

Assignment 1

Frequently Asked Questions

Last updated: Sunday 10th March 4:19pm

Most recent changes are shown in red ... older changes are shown in brown.

Frequently Asked Questions

The Forum has a search function, but nobody seems to use it, so here are answers to often-asked questions that people would have found if they'd searched for them:

I get messages about UTF8 and locale when I run `initdb`

Run it using the command: `initdb --no-locale --encoding=UTF8`

I get told that `pgxs` is missing when trying to compile

You most likely have an old installation of PostgreSQL (e.g. from COMP9311) in your `/localstorage/$USER/` directory. Remove it and install again.

```
$ cd /localstorage/$USER
$ rm -fr pgsql
$ cd postgresql-15.6
$ make install
```

Then follow the rest of the installation instructions from P01.

I get told that `postgres.h` is not found when trying to compile assignment code.

Add the following line to your `env` file:

```
C_INCLUDE_PATH=/localstorage/$USER/postgresql-15.6/src/include/
```

and then `source` it again.

Can I submit the assignment more than once?

You can submit as many times as you like. Only the final submission will be marked, unless you request otherwise. We keep the most recent three submissions.

Which titles (e.g. Dr, Mr, Ms, ...) do I have to check for?

None. Simply use the grammar to check for validity of names. Any title you can think of looks like a valid name. Treat it as such.

The size of my database is not exactly the same as the size expected by the testing script. Is this a problem?

If your disk usage is not more than 20% larger than the expected amount, that's ok. Not ideal, but ok. If your disk usage is twice as big as the expected amount, that is definitely **not** acceptable.

My code recognises valid and invalid names (sanity tests), but crashes the server as soon as I try to store and retrieve tuples containing names.

My `PersonName` data structure looks like:

```
typedef struct PersonName {
    char *family;
    char *given;
```

```
} PersonName;
```

Go and look at the Week 04 Thursday lecture notes again to see why this is **wrong** and to get some hints on how to do it properly.

I can't seem to get the system to use the hash index. It always uses a sequential scan rather than a "Bitmap Heap Scan".

You need to give additional information for the query optimiser when you define the equality operator:

```
restrict = eqsel, HASHES
```

I get error messages about compressed data (e.g. lz4).

This is caused by you messing up the length field in your `PersonName` object. If it turns into a large number (or maybe a negative number), PostgreSQL tries to store the object in a TOAST file and tries to compress it using `lz4` compression.

Make sure that you use the length field to store the whole length of the object and set it using the `SET_VARSIZE()` macro. Make sure that the size includes `VARHDRSZ` + the length of the name + 1 (to allow for the `'\0'` character at the end of the name string).

Note that the analysis below probably accounts for people who got the overall object length wrong by using two length fields.

I don't have a definitive answer, but I have seen this problem when people define their `PersonName` data structure as something like:

```
typedef struct {
    int family_name_length;
    int given_name_length;
    char name[...];
} PersonName;
```

Having a single length field for the whole array seems to work better. Of course, you then need to work out where the family name ends and the given name starts in the functions that manipulate `PersonName`s.