

Temporal and Spatial Data Management

Exercise Series 1

Get an IfI PostgreSQL account if you do not have one already. Follow the instructions at <http://www.ifi.uzh.ch/dbtg/sw.html>. Note that you might have to enable VPN to connect to PostgreSQL. Alternatively you can also install PostgreSQL on your own computer.

As an example relation to test your solutions for Task 1.1 - Task 1.4 use:

```
create table r(X char(1), S integer, E integer);

insert into r values ('a', 1, 5);
insert into r values ('a', 3, 7);
insert into r values ('a', 7, 9);
insert into r values ('a', 11, 12);
insert into r values ('b', 4, 12);
```

Task 1.1:

1. Use a user defined function with an iterated SQL statement to implement coalescing in PostgreSQL.
2. Use recursion to implement coalescing in PostgreSQL
3. Use SQL-89 (plain SQL, no user defined functions or advanced predicates/functions) to implement coalescing.
4. Use a user defined function with a loop and cursor to implement coalescing.
5. Use SQL's analytic functions to coalesce a temporal relation. Hint: Use cumulative counts of start and end points.

Assess your solutions.

Task 1.2: Assume relation *Employee*(*Name*, *Salary*, *Job*, *Start*, *End*) is split into two relations so that coalescing is not necessary for determining the salary and job history. Give a query that still requires coalescing with this schema.