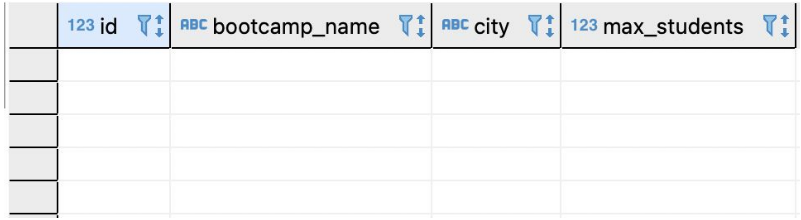
neue fische

School and Pool for Digital Talent

Modifying Tables

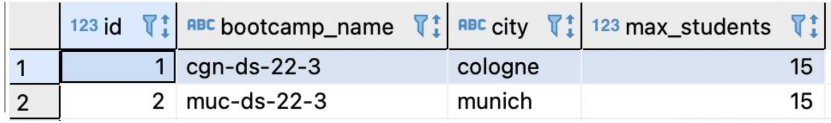
Creating Tables

Define schema of data upfront, including constraints.
No data inserted yet!



123 id	ABC bootcamp_name	ABC city	123 max_students

```
CREATE TABLE alte_fische.bootcamps
(
    id            INTEGER PRIMARY KEY NOT NULL,
    bootcamp_name VARCHAR             NOT NULL,
    city          VARCHAR,
    max_students  INTEGER
);
```

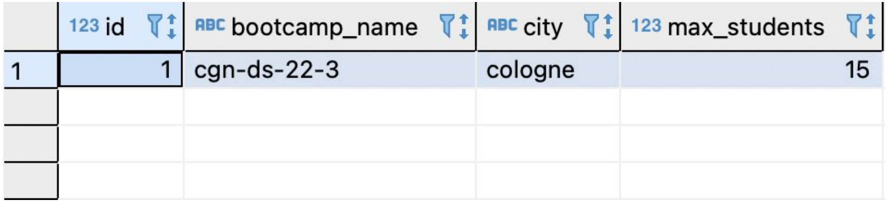


	id	bootcamp_name	city	max_students
1	1	cgn-ds-22-3	cologne	15
2	2	muc-ds-22-3	munich	15

Inserting Data

Rows are inserted via INSERT

```
INSERT INTO main.bootcamps
VALUES (1, 'cgn-ds-22-3', 'cologne', 15);
INSERT INTO main.bootcamps
VALUES (2, 'muc-ds-22-3', 'munich', 15);
```



	123 id	ABC bootcamp_name	ABC city	123 max_students
1	1	cgn-ds-22-3	cologne	15

Deleting Data

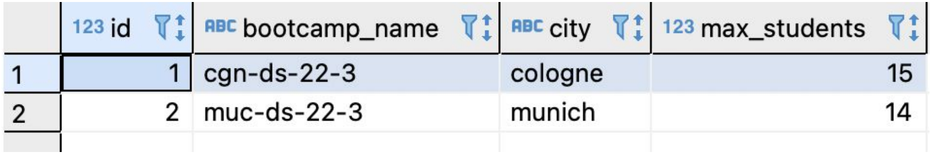
DELETE can have conditions. Make sure that they are precise!

```
DELETE
FROM
alte_fische.bootcamps
WHERE id = 2;
```

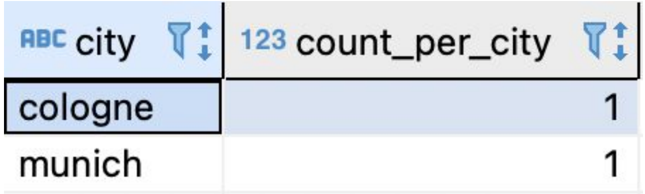
Updating Data

UPDATE can have conditions. Make sure that they are precise!

```
UPDATE
alte_fische.bootcamps
SET max_students = 14
WHERE id = 2;
```



	123 id	ABC bootcamp_name	ABC city	123 max_students
1	1	cgn-ds-22-3	cologne	15
2	2	muc-ds-22-3	munich	14



ABC city	123 count_per_city
cologne	1
munich	1

Creating Tables based on Select

We can create a new table from a Select Statement

```
CREATE TABLE alte_fische.bootcamps_per_city as (SELECT city, COUNT(*) as count_per_city
                                        FROM alte fische.bootcamps
                                        GROUP BY city);
```

If the table already exists, you can not create it again. First need to drop (CAUTION)

```
DROP TABLE alte_fische.bootcamps_per_city;
```

neue fische
School and Pool for Digital Talent

Extracting, Loading and Transforming Data

ELT

Making data accessible

Extract Extracting Data from a source → Loading a JSON from a Webserver

Integrating heterogeneous sources

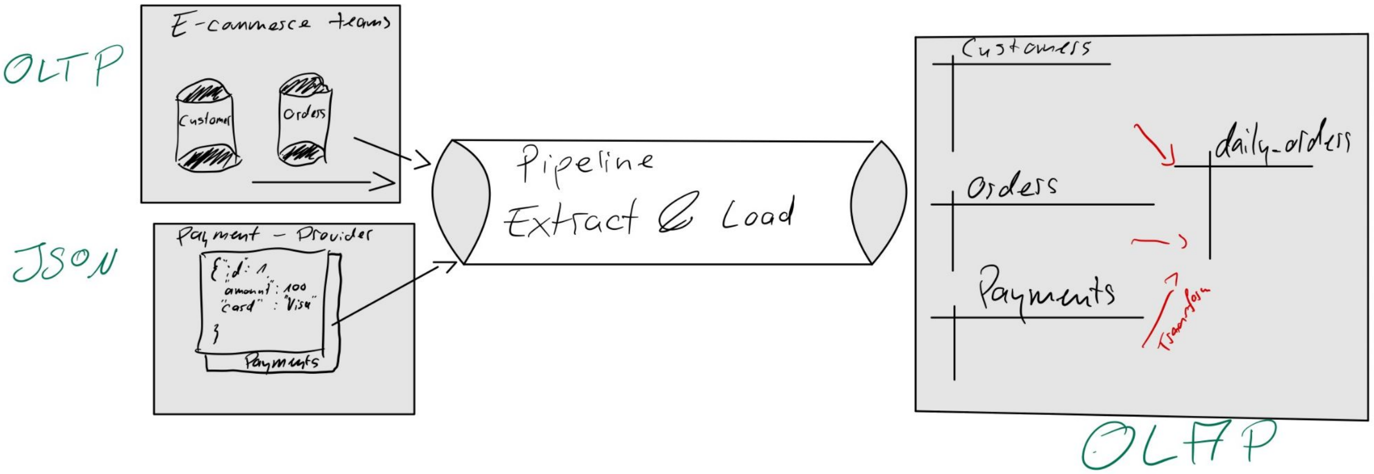
Load Moving the data to a central place → Data Lake, Data Warehouse, Analytical Database

Bringing Everything to a central place

Transform Transform the data inside the Data Lake → Aggregate single Events to aggregation on daily basis

Enabling Interpretation through abstractions

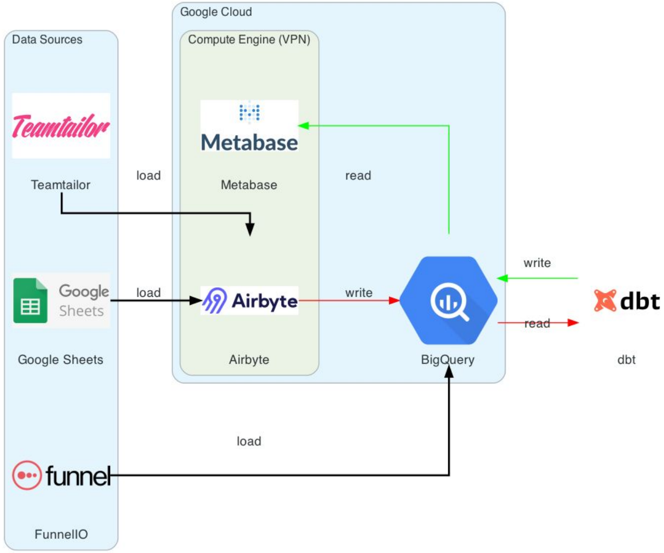
ELT Example



OLTP: Online Transactional Processing

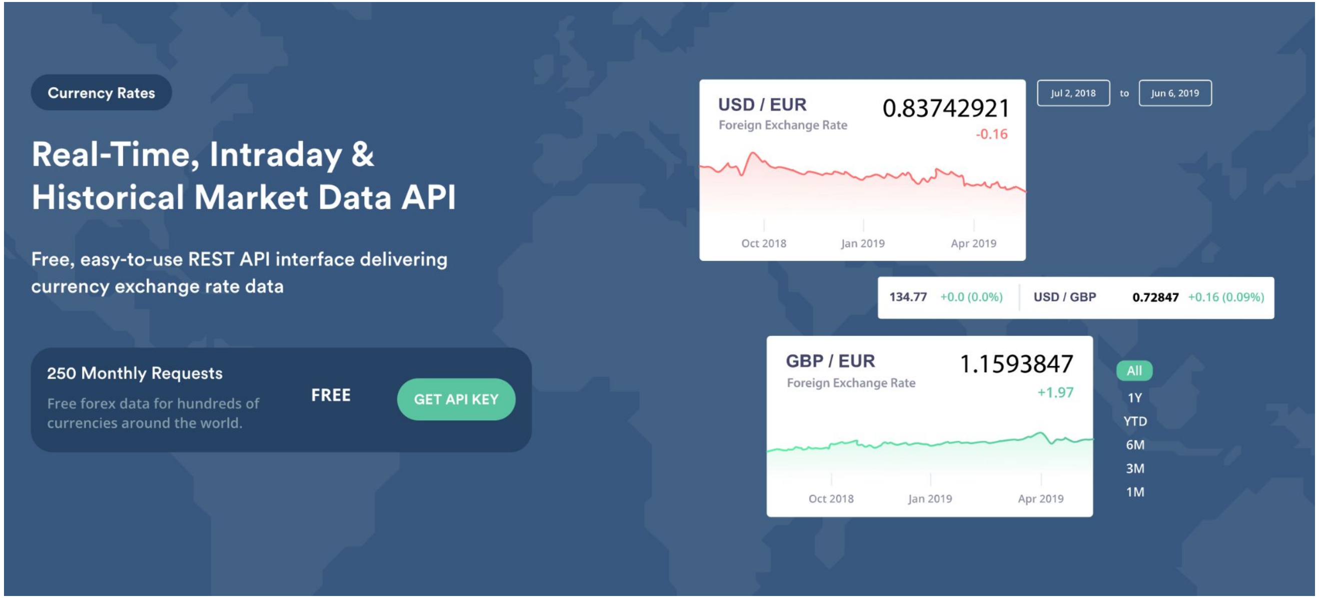
OLAP: Online Analytical Processing

ELT Example



NeueFische Data Stack

JSON API Example



Currency Rates

Real-Time, Intraday & Historical Market Data API

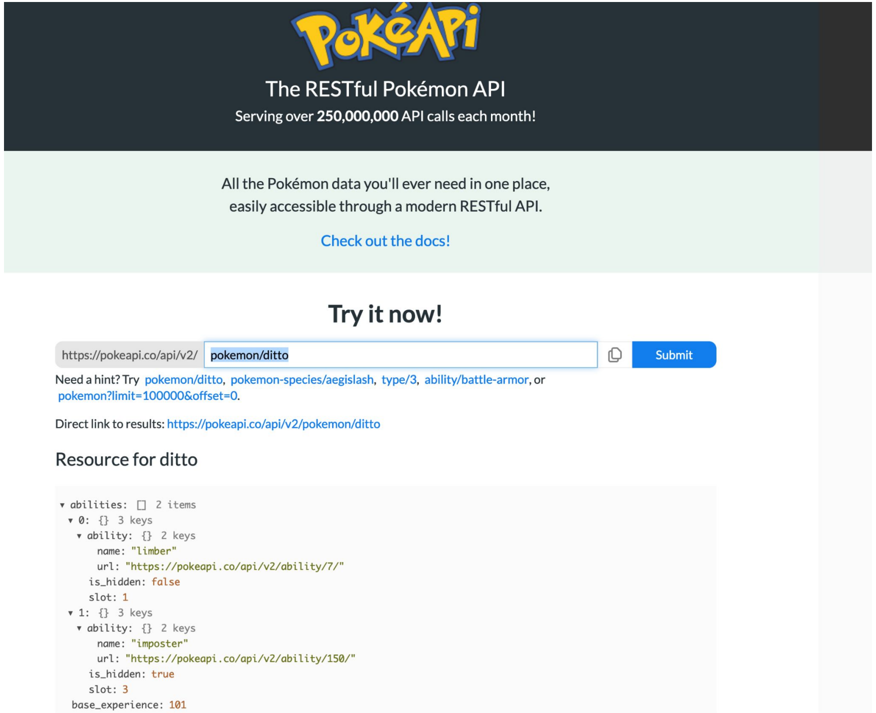
Free, easy-to-use REST API interface delivering currency exchange rate data

250 Monthly Requests

Free forex data for hundreds of currencies around the world.

```
{
   "success": true,
   "timestamp":
1519296206,
   "base": "EUR",
   "date": "2021-03-17",
   "rates": {
       "AUD": 1.566015,
       "CAD": 1.560132,
       "CHF": 1.154727,
       "CNY": 7.827874,
       "GBP": 0.882047,
       "JPY": 132.360679,
       "USD": 1.23396
   }
}
```

JSON API Example



The RESTful Pokémon API

Serving over **250,000,000** API calls each month!

All the Pokémon data you'll ever need in one place,
easily accessible through a modern RESTful API.

Check out the docs!

Try it now!

https://pokeapi.co/api/v2/ pokemon/ditto

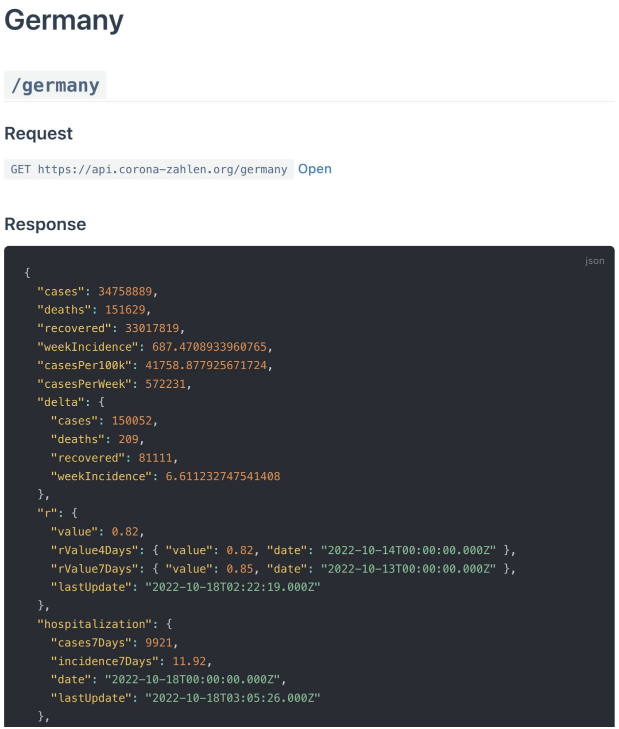
Need a hint? Try pokemon/ditto, pokemon-species/aegislash, type/3, ability/battle-armor, or pokemon?limit=100000&offset=0.

Direct link to results: https://pokeapi.co/api/v2/pokemon/ditto

Resource for ditto

```
▾ abilities: [] 2 items
 ▾ 0: {} 3 keys
  ▾ ability: {} 2 keys
     name: "limber"
     url: "https://pokeapi.co/api/v2/ability/7/"
    is_hidden: false
    slot: 1
 ▾ 1: {} 3 keys
  ▾ ability: {} 2 keys
     name: "imposter"
     url: "https://pokeapi.co/api/v2/ability/150/"
    is_hidden: true
    slot: 3
  base_experience: 101
```

JSON API Example



Germany

`/germany`

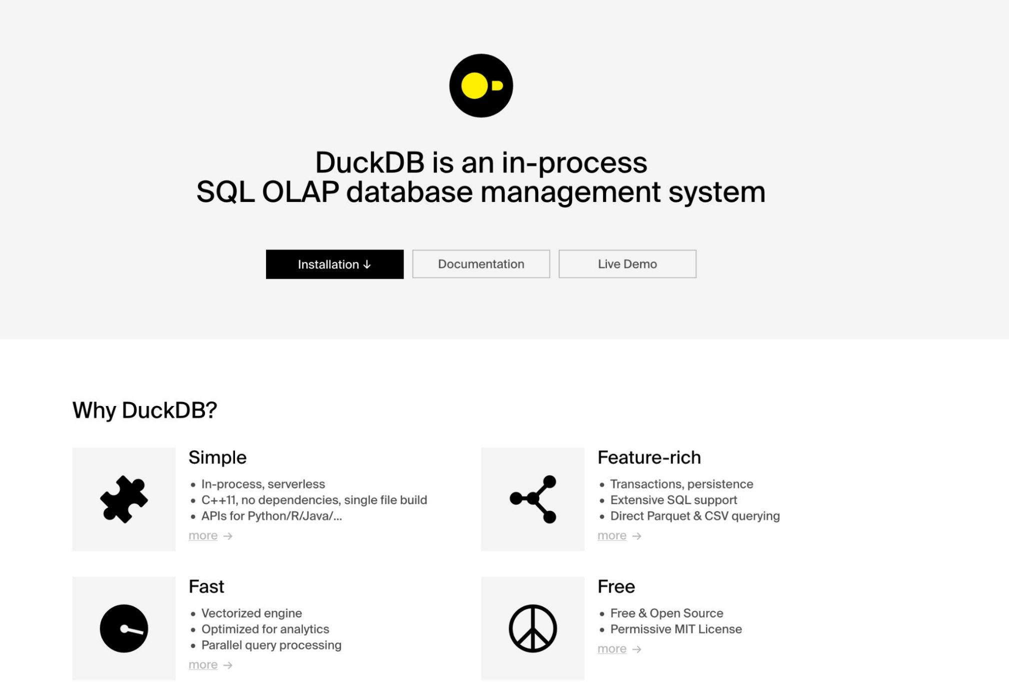
Request

`GET https://api.corona-zahlen.org/germany` Open

Response

```json
{
  "cases": 34758889,
  "deaths": 151629,
  "recovered": 33017819,
  "weekIncidence": 687.4708933960765,
  "casesPer100k": 41758.877925671724,
  "casesPerWeek": 572231,
  "delta": {
    "cases": 150052,
    "deaths": 209,
    "recovered": 81111,
    "weekIncidence": 6.611232747541408
  },
  "r": {
    "value": 0.82,
    "rValue4Days": { "value": 0.82, "date": "2022-10-14T00:00:00.000Z" },
    "rValue7Days": { "value": 0.85, "date": "2022-10-13T00:00:00.000Z" },
    "lastUpdate": "2022-10-18T02:22:19.000Z"
  },
  "hospitalization": {
    "cases7Days": 9921,
    "incidence7Days": 11.92,
    "date": "2022-10-18T00:00:00.000Z",
    "lastUpdate": "2022-10-18T03:05:26.000Z"
  },
```

Your Data Puddle



DuckDB is an in-process
SQL OLAP database management system

Why DuckDB?

Simple

- In-process, serverless
- C++11, no dependencies, single file build
- APIs for Python/R/Java/...

more →

Feature-rich

- Transactions, persistence
- Extensive SQL support
- Direct Parquet & CSV querying

more →

Fast

- Vectorized engine
- Optimized for analytics
- Parallel query processing

more →

Free

- Free & Open Source
- Permissive MIT License

more →