SET TRANSACTION

Name

SET TRANSACTION -- set the characteristics of the current transaction

Synopsis

```
SET TRANSACTION transaction_mode [, ...]
SET TRANSACTION SNAPSHOT snapshot_id
SET SESSION CHARACTERISTICS AS TRANSACTION transaction_mode [, ...]

where transaction_mode is one of:

    ISOLATION LEVEL { SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ
UNCOMMITTED }
    READ WRITE | READ ONLY
    [ NOT ] DEFERRABLE
```

Description

The `SET TRANSACTION` command sets the characteristics of the current transaction. It has no effect on any subsequent transactions. `SET SESSION CHARACTERISTICS` sets the default transaction characteristics for subsequent transactions of a session. These defaults can be overridden by `SET TRANSACTION` for an individual transaction.

The available transaction characteristics are the transaction isolation level, the transaction access mode (read/write or read-only), and the deferrable mode. In addition, a snapshot can be selected, though only for the current transaction, not as a session default.

The isolation level of a transaction determines what data the transaction can see when other transactions are running concurrently:

`READ COMMITTED`

A statement can only see rows committed before it began. This is the default.

`REPEATABLE READ`

All statements of the current transaction can only see rows committed before the first query or data-modification statement was executed in this transaction.

`SERIALIZABLE`

All statements of the current transaction can only see rows committed before the first query or data-modification statement was executed in this transaction. If a pattern of reads and writes among concurrent serializable transactions would create a situation which could not have occurred for any serial (one-at-a-time) execution of those transactions, one of them will be rolled back with a `serialization_failure` error.

The SQL standard defines one additional level, `READ UNCOMMITTED`. In PostgreSQL `READ UNCOMMITTED` is treated as `READ COMMITTED`.

The transaction isolation level cannot be changed after the first query or data-modification statement (`SELECT`, `INSERT`, `DELETE`, `UPDATE`, `FETCH`, or `COPY`) of a transaction has been executed. See Chapter 13 for more information about transaction isolation and concurrency control.

The transaction access mode determines whether the transaction is read/write or read-only. Read/write is the default. When a transaction is read-only, the following SQL commands are disallowed: `INSERT`, `UPDATE`, `DELETE`, and `COPY FROM` if the table they would write to is not a temporary table; all `CREATE`, `ALTER`, and `DROP` commands; `COMMENT`, `GRANT`, `REVOKE`, `TRUNCATE`; and `EXPLAIN ANALYZE` and `EXECUTE` if the command they would execute is among those listed. This is a high-level notion of read-only that does not prevent all writes to disk.

The `DEFERRABLE` transaction property has no effect unless the transaction is also `SERIALIZABLE` and `READ ONLY`. When all three of these properties are selected for a transaction, the transaction may block when first acquiring its snapshot, after which it is able to run without the normal overhead of a `SERIALIZABLE` transaction and without any risk of contributing to or being canceled by a serialization failure. This mode is well suited for long-running reports or backups.

The `SET TRANSACTION SNAPSHOT` command allows a new transaction to run with the same *snapshot* as an existing transaction. The pre-existing transaction must have exported its snapshot with the `pg_export_snapshot` function (see Section 9.26.5). That function returns a snapshot identifier, which must be given to `SET TRANSACTION SNAPSHOT` to specify which snapshot is to be imported. The identifier must be written as a string literal in this command, for example `'000003A1-1'`. `SET TRANSACTION SNAPSHOT` can only be executed at the start of a transaction, before the first query or data-modification statement (`SELECT`, `INSERT`, `DELETE`, `UPDATE`, `FETCH`, or `COPY`) of the transaction. Furthermore, the transaction must already be set to `SERIALIZABLE` or `REPEATABLE READ` isolation level (otherwise, the snapshot would be discarded immediately, since `READ COMMITTED` mode takes a new snapshot for each command). If the importing transaction uses `SERIALIZABLE` isolation level, then the transaction that exported the snapshot must also use that isolation level. Also, a non-read-only serializable transaction cannot import a snapshot from a read-only transaction.

Notes

If `SET TRANSACTION` is executed without a prior `START TRANSACTION` or `BEGIN`, it will appear to have no effect, since the transaction will immediately end.

It is possible to dispense with `SET TRANSACTION` by instead specifying the desired ***`transaction_modes`*** in `BEGIN` or `START TRANSACTION`. But that option is not available for `SET TRANSACTION SNAPSHOT`.

The session default transaction modes can also be set by setting the configuration parameters default_transaction_isolation, default_transaction_read_only, and default_transaction_deferrable. (In fact `SET SESSION CHARACTERISTICS` is just a verbose equivalent for setting these variables with `SET`.) This means the defaults can be set in the configuration file, via `ALTER DATABASE`, etc. Consult Chapter 18 for more information.

Examples

To begin a new transaction with the same snapshot as an already existing transaction, first export the snapshot from the existing transaction. That will return the snapshot identifier, for example:

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SELECT pg_export_snapshot();
 pg_export_snapshot
--------------------
 000003A1-1
(1 row)
```

Then give the snapshot identifier in a `SET TRANSACTION SNAPSHOT` command at the beginning of the newly opened transaction:

```
BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET TRANSACTION SNAPSHOT '000003A1-1';
```

Compatibility

These commands are defined in the SQL standard, except for the `DEFERRABLE` transaction mode and the `SET TRANSACTION SNAPSHOT` form, which are PostgreSQL extensions.

`SERIALIZABLE` is the default transaction isolation level in the standard. In PostgreSQL the default is ordinarily `READ COMMITTED`, but you can change it as mentioned above.

In the SQL standard, there is one other transaction characteristic that can be set with these commands: the size of the diagnostics area. This concept is specific to embedded SQL, and therefore is not implemented in the PostgreSQL server.

The SQL standard requires commas between successive ***`transaction_modes`***, but for historical reasons PostgreSQL allows the commas to be omitted.

Prev Home Next
SET SESSION AUTHORIZATION Up SHOW