--creating a table with appropriate data types and primary key

CREATE TABLE de\_challenge.dbc\_hotel\_ratings(

hotel\_address varchar(5000) NOT NULL,

hotel\_city varchar(50) NOT NULL,

hotel\_country varchar(50) NOT NULL,

keys serial not null,

hotel\_latitude float NOT NULL,

hotel\_longitude float NOT NULL,

hotel\_name varchar(1000) NOT NULL,

hotel\_postcode varchar(50) NOT NULL,

hotel\_province varchar(50) NOT NULL,

review\_date date NOT NULL,

review\_rating float NOT NULL,

review\_text varchar(100000) NOT NULL,

review\_title varchar(500) NOT NULL,

user\_city varchar(50) NOT NULL,

user\_province\_or\_country varchar(50) NOT NULL,

username varchar(50) NOT NULL,

PRIMARY KEY (keys)

);

-- Granting permission to users - studentX does not exist (it is just an example)

GRANT SELECT, INSERT, UPDATE on TABLE de\_challenge.dbc\_hotel\_ratings To "studentX";

--creating a rule for ignoring duplicates

CREATE OR REPLACE RULE ignore\_duplicates

AS ON INSERT

TO de\_challenge.dbc\_hotel\_ratings

WHERE (EXISTS ( SELECT 1

FROM de\_challenge.dbc\_hotel\_ratings dim

WHERE

dim.hotel\_address = NEW.hotel\_address and

dim.hotel\_name = NEW.hotel\_name and

dim.hotel\_city = NEW.hotel\_city and

dim.hotel\_country = NEW.hotel\_country and

dim.hotel\_latitude = NEW.hotel\_latitude and

dim.hotel\_longitude = NEW.hotel\_longitude and

dim.hotel\_postcode = NEW.hotel\_postcode and

dim.hotel\_province = NEW.hotel\_province and

dim.review\_date = NEW.review\_date and

dim.review\_rating = NEW.review\_rating and

dim.review\_text = NEW.review\_text and

dim.review\_title = NEW.review\_title

and dim.user\_city = NEW.user\_city and

dim.username = NEW.username and

dim.user\_province\_or\_country = NEW.user\_province\_or\_country))

DO INSTEAD NOTHING;

--USER TESTING: (using SELECT, UPDATE and INSERT as these keywords are the only ones available for users)

--UPDATE WORKS

UPDATE de\_challenge.dbc\_hotel\_ratings

SET hotel\_address = 'address'

WHERE keys=1;

SELECT keys, hotel\_address FROM de\_challenge.dbc\_hotel\_ratings WHERE keys=1;

--INSERT WORKS WHEN HAVING UNIQUE KEYS and new information for at least one row

INSERT INTO de\_challenge.dbc\_hotel\_ratings(hotel\_address, hotel\_city, hotel\_country, keys, hotel\_latitude, hotel\_longitude, hotel\_name, hotel\_postcode, hotel\_province, review\_date, review\_rating, review\_text, review\_title, user\_city, user\_province\_or\_country, username)

VALUES

('address',

'city',

'country',

27844,

-15.15,

20.15,

'name',

'postcode',

'province',

'2061-02-02',

2.3,

23,

'title',

'usercity',

'userprovince',

'usernameses'

);

SELECT \* FROM de\_challenge.dbc\_hotel\_ratings WHERE keys=27844;

--INSERT DOES NOT WORK WITHOUT UNIQUE KEYS

INSERT INTO de\_challenge.dbc\_hotel\_ratings(hotel\_address, hotel\_city, hotel\_country, keys, hotel\_latitude, hotel\_longitude, hotel\_name, hotel\_postcode, hotel\_province, review\_date, review\_rating, review\_text, review\_title, user\_city, user\_province\_or\_country, username)

VALUES

('address',

'city',

'country',

27844,

-15.15,

20.15,

'name',

'postcode',

'province',

'2061-02-02',

2.3,

23,

'title',

'usercity',

'userprovince',

'usernameses'

);

--INSERT does not work when attempting to duplicate rows under a unique key

INSERT INTO de\_challenge.dbc\_hotel\_ratings(hotel\_address, hotel\_city, hotel\_country, keys, hotel\_latitude, hotel\_longitude, hotel\_name, hotel\_postcode, hotel\_province, review\_date, review\_rating, review\_text, review\_title, user\_city, user\_province\_or\_country, username)

VALUES

('address',

'city',

'country',

27849,

-15.15,

20.15,

'name',

'postcode',

'province',

'2061-02-02',

2.3,

23,

'title',

'usercity',

'userprovince',

'usernameses'

);

--SELECTING THE LAST KEY (useful when wanting to insert new rows)

SELECT

max(last\_value) FROM (SELECT LAST\_value(keys) OVER(PARTITION BY hotel\_address) FROM de\_challenge.dbc\_hotel\_dim\_int) attempt;