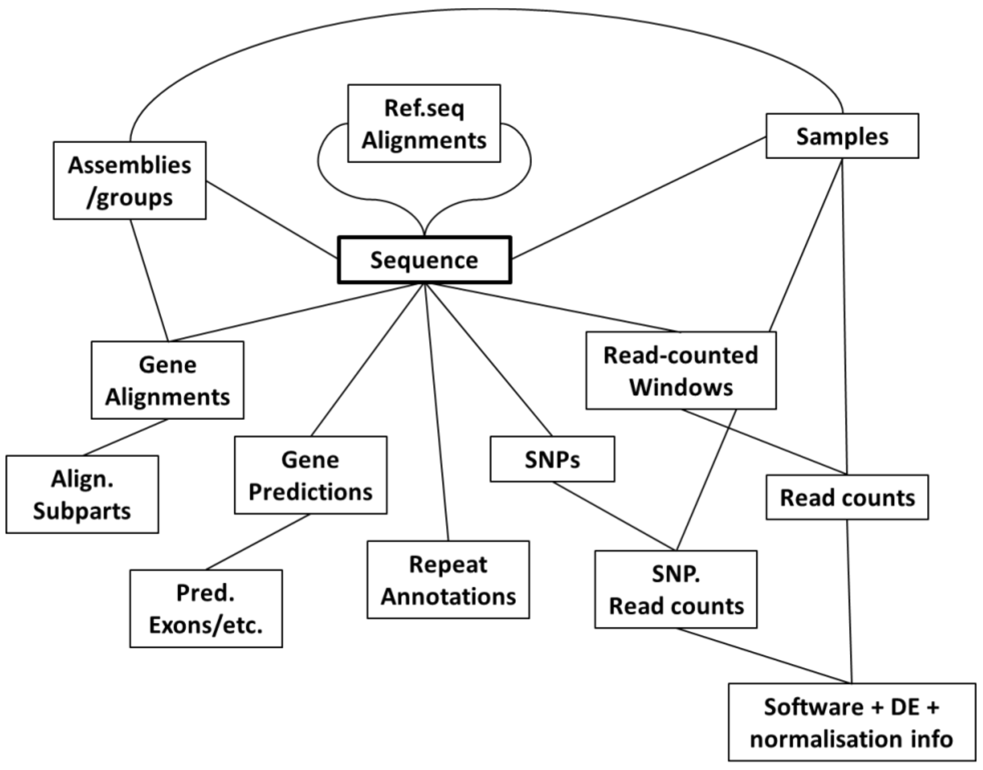
An initial simple representation of the data being stored, where each box is a separate table, may be:



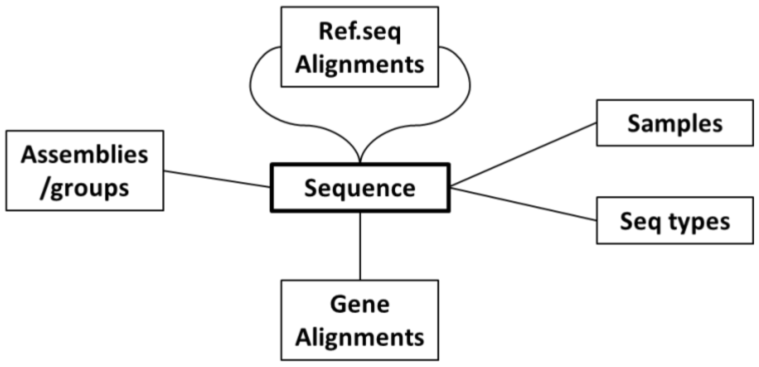
However, an additional important detail we wanted to capture was the relations between our sequence objects themselves. E.g. an RNA sequence may be fully contained within a longer DNA sequence, which is itself a sequence object in the database. Or an annotated protein sequence may have originated from a DNA sequence. To capture such hierarchical relationships between sequence objects, another table is needed that defines these connections.



The final database design for this project ended up looking like this:



However, for this example exercise, we're just going to play with a subset of data from a simplified subset of the database, which just looks like this:



The tables and their columns are as follows:

sequence
id (integer, primary key),
name (string),
length (integer),
belongsGroup (integer, FK seqgroup.id),
isSample (integer, FK sample.id),
isType (integer, FK seqtype.id)

sample
id (integer, primary key),
name (string),
species (string),
description (string)

seqgroup
id (integer, primary key),
name (string),
description (string)

seqtype
id (integer, primary key),
type (string)

seqrelation
id (integer, primary key),
parentSeq (integer, FK sequence.id),
childSeq (integer, FK sequence.id),
strand (boolean),
pStart (integer),
pEnd (integer),
cStart (integer),
cEnd (integer),
method (string)

alignedannot
id (integer, primary key),
onSequence (integer, FK sequence.id),
start (integer),
end (integer),
strand (boolean),
name (string),
annotation (string),
species (string),
source (string),
method (string),
score (string)

Notes:

1. In real use, the 'sequence' table would also contain a column for the *actual* DNA/RNA/protein sequences each sequence row represents (e.g. AGCATGCTAG…) but is left out of this demo to reduce file size. As is, the 'sequence' table just contains details *about* sequences.

2. FK = Foreign key, i.e. an ID referencing a row of a different table, creating association.

3. The 'seqRelation' has 2 foreign keys, both back to the sequence table, 'childSeq' and 'parentSeq'. This pair defines a relationship between two sequences- that a 'childSeq' is a sub-sequence of a 'parentSeq' or that a 'childSeq' aligns onto a 'parentSeq'. For example, a gene sequence that originates within a longer genome sequence, may be defined with a child/parent relationship.

Part 1 - Initialisation

The following tab-separated text files are included, which contain data ready to be loaded into the database described above (they correspond to tables as per the start of their filenames):

samples1.txt
seqgroups1.txt
seqtypes.txt
sequences1-D18-genomic.txt
sequences2-D36-RNA.txt
sequences3-D18-augustusGenePredict.txt
seqrelations1-D18augPred-vs-D18g.txt
seqrelations2-D36rna-vs-D18g.txt
alignedannot1-D18augPred-vs-NCBI-nr.txt

1. Create a new database to use

2. Write/run the CREATE TABLE statements required to create the tables described on the previous page. E.g. here's a couple to get you started*:

```
CREATE TABLE sample (
  id integer PRIMARY KEY,
  name text,
  species text,
  description text
);

CREATE TABLE sequence (
  id integer PRIMARY KEY,
  name text,
  length integer,
  belongsGroup integer,
  isSample integer,
  isType integer,
  FOREIGN KEY (belongsGroup) REFERENCES seqgroup(id),
  FOREIGN KEY (isSample) REFERENCES sample(id),
  FOREIGN KEY (isType) REFERENCES seqtype(id)
);
```

   Make sure you list columns in the same order as listed above, which is also the same order as in the data text files. When importing the data in the next step, *the column order must match*!

   *Note, if making use of foreign key definitions (good practice), the *order* that you create the tables will matter- *you can't reference another table until you've first created that other table*. I.e. create certain other tables before creating the sequence table.

3. Load the text files into the corresponding tables.
   Note, again, if making use of foreign keys, the order that you load tables may matter.

Our input files are tab-separated and each contain a header row. We can import them into our sqlite tables using a couple of tricks:

```
.separator "\t"
.import "|tail -n +2 samples1.txt" sample
.import "|tail -n +2 seqgroups1.txt" seqgroup
.import "|tail -n +2 seqtypes.txt" seqtype
.import "|tail -n +2 sequences1-D18-genomic.txt" sequence
.import "|tail -n +2 sequences2-D36-RNA.txt" sequence
.import "|tail -n +2 sequences3-D18-augustusGenePredict.txt"
sequence
.import "|tail -n +2 alignedannot1-D18augPred-vs-NCBI-nr.txt"
alignedannot
.import "|tail -n +2 seqrelations1-D18augPred-vs-D18g.txt"
seqrelation
.import "|tail -n +2 seqrelations2-D36rna-vs-D18g.txt" seqrelation
```

The '.separator' command sets our import file field separator to be tabs. 'tail -n +2' skips the first row of a file, i.e. our header row.

You may now also like to run:

```
.mode columns
.headers on
```

Part 2 - Queries

Try to answer each of the following by writing a *single* SELECT statement per question.

1. Can you list all details about all samples?
2. Can you list just the name and species for all samples?
3. Can you list details of the first 10 sequences?
4. How many loaded sequences are there?
5. How many sequences are of the type "genomeAssembly"?
6. How many loaded sequences are there of each different type?
7. How many sequences have a length greater than 1000?
8. How many sequences have a length greater than 1000 and are from sample "D36-s2-tr"?
9. What is the average length of all sequences?
10. What is the average length of sequences of type "genomeAssembly"?
11. What is the average length of sequences of type "genomeAssembly" AND what is the average length of sequences of type "protein"?
12. How many annotations (alignedannot.annotation) contain the phrase "hypothetical"?

13. Can you list details of the sequence named "D18-gDNA-s1638", replacing the foreign keys with sensible info (e.g. replace 'isSample' id with actual sample name)?

14. Does the sequence named "D18-gDNA-s1638" have any other sequences that align onto it (it'll appear in seqRelation.parentSeq)? List any such sequences.

    Hint: You'll need to make use of the 'AS' keyword (https://www.w3schools.com/sql/sql_alias.asp)