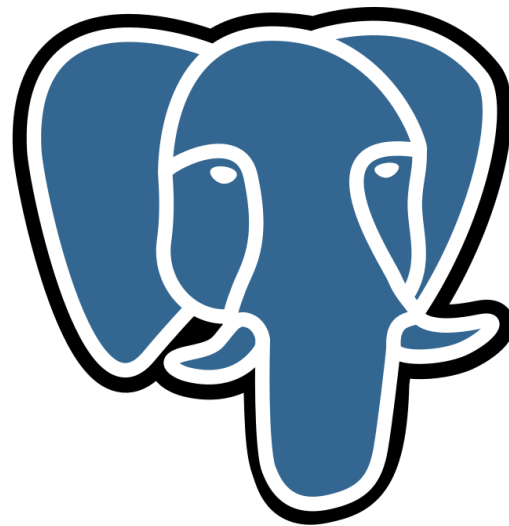


6

PostgreSQL

- PostgreSQL ist eine der beliebtesten relationalen Datenbanksysteme. Es ist open-source, performant und besitzt viele Features
- PostgreSQL gibt es seit über 30 Jahre (1989), seit 1997 open-source
- Kompatibilität mit SQL-Standard sehr hoch (170 von 179 Features des SQL:2016 Standards)
- viele Add-Ons verfügbar, die den Funktionsumfang nochmal erweitern. Z.B. PostGIS zur Verwaltung von Geodaten



- PGAdmin ist ein webbasiertes Tool, um mit der Datenbank zu kommunizieren
- PGAdmin wird mit PostgreSQL mitgeliefert
- Es können Benutzer und Prozesse verwaltet werden. Für das Arbeiten mit Daten ist es nicht so komfortabel wie DBeaver.



- PostgreSQL läuft meist auf einem Server, aber kann auch lokal installiert werden
- Herunterladen (www.postgresql.org) von Version 14.X (Windows x86-64) und Installation (siehe Anleitung)

Windows installers

Interactive installer by EDB

Download the installer certified by EDB for all supported PostgreSQL versions.

Note! This installer is hosted by EDB and not on the PostgreSQL community servers. If you have issues with the website it's hosted webmaster@enterprisedb.com.

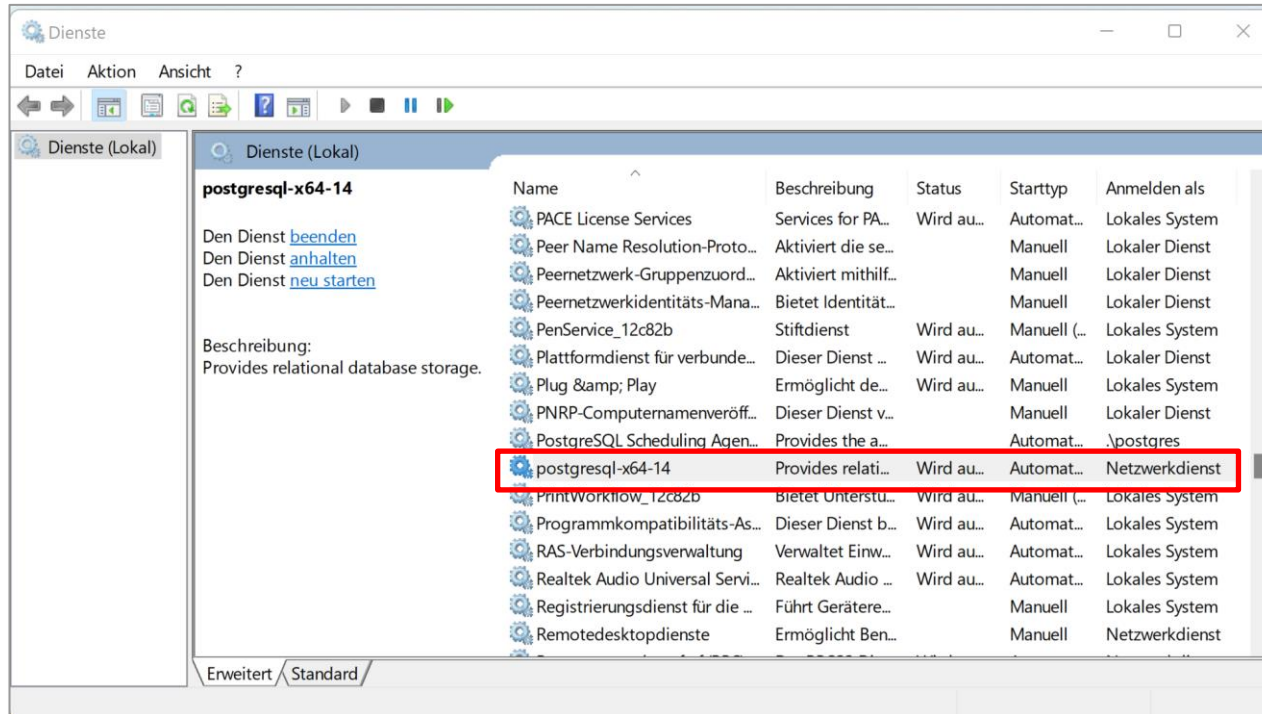
This installer includes the PostgreSQL server, pgAdmin; a graphical tool for managing and developing your databases, and StackBuilder to download and install additional PostgreSQL tools and drivers. Stackbuilder includes management, integration, migration, replication and backup tools.

This installer can run in graphical or silent install modes.

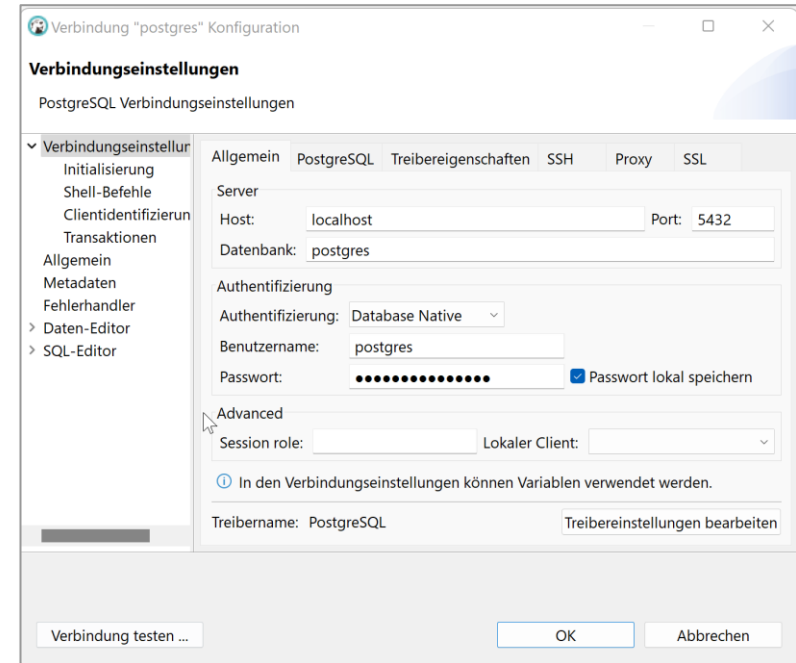
The installer is designed to be a straightforward, fast way to get up and running with PostgreSQL on Windows.

Advanced users can also download a **zip archive** of the binaries, without the installer. This download is intended for users who wish to use the application installer.

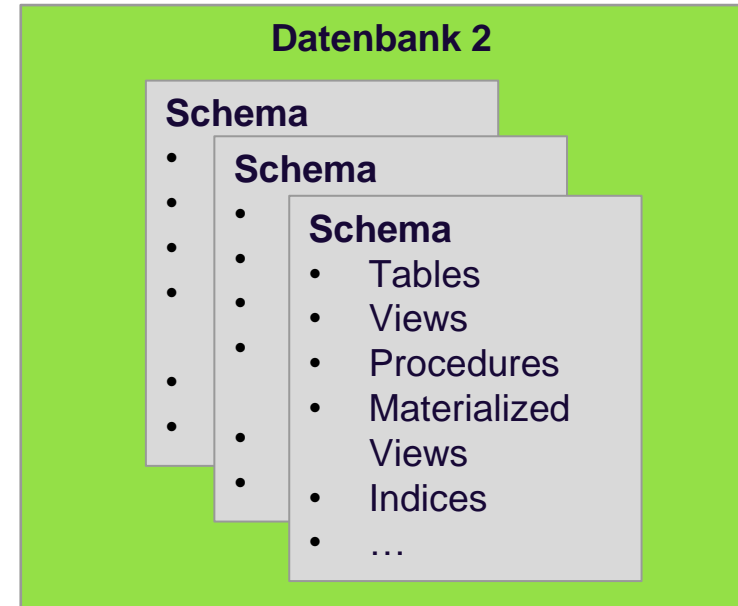
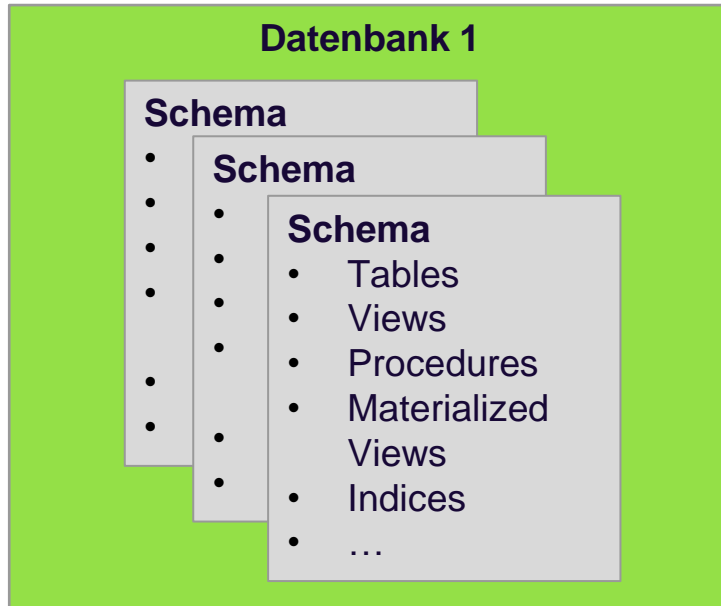
- PostgreSQL läuft als Windows Dienst mit Namen postgres. Dieser kann über die Applikation Dienste gesteuert werden



- Verbindung hinzufügen > PostgreSQL
 - Host: localhost
 - Port: 5432
 - Datenbank: postgres
 - Benutzername: postgres
 - Passwort



Der PostgreSQL-Server verwaltet mehrere Datenbanken. Diese enthalten Schemata, welche Tabellen und weitere Objekte enthalten.



PostgreSQL ist nahe am SQL-Standard, allerdings gibt es auch viele Funktionen, die nicht durch den SQL-Standard abgedeckt sind.

- In PostgreSQL (und fast allen anderen SQL-Dialekten) werden **'Strings'** mit einfachen Anführungszeichen geschrieben. SQLite versteht auch **"doppelte"**.
- Konvertierung von Datentypen mit `spalte::datentyp`. Es funktioniert aber auch `cast(spalte as datentyp)`, welches dem SQL-Standard entspricht.
- Besonderheit in SQLite: Es dürfen in einer Abfrage Spalten genannt werden, die weder in group by auftauchen noch eine Aggregationsfunktion haben. Es wird dann der erste Wert (gemäß der normalen Tabellenordnung) genommen. Den ersten/letzten Wert in der Gruppe zu bekommen, ist in PostgreSQL nicht ohne weiteres möglich

- Fast alle SQL-Dialekte unterscheiden nicht zwischen Groß- und Kleinschreibung. In PostgreSQL werden allerdings Spalten mit Großbuchstaben nicht "gefunden", z.B. circuitID. Um diese Spalten anzusprechen, muss der Spaltenname in doppelten Anführungszeichen stehen. **Generell sollten in PostgreSQL Spaltennamen nur Kleinbuchstaben enthalten.**

```
SELECT  
  "raceId" as race_id  
FROM results
```

- In PostgreSQL gibt es die Funktion `string_agg(spalte, trennzeichen)`, welche sich wie `group_concat` verhält

- In PostgreSQL gibt es für Datum- und Zeitangaben die Datentypen:
 - `date`
 - `timestamp (without timezone)`
 - `timestampz` (oder `timestamp with timezone`)
 - `time (without timezone)`
 - `time with timezone`
 - `interval`
- Umwandlung eines Strings in einen der Typen mittels doppeltem Doppelpunkt oder `cast`
 - `'2022-08-17'::date`
 - `'2022-08-17 12:23'::timestamp`
 - `'2022-08-17 12:00:22'::timestampz`
 - `cast('2022-08-17' as date)`
 - `cast('2022-08-17 12:23' as timestamp)`

- Umwandlung eines Strings in date mittels `to_date(datumstring, format)` oder in timestamptz mit `to_timestamp(datumstring, format)`

```
to_date('17.08.2022', 'DD.MM.YYYY')
```

```
to_timestamp('17.08.2022 10:30', 'DD.MM.YYYY HH24:MI') as mit_zeitzone,
```

```
to_timestamp('17.08.2022 10:30', 'DD.MM.YYYY HH24:MI')::timestamp as ohne_zeitzone
```

- Umwandlung eines Datumstyps in einen String mittels `to_char(datum, format)`

```
to_char(now(), 'DD.MM.YYYY')
```

```
to_char(now(), 'DD.MM.YYYY HH24:MI:SS')
```

- Alle Format-Möglichkeiten unter <https://www.postgresql.org/docs/current/functions-formatting.html>

- Extraktion einzelner Datumsteile mittels `date_part('datumsteil', datum)` oder mittels `extract(datumsteil from datum)`. Der Extract-Syntax entspricht dem SQL-Standard, `date_part` ist eine PostgreSQL-spezifische Funktion.

```
extract(day from now()),  
extract(isodow from '2022-08-17'::date),  
date_part('day', now()),  
date_part('isodow', '2022-08-17'::date)
```

- Alle Datumsteile unter <https://www.postgresql.org/docs/current/functions-datetime.html#FUNCTIONS-DATETIME-EXTRACT>

- Es gibt verschiedene Möglichkeiten, ein Datum zu verändern

```
'2022-08-17'::date + 1,  
'2022-08-17'::date + '1 hour'::interval,  
'2022-08-17'::timestamp + '03:00'::time,  
'2022-08-17'::date - '2022-01-01'::date,  
now() - '2 hours'::interval,  
now() - '2022-08-16 12:00:12.123'::timestamp
```
- Eine vollständige Liste der Datums- und Zeitstempel-Operatoren gibt es unter <https://www.postgresql.org/docs/current/functions-datetime.html>
- Die einzelnen Komponenten eines Intervalls können, wie bei date und timestamp, per to_char() oder mit extract() extrahiert werden. Die Anzahl Sekunden bekommt man mittels extract(epoch from intervall)