You can ask the system catalog. The tricky part is (as has been commented) that `CREATE DATABASE` can only be executed as a single statement. Per documentation:

> `CREATE DATABASE` cannot be executed inside a transaction block.

So it cannot be run inside a function or `DO` statement, where it would be inside a transaction block implicitly. That can be circumvented though by using a `dblink` connection back to the current database, which runs outside of the transaction block. Effects can therefore also not be rolled back.

You need to install the additional module dblink (once per db):

- How to use (install) dblink in PostgreSQL?

Then:

```
DO
$do$
BEGIN
   IF EXISTS (SELECT 1 FROM pg_database WHERE datname = 'mydb') THEN
      RAISE NOTICE 'Database already exists';
   ELSE
      PERFORM dblink_exec('dbname=' || current_database()  -- current db
                          , 'CREATE DATABASE mydb');
   END IF;
END
$do$;
```

A detailed explanation on how it works:

- How do I do large non-blocking updates in PostgreSQL?

Tested with Postgres 9.3. You could make this a function for repeated use.

Another alternative, just in case you want to have a shell script which creates the database if it does not exist and otherwise just keeps it as it is:

```
psql -U postgres -tc "SELECT 1 FROM pg_database WHERE datname = 'my_db'" | grep -q
1 || psql -U postgres -c "CREATE DATABASE my_db"
```

I found this to be helpful in devops provisioning scripts, which you might want to run multiple times over the same instance.

I had to use a slightly extended version @Erwin Brandstetter used:

```
DO
$do$
DECLARE
  _db TEXT := 'some_db';
  _user TEXT := 'posrgres';
  _password TEXT := 'posrgres';
BEGIN
  CREATE EXTENSION IF NOT EXISTS dblink; -- enable extension
  IF EXISTS (SELECT 1 FROM pg_database WHERE datname = _db) THEN
    RAISE NOTICE 'Database already exists';
  ELSE
    PERFORM dblink_connect('host=localhost user=' || _user || ' password=' ||
_password || ' dbname=' || current_database());
    PERFORM dblink_exec('CREATE DATABASE ' || _db);
  END IF;
END
$do$
```

I had to enable the `dblink` extension, plus i had to provide the credentials for dblink. Works with Postgres 9.4.