EPAM Systems, RD Dep., RD Dep.

POSTGRESQL DB FOR DWH AND ETL BUILDING

PostgreSQL Database Architecture

CONTENTS

1. PREREQUISITE TASK

Investigate more deeply:

1. Roles and privileges
2. Tablespace and Database creation

Read about PostgreSQL **Database Cluster Data Directory Layout**.

2. UNDERSTANDING DATABASE AND TABLESPACES

Tasks Results: Add your personal description of what is done with screenshots where needed.

2.1 TASK 1 CREATE NEW DATABASE

1. Connect to the postgres Database and create new one named “test_db”.
2. Run query, investigate result:

```
select d.oid, d.datname, d.datistemplate, d.datallowconn, t.spcname
from pg_database d
join pg_tablespace t on t.oid = d.dattablespace
```

2.2 TASK 2 CREATE NEW TABLESPACE

1. Create new tablespace “mytablespace” with location “[where postgresql located]/data/tblspc_test/”.
2. Check your tablespace exists in pg_tablespace table:

```
select *
from pg_tablespace
```

3. Move test_db into new tablespace:

```
ALTER DATABASE test_db SET TABLESPACE mytablespace
```

Run again 2.1.2 query and check your database. Check your directory where tablespace is located.

2.3 TASK 3 CREATE NEW SCHEMA

1. Connect to test_db and create new schema named “labs”.
2. In new schema create table named “person”:

```
CREATE TABLE labs.person (
  id integer NOT NULL,
  name varchar(15)
);
```

Check the table:

```
SELECT schemaname, tablename FROM pg_tables
WHERE tablename = 'person';
```

3. Insert into person table values (correct queries if needed):

```
INSERT INTO person VALUES(1, 'Bob');
INSERT INTO person VALUES(2, 'Alice');
INSERT INTO person VALUES(3, 'Robert');
```

4. Use `SHOW search_path` and `SET search_path` to perform INSERTS from previous task without any correction.

3. TRANSACTION AND VACUUMING

Tasks Results: Add your personal description of what happened with screenshots where needed.

3.1 TASK 4 INVESTIGATE MVCC*

*you need to install extension before: `CREATE EXTENSION pageinspect;`

Use queries:

```
select p.id, p.name, p.ctid, p.xmin, p.xmax from person p;

SELECT t_xmin, t_xmax, t_ctid,
tuple_data_split('labs.person'::regclass, t_data, t_infomask,
t_infomask2, t_bits)
FROM heap_page_items(get_raw_page('labs.person', 0));
```

And investigate what is happening with xmin and xmax while performing following in **different** transactions:

```
INSERT INTO person VALUES(4, 'John');

UPDATE person set name = 'Alex' where id = 2;

DELETE FROM person WHERE id = 3;

INSERT INTO person VALUES(999, 'Test');

DELETE FROM person WHERE id = 999;
```

3.2 TASK 5 INVESTIGATE VACUUM

To Check what happened use:

```
SELECT t_xmin, t_xmax, t_ctid,
tuple_data_split('labs.person'::regclass, t_data, t_infomask,
t_infomask2, t_bits)
FROM heap_page_items(get_raw_page('labs.person', 0));
```

1. Run:

   ```
   vacuum labs.person;
   ```

   Check results.

2. Run:

   ```
   INSERT INTO person VALUES(5, 'Sarah');
   ```

   Check results.

3. Run:

   ```
   vacuum full labs.person;
   ```

   Check results.