

Assignment 1 SQL Review

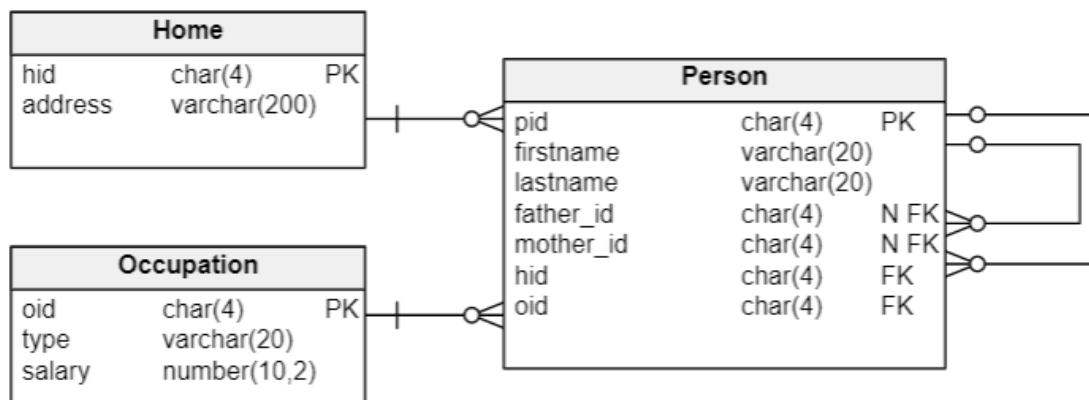
Objectives

- Review SQL syntax
- Create tables
- Write and execute queries

Instructions

Follow the instructions below to create an SQL file called assignment1.sql. This file should be run on PDBORA19C. Ensure that your file runs from start to finish without error and displays the correct output.

Part 1 Creating tables



The database we will be creating keeps track of people. Most of the information is held within the **person** table. Each person is uniquely represented by a *pid*, and they have a *firstname* and *lastname*. Each person has a *mother_id*, which holds the *pid* of their mother, and a *father_id*, which holds the *pid* of their father. Each person also knows where they live, represented by their *hid* and they know their occupation, represented by an *oid*.

The **home** table represents the households where people live. It has only 2 fields. The *hid* uniquely represents each residence, and each residence also has an address.

The **occupation** table represents the jobs that different people have. Each is uniquely represented by an *oid*. The job type describes the kind of job being performed, and the salary is the amount that job pays (if anything). (Note, in this scenario, all people who work the same job get the same salary)

Create all of the above tables in file called assignment1.sql. Everything you do in today's lab (adding tables/data/running queries, should all be done in this file).

Part 2 Adding Data

We'll start with 2 households:

- 'H001' at address '123 Easy St.'
- 'H002' at '56 Fake Ln.'.

Then add the following occupations:

- 'O000' should be for 'N/A', who get a salary of 0 (for people who are not employed)
- 'O001' should be for 'Student', who get a salary of 0
- 'O002' should be for 'Doctor', who get a salary of 100000
- 'O003' should be for 'Professor', who get a salary of 80000

Now add the following people (assign your own pid's to each):

- Zachary Aberny, who is not employed at lives at '56 Fake Ln.' (No mother and father in the db)
- Yanni Aberny who is not employed at lives at '56 Fake Ln.' (No mother and father in the db)
- Alice Aberny, who is a doctor and lives at 123 Easy St. Her mother is Yendell Aberny and her father is Zachary Aberny.
- Bob Bortelson, who is a professor that lives at 123 Easy St. (No mother and father in the db)
- Carl Aberny-Bortelson, who is a student and lives at 123 Easy Street. His mother is Alice Aberny and his father is Bob Bortelson.
- Denise Aberny-Bortelson, who is a student and lives at 123 Easy Street. Her mother is Alice Aberny and her father is Bob Bortelson.

With this, we should have created a simple family tree. We have Alice Aberny's parents, who are live together on 56 Fake Lane, as well as the 4 members of the Aberny-Bortelson family, who live on 123 Easy street.

To make sure you got the relationships right, try the following query:

```
SELECT gp.firstname FROM person gp JOIN person p ON gp.pid = p.mother_id OR gp.pid = p.father_id  
JOIN person c ON p.pid = c.mother_id OR p.pid = c.father_id WHERE c.firstname = 'Denise'
```

If you've done everything correctly, this should tell you the names of Denise's grandparents (you may need to alter it slightly depending on what you named your fields and tables). A couple of notes about this query: First, the three aliases used are gp for grandparent, p for parent, and c for child. For the ON clause, we are using OR because we want to join each person to both of their parents.

Note: To test that your queries are working as expected in the following questions, you will want to add more data to these tables, but for now that should be sufficient. I would highly recommend sharing your test data with classmates to make your tests as thorough and robust as possible!

Part 3 Running Queries

1. Write a query that, for each household, shows that household's address and the total number of people living at that household.
2. Write a query that prints the full names of all grandfathers.
3. List the full name of every student who lives with at least 1 of their parents.
4. Write a query that, prints the highest household income. (Household income is the sum of salaries of everyone living at a particular home).

Deliverable

Submit a file named assignment1.sql to Moodle. The sql file should run from start to finish without error and display the correct output. Ensure that any previously defined tables or types with the same name are dropped at the beginning of the script. For the due date see Moodle.