Cassandra Lab

Use the NixOS Database Systems Virtual Machine [../nixos-dbs-vm/] to run the code in this lab.

Demonstration Example

Run CQL Shell for Apache Cassandra [https://cassandra.apache.org/doc/latest/tools/cqlsh.html] by `cqlsh` and submit the following CQL statements.

Keyspace

Create a keyspace [https://cassandra.apache.org/doc/latest/cql/ddl.html#create-keyspace] with a simple strategy that defines a particular replication factor for data to be spread across the entire cluster.

```
CREATE KEYSPACE IF NOT EXISTS demo WITH replication = {'class':
'SimpleStrategy', 'replication_factor' : 3};
DESCRIBE KEYSPACE demo;
USE demo;
```

By default, the durable writes feature is enable to use the commit log for updates on this keyspace.

Tables

Create a table [https://cassandra.apache.org/doc/latest/cql/ddl.html#create-table] with a composed partition key [https://cassandra.apache.org/doc/latest/cql/ddl.html#partition-key] (building, room) and a simple clustering key [https://cassandra.apache.org/doc/latest/cql/ddl.html#clustering-columns] (time).

```
CREATE TABLE IF NOT EXISTS temps (
    building text,
    room text,
    time timestamp,
    temperature float,
    PRIMARY KEY (
        (building, room),
        time
    )
) WITH comment = 'Temperatures as measured by thermometers in particular buildins and their rooms.';
DESCRIBE temps;
```

Inserting Data

Insert data [https://cassandra.apache.org/doc/latest/cql/dml.html#insert] with a default or predefined time-to-live. (i.e., without/with `USING TTL` suffix).

```
BEGIN BATCH
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 11:00:00', 22);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 11:10:00', 23);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 11:20:00', 21);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 11:30:00', 20);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 11:40:00', 25);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 11:50:00', 20);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 12:00:00', 21);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N204', '2019-10-23 12:10:00', 22);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N205', '2019-10-23 11:00:00', 19);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N205', '2019-10-23 11:10:00', 20);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT VUT', 'N205', '2019-10-23 11:20:00', 21);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT
VUT', 'N205', '2019-10-23 11:30:00', 20);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT
VUT', 'N205', '2019-10-23 11:40:00', 18);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT
VUT', 'N205', '2019-10-23 11:50:00', 21);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT
VUT', 'N205', '2019-10-23 12:00:00', 22);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT
VUT', 'N205', '2019-10-23 12:10:00', 23);
INSERT INTO temps (building, room, time, temperature) VALUES ('FIT
VUT', 'N203', toTimestamp(now()), 23) USING TTL 3600;
APPLY BATCH;
SELECT * FROM temps;
```

Quering Data

Selecting data [https://cassandra.apache.org/doc/latest/cql/dml.html#select] by primary key and non-primary key columns.

Filtering by Primary Key Columns (Fast)

```
SELECT * FROM temps WHERE building = 'FIT VUT' AND room = 'N205';
SELECT * FROM temps WHERE building = 'FIT VUT' AND room = 'N205'
AND time = '2019-10-23 12:00:00';
SELECT MIN(temperature), MAX(temperature), AVG(temperature) FROM
temps WHERE building = 'FIT VUT' AND room = 'N205';
```

Filtering by Non-Primary Key Columns (Slow)

```
SELECT * FROM temps WHERE time = '2019-10-23 12:00:00' ALLOW
FILTERING;
SELECT * FROM temps WHERE temperature > 24 ALLOW FILTERING;
```

Secondary Indices

Create a secondary index [https://cassandra.apache.org/doc/latest/cql/indexes.html#create-index] to be able to query data also by non-primary keys.

```
DROP INDEX measurement_times;
SELECT * FROM temps WHERE time = '2019-10-23 12:00:00';
```

```
CREATE INDEX IF NOT EXISTS measurement_times ON temps (time);
SELECT * FROM temps WHERE time = '2019-10-23 12:00:00';
```

Materialized Views

Create a materialized view [https://cassandra.apache.org/doc/latest/cql/mvs.html#create-materialized-view] to provide an alternative partitioning and clustering of a table by different components of the original primary key.

```
CREATE MATERIALIZED VIEW temps_times AS
    SELECT * FROM temps
    WHERE time IS NOT NULL AND building IS NOT NULL AND room IS NOT
NULL
    PRIMARY KEY (time, building, room)
    WITH comment = 'Times and their temperature measurements in
individual locations.';
SELECT * FROM temps_times;
```

Practice

Create table(s) for an audit log of a building access control. In the log, usage of access cards to open doors in the building will be stored at least with the following details:

- who, when, and where utilized a card to open a door,
- result of the operation: success (the door was opened) or fail (the door could not be opened).

Design several queries to monitor the access control system, i.e., to query the system by various criteria.