NAME

pg_amcheck - checks for corruption in one or more PostgreSQL databases

SYNOPSIS

pg_amcheck [*option*...] [*dbname*]

DESCRIPTION

pg_amcheck supports running amcheck's corruption checking functions against one or more databases, with options to select which schemas, tables and indexes to check, which kinds of checking to perform, and whether to perform the checks in parallel, and if so, the number of parallel connections to establish and use.

Only ordinary and toast table relations, materialized views, sequences, and btree indexes are currently supported. Other relation types are silently skipped.

If dbname is specified, it should be the name of a single database to check, and no other database selection options should be present. Otherwise, if any database selection options are present, all matching databases will be checked. If no such options are present, the default database will be checked. Database selection options include **--all**, **--database** and **--exclude-database**. They also include **--relation**, **--exclude-relation**, **--table**, **--exclude-table**, **--index**, and **--exclude-index**, but only when such options are used with a three-part pattern (e.g. **mydb*.myschema*.myrel***). Finally, they include **--schema** and **--exclude-schema** when such options are used with a two-part pattern (e.g. **mydb*.myschema***).

dbname can also be a connection string.

OPTIONS

The following command-line options control what is checked:

-a
--all
Check all databases, except for any excluded via **--exclude-database**.

-d *pattern*
--database=*pattern*
Check databases matching the specified *pattern*, except for any excluded by **--exclude-database**. This option can be specified more than once.

-D *pattern*
--exclude-database=*pattern*

Exclude databases matching the given *pattern*. This option can be specified more than once.

-i *pattern*
--index=*pattern*
Check indexes matching the specified *pattern*, unless they are otherwise excluded. This option can be specified more than once.

This is similar to the **--relation** option, except that it applies only to indexes, not to other relation types.

-I *pattern*
--exclude-index=*pattern*
Exclude indexes matching the specified *pattern*. This option can be specified more than once.

This is similar to the **--exclude-relation** option, except that it applies only to indexes, not other relation types.

-r *pattern*
--relation=*pattern*
Check relations matching the specified *pattern*, unless they are otherwise excluded. This option can be specified more than once.

Patterns may be unqualified, e.g. myrel*, or they may be schema-qualified, e.g. myschema*.myrel* or database-qualified and schema-qualified, e.g. mydb*.myschema*.myrel*. A database-qualified pattern will add matching databases to the list of databases to be checked.

-R *pattern*
--exclude-relation=*pattern*
Exclude relations matching the specified *pattern*. This option can be specified more than once.

As with **--relation**, the *pattern* may be unqualified, schema-qualified, or database- and schema-qualified.

-s *pattern*
--schema=*pattern*
Check tables and indexes in schemas matching the specified *pattern*, unless they are otherwise excluded. This option can be specified more than once.

To select only tables in schemas matching a particular pattern, consider using something like --table=SCHEMAPAT.* --no-dependent-indexes. To select only indexes, consider using

something like --index=SCHEMAPAT.*.

A schema pattern may be database-qualified. For example, you may write --schema=mydb*.myschema* to select schemas matching myschema* in databases matching mydb*.

-S *pattern*
--exclude-schema=*pattern*
Exclude tables and indexes in schemas matching the specified *pattern*. This option can be specified more than once.

As with **--schema**, the pattern may be database-qualified.

-t *pattern*
--table=*pattern*
Check tables matching the specified *pattern*, unless they are otherwise excluded. This option can be specified more than once.

This is similar to the **--relation** option, except that it applies only to tables, materialized views, and sequences, not to indexes.

-T *pattern*
--exclude-table=*pattern*
Exclude tables matching the specified *pattern*. This option can be specified more than once.

This is similar to the **--exclude-relation** option, except that it applies only to tables, materialized views, and sequences, not to indexes.

--no-dependent-indexes
By default, if a table is checked, any btree indexes of that table will also be checked, even if they are not explicitly selected by an option such as --index or --relation. This option suppresses that behavior.

--no-dependent-toast
By default, if a table is checked, its toast table, if any, will also be checked, even if it is not explicitly selected by an option such as --table or --relation. This option suppresses that behavior.

--no-strict-names
By default, if an argument to --database, --table, --index, or --relation matches no objects, it is a fatal error. This option downgrades that error to a warning.

The following command-line options control checking of tables:

--exclude-toast-pointers
By default, whenever a toast pointer is encountered in a table, a lookup is performed to ensure that it references apparently-valid entries in the toast table. These checks can be quite slow, and this option can be used to skip them.

--on-error-stop
After reporting all corruptions on the first page of a table where corruption is found, stop processing that table relation and move on to the next table or index.

Note that index checking always stops after the first corrupt page. This option only has meaning relative to table relations.

--skip=*option*
If all-frozen is given, table corruption checks will skip over pages in all tables that are marked as all frozen.

If all-visible is given, table corruption checks will skip over pages in all tables that are marked as all visible.

By default, no pages are skipped. This can be specified as none, but since this is the default, it need not be mentioned.

--startblock=*block*
Start checking at the specified block number. An error will occur if the table relation being checked has fewer than this number of blocks. This option does not apply to indexes, and is probably only useful when checking a single table relation. See --endblock for further caveats.

--endblock=*block*
End checking at the specified block number. An error will occur if the table relation being checked has fewer than this number of blocks. This option does not apply to indexes, and is probably only useful when checking a single table relation. If both a regular table and a toast table are checked, this option will apply to both, but higher-numbered toast blocks may still be accessed while validating toast pointers, unless that is suppressed using **--exclude-toast-pointers**.

The following command-line options control checking of B-tree indexes:

--heapallindexed
For each index checked, verify the presence of all heap tuples as index tuples in the index using

amcheck's **heapallindexed** option.

--parent-check
For each btree index checked, use amcheck's **bt_index_parent_check** function, which performs additional checks of parent/child relationships during index checking.

The default is to use amcheck's **bt_index_check** function, but note that use of the **--rootdescend** option implicitly selects **bt_index_parent_check**.

--rootdescend
For each index checked, re-find tuples on the leaf level by performing a new search from the root page for each tuple using amcheck's **rootdescend** option.

Use of this option implicitly also selects the **--parent-check** option.

This form of verification was originally written to help in the development of btree index features. It may be of limited use or even of no use in helping detect the kinds of corruption that occur in practice. It may also cause corruption checking to take considerably longer and consume considerably more resources on the server.

Warning

The extra checks performed against B-tree indexes when the **--parent-check** option or the **--rootdescend** option is specified require relatively strong relation-level locks. These checks are the only checks that will block concurrent data modification from **INSERT**, **UPDATE**, and **DELETE** commands.

The following command-line options control the connection to the server:

-h *hostname*
--host=*hostname*
Specifies the host name of the machine on which the server is running. If the value begins with a slash, it is used as the directory for the Unix domain socket.

-p *port*
--port=*port*
Specifies the TCP port or local Unix domain socket file extension on which the server is listening for connections.

-U

--username=*username*
User name to connect as.

-w
--no-password
Never issue a password prompt. If the server requires password authentication and a password is not available by other means such as a .pgpass file, the connection attempt will fail. This option can be useful in batch jobs and scripts where no user is present to enter a password.

-W
--password
Force pg_amcheck to prompt for a password before connecting to a database.

This option is never essential, since pg_amcheck will automatically prompt for a password if the server demands password authentication. However, pg_amcheck will waste a connection attempt finding out that the server wants a password. In some cases it is worth typing **-W** to avoid the extra connection attempt.

--maintenance-db=*dbname*
Specifies a database or connection string to be used to discover the list of databases to be checked. If neither **--all** nor any option including a database pattern is used, no such connection is required and this option does nothing. Otherwise, any connection string parameters other than the database name which are included in the value for this option will also be used when connecting to the databases being checked. If this option is omitted, the default is postgres or, if that fails, template1.

Other options are also available:

-e
--echo
Echo to stdout all SQL sent to the server.

-j *num*
--jobs=*num*
Use *num* concurrent connections to the server, or one per object to be checked, whichever is less.

The default is to use a single connection.

-P
--progress

Show progress information. Progress information includes the number of relations for which checking has been completed, and the total size of those relations. It also includes the total number of relations that will eventually be checked, and the estimated size of those relations.

-v
--verbose
Print more messages. In particular, this will print a message for each relation being checked, and will increase the level of detail shown for server errors.

-V
--version
Print the pg_amcheck version and exit.

--install-missing
--install-missing=*schema*
Install any missing extensions that are required to check the database(s). If not yet installed, each extension's objects will be installed into the given *schema*, or if not specified into schema pg_catalog.

At present, the only required extension is amcheck.

-?
--help
Show help about pg_amcheck command line arguments, and exit.

NOTES

pg_amcheck is designed to work with PostgreSQL 14.0 and later.

SEE ALSO

amcheck