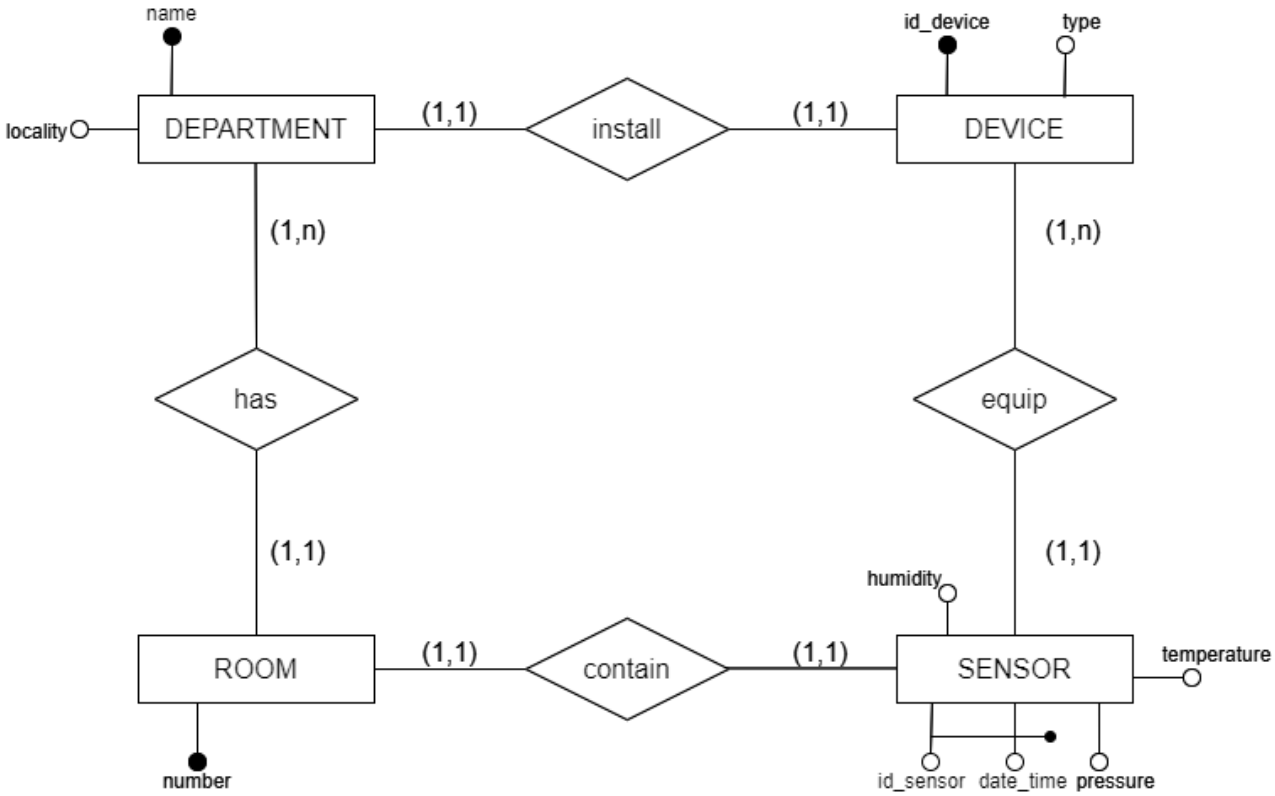
ADM PROJECT PART II

Author:

- Magno Alessandro: 4478234

Part 2.1

Conceptual schema for the domain identified in the first part.



Part 2.2

Workload related to the selected application:

1. Retrieve all the information of the sensors situated in the department named ‘Economia’
2. Retrieve the average temperature acquired by the sensors situated in the departments located in ‘Albaro’
3. Retrieve the maximum temperature acquired by the sensor contained in the room ‘701’
4. Retrieve the name of the departments which have devices of type ‘Arduino’
5. Retrieve the name of the departments and the rooms which has a temperature less than 15 degrees.

6. Retrieve the id sensor, the date and time, the temperature acquired by the sensors.
7. Retrieve the temperature, the humidity and the pressure acquired by the sensors in date timestamp 1643548195.
8. Retrieve all the rooms and id of the sensors which are equipped on devices of type ‘RaspberryPi’
9. Retrieve the humidity and pressure acquired by sensors in date timestamp 1643548325 situated in the departments named ‘DIMA’ or ‘DIBRIS’.
10. Retrieve the maximum and minimum temperature acquired by the sensors, the number of the rooms, the type of devices of all the departments located in ‘Darsena’

Part 2.3

I applied the methodology, having as input the previous ER schema and the workload.

Step 1

Each query in the workload is modeled in a formal and non-ambiguous way.

Q.1: E = Department

LS = [Department(name)_!]

LP = [Sensor_EI]

Q.2: E = Department

LS = [Department(locality)_!]

LP = [Sensor(temperature)_EI]

Q.3: E = Room

LS = [Room(number)_!]

LP = [Sensor(temperature)_C]

Q.4: E = Device

LS = [Device(type)_!]

LP = [Department(name)_I]

Q.5: E = Sensor

LS = [Sensor(temperature)_!]

LP = [Department(name)_HC, Room_C]

Q.6: E = Sensor

LS = []

LP = [Sensor(id_sensor, date_time, temperature)_!]

Q.7: E = Sensor

LS = [Sensor(date_time)_!]

LP = [Sensor(temperature, humidity, pressure)_!]

Q.8: E = Device

LS = [Device(type)_!]

LP = [Room_CE, Sensor(id_sensor)_E]

Q.9: E = Sensor

LS = [Sensor(date_time)_!, Department(name)_HC]

LP = [Sensor(humidity, pressure)_!]

Q.10: E = Department

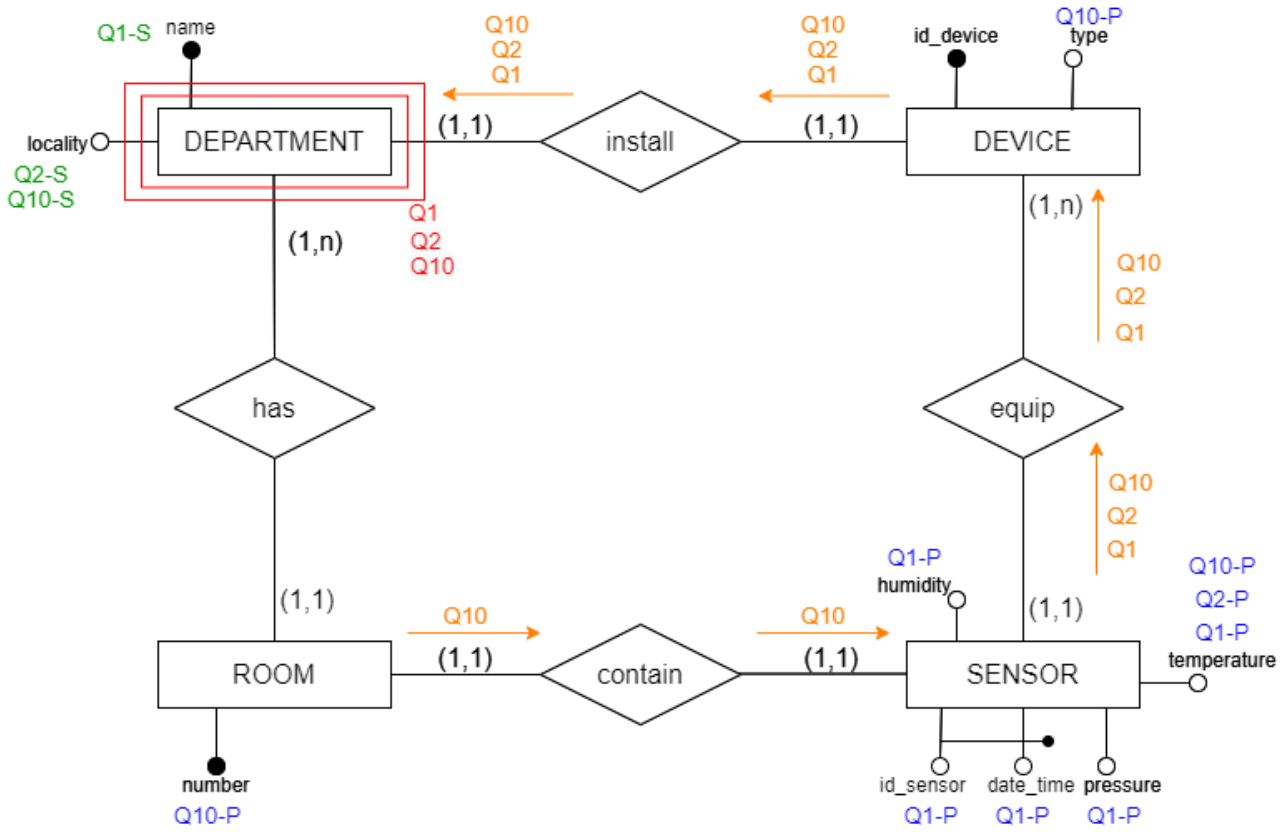
LS = [Department(locality)_!]

LP = [Sensor(temperature)_EI, Room_CEI, Device(type)_I]

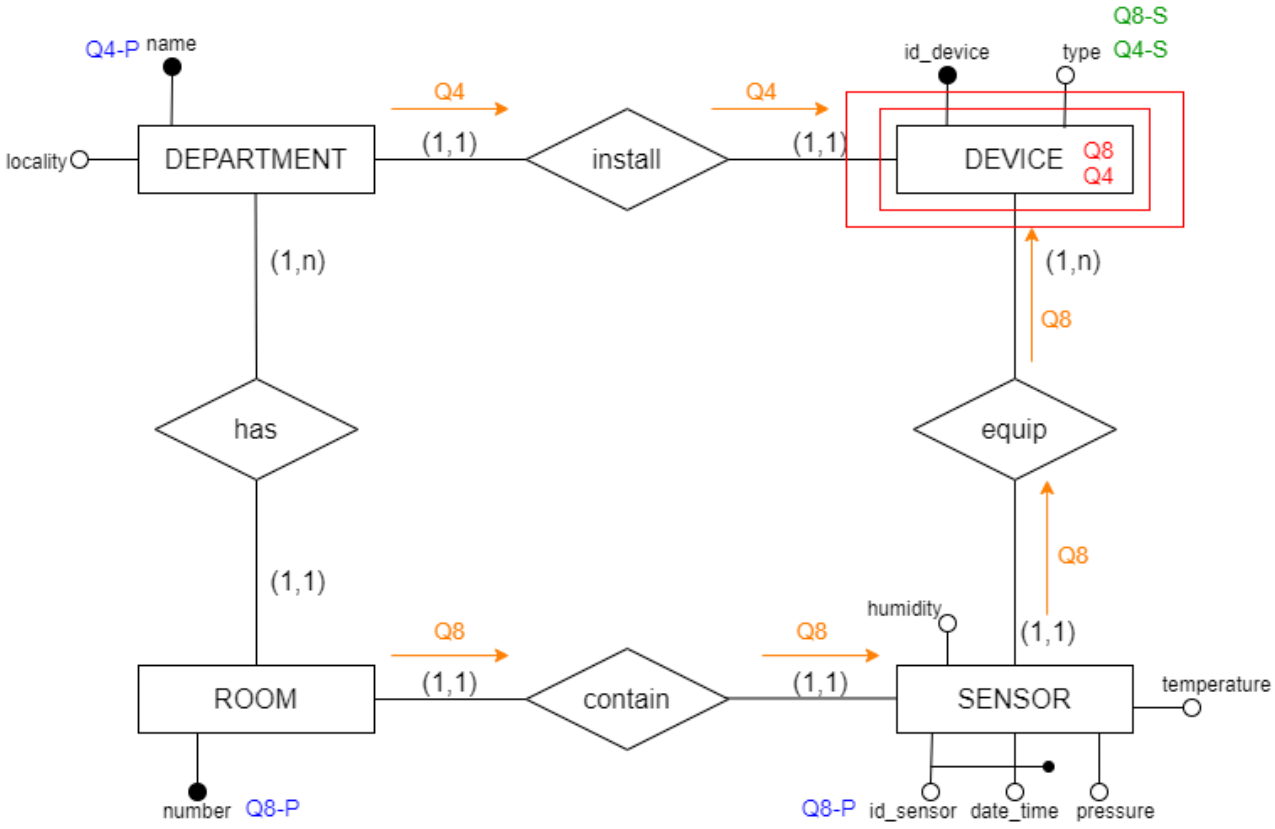
Step 2

The ER schema is annotated with query information and it is divided to make it more readable.

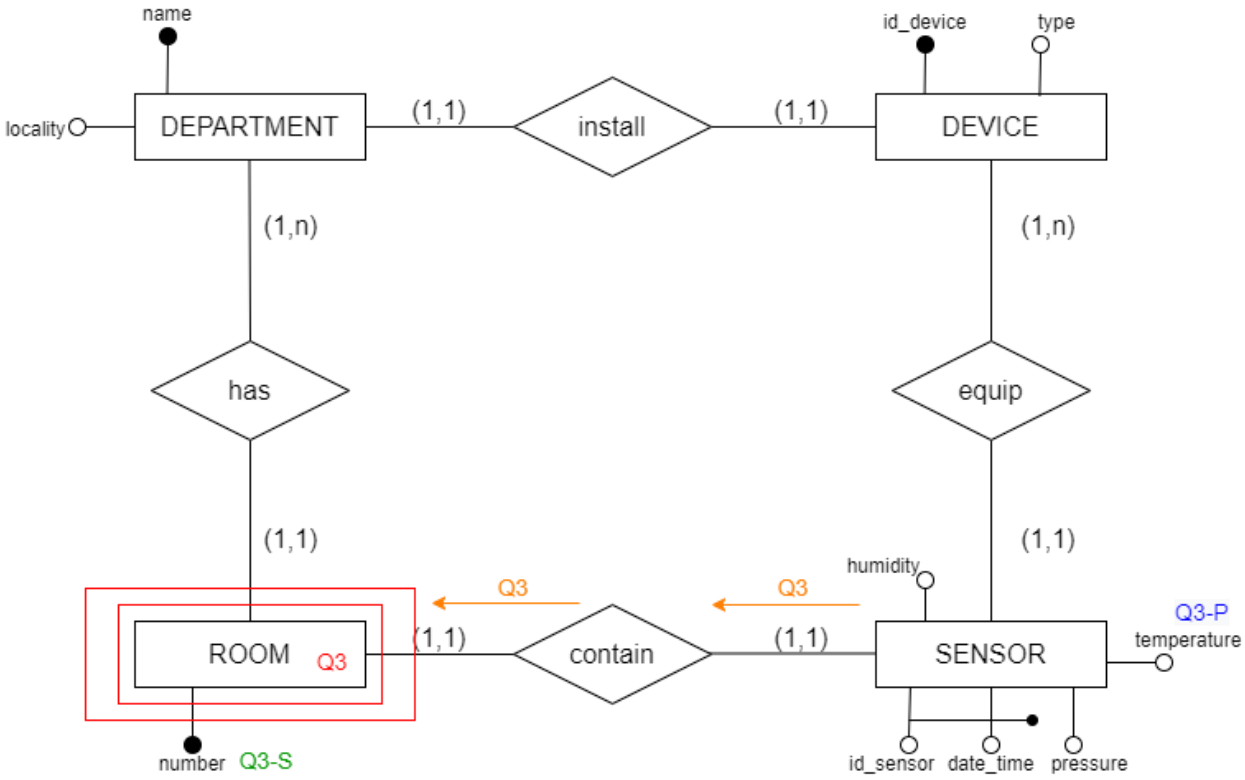
Q1, Q2, Q10 queries associated with entity Department.



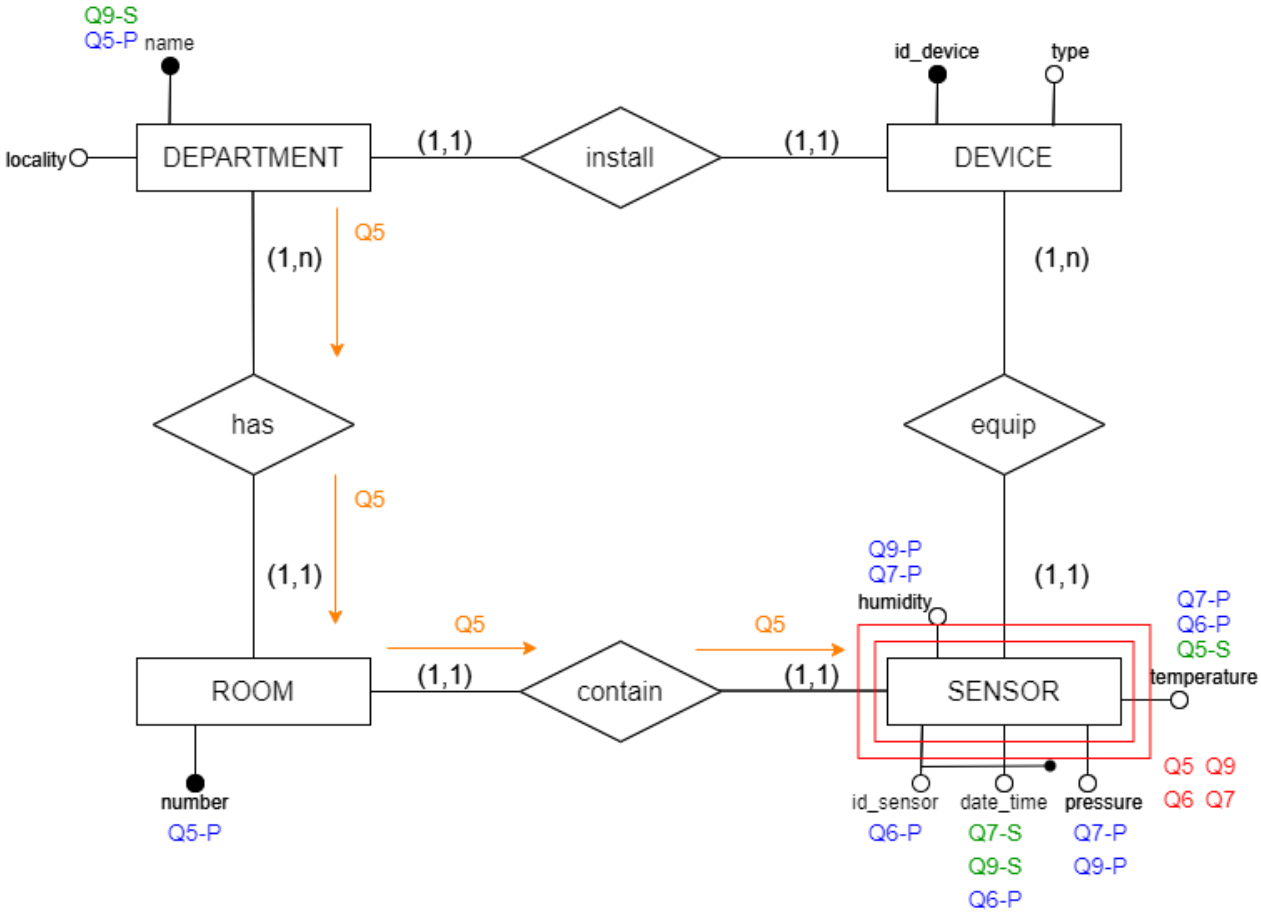
Q4, Q8 queries associated with entity Device.



Q3 query associated with entity Room.



Q5, Q6, Q7, Q9 queries associated with entity Sensor.



Step 3

The aggregated oriented logical schema is generated starting from the annotated ER schema.

department:

{name, locality, device_type, device_equips: [{id_sensor, date_time, temperature, pressure, humidity, nbrRoom}]}

device:

{type, name_department, equips: [{id_sensor, nbrRoom}]}

room:

{number, temperature}

sensor:

{id, date_time, pressure, temperature, humidity, nbrRoom, name_department}

Part 2.4

DEPARTMENT

Queries associated with Department: Q1, Q2, Q10

Selection attributes for Q1: {name}

Selection attributes for Q2: {locality}

Selection attributes for Q10: {locality}

Q1: Partition key = Primary key = {name}

Q2, Q10: it is not possible to create a table using the aggregate department in which we can execute the queries, because it requires to apply functions to subcomponent of device_equips. (we cannot access to temperature).

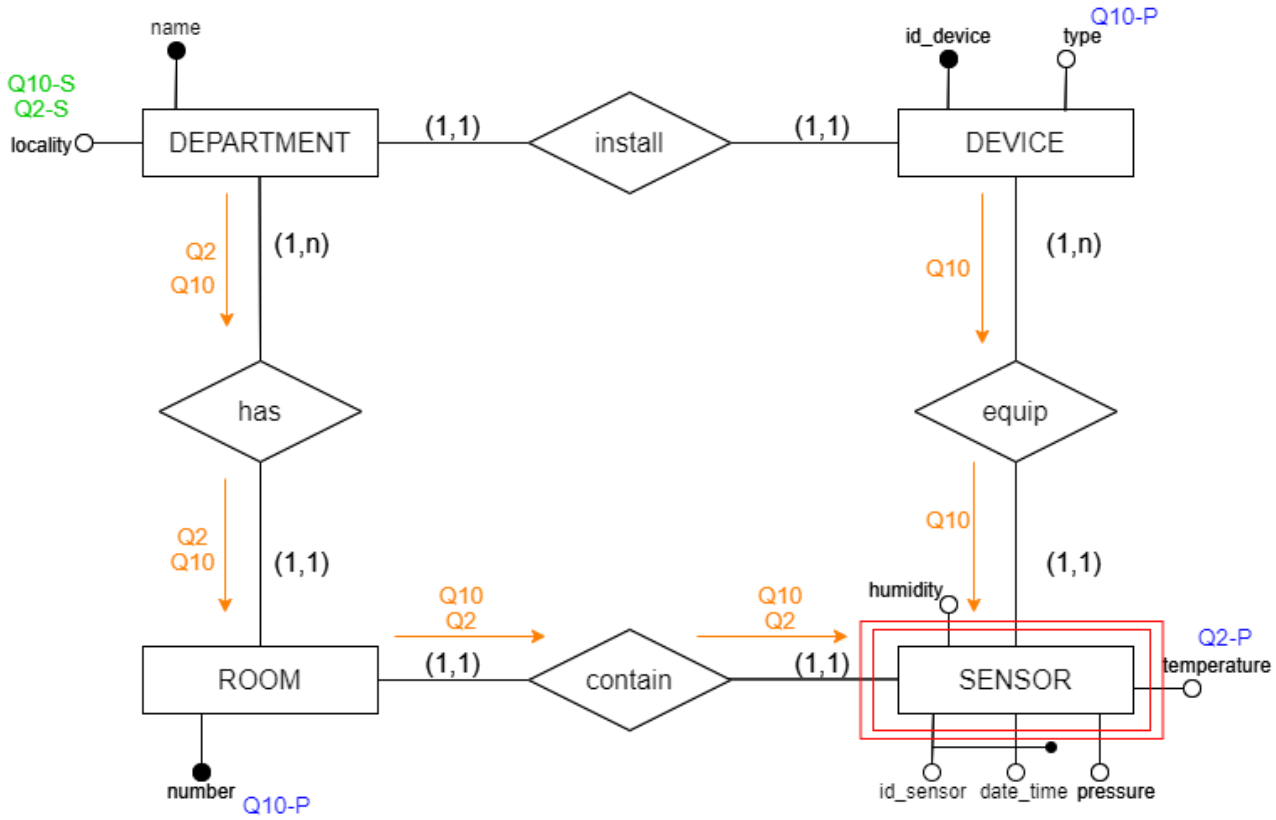
For Q1:

```
CREATE TYPE sensor_t (
        id_sensor float,
        datetime timestamp,
        temperature float,
        pressure float,
        humidity int,
        nbrRoom int
);
```

```
CREATE TABLE Department (
        name_department text PRIMARY KEY,
        device_type text,
        device_equips set<frozen<sensor_t>>
);
```

For Q2, Q10:

To solve the problem, we can go back to logical design and use as aggregation entity Sensor. In this way we are not incurring in (1, n) associations and we can avoid subcomponents.



Sensor: {temperature, nbrRoom, locality, device_type}

Partition key = {locality}

Primary key = {locality, name_department, id, datetime}

I added the name of the department, id_sensor, datetime so in this way the tuple is unique.

```
CREATE TABLE Sensor_by_locality (
        id_sensor float,
        datetime timestamp,
        temperature float,
        nbrRoom int,
        name_department text,
        locality text,
        device_type text,
        PRIMARY KEY(locality, name_department, id_sensor, datetime)
);
```

DEVICE

Queries associated with Device: Q4, Q8

Selection attributes for Q4: {type}

Selection attributes for Q8: {type}

Partition key = {type}

I added the id of device in the primary key, in this way it can be identified.

Primary key = {type, id}

```
CREATE TYPE sensorRoom_t (
        id_sensor float,
        nbrRoom int
);
```

```
CREATE TABLE Device (
        id_device int,
        device_type text,
        name_department text,
        equips set<frozen<sensorRoom_t>>,
        PRIMARY KEY(device_type, id_device)
);
```

ROOM

Queries associated with Room: Q3

Selection attributes for Q3: {nbrRoom}

Partition key = {nbrRoom}

I have to add in the primary key temperature because otherwise the insert would act as an update and I would lose almost all my data.

Primary key = {nbrRoom, temperature}

```
CREATE TABLE Room (
        nbrRoom int,
        temperature float,
        PRIMARY KEY(nbrRoom, temperature)
);
```

SENSOR

Queries associated with Sensor: Q5, Q6, Q7, Q9

Selection attributes for Q5: {temperature}

Selection attributes for Q6: {}

Selection attributes for Q7: {date_time}

Selection attributes for Q9: {date_time,name_department}

Q6: no selection attribute, no need for a specific partition key

Selection attributes of the query are disjoint, so to identify the partition key and primary key my idea was to build two different tables: one with PRIMARY KEY(date_time, name_department) for Q7, Q9, Q6 ; another with PRIMARY KEY(temperature, date_time, name_department) for Q5, Q6. I have tried this solution but did not work correctly, in the sense that also putting temperature as partition key, I could not filter it without the ALLOW FILTERING clause.

I realized that maybe I could define a single table, putting temperature in the clustering columns and creating an index for it, similar to an example in the slides. However also in this case, without the clause ALLOW FILTERING, I could not execute the query, because the previous columns are not restricted.

Partition key = {date_time}

Primary key = {date_time, name_depatment, temperature, id}

For Q5, Q6, Q7, Q9:

```
CREATE TABLE Sensor_by_datetime (
        id_sensor float,
        datetime timestamp,
        temperature float,
        pressure float,
        humidity int,
        nbrRoom int,
        name_department text,
        PRIMARY KEY(datetime, name_department, temperature, id_sensor)
);

CREATE INDEX ON Sensor_by_datetime(temperature);
```

Part 2.5

1. SELECT device_equips FROM department WHERE name_department = ‘Economia’;
2. SELECT AVG(temperature) FROM sensor_by_locality WHERE locality = ‘Albaro’;
3. SELECT MAX(temperature) FROM room WHERE room = 701;
4. SELECT name_department FROM device WHERE type = ‘Arduino’;
5. SELECT name_department, nbrRoom FROM sensor_by_datetime WHERE temperature < 15 ALLOW FILTERING;
6. SELECT id, datetime, temperature FROM sensor_by_datetime;
7. SELECT temperature, humidity, pressure FROM sensor_by_datetime WHERE datetime = 1643548195;
8. SELECT equips FROM device WHERE device_type = ‘Raspberry pi’;
9. SELECT humidity, pressure FROM sensor_by_datetime WHERE datetime = 1643548325 AND name_department IN (‘DIMA’, ‘DIBRIS’);
10. SELECT MAX(temperature), MIN(temperature), nbrRoom, device_type FROM department WHERE locality = ‘Darsena’ GROUP BY name_department, id_sensor;

Part 2.6

CREATE KEYSPACE ks_project_user17 WITH replication = {‘class’:’SimpleStrategy’, ‘replication factor’:’3’} AND durable_writes = true

SimpleStrategy is a basic replication strategy, used in a single datacenter to place replicas on subsequent nodes in a clockwise order. Replication factor equals to 3, means that data are replicated in 3 nodes which allow us to get high availability because we know that our data will always be written to at least three nodes. Setting durable_write to true, we reduce the speed of writes but also reduce the risk of data loss. I left the consistency to default (ONE), to have a low latency, which is another requirement with high availability, avoiding data loss, specified in part I.

Part 2.7

I generated my data randomly with a simple python script that produced the following columns: ['name_department', 'locality', 'id_device', 'device_type', 'id_sensor', 'temperature', 'humidity', 'pressure', 'datetime', 'numbRoom']. I split the dataset respect to the data presents in the different tables, then I have created in the cluster a folder adm_project and I uploaded my files with the following command:

C:\Users\Alessandro\Desktop\ADM\Progetto\PART_II\Data\dataset>scp name_file.csv user17@130.251.61.97:adm_project

After login in Cassandra, I have created the tables and copy the data from csv files to them using the COPY command.

An example:

COPY sensor_by_datetime (datetime, temperature, id_sensor, humidity, name_department, nbrroom, pressure) FROM 'adm_project/senso
r_by_datetime.csv' WITH HEADER = TRUE ;

Output

Using 7 child processes

Starting copy of ks_project_user17.sensor_by_datetime with columns [datetime, te
mperature, id_sensor, humidity, name_department, nbrroom, pressure].

Processed: 30000 rows; Rate: 18827 rows/s; Avg. rate: 10058 rows/s

30000 rows imported from 1 files in 2.983 seconds (0 skipped).

Part 2.8

The result of the implemented workload.

Q1:

user17@cqlsh:ks_project_user17> SELECT device_equips FROM department WHERE name_department = 'Economia';

 device_equips

{{id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 7.30852, pressure: 1013.49548, humidity: 41, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 10.20416, pressure: 1013.35126, humidity: 12, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 11.03704, pressure: 1013.45337, humidity: 15, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 13.64079, pressure: 1013.45374, humidity: 78, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 14.89998, pressure: 1013.31403, humidity: 85, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 23.60281, pressure: 1013.43341, humidity: 59, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 24.7974, pressure: 1013.48431, humidity: 99, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 27.16768, pressure: 1013.45892, humidity: 79, nbrroom: 701}, {id_sensor: 7.1, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 27.95645, pressure: 1013.54285, humidity: 37, nbrroom: 701}, {id_sensor: 7.2, datetime: '1970-01-20 00:32:28.290000+0000', temperature: 24.46399, pressure: 1013.37854, humidity: 39, nbrroom: 702}, {id_sensor: 7.2, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 18.6734, pressure: 1013.43097, humidity: 83, nbrroom: 702}, {id_sensor: 7.2, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 22.45013, pressure: 1013.37189, humidity: 43, nbrroom: 702}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 13.3944, pressure: 1013.25488, humidity: 12, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 13.70595, pressure: 1013.54114, humidity: 74, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 14.5528, pressure: 1013.39288, humidity: 64, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 17.44148, pressure: 1013.35571, humidity: 38, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 17.44383, pressure: 1013.29089, humidity: 10, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 21.0213, pressure: 1013.29156, humidity: 9, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 23.11766, pressure: 1013.51752, humidity: 3, nbrroom: 703}, {id_sensor: 7.3, datetime: '1970-01-20 00:32:28.291000+0000', temperature: 24.4779, pressure: 1013.38434, humidity: 38, nbrroom: 703}}

(1 rows)

Q2:

user17@cqlsh:ks_project_user17> SELECT AVG(temperature) FROM sensor_by_locality WHERE locality = 'Albaro';

system.avg(temperature)

15.72688

(1 rows)

Q3:

user17@cqlsh:ks_project_user17> SELECT MAX(temperature) FROM Room WHERE nbrroom = 701;

system.max(temperature)

27.9967

(1 rows)

Q4:

```
user17@cqlsh:ks_project_user17> SELECT name_department FROM device WHERE device_type =
'Arduino';


 name_department
-----------------
           DIFI
           DCCI
           DIME
        Economia
          DIRAAS
           DISPO


(6 rows)
```

Q5:

```
user17@cqlsh:ks_project_user17> SELECT name_department, nbrRoom FROM sensor_by_datetime
WHERE temperature < 15 ALLOW FILTERING;


 name_department | nbrroom
-----------------+---------
        Economia |     702
        Economia |     701
          DIRAAS |     902
        Economia |     703
             DAD |     603
             DAD |     602
             DAD |     601
            DIME |     501
            DIME |     502
           DISFOR |    1001
             DAD |     603
          DIBRIS |     401
        Economia |     703
```

DIMA | 203
DISFOR | 1003
DIFI | 102
Economia | 703
DAD | 601
DIRAAS | 901
DIBRIS | 401
DISFOR | 1002
Economia | 701
DISPO | 1101
DIRAAS | 903
DIME | 501
DIRAAS | 903
DIRAAS | 901
DIMA | 202
DCCI | 303
DIMA | 203
Economia | 701
DIME | 501
DAFIST | 801
DISPO | 1103
Economia | 702
Economia | 701
DIRAAS | 901
DISFOR | 1002
DIRAAS | 901
Economia | 703
Economia | 702
DISFOR | 1002
DAD | 601
DAD | 603
Economia | 701
DISFOR | 1001
DAD | 603
Economia | 701

DAFIST | 802

DCCI | 302

DISPO | 1102

DIME | 503

DIRAAS | 903

DIRAAS | 903

DISPO | 1102

DCCI | 301

Economia | 701

DISFOR | 1002

DISPO | 1102

DISFOR | 1002

Economia | 701

DIFI | 101

Economia | 701

DIME | 501

DISFOR | 1002

Economia | 703

DIMA | 202

DIME | 502

DISFOR | 1003

DAD | 602

DAFIST | 802

DISFOR | 1003

DAFIST | 801

DAD | 602

DIRAAS | 901

Economia | 702

Economia | 702

Economia | 701

DISFOR | 1003

DIME | 501

DIRAAS | 903

DIME | 503

DISPO | 1103

DISPO | 1103

DAD | 603

DCCI | 301

DIRAAS | 901

DAFIST | 803

DISFOR | 1003

DIMA | 201

DAD | 602

DIME | 501

DIRAAS | 902

DISPO | 1103

DIBRIS | 401

DAD | 602

DIBRIS | 401

DAFIST | 803

DAD | 602

DIME | 502

---MORE---

Q6:

user17@cqlsh:ks_project_user17> SELECT id_sensor, datetime, temperature FROM sensor_by_datetime;

```
 id_sensor | datetime                        | temperature
-----------+---------------------------------+-------------
       7.2 | 1970-01-20 00:32:28.305000+0000 |     7.05751
       7.1 | 1970-01-20 00:32:28.305000+0000 |     7.32222
       9.2 | 1970-01-20 00:32:28.305000+0000 |     7.66789
       7.3 | 1970-01-20 00:32:28.305000+0000 |     7.67951
       6.3 | 1970-01-20 00:32:28.305000+0000 |     7.69415
       6.2 | 1970-01-20 00:32:28.305000+0000 |     7.70957
       6.1 | 1970-01-20 00:32:28.305000+0000 |     7.79491
       5.1 | 1970-01-20 00:32:28.305000+0000 |     7.97912
       5.2 | 1970-01-20 00:32:28.305000+0000 |     8.20382
      10.1 | 1970-01-20 00:32:28.305000+0000 |     8.36267
```

6.3 | 1970-01-20 00:32:28.305000+0000 | 8.4613
4.1 | 1970-01-20 00:32:28.305000+0000 | 8.67684
7.3 | 1970-01-20 00:32:28.305000+0000 | 8.99808
2.3 | 1970-01-20 00:32:28.305000+0000 | 9.09893
10.3 | 1970-01-20 00:32:28.305000+0000 | 9.14018
1.2 | 1970-01-20 00:32:28.305000+0000 | 9.26506
7.3 | 1970-01-20 00:32:28.305000+0000 | 9.83184
6.1 | 1970-01-20 00:32:28.305000+0000 | 10.07575
9.1 | 1970-01-20 00:32:28.305000+0000 | 10.82056
4.1 | 1970-01-20 00:32:28.305000+0000 | 11.04441
10.2 | 1970-01-20 00:32:28.305000+0000 | 11.05214
7.1 | 1970-01-20 00:32:28.305000+0000 | 11.54043
11.1 | 1970-01-20 00:32:28.305000+0000 | 11.84195
9.3 | 1970-01-20 00:32:28.305000+0000 | 11.88149
5.1 | 1970-01-20 00:32:28.305000+0000 | 12.39474
9.3 | 1970-01-20 00:32:28.305000+0000 | 12.49174
9.1 | 1970-01-20 00:32:28.305000+0000 | 12.58923
2.2 | 1970-01-20 00:32:28.305000+0000 | 12.73151
3.3 | 1970-01-20 00:32:28.305000+0000 | 13.10441
2.3 | 1970-01-20 00:32:28.305000+0000 | 13.65034
7.1 | 1970-01-20 00:32:28.305000+0000 | 13.72166
5.1 | 1970-01-20 00:32:28.305000+0000 | 13.75566
8.1 | 1970-01-20 00:32:28.305000+0000 | 14.06926
11.3 | 1970-01-20 00:32:28.305000+0000 | 14.33739
7.2 | 1970-01-20 00:32:28.305000+0000 | 14.44853
7.1 | 1970-01-20 00:32:28.305000+0000 | 14.85003
9.1 | 1970-01-20 00:32:28.305000+0000 | 14.91875
10.2 | 1970-01-20 00:32:28.305000+0000 | 14.99263
7.2 | 1970-01-20 00:32:28.305000+0000 | 15.22577
7.1 | 1970-01-20 00:32:28.305000+0000 | 15.28915
6.3 | 1970-01-20 00:32:28.305000+0000 | 15.37688
7.2 | 1970-01-20 00:32:28.305000+0000 | 15.38704
5.2 | 1970-01-20 00:32:28.305000+0000 | 16.02216
7.3 | 1970-01-20 00:32:28.305000+0000 | 16.12416
10.1 | 1970-01-20 00:32:28.305000+0000 | 16.57281

 8.1 | 1970-01-20 00:32:28.305000+0000 | 16.63388
11.2 | 1970-01-20 00:32:28.305000+0000 | 17.01883
 9.1 | 1970-01-20 00:32:28.305000+0000 | 17.2193
 7.1 | 1970-01-20 00:32:28.305000+0000 | 17.25396
10.1 | 1970-01-20 00:32:28.305000+0000 | 17.41817
10.1 | 1970-01-20 00:32:28.305000+0000 | 17.46963
 3.2 | 1970-01-20 00:32:28.305000+0000 | 17.69282
 8.2 | 1970-01-20 00:32:28.305000+0000 | 17.85608
10.2 | 1970-01-20 00:32:28.305000+0000 | 17.91061
10.2 | 1970-01-20 00:32:28.305000+0000 | 17.95483
 7.3 | 1970-01-20 00:32:28.305000+0000 | 18.23798
 7.3 | 1970-01-20 00:32:28.305000+0000 | 18.50944
 7.1 | 1970-01-20 00:32:28.305000+0000 | 18.62832
11.2 | 1970-01-20 00:32:28.305000+0000 | 18.74299
 8.1 | 1970-01-20 00:32:28.305000+0000 | 18.76014
 5.2 | 1970-01-20 00:32:28.305000+0000 | 18.81331
 8.2 | 1970-01-20 00:32:28.305000+0000 | 18.87079
 6.3 | 1970-01-20 00:32:28.305000+0000 | 18.96017
 2.2 | 1970-01-20 00:32:28.305000+0000 | 19.03164
 7.3 | 1970-01-20 00:32:28.305000+0000 | 19.1524
 7.3 | 1970-01-20 00:32:28.305000+0000 | 19.19021
 2.3 | 1970-01-20 00:32:28.305000+0000 | 19.618
 9.1 | 1970-01-20 00:32:28.305000+0000 | 19.69877
 9.3 | 1970-01-20 00:32:28.305000+0000 | 19.73406
10.3 | 1970-01-20 00:32:28.305000+0000 | 19.94007
 7.1 | 1970-01-20 00:32:28.305000+0000 | 20.07427
11.2 | 1970-01-20 00:32:28.305000+0000 | 20.09658
 5.1 | 1970-01-20 00:32:28.305000+0000 | 20.40035
 4.1 | 1970-01-20 00:32:28.305000+0000 | 20.59781
 8.3 | 1970-01-20 00:32:28.305000+0000 | 20.65068
10.1 | 1970-01-20 00:32:28.305000+0000 | 20.84567
 1.2 | 1970-01-20 00:32:28.305000+0000 | 21.1361
10.2 | 1970-01-20 00:32:28.305000+0000 | 21.14792
 7.3 | 1970-01-20 00:32:28.305000+0000 | 21.51109
 1.3 | 1970-01-20 00:32:28.305000+0000 | 21.54437

2.1 | 1970-01-20 00:32:28.305000+0000 | 21.93248

10.1 | 1970-01-20 00:32:28.305000+0000 | 22.075

8.1 | 1970-01-20 00:32:28.305000+0000 | 22.19606

5.1 | 1970-01-20 00:32:28.305000+0000 | 22.19814

5.3 | 1970-01-20 00:32:28.305000+0000 | 22.75341

8.3 | 1970-01-20 00:32:28.305000+0000 | 22.79683

7.3 | 1970-01-20 00:32:28.305000+0000 | 23.25957

8.3 | 1970-01-20 00:32:28.305000+0000 | 23.86631

7.3 | 1970-01-20 00:32:28.305000+0000 | 24.08961

11.2 | 1970-01-20 00:32:28.305000+0000 | 24.11917

11.1 | 1970-01-20 00:32:28.305000+0000 | 24.5311

5.1 | 1970-01-20 00:32:28.305000+0000 | 24.53412

9.2 | 1970-01-20 00:32:28.305000+0000 | 24.60193

7.3 | 1970-01-20 00:32:28.305000+0000 | 24.65035

10.3 | 1970-01-20 00:32:28.305000+0000 | 24.88384

5.1 | 1970-01-20 00:32:28.305000+0000 | 25.55671

11.3 | 1970-01-20 00:32:28.305000+0000 | 25.7781

5.2 | 1970-01-20 00:32:28.305000+0000 | 26.15572

5.3 | 1970-01-20 00:32:28.305000+0000 | 26.27339

5.2 | 1970-01-20 00:32:28.305000+0000 | 26.54708

---MORE---

Q7:

user17@cqlsh:ks_project_user17> SELECT temperature, humidity, pressure FROM sensor_by_datetime WHERE datetime = 1643548195;

temperature | humidity | pressure

7.01485 | 74 | 1013.31085

7.56116 | 93 | 1013.38562

8.32734 | 83 | 1013.3996

8.37231 | 42 | 1013.4986

8.37942 | 46 | 1013.36023

8.59874 | 76 | 1013.36401

8.97567 | 5 | 1013.42877

9.89556 | 50 | 1013.26813

12.67719 | 36 | 1013.3114

13.23302 | 43 | 1013.3028

13.4419 | 16 | 1013.32379

15.42248 | 34 | 1013.52985

15.55293 | 24 | 1013.55035

17.6698 | 19 | 1013.51935

18.06746 | 57 | 1013.37646

18.31624 | 96 | 1013.46161

18.92161 | 79 | 1013.54401

19.80246 | 53 | 1013.33215

20.39707 | 45 | 1013.45685

20.41026 | 92 | 1013.49384

21.04141 | 30 | 1013.59229

21.29911 | 97 | 1013.49915

21.8527 | 62 | 1013.43237

22.08734 | 80 | 1013.46594

22.38761 | 91 | 1013.46606

22.57025 | 20 | 1013.57648

23.13941 | 80 | 1013.50836

24.80265 | 35 | 1013.39459

27.00056 | 41 | 1013.47363

27.28751 | 3 | 1013.44946

27.57076 | 64 | 1013.45135

27.67275 | 4 | 1013.34125

(32 rows)

Q8:

user17@cqlsh:ks_project_user17> SELECT equips FROM device WHERE device_type = 'Raspberry pi';

equips

{{id_sensor: 2.2, nbrroom: 202}}

{{id_sensor: 4.2, nbrroom: 402}}

{{id_sensor: 6.1, nbrroom: 601}}

{{id_sensor: 8.1, nbrroom: 801}}

{{id_sensor: 10.1, nbrroom: 1001}, {id_sensor: 10.2, nbrroom: 1002}, {id_sensor: 10.3, nbrroom: 1003}}

(5 rows)

Q9:

user17@cqlsh:ks_project_user17> SELECT humidity, pressure FROM sensor_by_datetime WHERE datetime = 1643548325 AND name_department IN ('DIMA','DIBRIS');

```
 humidity | pressure
----------+------------
       90 | 1013.44397
       77 | 1013.35651

(2 rows)
```

Q10:

user17@cqlsh:ks_project_user17> SELECT MAX(temperature), MIN(temperature), nbrRoom, device_type FROM sensor_by_locality WHERE locality = 'Darsena' GROUP BY name_department,id_sensor;

```
 system.max(temperature) | system.min(temperature) | nbrroom | device_type
-------------------------+-------------------------+---------+-------------
               27.79669 |                  7.2763 |     701 |     Arduino
               27.92086 |                 7.28155 |     702 |     Arduino
               27.87935 |                 7.18826 |     703 |     Arduino

(3 rows)
```