

# Frequently Asked Questions for FUJITSU Enterprise Postgres

## Contents

Q The application server is configured to time out SQL that takes more than 30 minutes to perform. SQL time out on the application server, but it does not time out on the DB server. How do I time out SQL on the DB server?.....	3
Q I can use the DBLINK feature to connect to PostgreSQL, but can I also connect to SQL Server or Oracle databases? .....	3
Q About the DB connection password, I want to perform password input automatically without interactive operation, so that the operator (non-administrator user) does not know the operator's password. How to do it?.....	3
Q What is the maximum size of one table for a database? .....	3
Q About postgresql.conf configuration, do the max_connections and max_prepared_transactions settings use memory even when no DB access is performed? Or will it only use memory for DB connections/prepared transactions?.....	3
Q How do I store binary data in bytea type (hex)?.....	4
Q What is the normal number of processes? .....	5
Q What is the unit of the parameter setting for the estimation of the database memory?.....	6
Q Can I specify an expiration date for a database user account like Oracle?.....	6
Q How do I measure the timing of VACUUM operation and the processing time?.....	6
Q When setting the password expiration date for a database user, the expiration date can be specified with VALID UNTIL. But, is it possible to specify a time period such as "3 months"? .....	6
Q How do I change the client encoding in an application accessing a database?.....	7
Q Is it required to rebuild an index of database regularly?.....	8
Q What are the criteria for running VACUUM? What are the business impacts and risks of running VACUUM during production? .	8
Q How is the data sorted when sending query using SQL statement?.....	9
Q How do I lock the statistics which is set in a table?.....	9

**Version and issue date**

Edition 1.0: July 2021

**Copyright**

Copyright 2021 FUJITSU LIMITED

**Legal Notice**

PostgreSQL is Copyright © 1996–2020 by the PostgreSQL Global Development Group.

Postgres95 is Copyright © 1994–5 by the Regents of the University of California.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS-IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

**Q** The application server is configured to time out SQL that takes more than 30 minutes to perform. SQL time out on the application server, but it does not time out on the DB server. How do I time out SQL on the DB server?

**A** SQL can also time out on the DB server by setting the timeout period (milliseconds) in `statement_timeout` by `SET` command before executing SQL.

**Q** I can use the DBLINK feature to connect to PostgreSQL, but can I also connect to SQL Server or Oracle databases?

**A** The DBLINK feature is for connecting to remote PostgreSQL databases. It cannot be used to connect to SQL Server or Oracle databases.

**Q** About the DB connection password, I want to perform password input automatically without interactive operation, so that the operator (non-administrator user) does not know the operator's password. How to do it?

**A** It is not possible to keep the operators (non-administrator user) from knowing their own passwords.

**Q** What is the maximum size of one table for a database?

**A** The table size has a quantitative limit of 32 terabytes or less. Both "Total table size" and "Tablespace size" are available up to the OS acquired sizes.

Table G.3 Schema element

Item	Limit
Table size	Up to 32 terabytes

**[Reference Manual]**

Installation and Setup Guide for Server  
Appendix G Quantitative Limits

**Q** About `postgresql.conf` configuration, do the `max_connections` and `max_prepared_transactions` settings use memory even when no DB access is performed? Or will it only use memory for DB connections/prepared transactions?

**A** The `max_connections` and `max_prepared_transactions` settings affect shared memory usage, and shared memory is acquired at database server startup. The acquired shared memory is used for DB connections/prepared transactions, etc. Therefore, memory is used even when DB access is not performed.

## Q How do I store binary data in bytea type (hex)?

A Convert the binary string to bytea hex format or bytea escape format and then create an SQL string.

### bytea Hex Format

The "hex" format encodes binary data as 2 hexadecimal digits per byte, most significant nibble first.

The entire string is preceded by the sequence `¥x` (to distinguish it from the escape format).

Example:

```
SELECT '¥xDEADBEEF';
```

### bytea Escape Format

The "escape" format takes the approach of representing a binary string as a sequence of ASCII characters, while converting those bytes that cannot be represented as an ASCII character into special escape sequences.

Example:

```
SELECT '¥000'::bytea;
```

### [Reference Manual]

PostgreSQL 13.1 Documentation *\*Replace version as appropriate.*

Chapter 8. Data Types

8.4. Binary Data Types

8.4.1. bytea Hex Format

8.4.2. bytea Escape Format

You can also use the `PQescapeByteaConn` function for conversion.

However, when you use this function, this is only used when inserting data directly into an SQL command string.

```
unsigned char *PQescapeByteaConn(PGconn *conn,  
                                const unsigned char *from,  
                                size_t from_length,  
                                size_t *to_length);
```

The `from` parameter points to the first byte of the string that is to be escaped, and the `from_length` parameter gives the number of bytes in this binary string. (A terminating zero byte is neither necessary nor counted.) The `to_length` parameter points to a variable that will hold the resultant escaped string length. This result string length includes the terminating zero byte of the result.

`PQescapeByteaConn` returns an escaped version of the `from` parameter binary string in memory allocated with `malloc()`. This memory should be freed using `PQfreemem()` when the result is no longer needed. The return string has all special characters replaced so that they can be properly processed by the PostgreSQL string literal parser, and the `bytea` input function. A terminating zero byte is also added. The single quotes that must surround PostgreSQL string literals are not part of the result string.

On error, a null pointer is returned, and a suitable error message is stored in the `conn` object.

### [Reference Manual]

PostgreSQL 13.1 Documentation *\*Replace version as appropriate.*

Chapter 33. libpq - C Library

33.3. Command Execution Functions

33.3.4. Escaping Strings for Inclusion in SQL Commands

## Q What is the normal number of processes?

A A total of six processes, a master process at startup and five processes launched from the master, are created, and one back-end process runs for each application when the application runs.

The following processes are created at startup:

- Master process
- Writer process
- Check pointer process
- WAL writer process
- Automatic vacuum launcher process
- Statistics collector process

In addition, one process is created for each application as the back-end of the application.

Since the above is the process configuration when using the basic functions, the number of processes may be further increased depending on the utilized functions such as replication.

Please refer to the following manual for details.

On most Unix platforms, PostgreSQL modifies its command title as reported by `ps`, so that individual server processes can readily be identified. A sample display is

```
$ ps auxww | grep ^postgres
postgres 15551 0.0 0.1 57536 7132 pts/0 S 18:02 0:00 postgres-i
postgres 15554 0.0 0.0 57536 1184 ? Ss 18:02 0:00 postgres: background writer
postgres 15555 0.0 0.0 57536 916 ? Ss 18:02 0:00 postgres: checkpointer
postgres 15556 0.0 0.0 57536 916 ? Ss 18:02 0:00 postgres: walwriter
postgres 15557 0.0 0.0 58504 2244 ? Ss 18:02 0:00 postgres: autovacuum launcher
postgres 15558 0.0 0.0 17512 1068 ? Ss 18:02 0:00 postgres: stats collector
postgres 15582 0.0 0.0 58772 3080 ? Ss 18:04 0:00 postgres: joe runbug 127.0.0.1 idle
postgres 15606 0.0 0.0 58772 3052 ? Ss 18:07 0:00 postgres: tgl regression [local] SELECT waiting
postgres 15610 0.0 0.0 58772 3056 ? Ss 18:07 0:00 postgres: tgl regression [local] idle in transaction
```

(The appropriate invocation of `ps` varies across different platforms, as do the details of what is shown. This example is from a recent Linux system.) The first process listed here is the master server process. The command arguments shown for it are the same ones used when it was launched. The next five processes are background worker processes automatically launched by the master process. (The "stats collector" process will not be present if you have set the system not to start the statistics collector; likewise the "autovacuum launcher" process can be disabled.)

### [Reference Manual]

PostgreSQL 13.1 Documentation \*Replace version as appropriate.  
Chapter 27. Monitoring Database Activity  
27.1. Standard Unix Tools

**Q** What is the unit of the parameter setting for the estimation of the database memory?

---

**A** Please refer to following manuals for details:

**[Reference Manual]**

PostgreSQL 13.1 Documentation *\*Replace version as appropriate.*

Chapter 19. Server Configuration

19.4. Resource Consumption

19.4.1. Memory

Installation and Setup Guide for Server

Appendix D Configuring Parameters

**Q** Can I specify an expiration date for a database user account like Oracle?

---

**A** You can set the password expiration with VALID UNTIL of CREATE ROLE.  
For database roles that have already been created, you can change the password expiration with VALID UNTIL of ALTER ROLE.

**SQL Commands**

ALTER ROLE

VALID UNTIL 'timestamp'

These clauses alter attributes originally set by CREATE ROLE. For more information, see the CREATE ROLE reference page.

CREATE ROLE

VALID UNTIL 'timestamp'

The VALID UNTIL clause sets a date and time after which the role's password is no longer valid. If this clause is omitted the password will be valid for all time.

**[Reference Manual]**

PostgreSQL 13.1 Documentation *\*Replace version as appropriate.*

Part VI. Reference

**Q** How do I measure the timing of VACUUM operation and the processing time?

---

**A** You can see the vacuumed time by checking the pg\_stat\_all\_tables view. However, you should manually measure how long it takes to run VACUUM.

**Q** When setting the password expiration date for a database user, the expiration date can be specified with VALID UNTIL. But, is it possible to specify a time period such as "3 months"?

---

**A** When setting the password expiration date for a database user, you cannot specify a time period such as "3 months".

## **Q** How do I change the client encoding in an application accessing a database?

**A** **Note:** Excludes FUJITSU Enterprise Postgres on IBM LinuxONE™

Character codes (coding scheme) used in applications are specified differently depending on the application type. For details, refer to the following manuals. Note that for JDBC drivers and .NET Data Provider drivers, the encoding is fixed in UTF-8 and cannot be changed.

Ensure that the encoding system that is embedded in the application and passed to the database, and the encoding system setting of the runtime environment, are the same. Otherwise, the encoding system cannot be converted correctly on the database server.

Refer to "Supported Character Sets" in "Server Administration" in the PostgreSQL Documentation for information on the strings that represent the encoding system that can be set.

### **For ODBC Driver or C Library (libpq)**

Use one of the following methods to set the encoding system for the application:

- Set the PGCLIENTENCODING environment variable in the runtime environment.
- Set the client\_encoding keyword in the connection string.
- Use the PQsetClientEncoding function.

### **For Embedded SQL in C or Embedded SQL in COBOL**

Use one of the following methods to set the encoding system for the application:

- Set the PGCLIENTENCODING environment variable in the runtime environment.
- Set the client\_encoding keyword in the connection string.

#### **[Reference Manual]**

Application Development Guide

Chapter 2 JDBC Driver

2.2 Setup

2.2.2 Message Language and Encoding System Used by Applications Settings

Chapter 3 ODBC Driver

3.2 Setup

3.2.4 Message Language and Encoding System Used by Applications Settings

Chapter 4 .NET Data Provider

4.2 Setup

4.2.5 Message Language Settings

Chapter 5 C Library (libpq)

5.2 Setup

5.2.2 Message Language and Encoding System Used by Applications Settings

Chapter 6 Embedded SQL in C

6.2 Setup

6.2.2 Message Language and Encoding System Used by Applications Settings

Chapter 7 Embedded SQL in COBOL

7.2 Setup

7.2.2 Message Language and Encoding System Used by Applications Settings

PostgreSQL 13.1 Documentation \*Replace version as appropriate

Chapter 23. Localization

23.3. Character Set Support

23.3.1. Supported Character Sets

**Q Is it required to rebuild an index of database regularly?**

---

**A** It is recommended that indexes should be reorganized on a regular basis.

Normally, a database defines indexes in tables, but if data is frequently updated, indexes can no longer use free space in the disk efficiently. This situation can also cause a gradual decline in database access performance.

To rearrange used space on the disk and prevent the database access performance from declining, it is recommended that you periodically execute the REINDEX command to reorganize indexes.

**[Reference Manual]**

Operation Guide

Chapter 7 Periodic Operations

7.5 Reorganizing Indexes

**Q What are the criteria for running VACUUM?  
What are the business impacts and risks of running VACUUM during production?**

---

**A** - Criteria

The PostgreSQL VACUUM command is needed to be processed on each table periodically due to below reasons.

1. Recovery or reuse of disk space occupied by updated or deleted rows
2. Update of data statistics used by the PostgreSQL query planner
3. Update of visibility maps to speed up index-only scans.
4. Loss prevention of very old data due to Transaction ID wraparound or multi-transaction ID wraparound

The timing of execution is environment dependent.

It is executed at an opportunity when detecting performance delay or when it can be assumed that the enlargement of the table region hinders the operation.

- Business impacts and Risks

VACUUM

It can be executed at the same time as operations on a production database.

Commands such as SELECT, INSERT, UPDATE, and DELETE can continue to work normally.

However, the table definition cannot be changed with commands such as ALTER TABLE ADD COLUMN.

VACUUM FULL

It requires exclusive locks on all working tables.

Therefore, it cannot be done in parallel with other uses of those tables.

As VACUUM generates a significant amount of I/O traffic, this may degrade the performance of other running sessions.



**Q** How is the data sorted when sending query using SQL statement?

---

**A** It is sorted in ascending order if ASC is specified in the ORDER BY clause, and in descending order if DESC is specified.  
If the ORDER BY clause is not specified, unsorted rows are returned without any rule.

**Q** How do I lock the statistics which is set in a table?

---

**A** If you want to lock statistics on a table-by-table basis, please specify '<schema name>.<table name>' in dbms\_stats.lock\_table\_stats.