n8n

Bei n8n handelt es sich um ein Tool zur Workflow Automatisierung. Es gibt diverse Trigger (z. B. Zeit, Mail erhalten, Postgres Change) und Integrationen (MS SQL, MySQL, Postgres, M365, Mongo, Redis, ...) um Aufgaben automatisiert zu erledigen. Daten sammeln, verarbeiten und wegschreiben.

- Installation
- Beispiel Workflow MS SQL zu Mongo

Installation

8n8 wird als Webseite bereitgestellt und auf einem Server gehostet oder aus der Cloud direkt im Browser ausgeführt.

Hier wird die Docker Installation beschrieben. Dabei wird PostgreSQL als Datenbank verwendet. Es kann auch statt einer PostgreSQL die integrierte SQLite verwendet werden, jedoch wird aus Perfomance Gründen PostgreSQL empfohlen.

> Alle benötigten Dateien sind auch im GitHub von n8n verfügbar: n8n GitHub - Config Files

Folgend die Docker Compose Konfiguration. (`docker-compose.yml`)

> Bitte die Pfade bei den beiden Volumes für die Daten anpassen.

```
services:
  n8n-db:
    image: postgres
    restart: always
    hostname: n8n-db
    container_name: n8n-db
    environment:
      - POSTGRES_USER
      - POSTGRES_PASSWORD
      - POSTGRES_DB
      - POSTGRES_NON_ROOT_USER
      - POSTGRES_NON_ROOT_PASSWORD
    volumes:
      - /path/to/n8n/data-db:/var/lib/postgresql/data
      - ./init-data.sh:/docker-entrypoint-initdb.d/init-data.sh
    healthcheck:
      test: ['CMD-SHELL', 'pg_isready -h localhost -U ${POSTGRES_USER} -d ${POSTGRES_DB}']
      interval: 5s
      timeout: 5s
      retries: 10
  n8n:
    image: docker.n8n.io/n8nio/n8n
```

```
    restart: always
    hostname: n8n
    container_name: n8n
    ports:
      - 5678:5678
    volumes:
      - /path/to/n8n/data-n8n:/home/node/.n8n
    environment:
      - DB_TYPE=postgresdb
      - DB_POSTGRESDB_HOST=n8n-db
      - DB_POSTGRESDB_PORT=5432
      - DB_POSTGRESDB_DATABASE=${POSTGRES_DB}
      - DB_POSTGRESDB_USER=${POSTGRES_NON_ROOT_USER}
      - DB_POSTGRESDB_PASSWORD=${POSTGRES_NON_ROOT_PASSWORD}
    depends_on:
      n8n-db:
        condition: service_healthy
```

Zusätzlich wird noch eine `.env` Konfiguration für die Benutzer und Passwörter benötigt, welche wie folgt aussieht. Diese Datei bitte in den selben Ordner wie die Compose Konfiguration platzieren.

> Bitte die Passwörter durch eigene und sichere Passwörter austauschen.

```
POSTGRES_USER=root
POSTGRES_PASSWORD=Ganz-Super-Dolles-Passwort
POSTGRES_DB=n8n
POSTGRES_NON_ROOT_USER=n8n
POSTGRES_NON_ROOT_PASSWORD=Weniger-Dolles-Passwort
```

Außerdem ist noch ein Shell-Skript hinzuzufügen, welches ebenfalls im selben Ordner liegen muss. `(init-data.sh)`

```
#!/bin/bash
set -e;


if [ -n "${POSTGRES_NON_ROOT_USER:-}" ] && [ -n "${POSTGRES_NON_ROOT_PASSWORD:-}" ]; then
	psql -v ON_ERROR_STOP=1 --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" <<-EOSQL
		CREATE USER ${POSTGRES_NON_ROOT_USER} WITH PASSWORD '${POSTGRES_NON_ROOT_PASSWORD}';
		GRANT ALL PRIVILEGES ON DATABASE ${POSTGRES_DB} TO ${POSTGRES_NON_ROOT_USER};
```

```
    GRANT CREATE ON SCHEMA public TO ${POSTGRES_NON_ROOT_USER};
    EOSQL
else
    echo "SETUP INFO: No Environment variables given!"
fi
```

Damit wären alle notwendigen Konfigurationen abgeschlossen und das Setup kann gestartet werden.

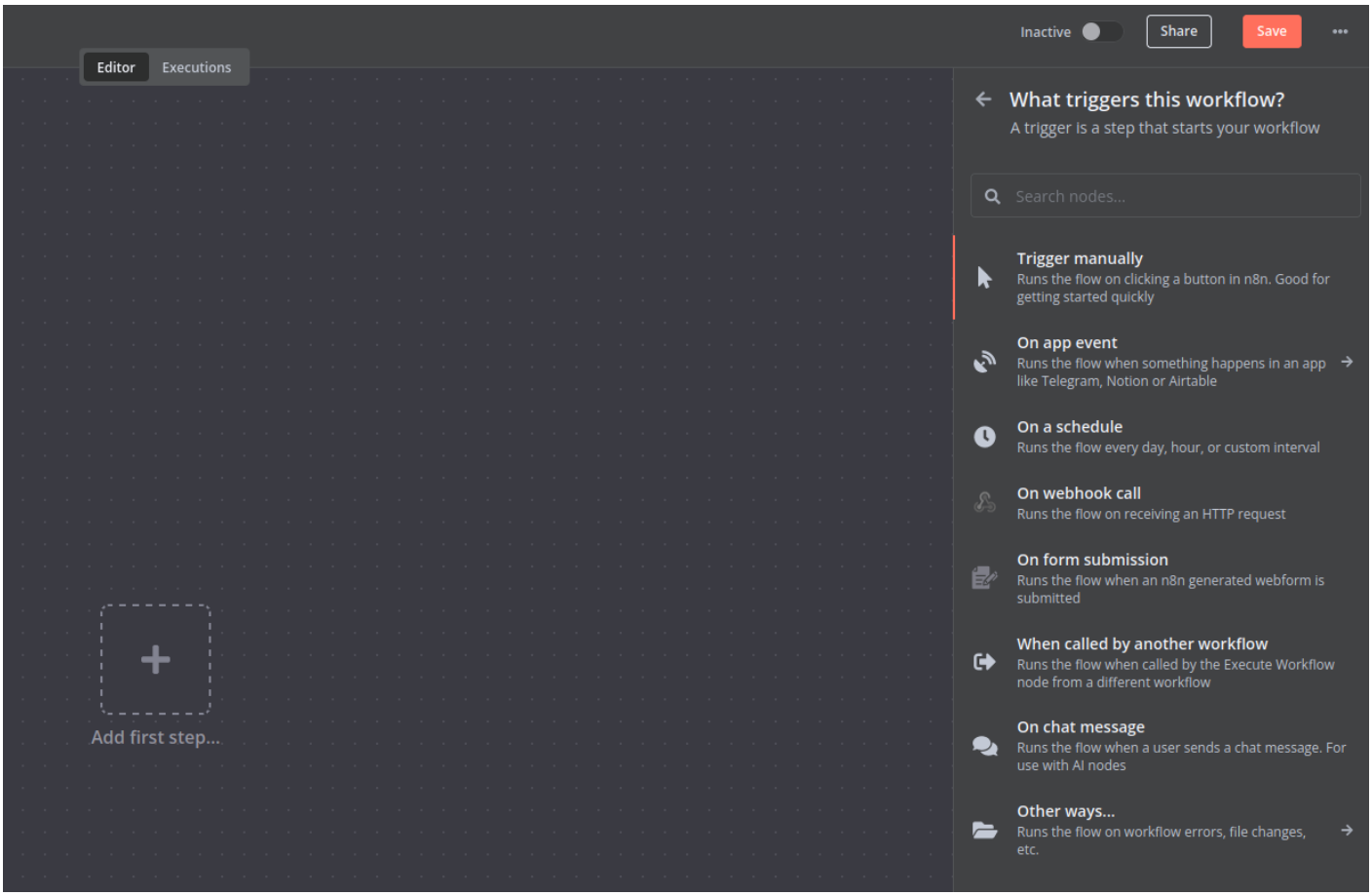
Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegeben Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Nach einigen Sekunden sollte n8n einsatzbereit sein. Die Seite von n8n ist über http://localhost:5678 erreichbar.

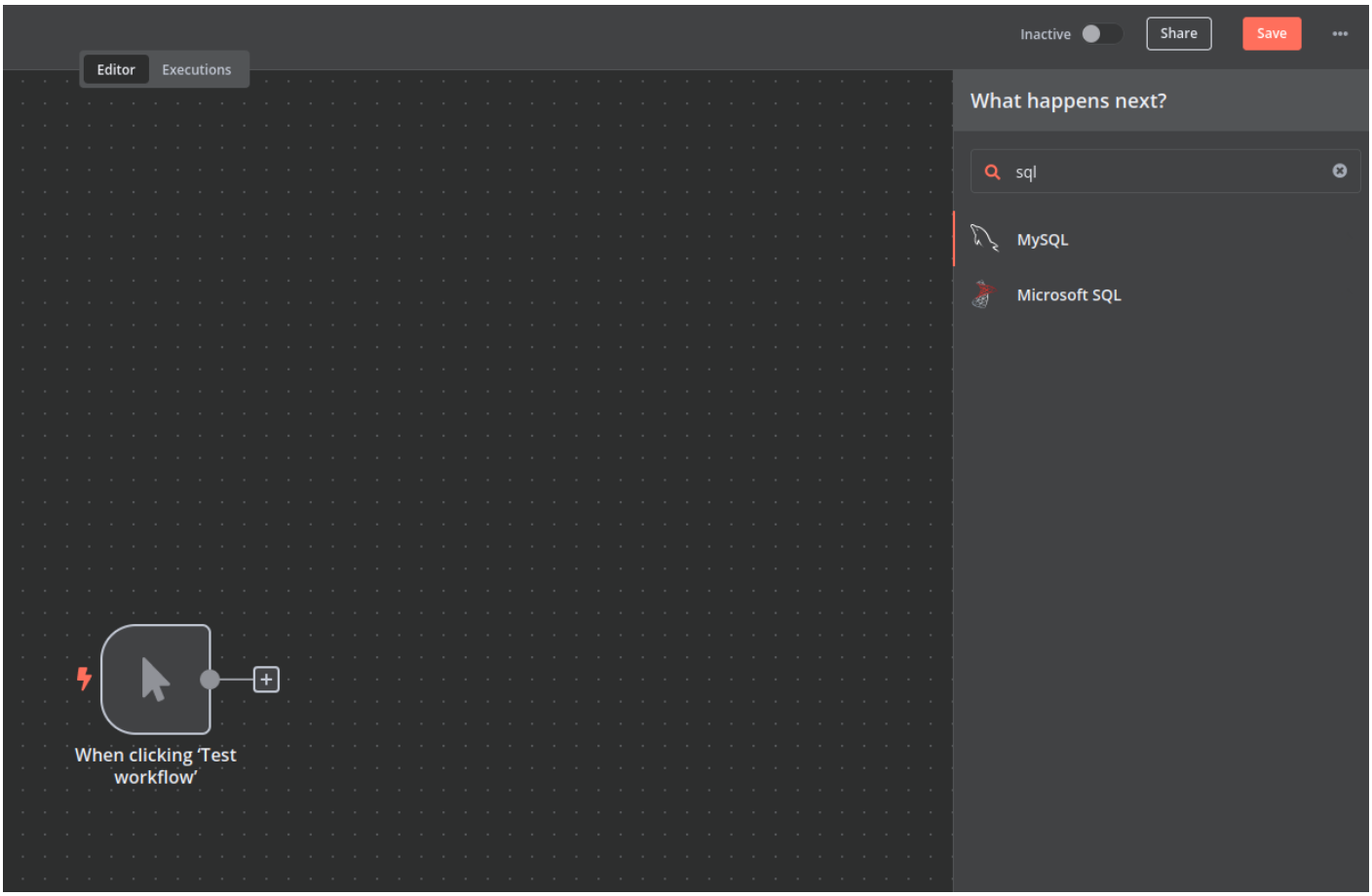
Beispiel Workflow MS SQL zu Mongo

In diesem n8n Beispiel Workflow werden 2 SELECTs auf einem Microsoft SQL Server ausgeführt, die Daten zusammengeführt und dann in eine Mongo Datenbank geschrieben.

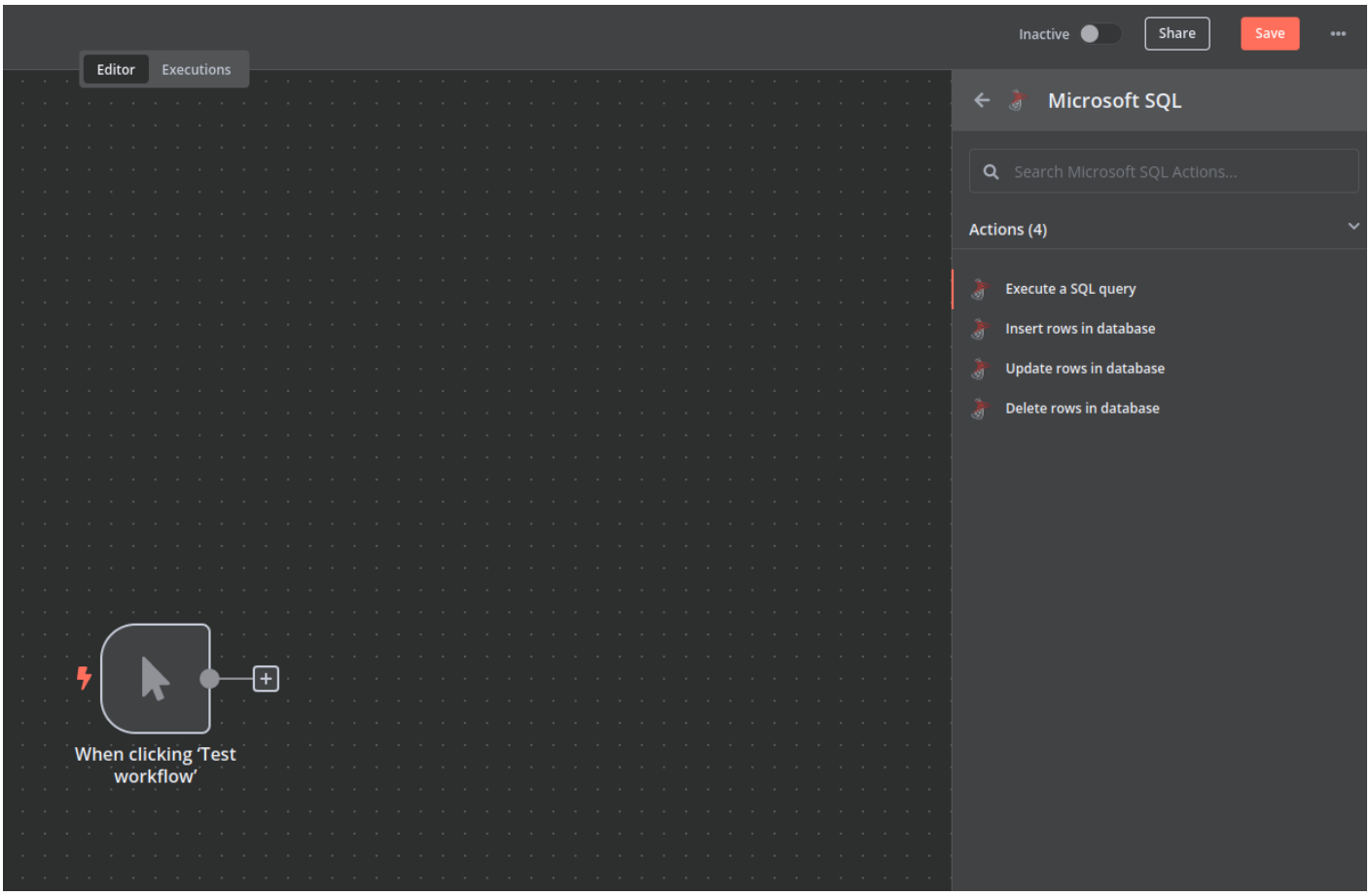
Zuerst wird ein Trigger gewählt, in diesem Fall wähle ich *Trigger manuelly*, da es einfach auf Knopfdruck zum Testen ausgeführt werden soll.



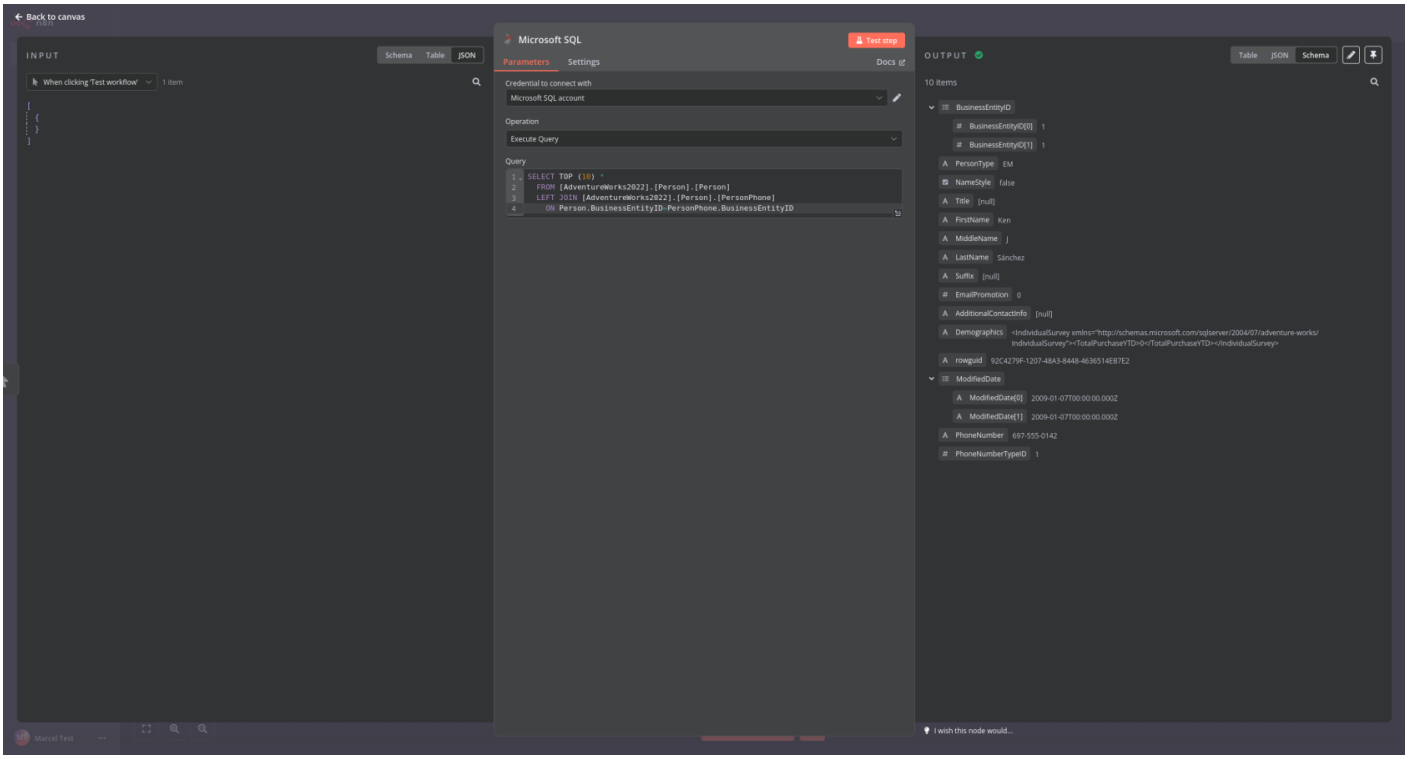
In diesem Test sollen aus 2 SQL Abfragen Daten gesammelt und zusammengeführt werden. Zuerst wird eine Action für das 1. SQL ausgewählt, dazu auf das kleine Plus neben dem Trigger klicken und nach SQL suchen. Für diesen Test wird Microsoft SQL gewählt, da eine Verbindung zu einem Microsoft SQL Server aufgebaut werden soll. Dadurch wird die Integration gewählt und es werden die verfügbaren Actions angezeigt.



Aus den Actions wird *Execute a SQL query* gewählt.

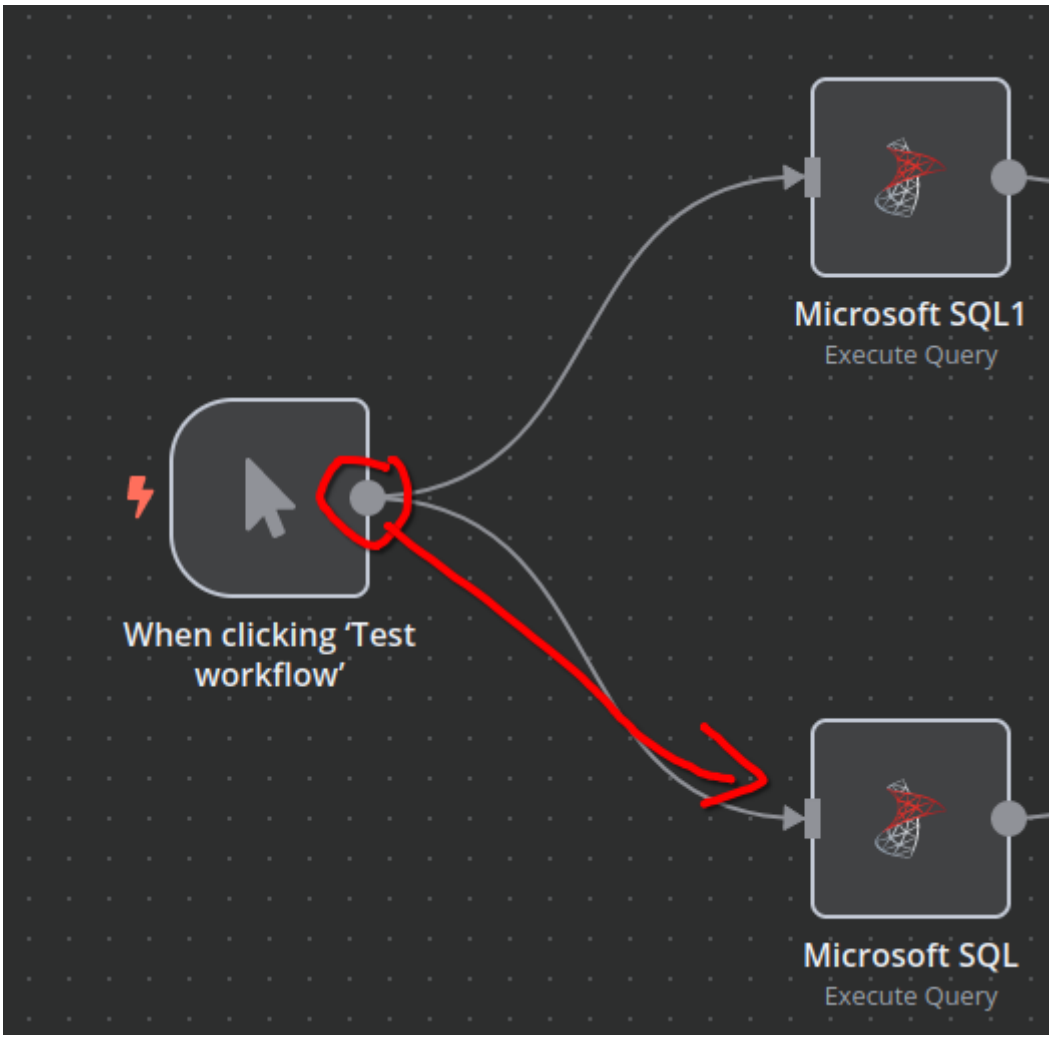


Anschließend wird eine neue Action zum Workflow hinzugefügt und es öffnet sich direkt die Konfiguration für diese Action. Im folgenden Screenshot wurden schon Credentials eingegeben (bei einem anderen Workflow), sodass diese direkt ausgewählt werden können. Sollten noch keine Credentials vorhanden sein, erscheint direkt eine Abfrage, wo Server, Benutzer, Passwort usw. einzugeben sind. Außerdem wurde folgend schon ein SQL in das Query Fenster eingetragen und auf Test step geklickt.

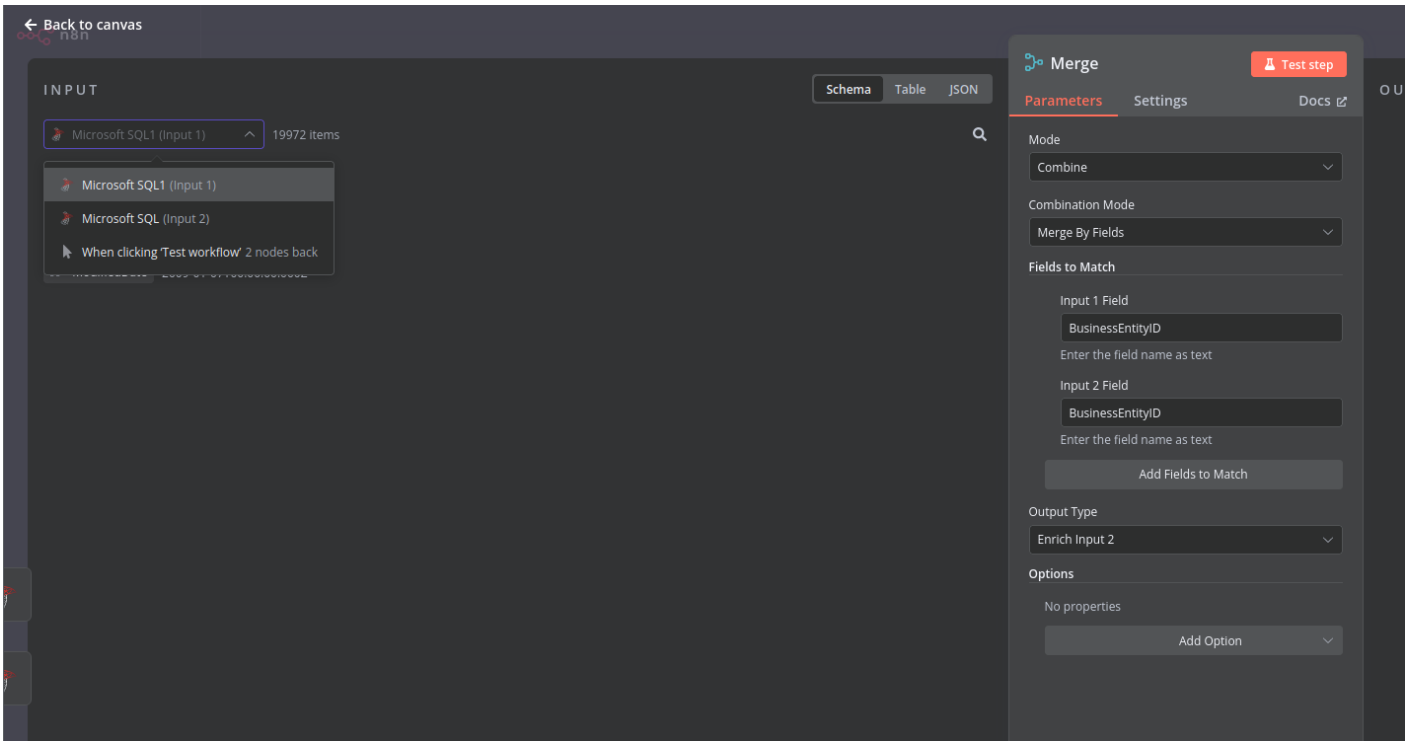


Nachdem alles geklappt hat, wird nun oben links auf Back to canvas geklickt, um zur Workflow Übersicht zurückzukehren.

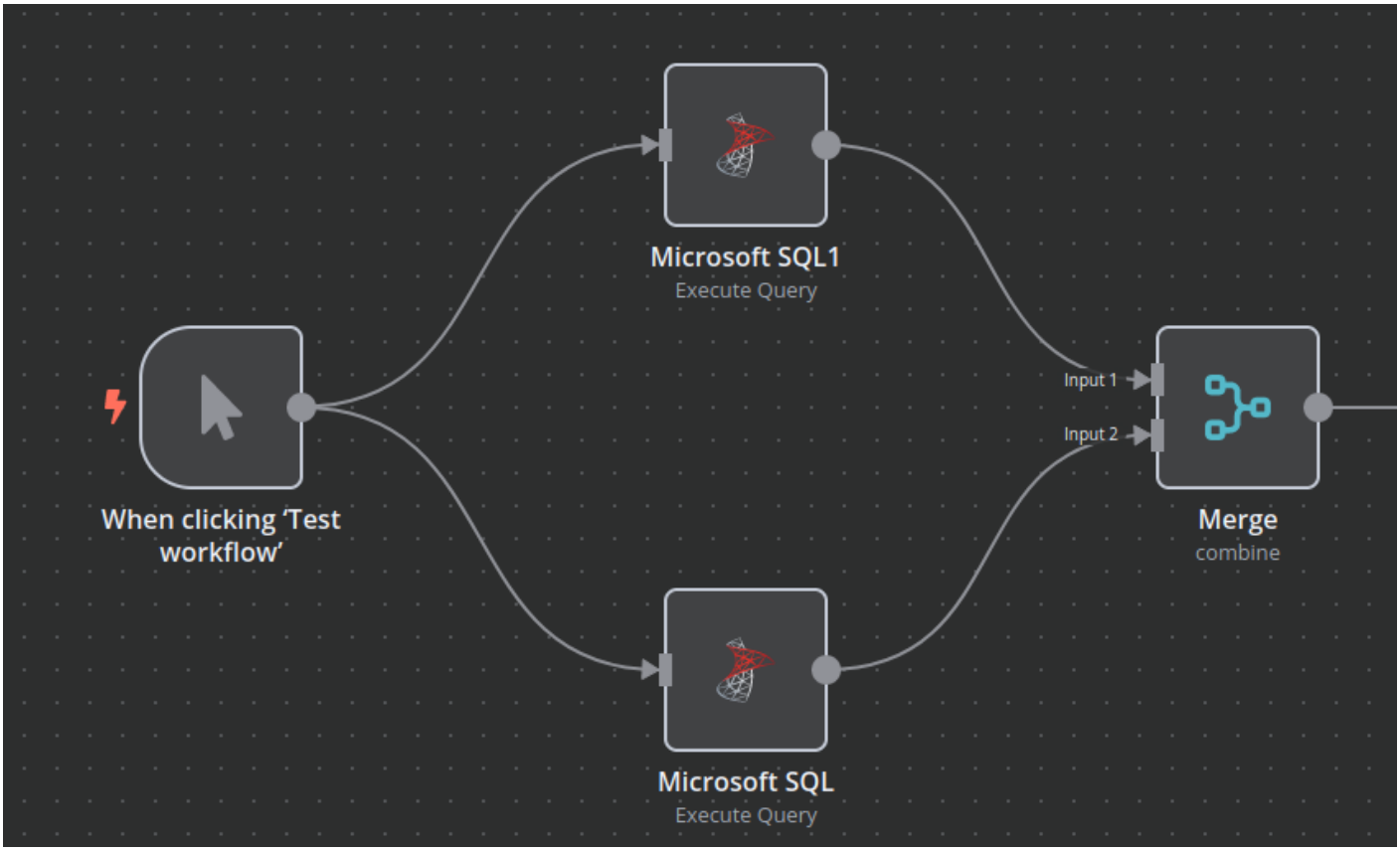
Um das 2. SQL Statement parallel zum 1. ausführen zu lassen, wird der kleine Punkt beim Trigger angeklickt und gedrückt gehalten. Dann wird der Mauszeiger weg bewegt, an eine gewünschte Position und losgelassen. Dadurch wird eine weitere parellele Aktion ermöglicht und es gehen nun 2 Pfeile aus dem Trigger raus. Dadurch wird wieder das Fenster zum Durchsuchen der Integrationen/Aktionen geöffnet und es wird wieder Microsoft SQL ausgewählt. Dort können dann die selben Credentials verwendet werden (wenn es der gleiche SQL Server ist) oder neue Credentials für einen neuen Server ausgewählt werden.



Da nun 2 Pipelines laufen, müssen diese mit einem *Merge* zusammengeführt werden. Hier wird nun der Mode Combine ausgewählt, um die beiden Tabellen zu verbinden (Join). Zusätzlich sind dann die Schlüssel auszuwählen.

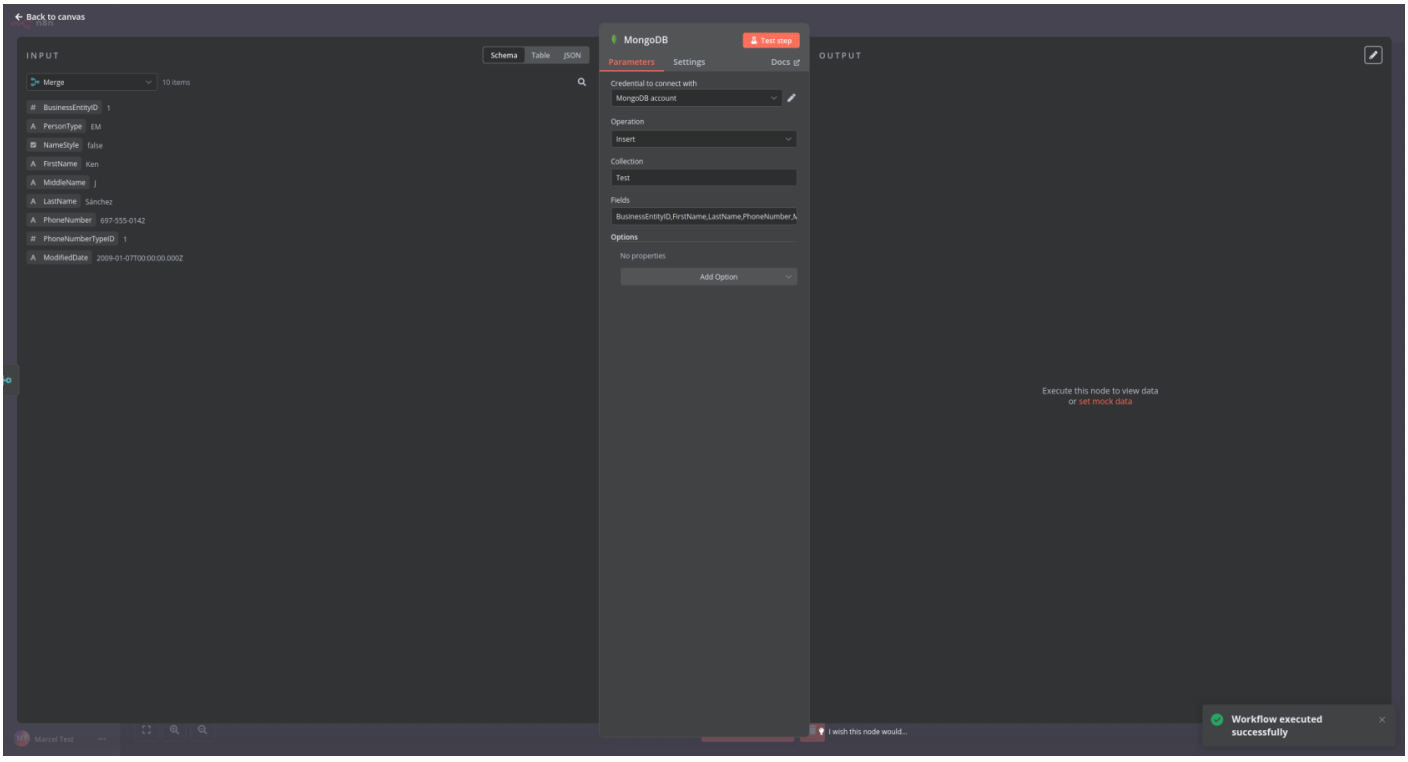


Der Workflow sieht nun wie folgt aus.



Nun kann auf das Plus neben Merge geklickt werden, um eine Aktion für die Mongo Datenbank hinzuzufügen.

> Wer jetzt auf *Test step* klickt, schreibt direkt in die Mongo!



Der finale Workflow sieht nun wie folgt aus.

