DuckDB Analytical SQL shell

CONCEPTS

With no file argument DuckDB runs in memory and everything is lost on exit; pass a file to persist.

START AND RUN

duckdb
Start an in-memory session

duckdb *<file.db>*
Open or create a persistent database

duckdb -c "*<sql>***"**
Run one statement and exit

duckdb -csv -c "*<sql>***"**
Print the result as CSV

duckdb -json -c "*<sql>***"**
Print the result as JSON

duckdb *<file.db>* **-f** *<script.sql>*
Run a script file and exit

DOT COMMANDS

.help
List all dot commands

.tables
List tables with columns and row counts

.schema *<pattern>*
Show CREATE statements

.open *<file.db>*
Switch to another database

.quit
Leave the shell

OUTPUT

.mode *<mode>*
Set the output format

.output *<file>*
Send all output to a file

EXPLORE DATA

FROM *<table>*;
FROM-first query, SELECT * implied

EXPLORE DATA (cont.)

DESCRIBE *<table>*;
Columns and types

DESCRIBE FROM '*<file>***';**
Infer the schema of a file

SUMMARIZE *<table>*;
Min, max, nulls, quantiles per column

READ FILES

FROM '*<file.csv>***';**
Query a CSV directly

FROM '*<file.parquet>***';**
Query a Parquet file

FROM '*<file.json>***';**
Query a JSON file

FROM '*<dir>***/*.parquet';**
Glob many files as one table

FROM read_csv('*<f>***', header=true, delim=';');**
CSV with explicit options

TABLES AND IMPORT

CREATE TABLE *<t>* **AS FROM '***<file>***';**
Load a file into a table

EXPORT

COPY *<table>* **TO '***<file.csv>***' (HEADER);**
Export a table as CSV

COPY (*<query>***) TO '***<file.csv>***';**
Export a query result

COPY *<table>* **TO '***<file.parquet>***' (FORMAT parquet);**
Export as Parquet

SQL EXTRAS

SELECT * EXCLUDE (*<col>***) FROM** *<t>*;
All columns except some

SELECT *<col>*, **count(*) FROM** *<t>* **GROUP BY ALL;**
Group by every non-aggregate column