

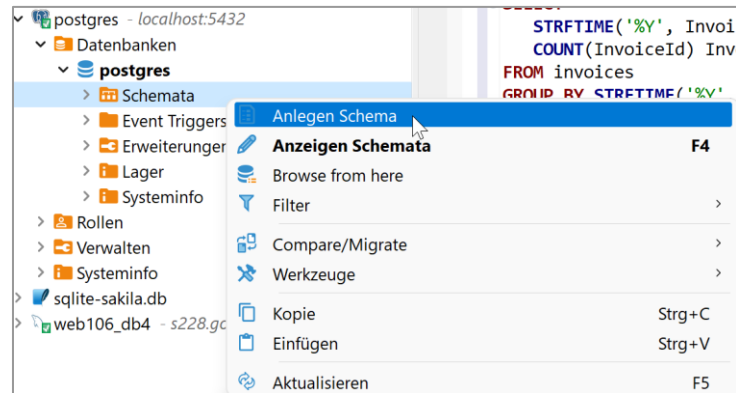
7

Tabellen erstellen und modifizieren

Schema erstellen

```
create schema schema_name;
```

```
create schema schema_name AUTHORIZATION  
benutzer;
```



Datentyp	Größe (Bytes)	Beschreibung
smallint	2	-32.768 bis +32.767
integer	4	-2.147.483.648 bis +2.147.483.647
bigint	8	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807
decimal / numeric	variabel	bis zu 131.072 Stellen vor dem Komma, bis zu 16.383 Stellen nach dem Komma
real / float	4	6 Stellen nach dem Komma
double precision	8	15 Stellen nach dem Komma
smallserial	2	1 bis 32.767, erhöht sich automatisch
serial	4	1 bis 2.147.483.647, erhöht sich automatisch
bigserial	8	1 bis 9.223.372.036.854.775.807, erhöht sich automatisch
money	8	-92.233.720.368.547.758,08 bis +92.233.720.368.547.758,07
boolean	1	true oder false

- **varchar(n)**: Variable Länge bis zu n Stellen
- **char(n)**: Fixe Länge mit Leerzeichen
- **text**: beliebige variable Länge

Datentyp	Größe (Bytes)	Beschreibung
timestamp [without time zone]	8	Zeitstempel ohne Zeitzone
timestamptz	8	Zeitstempel mit Zeitzone
date	4	Datum
time without time zone	8	Zeit
time with time zone	12	Zeit mit Zeitzone
interval	12	Zeitintervall

- **UUID:** Universally Unique Identifiers, z.B. 550e8400-e29b-41d4-a716-446655440000
- **JSON:** JSON-Objekte, z.B. {"Farbe":rot, "Alter":11}
- **Arrays:** Listen eines Typs, z.B. integer[], text[][]

- **eigene Typen**

```
CREATE TYPE week AS  
    ENUM('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun');
```

```
CREATE TYPE inventory_item AS (  
    name text,  
    supplier_id integer,  
    price numeric );
```

- Tabellen erzeugen:

```
CREATE TABLE produkte (  
    id serial primary key,  
    name varchar(100) not null,  
    preis numeric(5,2)  
);
```

- Tabelle löschen

```
drop table produkte;  
drop table if exists produkte;
```

- Tabelle leeren

```
truncate table produkte;
```

- temporäre Tabellen werden mit dem Schlüsselwort TEMPORARY oder TEMP nach create erzeugt. Diese werden am Ende einer Session automatisch gelöscht

- **primary key:** nur eindeutige, nicht-null Werte. Es darf nur einen primary key pro Tabelle geben
- **not null:** Es muss immer ein Wert vorhanden sein
- **DEFAULT xy:** Gibt der Spalte einen Defaultwert
- **GENERATED [BY DEFAULT] AS IDENTITY:** Variante von SERIAL, mit optionaler Angabe und Definition der Sequenz

```
color_id INT GENERATED BY DEFAULT AS IDENTITY  
(START WITH 10 INCREMENT BY 10)
```


- Tabellen umbenennen:

```
ALTER TABLE tabelle RENAME TO neue_tabelle;
```

- Spalte hinzufügen

```
ALTER TABLE tabelle ADD COLUMN ort text;
```

- Spalte umbenennen

```
ALTER TABLE tabelle RENAME COLUMN alte_spalte TO neue_spalte;
```

- Werte einfügen:

```
insert into produkte values (Wert1, Wert2, Wert3);  
insert into produkte (Spalte2) values (Wert2);
```

- Abfrage hinzufügen:

```
insert into produkte (Spalte1, Spalte2, Spalte3)  
select * from tabelle;
```

- Zeilen löschen:

```
delete from tabelle  
where preis is null;
```

- Werte ändern:

```
update tabelle  
set spalte1 = wert1, spalte2 = wert2  
where preis > 10;
```

Views sind gespeicherte Abfragen auf Tabellen. Auf eine View kann wieder per select zugegriffen werden.

- View erzeugen:

```
create view produkte_view as
select
    produkt_id,
    produkt
from produkte;
```

- View löschen

```
drop view produkte_view;
drop view if exists produkte_view;
```