

Quack Attack: Bringing DuckDB to the Dart side

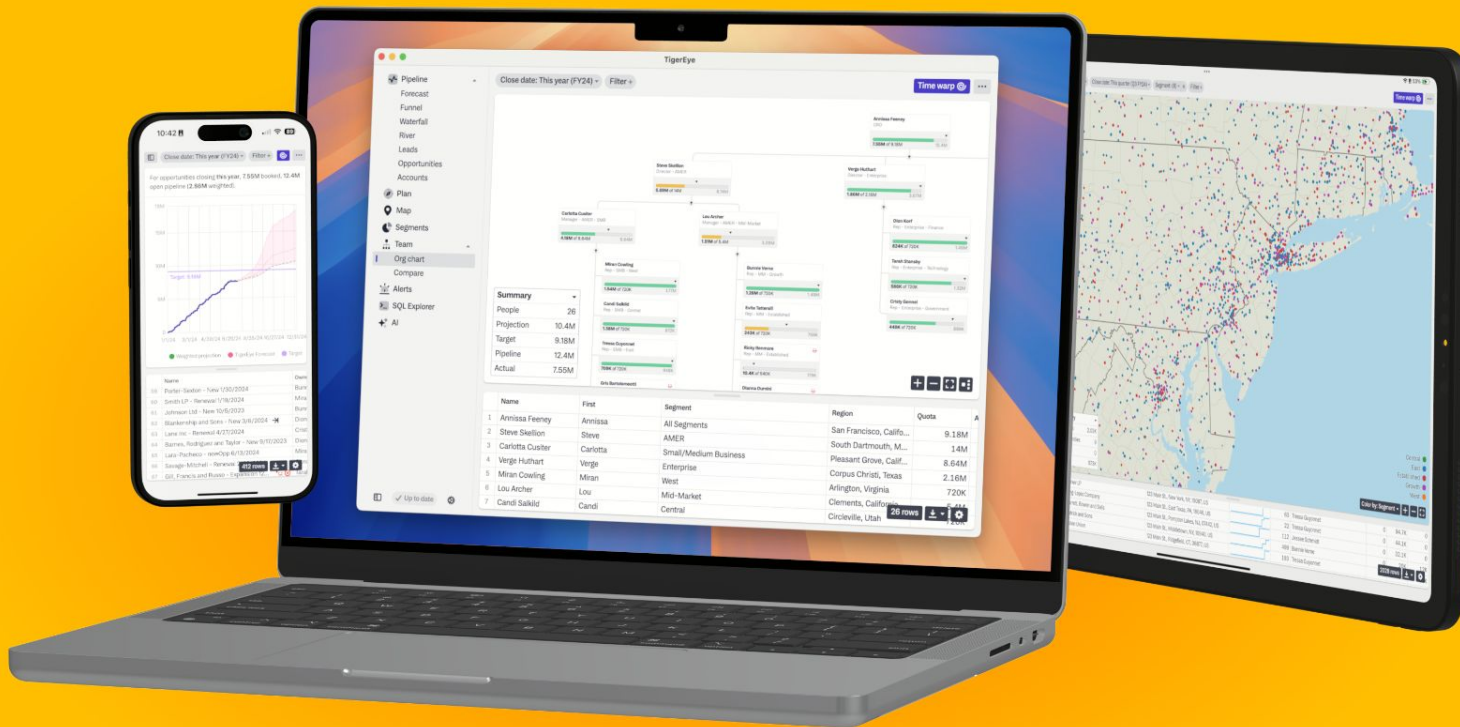


Hello



Andy Prock

Software Engineer @ TigerEye



MacOS | iOS | Android | Windows

DuckDB and Dart

1. Technology (Features)
2. Usage (Examples, Benchmarks)
3. Community (Sample App, Open Source)

What are Dart and Flutter?

“Dart is a client-optimized language for developing fast apps on any platform.”
–Google

Flutter is the UX layer built on top of Dart.

Surface™



ByteDance



Google Ads



ebay™



GROUPON

PHILIPS
hue



Square



DuckDB.Dart

Native Dart Integration

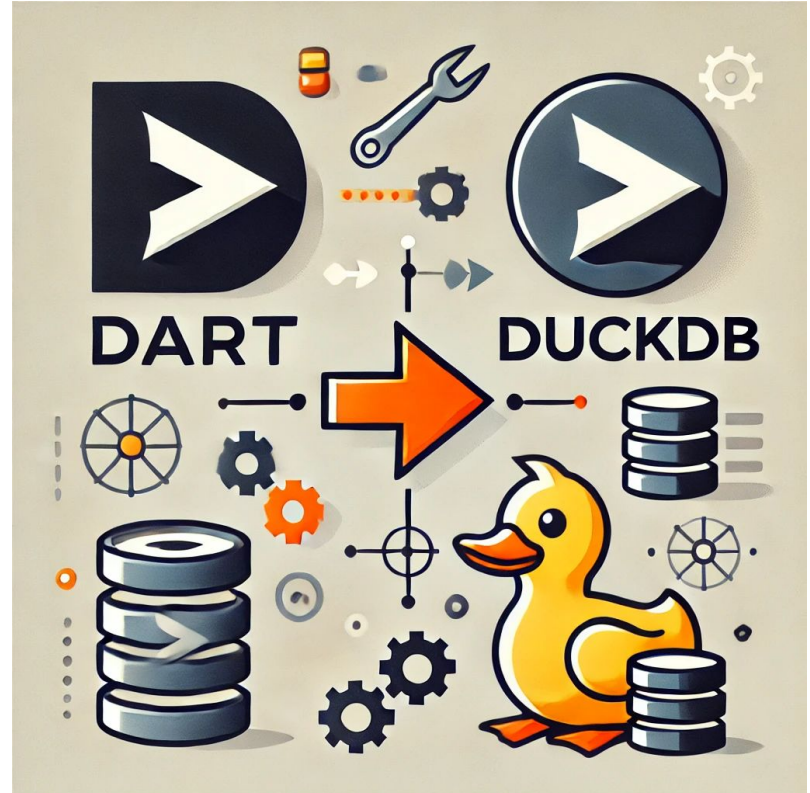
- Direct access to DuckDB's capabilities
- Custom Dart data types for seamless use

High Performance

- Built on DuckDB's C APIs
- Multithreaded connection support

Community-Inspired

- Inspired by other language wrappers



Features

New Dart Support

- Decimal, Date, Time, Interval

Operations

- Prepared Statements
- Appenders

Included Extensions

- icu; parquet; json
- fts; autocomplete

Test Coverage

- 80% line and branch

Querying Data

```
import 'package:dart_duckdb/dart_duckdb.dart' ;

final db = duckdb.open(":memory:");
final connection = duckdb.connect(db);

// Create a table and insert some data
connection.execute('''
    CREATE TABLE users (id INTEGER, name VARCHAR, age INTEGER);
    INSERT INTO users VALUES (1, 'Alice', 30), (2, 'Bob', 25);
''');

// Run a query and print results
final result = connection.query("SELECT * FROM users WHERE age > 28");
for (final row in result.fetchAll()) {
    print("User: ${row[1]}, Age: ${row[2]}");
}
```


Bulk Insertion with Appenders

```
final appender = connection.append("users")

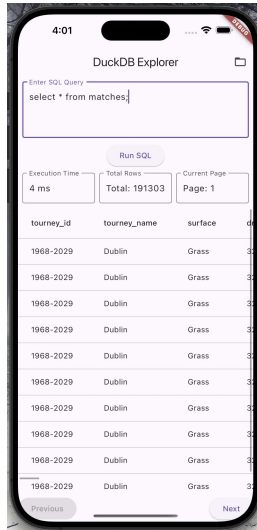
for (final user in [
    (id: 1, name: 'Alice', age: 30),
    (id: 2, name: 'Bob', age: 25)
]) {
    appender.append(user.id);
    appender.append(user.name);
    appender.append(user.age);
    appender.endRow();
}

appender.flush();
```

Transferable Databases

```
void main() async {  
    final db = duckdb.open(':memory:');  
  
    await Isolate.spawn(backgroundTask, db.transferrable);  
    ...  
}  
  
void backgroundTask(TransferableDatabase transferableDb) {  
    final connection = duckdb.connectWithTransferred(transferableDb);  
    ...  
}
```

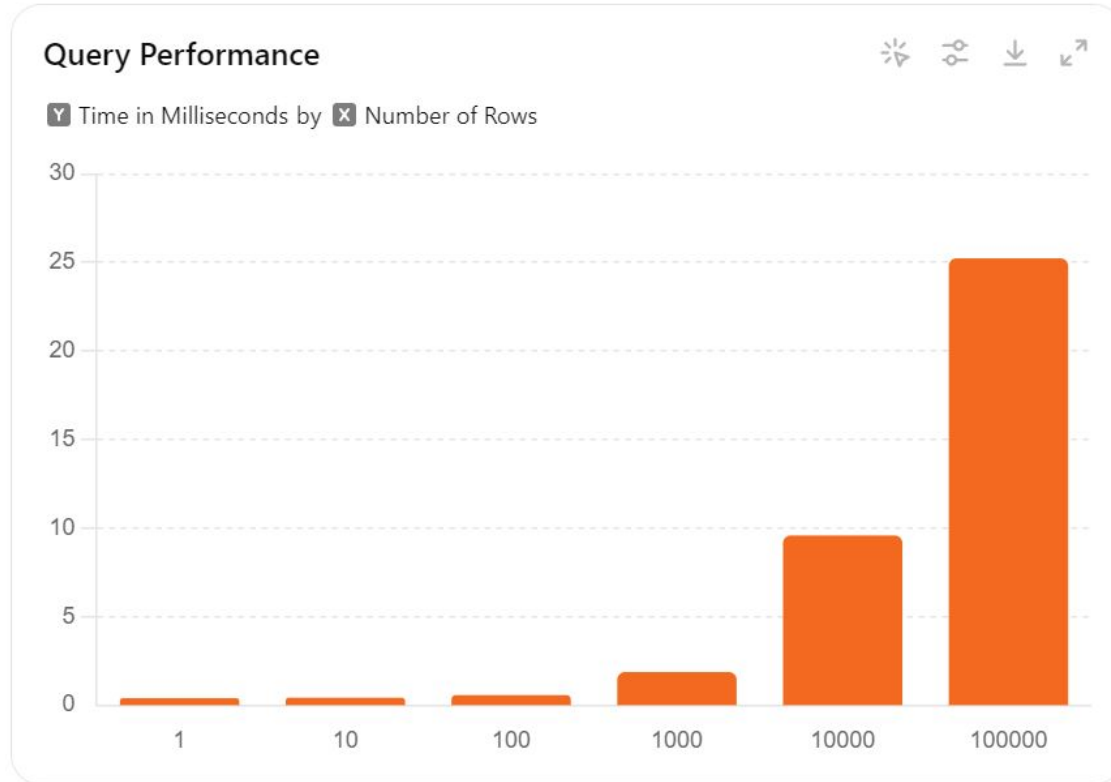
DuckDB Explorer



The desktop view of DuckDB Explorer shows the same interface. The title bar reads "duckdbexplorer". The main window title is "DuckDB Explorer". The SQL query input field contains "select * from matches;". A "Run SQL" button is located below the input field. Below the button are three summary boxes: "Execution Time: 3 ms", "Total Rows: 191303", and "Current Page: Current Page: 11". The table below has the same columns as the mobile view. The table contains 12 rows of data, all for the year 1968-347 and the location Bournemouth. At the bottom, there are "Previous" and "Next" navigation buttons.

tourney_id	tourney_name	surface	draw_size	tourney_level	tourney_date	match_num	winner_id	winner_seed	winner_entry	winner_name
1968-347	Bournemouth	Clay	32	A	1968-04-22	3	109775	null	null	Robert Wilson
1968-347	Bournemouth	Clay	32	A	1968-04-22	4	110047	null	Q	John Ward
1968-347	Bournemouth	Clay	32	A	1968-04-22	5	100005	null	null	Richard Gonzalez
1968-347	Bournemouth	Clay	32	A	1968-04-22	6	100073	null	null	Mark Cox
1968-347	Bournemouth	Clay	32	A	1968-04-22	7	100022	5	null	Roy Emerson
1968-347	Bournemouth	Clay	32	A	1968-04-22	8	100152	null	Q	David Lloyd
1968-347	Bournemouth	Clay	32	A	1968-04-22	9	112374	null	null	Keith Wooldridge
1968-347	Bournemouth	Clay	32	A	1968-04-22	10	100030	6	null	Fred Stolle
1968-347	Bournemouth	Clay	32	A	1968-04-22	11	100010	null	null	Luis Ayala
1968-347	Bournemouth	Clay	32	A	1968-04-22	12	100026	3	null	Andres Gimeno

Benchmarks



*MacBook Pro M2
with 16GB RAM*

Supported Platforms



Currently No WASM

DuckDB.Dart Getting Started

Source code available on github

<https://github.com/TigerEyeLabs/duckdb-dart>

Distributed on pub.dev

https://pub.dev/packages/dart_duckdb

MRs are welcome!



Q&A