# DeepDB Project README

## Table of Contents

## Prerequisites

- **Operating System:** Linux/MacOS/Windows (ensure PostgreSQL and Python are supported)
- **Python Version:** 3.10
- **PostgreSQL Version:** ≥ 13
- **pgAdmin:** For database table management

## Installation

1. **Install Python 3.10:** Refer to the official Python website to download and install Python 3.10: `https://www.python.org/downloads/`
2. **Install Required Dependencies:**

```
pip install -r requirements.txt
```

## Database Setup

1. **Create a New Database in PostgreSQL:** Use `pgAdmin` or `psql` to create a database (e.g., `flights_db`).
2. **Create the `flights` Table:** Use the following SQL command in `pgAdmin` to create the table structure for the dataset:

```
CREATE TABLE flights (
    year INT,
    month INT,
    day INT,
    day_of_week INT,
    airline VARCHAR,
        flight_number INT,
        tail_number VARCHAR,
        origin_airport VARCHAR,
        destination_airport VARCHAR,
        scheduled_departure INT,
        departure_time INT,
        departure_delay FLOAT,
        taxi_out FLOAT,
        wheels_off INT,
        scheduled_time INT,
        elapsed_time INT,
        air_time INT,
        distance INT,
        wheels_on INT,
        taxi_in FLOAT,
        scheduled_arrival INT,
        arrival_time FLOAT,
        arrival_delay FLOAT,
        diverted INT,
        cancelled INT,
        cancellation_reason CHAR,
        air_system_delay FLOAT,
        security_delay FLOAT,
        airline_delay FLOAT,
        late_aircraft_delay FLOAT,
        weather_delay FLOAT
);
```

3. **Load Dataset into the Table:**
   - Open `pgAdmin`.
   - Right-click the `flights` table and select **Import/Export**.
   - In the Import dialog:
     - Select **CSV file** as the format.
     - Provide the path to your dataset file (e.g., `flights.csv`).
     - Map the columns to the dataset structure.
   - Click **Start** to load the data.

# Schema File Creation

Create a `schema.py` file in the appropriate directory, defining the schema based on your PostgreSQL database structure. This is essential for generating HDF files.

# Commands to Run the Project

## 1. Generate HDF Files

Run the following command to create HDF files from the loaded PostgreSQL database:

```
python3 maqp.py \
    —dataset flights \
    —generate_hdf \
    —hdf_path ./mqp–data/flights–benchmark/gen_hdf \
    —csv_path ./mqp–data/flights–benchmark/flights.csv
```

### 2. Generate Ensembles

Use the command below to generate SPN ensembles:

```
python3 maqp.py \
    --dataset flights \
    --generate_ensemble \
    --ensemble_strategy rdc_based \
    --hdf_path ./mqp-data/flights-benchmark/gen_hdf \
    --ensemble_path ./mqp-data/flights-benchmark/spn_ensembles/ensemble_join_5_budget_1.
    --pairwise_rdc_path ./mqp-data/flights-benchmark/spn_ensembles/pairwise_rdc_file.pkl
    --post_sampling_factor 1 1 1 1 \
    --ensemble_budget_factor 1 \
    --ensemble_max_no_joins 5 \
    --rdc_threshold 0.15 \
    --bloom_filters \
    --samples_rdc_ensemble_tests 10000
```

### 3. Evaluate Results

Run the following command to evaluate queries:

```
python3 maqp.py --results
```

## Running Queries in PostgreSQL

To run a query directly in PostgreSQL:

1. Open `pgAdmin` or connect via `psql`.
2. Run your SQL query. For example:

```
SELECT origin_airport, destination_airport, AVG(departure_delay)
FROM flights
WHERE month = 1 AND distance > 1000
GROUP BY origin_airport, destination_airport
LIMIT 100;
```

This will compute the average departure delay for flights in January with distances greater than 1000 miles.