PassQueen
a.k.a dumpsqueen.com

PostgreSQL Essentials Certification v13

EnterpriseDB PostgreSQL-Essentials

Version Demo

Total Demo Questions: 6

Total Premium Questions: 64

Buy Premium PDF

https://passqueen.com

support@passqueen.com

QUESTION NO: 1

You want to identify sessions that are waiting for a lock and determine the relation on which the lock is held. Which system view should be queried together with `pg_stat_activity`?

A. pg_stat_database

B. pg_locks

C. pg_indexes

D. pg_stat_progress_vacuum

ANSWER: B

Explanation:

`pg_locks` is correct because it provides information about locks currently held or awaited by server processes. Its `pid` column can be joined to `pg_stat_activity.pid` to associate lock information with sessions and their queries. The view can also be joined to catalog information where needed to identify locked database objects. `pg_stat_database` provides database-level statistics, `pg_indexes` describes indexes, and `pg_stat_progress_vacuum` reports active vacuum progress rather than general lock status.

QUESTION NO: 2

You want to take a data-only dump of the edbstore database, disable all triggers for faster restore, and use the INSERT command instead of COPY. Which pg_dump command options should you use?

A. -a --inserts --disable-triggers

B. -d --inserts --disable-triggers

C. --data-only --inserts

D. -a --copy --disable-triggers

ANSWER: A

Explanation:

`-a --inserts --disable-triggers` is the correct combination. The `-a` (or `--data-only`) option instructs `pg_dump` to dump only database data, omitting schema-definition objects such as tables, functions, and indexes. The `--inserts` option changes the generated data-loading statements from the default `COPY`-based form to SQL `INSERT` statements. This is useful when a plain SQL representation using individual insert commands is specifically required.

The `--disable-triggers` option causes the dump output to include commands that disable triggers on user tables before data is loaded and re-enable them afterward. During a restore, this can reduce trigger-related overhead and avoid trigger execution while bulk data is being inserted. PostgreSQL documents that this behavior applies to a data-only dump and generally requires restoring as a superuser because disabling triggers requires elevated privileges. The target database can then be supplied separately, for example with `-d edbstore` or as the final database-name argument, while these three options express the requested dump behavior. See the PostgreSQL pg_dump documentation and the backup and restore documentation.

QUESTION NO: 3

Which of the following system functions can be used to track the status of the vacuum process?

A. pg_stat_vacuum

B. pg_stat_vacuumdb

C. pg_stat_user_vacuum

D. pg_stat_progress_vacuum

ANSWER: D

Explanation:

`pg_stat_progress_vacuum` is the PostgreSQL system view used to monitor the progress and current status of vacuum operations while they are running. It exposes one row for each active VACUUM worker and includes useful operational details such as the database and relation being processed, the VACUUM phase, heap blocks scanned and vacuumed, index vacuuming progress, and tuple counts. This makes it particularly useful to DBAs who need to determine whether a long-running manual VACUUM or autovacuum operation is actively progressing and which stage it has reached. The view was introduced in PostgreSQL 9.6 and is therefore available in PostgreSQL 13, the version covered by this certification. Querying it during an active operation can provide a direct, real-time progress snapshot, for example: `SELECT * FROM pg_stat_progress_vacuum;`. PostgreSQL documents the columns and phases provided by this view in its monitoring statistics reference: VACUUM Progress Reporting. For related details about VACUUM behavior and its role in reclaiming storage and maintaining table health, see the PostgreSQL 13 VACUUM documentation.

QUESTION NO: 4

You need to restore a database from a compressed format backup file created with pg_dump -Fc option. Which utility should you use?

A. psql

B. pg_restore

C. pg_dump

D. pg_basebackup

ANSWER: B

Explanation:

`pg_restore` is the correct utility for restoring an archive produced by `pg_dump -Fc`. The `-Fc` flag creates a custom-format archive, which is compressed by default and is designed for use with `pg_restore`. This utility reads the archive's table-of-contents information and can restore the database schema, table data, functions, indexes, triggers, and other dumped objects. It also supports useful restore controls, such as selecting particular database objects, restoring only schema or data, setting parallel jobs, and controlling restore order. A typical restore first creates an empty target database, then runs a command such as `pg_restore -d target_database backup.dump`. If the archived dump includes commands to create the database, `pg_restore --create -d postgres backup.dump` can create and connect to the target database as part of the restore process. PostgreSQL documents that custom-format archives must be restored with `pg_restore`, rather than being treated as plain SQL scripts. See the PostgreSQL pg_dump documentation and pg_restore documentation.

QUESTION NO: 5

A DBA needs to review the effective value, source, and source file of a configuration parameter while connected to the server. Which query provides this information for the `work_mem` parameter?

A. SHOW work_mem;

B. SELECT name, setting, source, sourcefile FROM pg_settings WHERE name = 'work_mem';

C. SELECT * FROM pg_stat_activity WHERE name = 'work_mem';

D. SELECT setting FROM pg_database WHERE name = 'work_mem';

ANSWER: B

Explanation:

`SELECT name, setting, source, sourcefile FROM pg_settings WHERE name = 'work_mem';` is correct because `pg_settings` exposes run-time configuration parameters and metadata about their current values. The `source` column identifies how the value was set, and `sourcefile` can identify the configuration file when applicable. `SHOW work_mem;` displays only the current effective value and does not provide the requested source details. `pg_stat_activity` reports session activity, while `pg_database` contains database-level catalog information.

QUESTION NO: 6

You need to determine the names and definitions of all views in the edbuser schema. Which query should you use?

A. SELECT * FROM pg_views;

B. SELECT * FROM pg_views WHERE schemaname = 'edbuser';

C. SELECT * FROM information_schema.views WHERE table_schema = 'edbuser';

D. Both B and C

ANSWER: D

Explanation:

Both `pg_views` and `information_schema.views` provide the information required to identify views and retrieve their definitions when filtered to the `edbuser` schema. The PostgreSQL system view `pg_catalog.pg_views` includes one row per view and exposes `schemaname`, `viewname`, and `definition`. Therefore, filtering `schemaname = 'edbuser'` returns the names and SQL definitions of views belonging to that schema.

The SQL-standard `information_schema.views` view similarly exposes view metadata, including `table_schema`, `table_name`, and `view_definition`. Filtering it with `table_schema = 'edbuser'` produces the same required scope: all views in that schema together with their definitions. The schema name string must be written without unintended leading or trailing spaces, because those spaces are part of the literal value used for comparison. PostgreSQL documents `pg_views` as a system catalog view for accessing view definitions, while `information_schema.views` is the portable information-schema interface for view metadata. See the PostgreSQL pg_views documentation and information_schema.views documentation.