MONTANA
STATE UNIVERSITY

Paging & Ordering

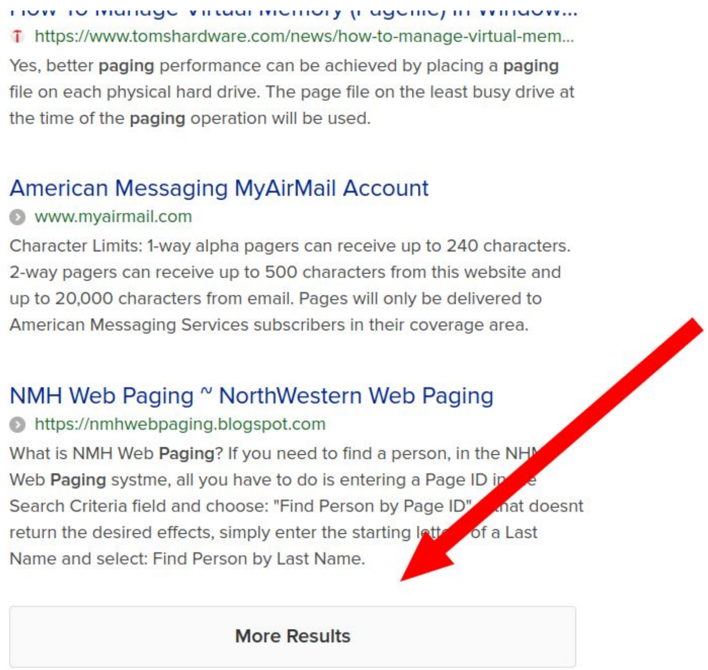
•••

Paging

- If you have used the internet, you are familiar with the concept of paging
- No, not that kind of paging..

Paging

- This kind of paging



How To Manage Virtual Memory (Pagefile) in Window...

https://www.tomshardware.com/news/how-to-manage-virtual-mem...

Yes, better **paging** performance can be achieved by placing a **paging** file on each physical hard drive. The page file on the least busy drive at the time of the **paging** operation will be used.

American Messaging MyAirMail Account

www.myairmail.com

Character Limits: 1-way alpha pagers can receive up to 240 characters. 2-way pagers can receive up to 500 characters from this website and up to 20,000 characters from email. Pages will only be delivered to American Messaging Services subscribers in their coverage area.

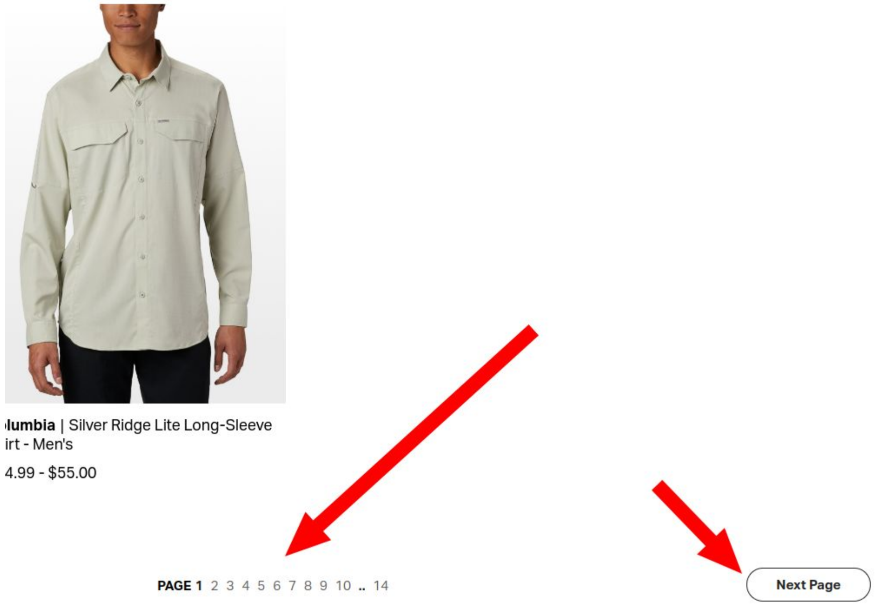
NMH Web Paging ~ NorthWestern Web Paging

https://nmhwebpaging.blogspot.com

What is NMH Web **Paging**? If you need to find a person, in the NHM Web **Paging** systme, all you have to do is entering a Page ID in the Search Criteria field and choose: "Find Person by Page ID". If that doesnt return the desired effects, simply enter the starting letters of a Last Name and select: Find Person by Last Name.

Paging

- And this kind of paging

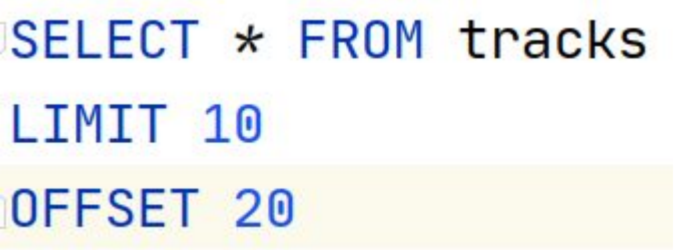


lumbia | Silver Ridge Lite Long-Sleeve
irt - Men's

4.99 - $55.00

Paging

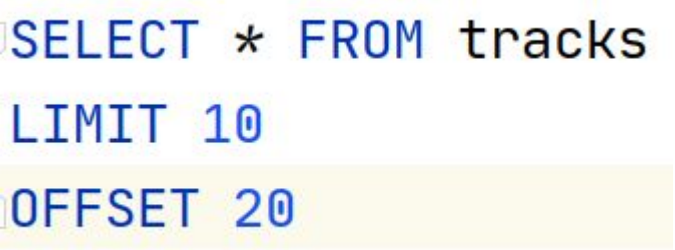
- SQL Database typically have support for a notion of paging
- It is not, however, part of the SQL standard
- In SQLite there are two optional clauses in the SELECT statement:
    - LIMIT - limits the number of results
    - OFFSET - takes an offset to start at



```
SELECT * FROM tracks
LIMIT 10
OFFSET 20
```

Paging

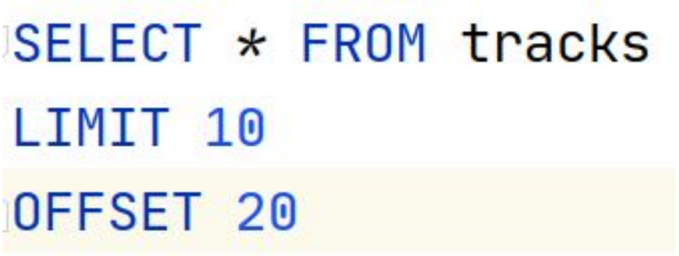
- Note that these can be used independently
- LIMIT, in particular, might be used to limit a search page to only N results



```
SELECT * FROM tracks
LIMIT 10
OFFSET 20
```

Paging

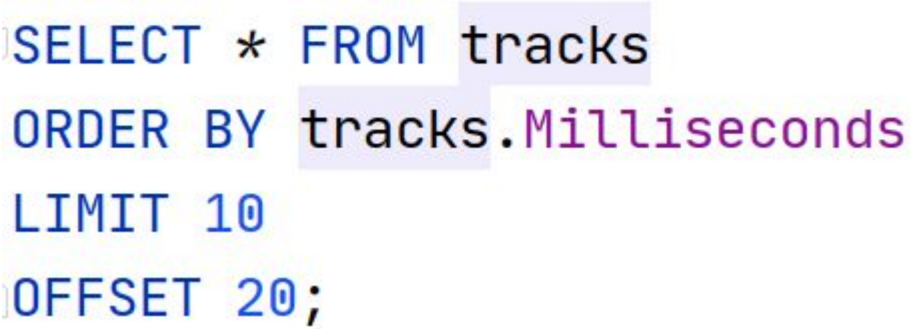
- Offset is the total offset in records, not pages
- So in this case we are looking at page 3 of a system that is showing 10 items at a time:
    - OFFSET 0 - page 1
    - OFFSET 10 - page 2
    - OFFSET 20 - page 3



```
SELECT * FROM tracks
LIMIT 10
OFFSET 20
```

Ordering

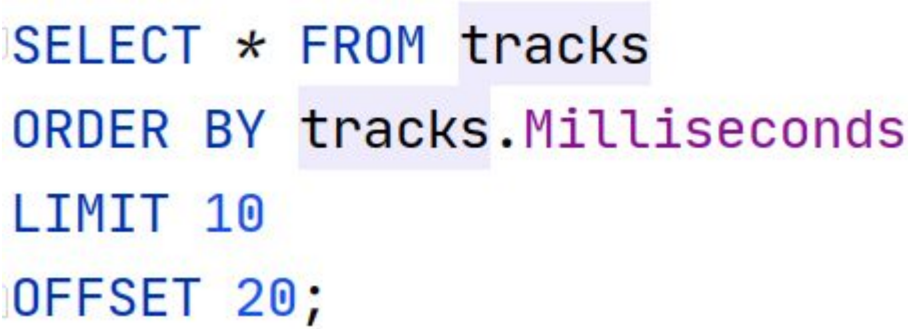
- You often wish to order the data you are displaying
- For example, maybe you allow sorting by different columns
- This is accomplished with the ORDER BY clause in SQL



```sql
SELECT * FROM tracks
ORDER BY tracks.Milliseconds
LIMIT 10
OFFSET 20;
```

Ordering

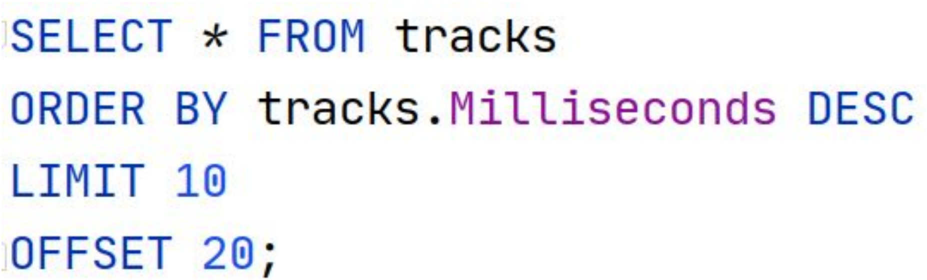
- You can order by any column in your query
- Note that in aggregate queries you must order by an attribute in the final results



```
SELECT * FROM tracks
ORDER BY tracks.Milliseconds
LIMIT 10
OFFSET 20;
```

Ordering - Direction

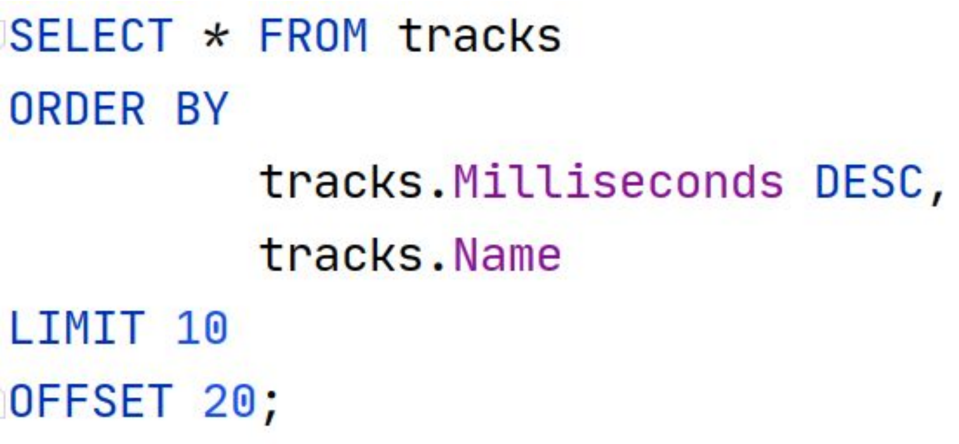
- By default, ordering is done in an ascending direction
- If you want to order in the descending direction, you can add the DESC
- You can also use the ASC keyword if you wish to be explicit about ascending order



```
SELECT * FROM tracks
ORDER BY tracks.Milliseconds DESC
LIMIT 10
OFFSET 20;
```

Ordering - Multiple Columns

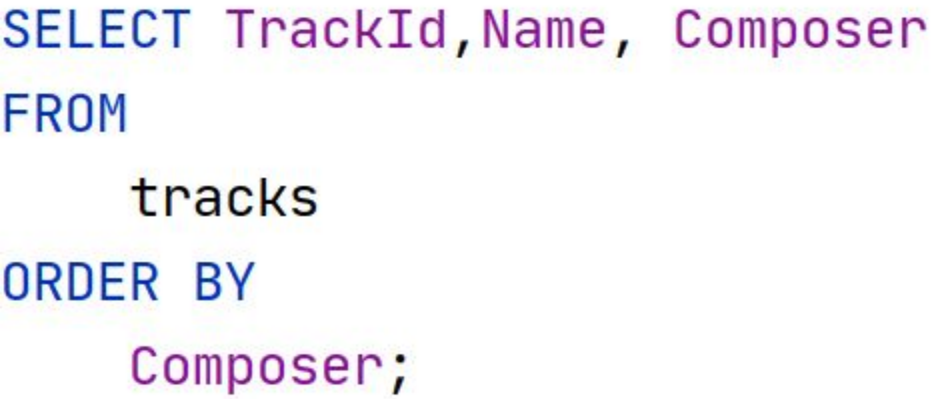
- If you wish to order by multiple columns, you can use comma separated ordering specifications
- *Order by milliseconds, descending and if two tracks are equal in length order them by name, ascending*



```
SELECT * FROM tracks
ORDER BY
         tracks.Milliseconds DESC,
         tracks.Name
LIMIT 10
OFFSET 20;
```

Ordering - Nulls

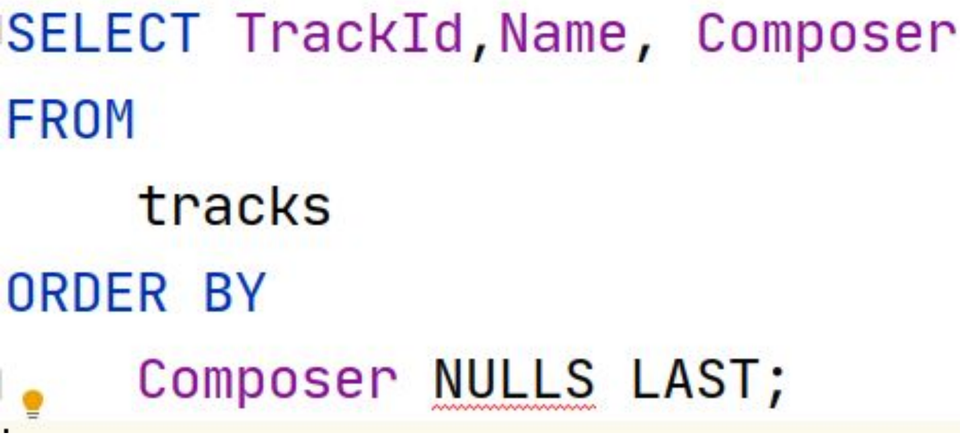
- Our old friend NULL
- What does NULL mean when ordering values?
- Kinda depends…
- By default NULL is considered “first” in SQLite



```
SELECT TrackId,Name, Composer
FROM
    tracks
ORDER BY
    Composer;
```

Ordering - Nulls

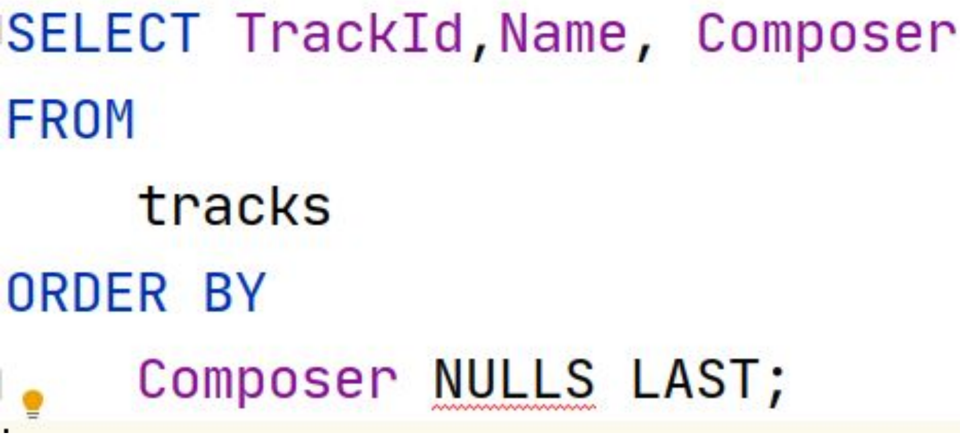
- If you want the opposite behavior you can use the NULLS LAST qualifier in an ORDER BY clause



```
SELECT TrackId,Name, Composer
FROM
    tracks
ORDER BY
    Composer NULLS LAST;
```

Ordering - Nulls

- Note that NULLS LAST means literally: NULLS LAST
- If you run a DESC order with the clause, nulls will still show up at the end of the query
- This is SQLite specific stuff, a different DB may have dramatically different behavior



```
SELECT TrackId,Name, Composer
FROM
    tracks
ORDER BY
    Composer NULLS LAST;
```

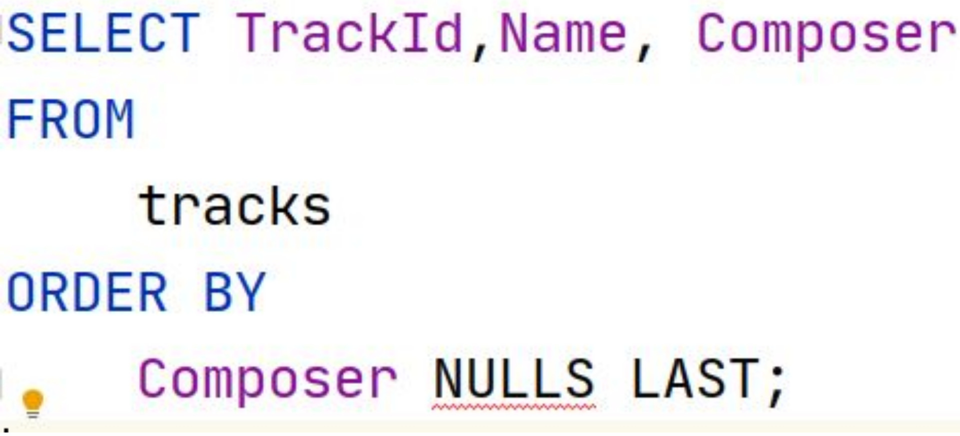
EXPLAIN

•••

Learning how a query will execute

Explain

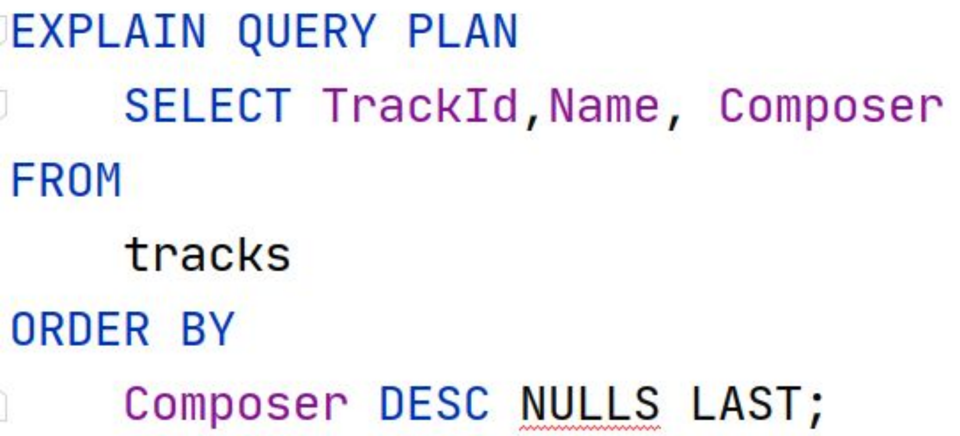
- As we have discussed before, SQL is a declarative rather than an imperative language
- SQL tells the database what you want, not how to get it
- But sometimes you want to know how the DB will get it
  - Mainly for perf reasons



```
SELECT TrackId,Name, Composer
FROM
    tracks
ORDER BY
    Composer NULLS LAST;
```

Explain

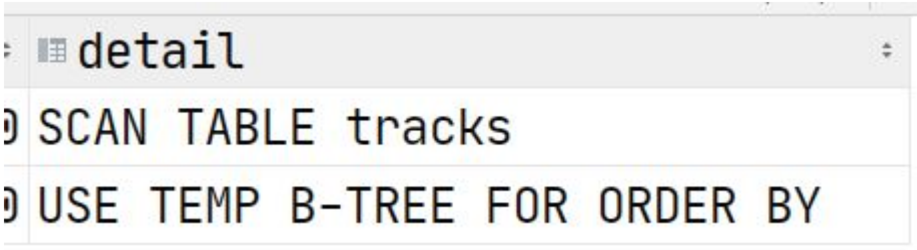
- To learn how a query is going to execute, you can use the EXPLAIN QUERY PLAN statement



```
EXPLAIN QUERY PLAN
    SELECT TrackId,Name, Composer
FROM
    tracks
ORDER BY
    Composer DESC NULLS LAST;
```

Explain

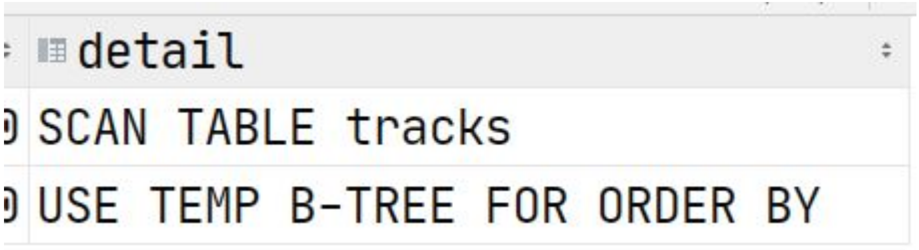
- The results for this query are interesting!
- The DB will do a table scan
- The DB will then create a temporary B-Tree (a data structure) for ordering the results



detail
SCAN TABLE tracks
USE TEMP B-TREE FOR ORDER BY

Explain

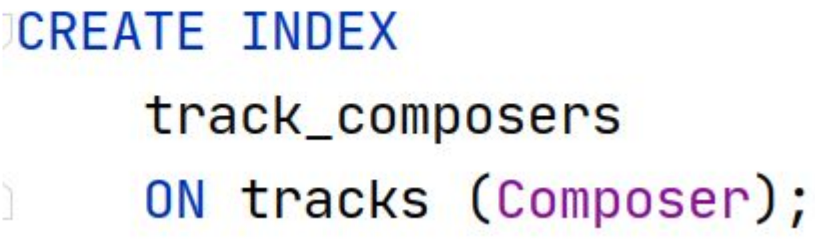
- That’s unfortunate that we need to create a temporary B-Tree
- We can fix that by adding an index on that column
- We will discuss indexes (and B-Trees) in more depth later
- For now, just assume they mean “fast access”



detail
SCAN TABLE tracks
USE TEMP B-TREE FOR ORDER BY

Explain

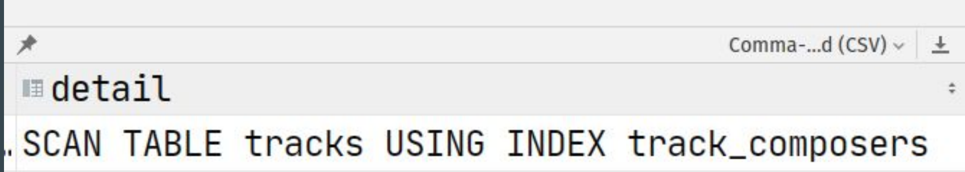
- Adding an index is pretty simple



```
CREATE INDEX
    track_composers
    ON tracks (Composer);
```

Explain

- And now our query is more efficient!
- No temporary B-Trees needed!
- Awesome!

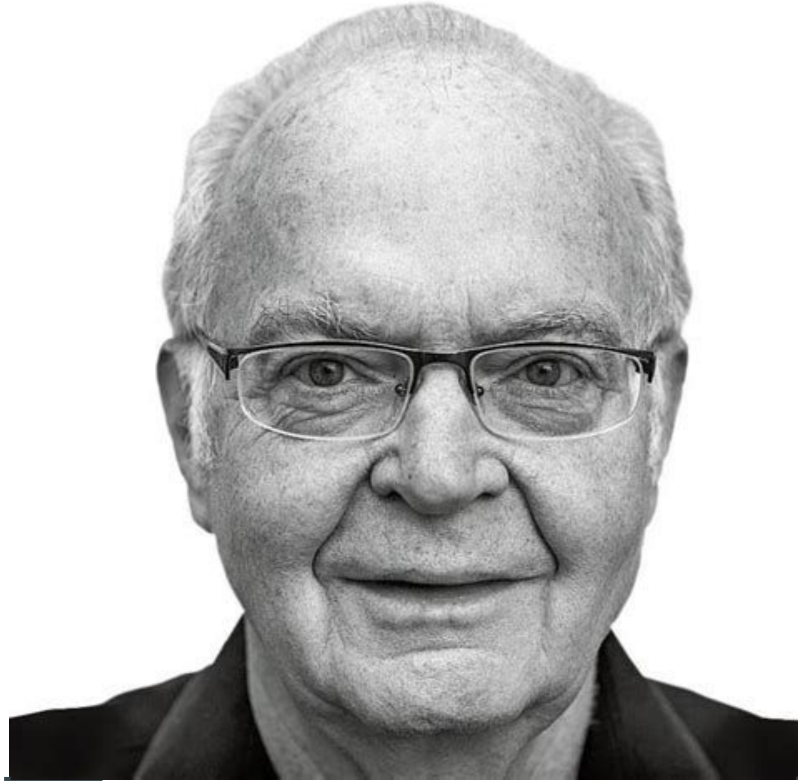


```
detail
SCAN TABLE tracks USING INDEX track_composers
```

Optimization

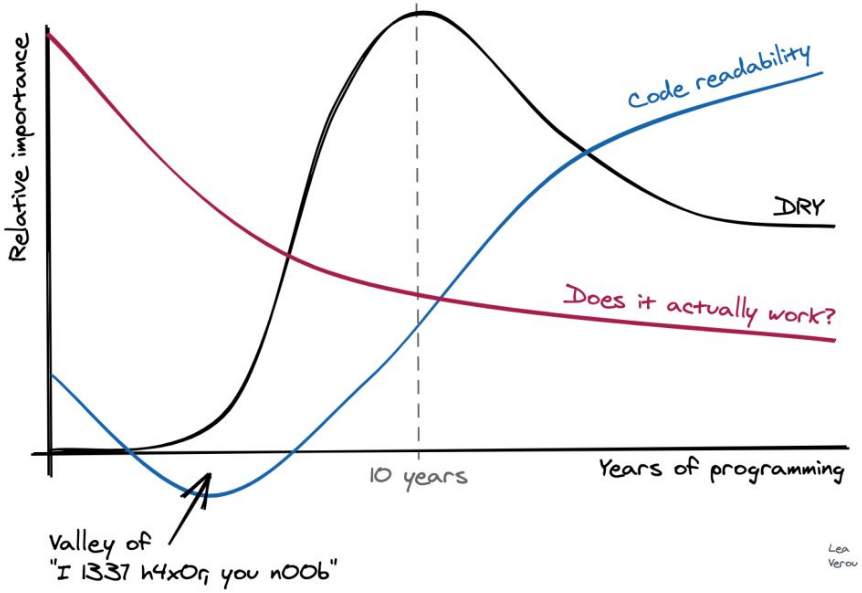
“The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming.”

-- Donald Knuth



Optimization

> *“The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming.”*
>
> *-- Donald Knuth*



Paging, Ordering & Explain Summary

- You can page data in SQLite (not SQL) using the LIMIT/OFFSET clauses
- You can order your results with the ORDER BY clause (standard SQL)
    - Null hanlding can be tricky, definitely non-standard
- If you want to see the implementation of a query, the EXPLAIN QUERY PLAN statement will tell you how SQLite is going to implement a given query
- With that information, you can make optimization decisions

MONTANA
STATE UNIVERSITY