1. Query for a song:
note: when multiple versions of the song are found, use the more specific name until only one version is found

```
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX mid: <http://purl.org/midi-ld/midi#>

SELECT ?filename ?pattern

WHERE {
?pattern prov:wasDerivedFrom ?filename .
FILTER (regex(?filename, "song title/author", "i")) .
}
```

2. Query for the tempo values of selected track:

```
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX mid: <http://purl.org/midi-ld/midi#>

SELECT *

WHERE {
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
?track mid:hasEvent ?timesignatureevent .
?timesignatureevent a mid:TimeSignatureEvent .
?timesignatureevent mid:denominator ?denominator .
?timesignatureevent mid:metronome ?metronome .
?timesignatureevent mid:numerator ?numerator .
?timesignatureevent mid:thirtyseconds ?thirtyseconds .
FILTER (regex(?filename, "song title (exact)", "i")) .
}
```

3. Query to search for a track with certain tempo values:
note: you can filter for a certain song, or just remove this to select a song from the first 1000 songs that are found. LIMIT can also be changed if you want more results, but this is more time consuming:

```
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX mid: <http://purl.org/midi-ld/midi#>

SELECT DISTINCT ?pattern ?filename

WHERE {
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
?track mid:hasEvent ?timesignatureevent .
?timesignatureevent a mid:TimeSignatureEvent .
?timesignatureevent mid:denominator value1 .
?timesignatureevent mid:metronome value2 .
?timesignatureevent mid:numerator value3 .
?timesignatureevent mid:thirtyseconds value4 .
FILTER (regex(?filename, "song title/author", "i")) . #optional

}

LIMIT 1000
```

4. Query to combine two songs using all tracks:

```
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX mid: <http://purl.org/midi-ld/midi#>

CONSTRUCT { <crap> a mid:Pattern ;
mid:hasTrack ?track .
<crap> mid:format ?format .
<crap> mid:resolution ?resolution .
?track mid:hasEvent ?event .
?track a mid:Track .
?event a ?type .
?event ?property ?value .
}

WHERE {
{
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
?pattern mid:format ?format .
?pattern mid:resolution ?resolution1 .
?track mid:hasEvent ?event .
?event a ?type .
?event ?property ?value .
FILTER (regex(?filename, "song1", "i")) .
} UNION {
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
?pattern mid:format ?format .
?pattern mid:resolution ?resolution2 .
?track mid:hasEvent ?event .
?event a ?type .
?event ?property ?value .
FILTER (regex(?filename, "song2", "i")) .
}
}
```

5. Query for the tracks of a song:

```
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX mid: <http://purl.org/midi-ld/midi#>

SELECT ?track

WHERE {
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
FILTER (regex(?filename, “song title (exact)“, "i")) .
}
```

6. Query to combine the songs, using only desired tracks:

```
PREFIX prov: <http://www.w3.org/ns/prov#>
PREFIX mid: <http://purl.org/midi-ld/midi#>

CONSTRUCT { <newsong> a mid:Pattern ;
mid:hasTrack ?track .
<newsong> mid:format ?format .
?track mid:hasEvent ?event .
?track a mid:Track .
?event a ?type .
?event ?property ?value .
}

WHERE {
{
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
?pattern mid:format ?format .
?pattern mid:resolution ?resolution .
?track mid:hasEvent ?event .
?event a ?type .
?event ?property ?value .
FILTER (regex(?filename, "song1", "i")) .
FILTER (?track IN (<track1>, <track2>, <and so on>))
} UNION {
?pattern prov:wasDerivedFrom ?filename .
?pattern mid:hasTrack ?track .
?pattern mid:format ?format .
?pattern mid:resolution ?resolution .
?track mid:hasEvent ?event .
?event a ?type .
?event ?property ?value .
FILTER (regex(?filename, "song2", "i")) .
FILTER (?track IN (<track1>, <track2>, <and so on>))
}
}
```