Accessing_DataBase_with_SQL_Magic

February 14, 2022

0.1 Accessing Database with SQL Magic

In this section we will see the use of SQL magic functions that are utilized in python Jupyter notebook to perform SQL queries. For single line comment we use % sign and for whole cell to be converted to SQl format we use %%. Firstly, we will use *load_ext* magic to load the ipython sql extension.

[34]:
```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

[35]:
```
# Enter your Db2 credentials in the connection string below
# i.e. from the uri field in the Service Credentials copy everything after db2:/
 ↪/ (but remove the double quote at the end)
# for example, if your credentials are as in the screenshot above, you would␣
 ↪write:
# %sql ibm_db_sa://my-username:my-password@hostname:port/BLUDB?security=SSL
# Note the ibm_db_sa:// prefix instead of db2://
# This is because JupyterLab's ipython-sql extension uses sqlalchemy (a python␣
 ↪SQL toolkit)
# which in turn uses IBM's sqlalchemy dialect: ibm_db_sa

%sql ibm_db_sa://trc40191:ptmIhOTtWcaA8SsK@9938aec0-8105-433e-8bf9-0fbb7e483086.
 ↪c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUDB?security=SSL
```

Creating a new table called INTERNATIONAL_STUDENT_TEST_SCORES on IBM cloud instace through using SQL magic operator. Following table data is provided by IBM coursework lab.

[36]:
```
%%sql

CREATE TABLE INTERNATIONAL_STUDENT_TEST_SCORES (
        country VARCHAR(50),
        first_name VARCHAR(50),
        last_name VARCHAR(50),
        test_score INT
);
```

```
INSERT INTO INTERNATIONAL_STUDENT_TEST_SCORES (country, first_name, last_name,␣
↪test_score)
VALUES
('United States', 'Marshall', 'Bernadot', 54),
('Ghana', 'Celinda', 'Malkin', 51),
('Ukraine', 'Guillermo', 'Furze', 53),
('Greece', 'Aharon', 'Tunnow', 48),
('Russia', 'Bail', 'Goodwin', 46),
('Poland', 'Cole', 'Winteringham', 49),
('Sweden', 'Emlyn', 'Erricker', 55),
('Russia', 'Cathee', 'Sivewright', 49),
('China', 'Barny', 'Ingerson', 57),
('Uganda', 'Sharla', 'Papaccio', 55),
('China', 'Stella', 'Youens', 51),
('Poland', 'Julio', 'Buesden', 48),
('United States', 'Tiffie', 'Cosely', 58),
('Poland', 'Auroora', 'Stiffell', 45),
('China', 'Clarita', 'Huet', 52),
('Poland', 'Shannon', 'Goulden', 45),
('Philippines', 'Emylee', 'Privost', 50),
('France', 'Madelina', 'Burk', 49),
('China', 'Saunderson', 'Root', 58),
('Indonesia', 'Bo', 'Waring', 55),
('China', 'Hollis', 'Domotor', 45),
('Russia', 'Robbie', 'Collip', 46),
('Philippines', 'Davon', 'Donisi', 46),
('China', 'Cristabel', 'Radeliffe', 48),
('China', 'Wallis', 'Bartleet', 58),
('Moldova', 'Arleen', 'Stailey', 38),
('Ireland', 'Mendel', 'Grumble', 58),
('China', 'Sallyann', 'Exley', 51),
('Mexico', 'Kain', 'Swaite', 46),
('Indonesia', 'Alonso', 'Bulteel', 45),
('Armenia', 'Anatol', 'Tankus', 51),
('Indonesia', 'Coralyn', 'Dawkins', 48),
('China', 'Deanne', 'Edwinson', 45),
('China', 'Georgiana', 'Epple', 51),
('Portugal', 'Bartlet', 'Breese', 56),
('Azerbaijan', 'Idalina', 'Lukash', 50),
('France', 'Livvie', 'Flory', 54),
('Malaysia', 'Nonie', 'Borit', 48),
('Indonesia', 'Clio', 'Mugg', 47),
('Brazil', 'Westley', 'Measor', 48),
('Philippines', 'Katrinka', 'Sibbert', 51),
('Poland', 'Valentia', 'Mounch', 50),
('Norway', 'Sheilah', 'Hedditch', 53),
('Papua New Guinea', 'Itch', 'Jubb', 50),
```

```
('Latvia', 'Stesha', 'Garnson', 53),
('Canada', 'Cristionna', 'Wadmore', 46),
('China', 'Lianna', 'Gatward', 43),
('Guatemala', 'Tanney', 'Vials', 48),
('France', 'Alma', 'Zavittieri', 44),
('China', 'Alvira', 'Tamas', 50),
('United States', 'Shanon', 'Peres', 45),
('Sweden', 'Maisey', 'Lynas', 53),
('Indonesia', 'Kip', 'Hothersall', 46),
('China', 'Cash', 'Landis', 48),
('Panama', 'Kennith', 'Digance', 45),
('China', 'Ulberto', 'Riggeard', 48),
('Switzerland', 'Judy', 'Gilligan', 49),
('Philippines', 'Tod', 'Trevaskus', 52),
('Brazil', 'Herold', 'Heggs', 44),
('Latvia', 'Verney', 'Note', 50),
('Poland', 'Temp', 'Ribey', 50),
('China', 'Conroy', 'Egdal', 48),
('Japan', 'Gabie', 'Alessandone', 47),
('Ukraine', 'Devlen', 'Chaperlin', 54),
('France', 'Babbette', 'Turner', 51),
('Czech Republic', 'Virgil', 'Scotney', 52),
('Tajikistan', 'Zorina', 'Bedow', 49),
('China', 'Aidan', 'Rudeyeard', 50),
('Ireland', 'Saunder', 'MacLice', 48),
('France', 'Waly', 'Brunstan', 53),
('China', 'Gisele', 'Enns', 52),
('Peru', 'Mina', 'Winchester', 48),
('Japan', 'Torie', 'MacShirrie', 50),
('Russia', 'Benjamen', 'Kenford', 51),
('China', 'Etan', 'Burn', 53),
('Russia', 'Merralee', 'Chaperlin', 38),
('Indonesia', 'Lanny', 'Malam', 49),
('Canada', 'Wilhelm', 'Deeprose', 54),
('Czech Republic', 'Lari', 'Hillhouse', 48),
('China', 'Ossie', 'Woodley', 52),
('Macedonia', 'April', 'Tyer', 50),
('Vietnam', 'Madelon', 'Dansey', 53),
('Ukraine', 'Korella', 'McNamee', 52),
('Jamaica', 'Linnea', 'Cannam', 43),
('China', 'Mart', 'Coling', 52),
('Indonesia', 'Marna', 'Causbey', 47),
('China', 'Berni', 'Daintier', 55),
('Poland', 'Cynthia', 'Hassell', 49),
('Canada', 'Carma', 'Schule', 49),
('Indonesia', 'Malia', 'Blight', 48),
('China', 'Paulo', 'Seivertsen', 47),
```

```
('Niger', 'Kaylee', 'Hearley', 54),
('Japan', 'Maure', 'Jandak', 46),
('Argentina', 'Foss', 'Feavers', 45),
('Venezuela', 'Ron', 'Leggitt', 60),
('Russia', 'Flint', 'Gokes', 40),
('China', 'Linet', 'Conelly', 52),
('Philippines', 'Nikolas', 'Birtwell', 57),
('Australia', 'Eduard', 'Leipelt', 53)
```

```
 * ibm_db_sa://trc40191:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32459/BLUDB
Done.
99 rows affected.
```

[36]:
```
[]
```

[37]:
```
country = "Canada"
%sql select * from INTERNATIONAL_STUDENT_TEST_SCORES where country = :country
```

```
 * ibm_db_sa://trc40191:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32459/BLUDB
Done.
```

[37]:
```
[('Canada', 'Cristionna', 'Wadmore', 46),
 ('Canada', 'Wilhelm', 'Deeprose', 54),
 ('Canada', 'Carma', 'Schule', 49)]
```

0.1.1 Assigning results of Queries to python variable

[38]:
```
test_score_distribution = %sql SELECT test_score as "score", count(*) as␣
 ↪"frequency" from INTERNATIONAL_STUDENT_TEST_SCORES GROUP BY test_score;
test_score_distribution
```

```
 * ibm_db_sa://trc40191:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32459/BLUDB
Done.
```

[38]:
```
[(38, 2),
 (40, 1),
 (43, 2),
 (44, 2),
 (45, 8),
 (46, 7),
 (47, 4),
 (48, 14),
 (49, 8),
 (50, 10),
```

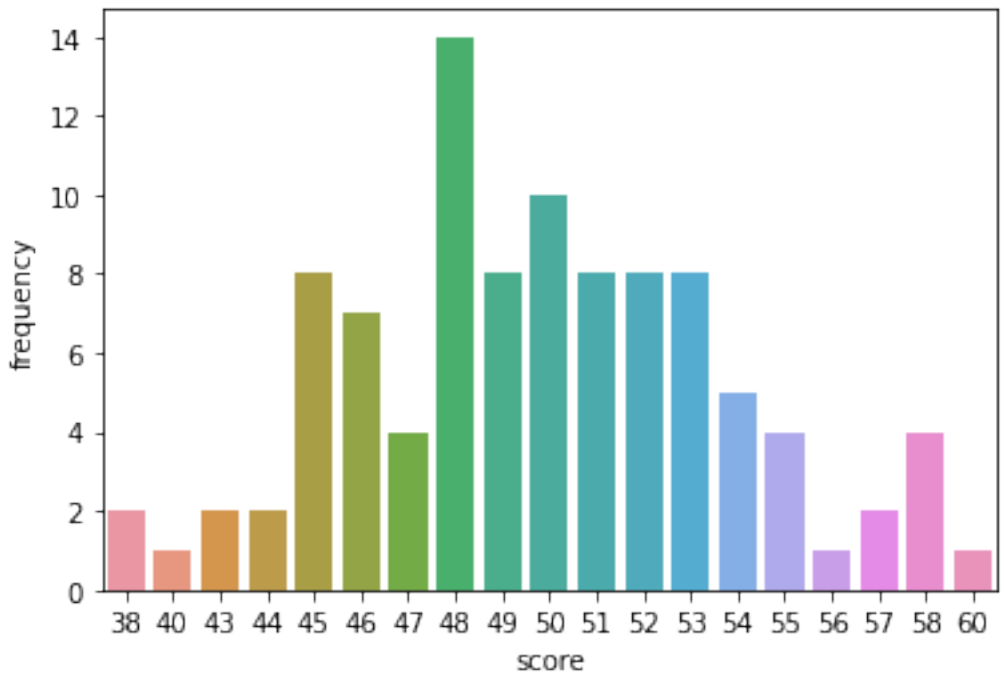
```
 (51, 8),
 (52, 8),
 (53, 8),
 (54, 5),
 (55, 4),
 (56, 1),
 (57, 2),
 (58, 4),
 (60, 1)]
```

0.1.2 Converting Query results to Dataframe

[39]:
```
df = test_score_distribution.DataFrame()

%matplotlib inline
# uncommment the following line if you get an module error saying seaborn not␣
 ↪found
# !pip install seaborn==0.9.0
import seaborn

plot = seaborn.barplot(x=df['score'],y='frequency', data=df)
```



[45]:
```
%%sql
```

```
-- displaying more data with magic functions
SELECT country, first_name, last_name, test_score FROM␣
 ↪INTERNATIONAL_STUDENT_TEST_SCORES LIMIT 5;
```

```
 * ibm_db_sa://trc40191:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32459/BLUDB
Done.
```

[45]:
```
[('United States', 'Marshall', 'Bernadot', 54),
 ('Ghana', 'Celinda', 'Malkin', 51),
 ('Ukraine', 'Guillermo', 'Furze', 53),
 ('Greece', 'Aharon', 'Tunnow', 48),
 ('Russia', 'Bail', 'Goodwin', 46)]
```

[ ]: