User .Pixels files instead of .Parquet

Derek Snow, 6/21/25

You have probably heard of .Parquet, but I am sure you have never heard of .Pixels? Recall how Parquet is ~100x faster than CSV, well Pixels is ~10,000x faster than CSV (yes, I am slightly exagerating here).  
  
The biggest speedup comes when filtering, scanning, or aggregating large tables—Pixels’ advanced, thread-safe indexing lets the query engine skip over irrelevant data far more efficiently than Parquet.  
  
You can use DuckDB’s Python API (with Pixels support) like this:  
  
import duckdb  
con = duckdb.connect()  
con.execute("SELECT \* FROM 'yourfile.pixels'")  
df = con.fetchdf()  
  
Pixels has quickly become one of my favorite storage formats—powerful, efficient, and still largely undiscovered. Because Pixels is just a highly-tuned columnar file format (similar in spirit to Parquet), any table that has a timestamp column can be treated as a time series. The real gains come from two things:  
  
Range filtering on the timestamp: Pixels’ thread-safe primary index can jump straight to the row-groups that cover the time window you ask for, so scans touch far fewer bytes than in Parquet.  
  
Aggregations or roll-ups.(e.g., GROUP BY date\_trunc('hour', ts)): Adaptive column layout keeps the timestamp and the one or two value columns you’re aggregating contiguous in memory, so the engine streams through them at CPU-cache speed.  
  
import duckdb, pandas as pd  
con = duckdb.connect()  
df = con.execute("""  
SELECT \*  
FROM 's3://bucket/trades/2025/06/18/\*.pixels'  
WHERE ts BETWEEN '2025-06-18 09:30' AND '2025-06-18 16:00'  
""").fetchdf()  
  
It is great for quant large time series workflows, give it a shot.

References

[1] [Original post by Derek Snow on Linkedin, 6/21/2025](https://www.linkedin.com/posts/snowderek_you-have-probably-heard-of-parquet-but-activity-7341485343306674176--t1z?utm_source=share&utm_medium=member_desktop&rcm=ACoAAAFZfUoBgPoGUucdnvtwuzPv79P8VHj6uvk)

[2] <https://github.com/pixelsdb/pixels>

[3] <https://duckdb.org/docs/stable/core_extensions/vss>

[4] <https://github.com/pixelsdb/duckdb>