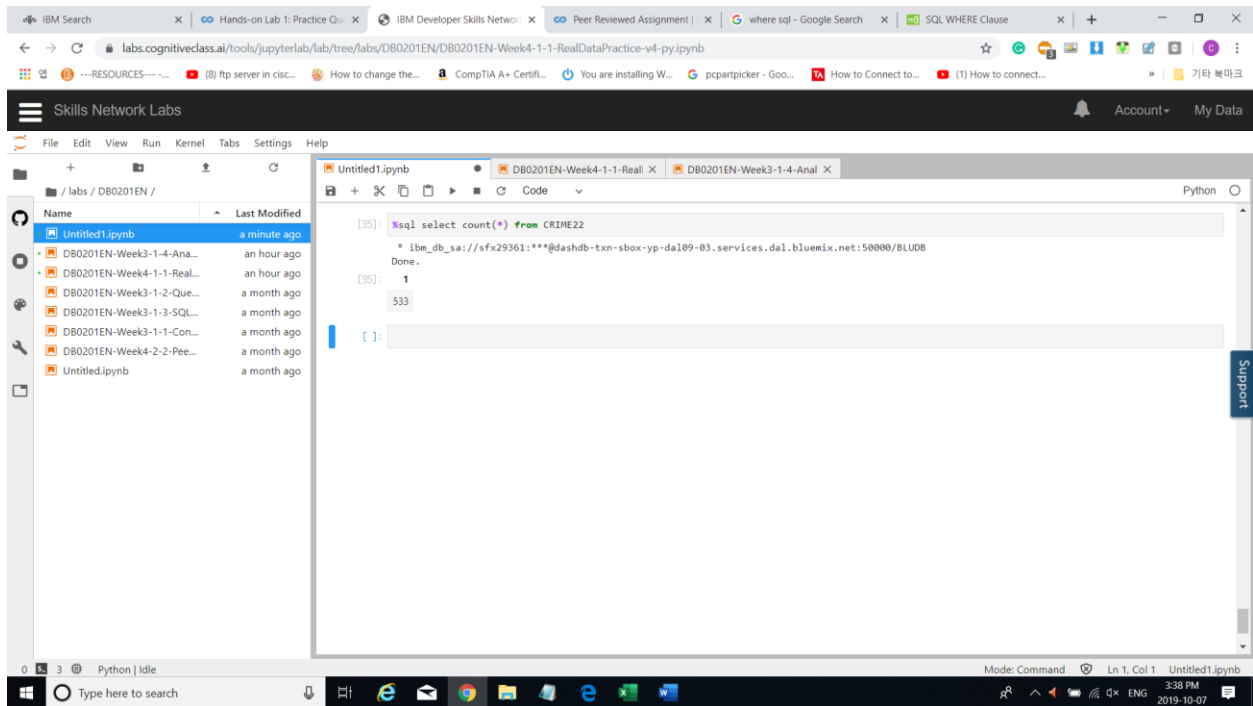


Problem 1: Find the total number of crimes recorded in the crime table.

```
%sql select count(*) from CRIME22
```



Problem 2: Retrieve first 10 rows from the CRIME table.

```
%sql select * from CRIME22 fetch first 10 rows only
```

The screenshot shows a Jupyter Notebook titled 'Untitled1.ipynb' in a browser window. The notebook is running a SQL query to fetch the first 10 rows from the CRIME22 table. The query is: `%sql select * from CRIME22 fetch first 10 rows only`. The results are displayed as a table with the following columns: `id`, `case_number`, `DATE`, `block`, `lucr`, `primary_type`, `description`, `location_description`, `arrest`, `domestic`, `beat`, `district`, `ward`, `community_area_number`, `fbcode`, and `x_coordinate`.

id	case_number	DATE	block	lucr	primary_type	description	location_description	arrest	domestic	beat	district	ward	community_area_number	fbcode	x_coordinate
3512276	HK587712	08/28/2004 05:50:56 PM	047XX S KEDZIE AVE	890	THEFT	FROM BUILDING	SMALL RETAIL STORE	FALSE	FALSE	911	9	14	58	6	1155838
3406613	HK456306	06/26/2004 12:40:00 PM	009XX N CENTRAL PARK AVE	820	THEFT	\$500 AND UNDER	OTHER	FALSE	FALSE	1112	11	27	23	6	1152206
8002131	HT233595	04/04/2011 05:45:00 AM	043XX S WABASH AVE	820	THEFT	\$500 AND UNDER	NURSING HOME/RETIREMENT HOME	FALSE	FALSE	221	2	3	38	6	1177436
7903289	HT133522	12/30/2010 04:30:00 PM	083XX S KINGSTON AVE	840	THEFT	FINANCIAL ID THEFT, OVER \$300	RESIDENCE	FALSE	FALSE	423	4	7	46	6	1194622
10402076	HZ138551	02/02/2016 07:30:00 PM	033XX W 66TH ST	820	THEFT	\$500 AND UNDER	ALLEY	FALSE	FALSE	831	8	15	66	6	1155240
7732712	H5540106	09/29/2010 07:59:00 AM	006XX W CHICAGO AVE	810	THEFT	OVER \$500	PARKING LOT/GARAGE(NON-RESID.)	FALSE	FALSE	1323	12	27	24	6	1171668
10769475	HZ534771	11/30/2016 01:15:00 AM	050XX N KEDZIE AVE	810	THEFT	OVER \$500	STREET	FALSE	FALSE	1713	17	33	14	6	1154133
		12/16/2005	005XX E												

Problem 3: How many crimes involve an arrest.

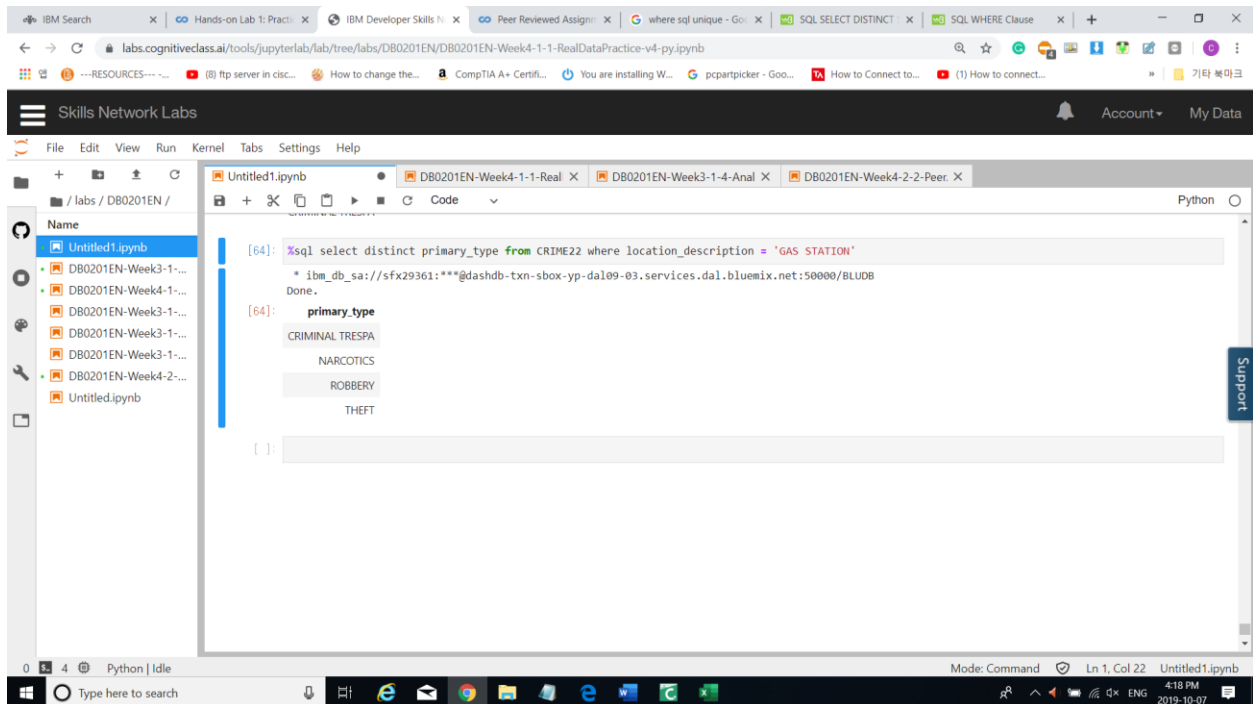
`%sql select count(*) from CRIME22 where ARREST = TRUE`

The screenshot shows a Jupyter Notebook titled 'Untitled1.ipynb' in a browser window. The notebook is running a SQL query to count the number of crimes where the ARREST column is TRUE. The query is: `%sql select count(*) from CRIME22 where ARREST = TRUE`. The result is displayed as a single row with the value 163.

count(*)
163

Problem 4: Which unique types of crimes (e.g. THEFT) have been recorded at a GAS STATION locations?

```
%sql select distinct primary_type from CRIME22 where location_description = 'GAS STATION'
```



The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The code editor contains a SQL query and its output. The query is: `%sql select distinct primary_type from CRIME22 where location_description = 'GAS STATION'`. The output shows a table with one column, `primary_type`, and four rows of data: `CRIMINAL TRESPA`, `NARCOTICS`, `ROBBERY`, and `THEFT`.

```
[64]: %sql select distinct primary_type from CRIME22 where location_description = 'GAS STATION'
* ibm_db_sa://sf29361:***@dashdb-txn-sbox-yp-dal09-03.services.dal.ibm.com:50000/BLUDB
Done.

[64]:
primary_type
CRIMINAL TRESPA
NARCOTICS
ROBBERY
THEFT
```

Problem 5: In the CENSUS_DATA table list all community areas whose names start with the letter 'B'.

```
%sql select * from CENSUS11 where COMMUNITY_AREA_NAME like 'B%'
```

The screenshot shows a Jupyter Notebook interface within the IBM Skills Network Labs environment. The notebook is titled 'Untitled1.ipynb'. The code cell contains a SQL query: `%sql select * from CENSUS11 where COMMUNITY_AREA_NAME like 'B%'`. The output shows the results of the query, which are displayed as a table with columns: `community_area_number`, `community_area_name`, `percent_of_housing_crowded`, `percent_households_below_poverty`, `percent_aged_16_unemployed`, and `percent_aged_25_without_high_school`. The results show data for community areas 19, 47, 58, 60, and 72.

community_area_number	community_area_name	percent_of_housing_crowded	percent_households_below_poverty	percent_aged_16_unemployed	percent_aged_25_without_high_school
19	Belmont Cragin	10.8	18.7	14.6	
47	Burnside	6.8	33.0	18.6	
58	Brighton Park	14.4	23.6	13.9	
60	Bridgeport	4.5	18.9	13.7	
72	Beverly	0.9	5.1	8.0	

Problem 6: List the schools in community areas 10 to 15 that are healthy school certified.

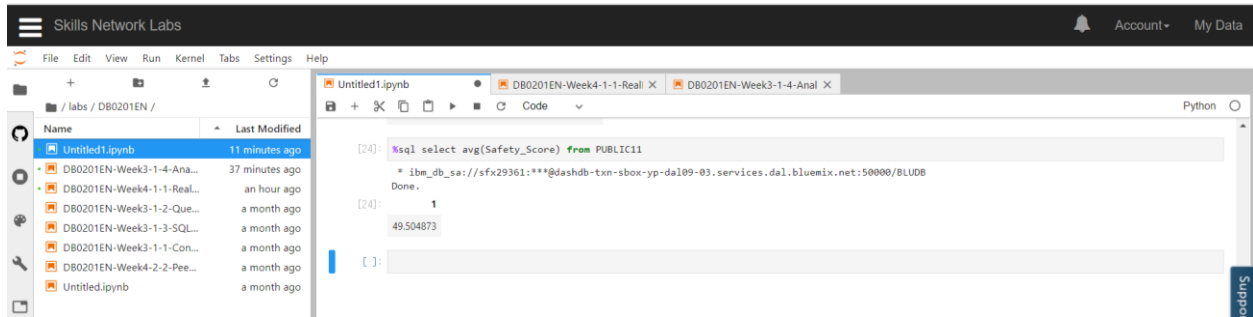
```
%sql select * from PUBLIC11 where HEALTHY_SCHOOL_CERTIFIED = 'Yes' and
COMMUNITY_AREA_NUMBER in (10,11,12,13,14,15)
```

The screenshot shows a Jupyter Notebook interface within the IBM Skills Network Labs environment. The notebook is titled 'Untitled1.ipynb'. The code cell contains a SQL query: `%sql select * from PUBLIC11 where HEALTHY_SCHOOL_CERTIFIED = 'Yes' and COMMUNITY_AREA_NUMBER in (10,11,12,13,14,15)`. The output shows the results of the query, which are displayed as a table with columns: `School_ID`, `name_of_school`, `Elementary, Middle, or High School`, `Street_Address`, `City`, `State`, `ZIP_Code`, and `Phone_Number`. The results show data for school 609995, Rufus M Hitch Elementary School.

School_ID	name_of_school	Elementary, Middle, or High School	Street_Address	City	State	ZIP_Code	Phone_Number
609995	Rufus M Hitch Elementary School	ES	5625 N McVicker Ave	Chicago	IL	60646	(773) 534-1189

Problem 7: What is the average school Safety Score?

```
%sql select avg(Safety_Score) from PUBLIC11
```



Problem 8: Find the top 5 Community Areas by average College Enrollment [number of students].

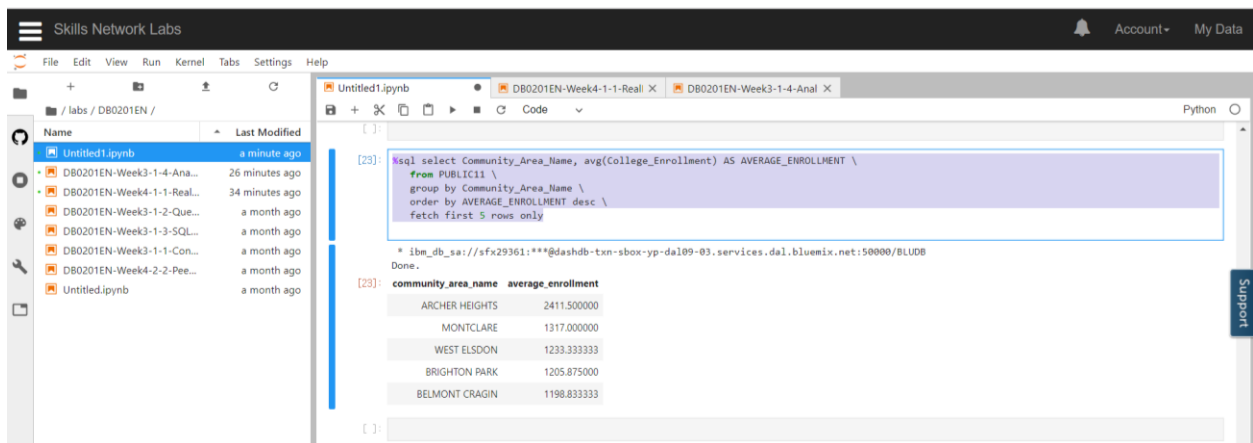
```
%sql select Community_Area_Name, avg(College_Enrollment) AS AVERAGE_ENROLLMENT \

from PUBLIC11 \

group by Community_Area_Name \

order by AVERAGE_ENROLLMENT desc \

fetch first 5 rows only
```



Problem 9: Use a sub-query to determine which Community Area has the least value for school Safety Score.

```
%sql select Community_Area_Name, Safety_score \

from PUBLIC11 \

order by Safety_score asc

fetch first 5 rows only
```

```
%sql select community_area_name from PUBLIC11 where safety_score < 20
```

```
%sql select distinct primary_type from CRIME22
```

```
\ where (select from PUBLIC11 where safety_score < 90)
```

Problem 10: [Without using an explicit JOIN operator] Find the Per Capita Income of the Community Area which has a school Safety Score of 1.

1. **SELECT** column-names
2. **FROM** table-name1
3. **WHERE** value **IN** (**SELECT** column-name
4. **FROM** table-name2
5. **WHERE** condition)

```
%sql select communy_area_name, per_capita_income
```

```
from CENSUS11
```

```
where community_area_name in ( select community_area_name \
```

```
from PUBLIC11 \
```

```
where safey_score = 1 )
```