Individual Project 6 (SMP)

Author: Sheridan Payne
Version: 1.0
Semester: Spring 2021

```
USE crime_data;
```

1. Write a query to extract the number of different crime types are in the incident_reports table. Name the output column unique_crimes.

```
SELECT DISTINCT(crime_type) AS unique_crimes
FROM incident_reports
GROUP BY crime_type;
```

16 rows returned

2. Write a query to extract the number of incidents for each crime type in the incident_reports table. The output column for count of each crime type should be called num_crimes and the results should be shorted alphabetically (ascending) by crime type

```
SELECT DISTINCT(crime_type) AS unique_crimes, COUNT(crime_type) AS num_crimes
FROM incident_reports
GROUP BY crime_type
ORDER BY crime_type ASC;
```

16 rows returned

3. Write a query to extract the number of incidents that were reported on the same day they occurred. You will need to determine how to get the difference of date columns in MySQL.

```
SELECT COUNT(date_occured)
FROM incident_reports
WHERE DATEDIFF(date_reported, date_occured) = 0;
```

Answer: 961309 incidents

4. Write a query to find the amount of time between the date an incident was reported and the date it occurred. The result should include the date it was reported, the date it occurred, the crime type and the difference in years. In addition, the result should only include differences that were one year or more. Sort the result so the longest distance appears at the top.

```
SELECT date_occured, date_reported, crime_type, DATEDIFF(date_reported, date_occured)/365
AS year_diff
FROM incident_reports
WHERE DATEDIFF(date_reported, date_occured)/365 >= 1
ORDER BY DATEDIFF(date_reported, date_occured)/365 DESC;
```

9928 rows returned

5. Write the query to get a count by year of all crimes from the past 10 years (e.g. 2021 – 2012). Your results should be sorted so 2021 appears at the top and your columns should be named year and num_incidents.

```
SELECT YEAR(date_occured) AS Year, COUNT(YEAR(date_occured)) AS num_incidents
FROM incident_reports
WHERE YEAR(date_occured) >= '2012'
GROUP BY YEAR(date_occured)
ORDER BY YEAR(date_occured) DESC;
```

10 rows returned

6. Write the query to return all columns in rows in which the crime type is robbery.

```
SELECT *
FROM incident_reports
WHERE crime_type LIKE 'ROBBERY';
```

27411 rows returned

7. Write the query to return the LMPD division, Incident Number, and date the incident occurred (in that order) for attempted robberies. Order the results by the LMPD Division and the date (ascending for both).

```
SELECT lmpd_division, incident_number, date_occured
FROM incident_reports
WHERE att_comp LIKE 'ATTEMPTED' AND crime_type LIKE 'ROBBERY'
ORDER BY lmpd_division ASC, date_occured;
```

2575 rows returned

8. Write the query to return the date the incident occurred and the type of crime for the zip code 40202. Order the results by the type of crime and then by the date the incident occurred.

```
SELECT date_occured, crime_type
FROM incident_reports
WHERE zip_code = '40202'
ORDER BY crime_type ASC, date_occured DESC;
```

60295 rows returned

9. Write the query to determine which zip code has the highest number of vehicle thefts. In your query, include the zip code and a number of incidents in that zip code in a column named num_thefts. Sort the output so the zip code with the highest number of thefts appears at the top. For this, report the query, the number of results, and the zip code with the highest number of thefts.

```
SELECT zip_code, COUNT(crime_type) AS num_thefts
FROM incident_reports
WHERE crime_type LIKE 'MOTOR%' AND LENGTH(zip_code)=5
GROUP BY zip_code
ORDER BY num_thefts DESC
LIMIT 1;
```

Answer: 4566 thefts for zip code 40214

10. Write the query to determine how many different cities are the reported in the Incident Reports.

```
SELECT COUNT(DISTINCT(city))
FROM incident_reports;
```

Answer: 234 cities

11. Write the query to determine which city had the second highest number of incidents behind Louisville. Your query should only include the city and the number of incidents and be sorted by the number of incidents. For this report the query, the city with the second highest number of incidents and the number. Comment on if you see anything weird in your results.

```
SELECT city, COUNT(incident_number)
FROM incident_reports
GROUP BY city
ORDER BY COUNT(incident_number) DESC;
```

Answer: LVIL with 37299 incidents
It was strange was that there were two different Louisville's in the data set (Louisville and LVIL)

12. Write the query to return Uniform Offense Reporting code and the type of crime order in which the type of crime is not OTHER. Order the results by the UOR code and then by crime type. In your result, explain what the difference between the two columns seem to be.

```
SELECT DISTINCT(nibrs_code), crime_type
FROM incident_reports
WHERE crime_type NOT LIKE 'OTHER'
ORDER BY nibrs_code ASC, crime_type DESC;
```

48 rows returned

The first column (nibrs_code) is the official code that classifies what crime type it is, but I believe there are different levels for each code, since “Homicide” is classified under three different codes (09A, 09B, 09C).

13. Write the query to determine how many LMPD beats there are.

```
SELECT COUNT(DISTINCT(lmpd_beat))
FROM incident_reports;
```

Answer: 60 LMPD beats

14. Write the query to determine how many NIBRS codes exist (in the nibrs_codes table).

```
SELECT COUNT(DISTINCT(offense_code))
FROM nibrs_codes;
```

Answer: 61 NIBRS codes

15. Write the query to determine how many of the NIBRS codes appear in the Incident Reports table. You are finding UNIQUE NIBRS codes.

```
SELECT COUNT(DISTINCT(nibrs_code))
FROM incident_reports;
```

Answer: 54 NIBRS codes

16. Write the query to list the date the incident occurred, the Block address, the zip code and the NIBRS Offense Description. Retrieve only rows for NIBRS codes 240, 250, 270, and 280. Sort the results by the Block address.

```
SELECT date_occured, block_address, zip_code, uor_desc
FROM incident_reports
WHERE nibrs_code = '240' OR '250' OR '270' OR '280'
ORDER BY block_address ASC;
```

1377767 rows returned

17. Write the query to show the zip code and the type of entity the offense was against. In your results, remove any rows with the NIBRS code = 999 or an invalid zip code. You can do that with the LENGTH function in MySQL to find all zip codes that 5 digits or more. Sort the results by zip code.

```
SELECT zip_code, offense_against
FROM nibrs_codes
JOIN incident_reports ON nibrs_codes.offense_code = incident_reports.nibrs_code
WHERE LENGTH(zip_code) = 5 AND LENGTH(offense_against) > 0 AND offense_code NOT LIKE
999
ORDER BY zip_code DESC;
```

1309550 rows returned

18. Write the query show a count of each number of offense against various entities (e.g. Not a Crime, Property, etc.) Order the results by the offense against column in the nibrs_codes table. Remove any rows for which the offense against value is empty.

```
SELECT offense_against, COUNT(offense_against)
FROM nibrs_codes
WHERE LENGTH(offense_against) > 0
GROUP BY offense_against;
```

6 rows returned

19. Write a single table query of your own choosing that I haven't already asked for that performs an aggregate query of some type and that restricts the result row with a HAVING clause.

```
SELECT DISTINCT(offense_category), COUNT(offense_category)
FROM nibrs_codes
GROUP BY offense_category
HAVING COUNT(offense_category) > 4;
```

2 rows returned

20. Write a multi-table query of your own choosing that I haven't already asked for that show data from both tables, which applies a sort of some type and restricts rows by some criteria.

```
SELECT incident_reports.date_occured, nibrs_codes.offense_description
FROM incident_reports
JOIN nibrs_codes ON nibrs_codes.offense_code = incident_reports.nibrs_code
WHERE YEAR(date_occured) = 2021
ORDER BY date_occured DESC;
```

7818 rows returned