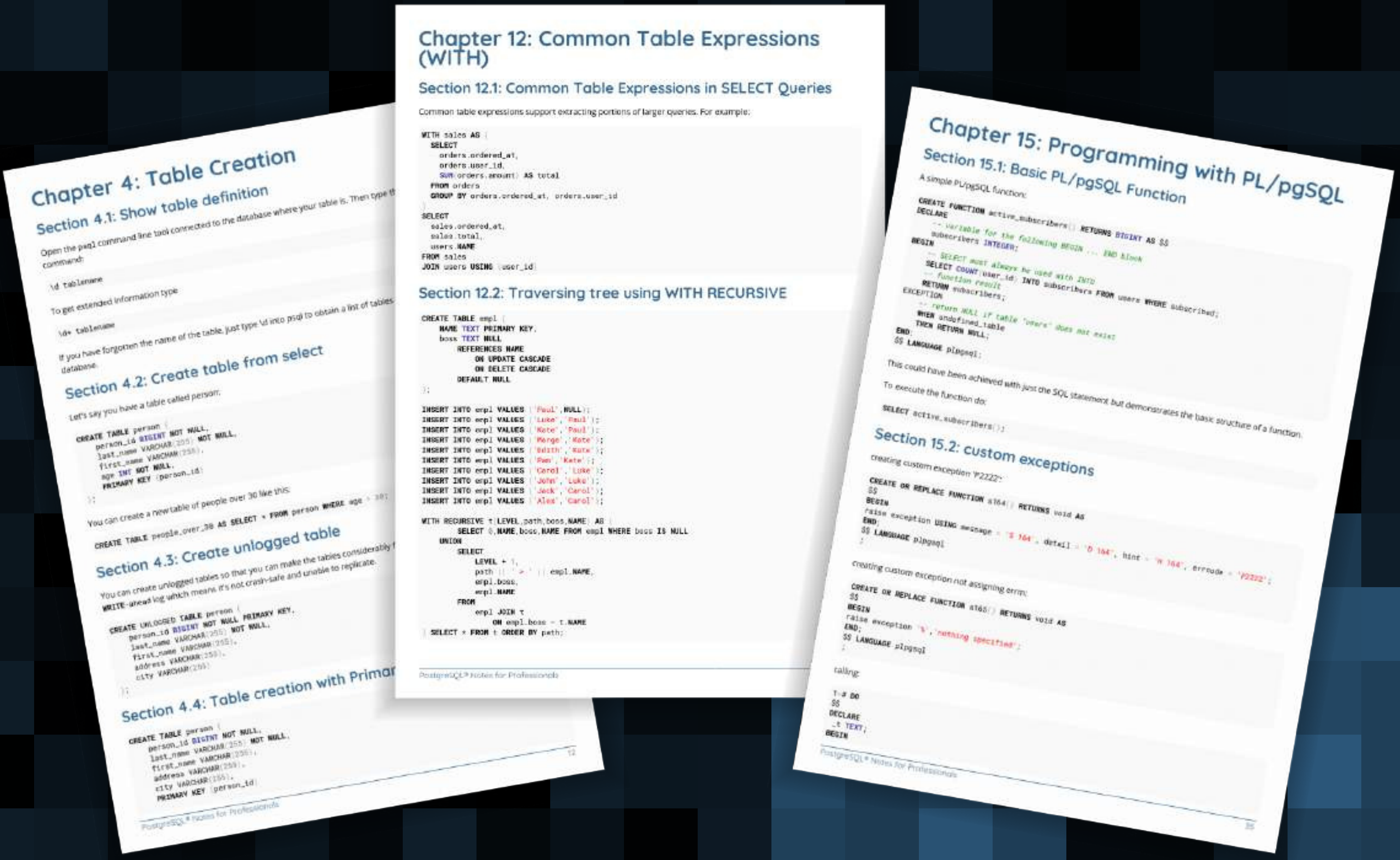
# PostgreSQL®

## Notes for Professionals



### Chapter 4: Table Creation

#### Section 4.1: Show table definition

Open the psql command line tool connected to the database where your table is. Then type th
command:

```
\d tablename
```

To get extended information type

```
\d+ tablename
```

If you have forgotten the name of the table, just type \d into psql to obtain a list of tables
database.

#### Section 4.2: Create table from select

Let's say you have a table called person:

```
CREATE TABLE person (
    person_id BIGINT NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    first_name VARCHAR(255),
    age INT NOT NULL,
    PRIMARY KEY (person_id)
);
```

You can create a new table of people over 30 like this:

```
CREATE TABLE people_over_30 AS SELECT * FROM person WHERE age > 30;
```

#### Section 4.3: Create unlogged table

You can create unlogged tables so that you can make the tables considerably f
WRITE-ahead log which means it's not crash-safe and unable to replicate.

```
CREATE UNLOGGED TABLE person (
    person_id BIGINT NOT NULL PRIMARY KEY,
    last_name VARCHAR(255) NOT NULL,
    first_name VARCHAR(255),
    address VARCHAR(255),
    city VARCHAR(255)
);
```

#### Section 4.4: Table creation with Primar

```
CREATE TABLE person (
    person_id BIGINT NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    first_name VARCHAR(255),
    address VARCHAR(255),
    city VARCHAR(255),
    PRIMARY KEY (person_id)
```

PostgreSQL® Notes for Professionals 12

### Chapter 12: Common Table Expressions (WITH)

#### Section 12.1: Common Table Expressions in SELECT Queries

Common table expressions support extracting portions of larger queries. For example:

```
WITH sales AS (
  SELECT
    orders.ordered_at,
    orders.user_id,
    SUM(orders.amount) AS total
  FROM orders
  GROUP BY orders.ordered_at, orders.user_id
)
SELECT
  sales.ordered_at,
  sales.total,
  users.NAME
FROM sales
JOIN users USING (user_id)
```

#### Section 12.2: Traversing tree using WITH RECURSIVE

```
CREATE TABLE empl (
    NAME TEXT PRIMARY KEY,
    boss TEXT NULL
        REFERENCES NAME
            ON UPDATE CASCADE
            ON DELETE CASCADE
        DEFAULT NULL
);

INSERT INTO empl VALUES ('Paul',NULL);
INSERT INTO empl VALUES ('Luke','Paul');
INSERT INTO empl VALUES ('Kate','Paul');
INSERT INTO empl VALUES ('Marge','Kate');
INSERT INTO empl VALUES ('Edith','Kate');
INSERT INTO empl VALUES ('Pam','Kate');
INSERT INTO empl VALUES ('Carol','Luke');
INSERT INTO empl VALUES ('John','Luke');
INSERT INTO empl VALUES ('Jack','Carol');
INSERT INTO empl VALUES ('Alex','Carol');

WITH RECURSIVE t(LEVEL,path,boss,NAME) AS (
        SELECT 0,NAME,boss,NAME FROM empl WHERE boss IS NULL
    UNION
        SELECT
            LEVEL + 1,
            path || ' > ' || empl.NAME,
            empl.boss,
            empl.NAME
        FROM
            empl JOIN t
                ON empl.boss = t.NAME
) SELECT * FROM t ORDER BY path;
```

PostgreSQL® Notes for Professionals

### Chapter 15: Programming with PL/pgSQL

#### Section 15.1: Basic PL/pgSQL Function

A simple PL/pgSQL function:

```
CREATE FUNCTION active_subscribers() RETURNS BIGINT AS $$
DECLARE
    -- variable for the following BEGIN ... END block
    subscribers INTEGER;
BEGIN
    -- SELECT must always be used with INTO
    SELECT COUNT(user_id) INTO subscribers FROM users WHERE subscribed;
    -- function result
    RETURN subscribers;
EXCEPTION
    -- return NULL if table "users" does not exist
    WHEN undefined_table
    THEN RETURN NULL;
END;
$$ LANGUAGE plpgsql;
```

This could have been achieved with just the SQL statement but demonstrates the basic structure of a function.

To execute the function do:

```
SELECT active_subscribers();
```

#### Section 15.2: custom exceptions

creating custom exception 'P2222':

```
CREATE OR REPLACE FUNCTION s164() RETURNS void AS
$$
BEGIN
raise exception USING message = 'S 164', detail = 'D 164', hint = 'H 164', errcode = 'P2222';
END;
$$ LANGUAGE plpgsql
;
```

creating custom exception not assigning errm:

```
CREATE OR REPLACE FUNCTION s165() RETURNS void AS
$$
BEGIN
raise exception '%','nothing specified';
END;
$$ LANGUAGE plpgsql
;
```

calling:

```
t=# do
$$
DECLARE
_t TEXT;
BEGIN
```

PostgreSQL® Notes for Professionals 55

**60+ pages**

of professional hints and tricks

GoalKicker.com
Free Programming Books

**Disclaimer**
This is an unofficial free book created for educational purposes and is not affiliated with official PostgreSQL® group(s) or company(s).
All trademarks and registered trademarks are the property of their respective owners

# Contents

About ... 1

Chapter 1: Getting started with PostgreSQL ... 2

- Section 1.1: Installing PostgreSQL on Windows ... 2
- Section 1.2: Install PostgreSQL from Source on Linux ... 3
- Section 1.3: Installation on GNU+Linux ... 4
- Section 1.4: How to install PostgreSQL via MacPorts on OSX ... 5
- Section 1.5: Install postgresql with brew on Mac ... 7
- Section 1.6: Postgres.app for Mac OSX ... 7

Chapter 2: Data Types ... 8

- Section 2.1: Numeric Types ... 8
- Section 2.2: Date/ Time Types ... 8
- Section 2.3: Geometric Types ... 9
- Section 2.4: Network Adress Types ... 9
- Section 2.5: Character Types ... 9
- Section 2.6: Arrays ... 9

Chapter 3: Dates, Timestamps, and Intervals ... 11

- Section 3.1: SELECT the last day of month ... 11
- Section 3.2: Cast a timestamp or interval to a string ... 11
- Section 3.3: Count the number of records per week ... 11

Chapter 4: Table Creation ... 12

- Section 4.1: Show table definition ... 12
- Section 4.2: Create table from select ... 12
- Section 4.3: Create unlogged table ... 12
- Section 4.4: Table creation with Primary Key ... 12
- Section 4.5: Create a table that references other table ... 13

Chapter 5: SELECT ... 14

- Section 5.1: SELECT using WHERE ... 14

Chapter 6: Find String Length / Character Length ... 15

- Section 6.1: Example to get length of a character varying field ... 15

Chapter 7: COALESCE ... 16

- Section 7.1: Single non null argument ... 16
- Section 7.2: Multiple non null arguments ... 16
- Section 7.3: All null arguments ... 16

Chapter 8: INSERT ... 17

- Section 8.1: Insert data using COPY ... 17
- Section 8.2: Inserting multiple rows ... 18
- Section 8.3: INSERT data and RETURING values ... 18
- Section 8.4: Basic INSERT ... 18
- Section 8.5: Insert from select ... 18
- Section 8.6: UPSERT - INSERT ... ON CONFLICT DO UPDATE.. ... 19
- Section 8.7: SELECT data into file ... 19

Chapter 9: UPDATE ... 21

- Section 9.1: Updating a table based on joining another table ... 21
- Section 9.2: Update all rows in a table ... 21
- Section 9.3: Update all rows meeting a condition ... 21
- Section 9.4: Updating multiple columns in table ... 21

Chapter 10: JSON Support ... 22
Section 10.1: Using JSONb operators ... 22
Section 10.2: Querying complex JSON documents ... 26
Section 10.3: Creating a pure JSON table ... 27
Chapter 11: Aggregate Functions ... 28
Section 11.1: Simple statistics: min(), max(), avg() ... 28
Section 11.2: regr_slope(Y, X) : slope of the least-squares-fit linear equation determined by the (X, Y) pairs ... 28
Section 11.3: string_agg(expression, delimiter) ... 29
Chapter 12: Common Table Expressions (WITH) ... 31
Section 12.1: Common Table Expressions in SELECT Queries ... 31
Section 12.2: Traversing tree using WITH RECURSIVE ... 31
Chapter 13: Window Functions ... 32
Section 13.1: generic example ... 32
Section 13.2: column values vs dense_rank vs rank vs row_number ... 33
Chapter 14: Recursive queries ... 34
Section 14.1: Sum of Integers ... 34
Chapter 15: Programming with PL/pgSQL ... 35
Section 15.1: Basic PL/pgSQL Function ... 35
Section 15.2: custom exceptions ... 35
Section 15.3: PL/pgSQL Syntax ... 36
Section 15.4: RETURNS Block ... 36
Chapter 16: Inheritance ... 37
Section 16.1: Creating children tables ... 37
Chapter 17: Export PostgreSQL database table header and data to CSV file ... 38
Section 17.1: copy from query ... 38
Section 17.2: Export PostgreSQL table to csv with header for some column(s) ... 38
Section 17.3: Full table backup to csv with header ... 38
Chapter 18: Triggers and Trigger Functions ... 39
Section 18.1: Type of triggers ... 39
Section 18.2: Basic PL/pgSQL Trigger Function ... 40
Chapter 19: Event Triggers ... 42
Section 19.1: Logging DDL Command Start Events ... 42
Chapter 20: Role Management ... 43
Section 20.1: Create a user with a password ... 43
Section 20.2: Grant and Revoke Privileges ... 43
Section 20.3: Create Role and matching database ... 44
Section 20.4: Alter default search_path of user ... 44
Section 20.5: Create Read Only User ... 45
Section 20.6: Grant access privileges on objects created in the future ... 45
Chapter 21: Postgres cryptographic functions ... 46
Section 21.1: digest ... 46
Chapter 22: Comments in PostgreSQL ... 47
Section 22.1: COMMENT on Table ... 47
Section 22.2: Remove Comment ... 47
Chapter 23: Backup and Restore ... 48
Section 23.1: Backing up one database ... 48
Section 23.2: Restoring backups ... 48

Section 23.3: Backing up the whole cluster ..... 48
Section 23.4: Using psql to export data ..... 49
Section 23.5: Using Copy to import ..... 49
Section 23.6: Using Copy to export ..... 50
Chapter 24: Backup script for a production DB ..... 51
Section 24.1: saveProdDb.sh ..... 51
Chapter 25: Accessing Data Programmatically ..... 52
Section 25.1: Accessing PostgreSQL with the C-API ..... 52
Section 25.2: Accessing PostgreSQL from python using psycopg2 ..... 55
Section 25.3: Accessing PostgreSQL from .NET using the Npgsql provider ..... 55
Section 25.4: Accessing PostgreSQL from PHP using Pomm2 ..... 56
Chapter 26: Connect to PostgreSQL from Java ..... 58
Section 26.1: Connecting with java.sql.DriverManager ..... 58
Section 26.2: Connecting with java.sql.DriverManager and Properties ..... 58
Section 26.3: Connecting with javax.sql.DataSource using a connection pool ..... 59
Chapter 27: PostgreSQL High Availability ..... 61
Section 27.1: Replication in PostgreSQL ..... 61
Chapter 28: EXTENSION dblink and postgres_fdw ..... 64
Section 28.1: Extention FDW ..... 64
Section 28.2: Foreign Data Wrapper ..... 64
Section 28.3: Extention dblink ..... 65
Chapter 29: Postgres Tip and Tricks ..... 66
Section 29.1: DATEADD alternative in Postgres ..... 66
Section 29.2: Comma separated values of a column ..... 66
Section 29.3: Delete duplicate records from postgres table ..... 66
Section 29.4: Update query with join between two tables alternative since Postresql does not support join in update query ..... 66
Section 29.5: Difference between two date timestamps month wise and year wise ..... 66
Section 29.6: Query to Copy/Move/Transafer table data from one database to other database table with same schema ..... 67
Credits ..... 68
You may also like ..... 70

# About

Please feel free to share this PDF with anyone for free,
latest version of this book can be downloaded from:
https://goalkicker.com/PostgreSQLBook

This *PostgreSQL® Notes for Professionals* book is compiled from Stack Overflow Documentation, the content is written by the beautiful people at Stack Overflow. Text content is released under Creative Commons BY-SA, see credits at the end of this book whom contributed to the various chapters. Images may be copyright of their respective owners unless otherwise specified

This is an unofficial free book created for educational purposes and is not affiliated with official PostgreSQL® group(s) or company(s) nor Stack Overflow. All trademarks and registered trademarks are the property of their respective company owners

The information presented in this book is not guaranteed to be correct nor accurate, use at your own risk

Please send feedback and corrections to web@petercv.com

Click here to download full PDF material