Data Definition Language (DDL) commands:

```
CREATE TABLE Crime (CrimeID INT NOT NULL AUTO_INCREMENT,
LocationID INT, VictimID INT, TypeID INT, Time INT, PRIMARY KEY
(CrimeID), FOREIGN KEY (LocationID) REFERENCES
Location(LocationID), FOREIGN KEY (VictimID) REFERENCES
Victim(VictimID), FOREIGN KEY (TypeID) REFERENCES
CrimeDesc(TypeID));

CREATE TABLE Location (LocationID INT NOT NULL AUTO_INCREMENT,
Address VARCHAR(255), Longitude DOUBLE, Latitude DOUBLE,
AreaName VARCHAR(255), PRIMARY KEY (LocationID));

CREATE TABLE Victim (VictimID INT NOT NULL, Age INT, Sex
VARCHAR(10), Ethnicity VARCHAR(255), PRIMARY KEY (VictimID));

CREATE TABLE CrimeDesc (TypeID INT NOT NULL, TypeName
VARCHAR(255), Weapon VARCHAR(255), PRIMARY KEY (TypeID));
```

Main tables:

```
mysql> show tables;
+----------------------+
| Tables_in_411project |
+----------------------+
| Crime                |
| CrimeDesc            |
| Location             |
| Victim               |
+----------------------+
4 rows in set (0.01 sec)
```

Main table lengths:

```
mysql> show tables;
+----------------------+
| Tables_in_411project |
+----------------------+
| Crime                |
| CrimeDesc            |
| Location             |
| Victim               |
+----------------------+
4 rows in set (0.01 sec)

mysql> SELECT COUNT(CrimeID) FROM Crime;
+-----------------+
| COUNT(CrimeID) |
+-----------------+
|           1775 |
+-----------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(TypeID) FROM CrimeDesc;
+---------------+
| COUNT(TypeID) |
+---------------+
|            75 |
+---------------+
1 row in set (0.00 sec)

mysql> SELECT COUNT(LocationID) FROM Location;
+-------------------+
| COUNT(LocationID) |
+-------------------+
|              1775 |
+-------------------+
1 row in set (0.01 sec)

mysql> SELECT COUNT(VictimID) FROM Victim;
+-----------------+
| COUNT(VictimID) |
+-----------------+
|            2052 |
+-----------------+
1 row in set (0.01 sec)
```

SQL Queries:

```
-- First query is demonstrating use of JOIN
-- Second query is demonstrating use of GROUP BY and JOIN
-- Third query is demonstrating use of Set operations,
Subqueries, and Join of multiple relations.
```

First Query: JOIN

```
SELECT loc.Latitude, loc.Longitude
FROM Crime c NATURAL JOIN Location loc
WHERE c.Time BETWEEN 0000 AND 0600;
```

Top 15 rows:

```
mysql> SELECT loc.Latitude, loc.Longitude
    -> FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID
    -> WHERE c.Time BETWEEN 0000 AND 0600
    -> Limit 15;
+----------+-----------+
| Latitude | Longitude |
+----------+-----------+
|  34.0459 | -118.2545 |
|  34.2198 | -118.4468 |
|  34.0452 | -118.2534 |
|  34.0448 | -118.2474 |
|  34.0677 | -118.2398 |
|  33.9019 | -118.2916 |
|  33.9144 | -118.2894 |
|  33.9451 | -118.4029 |
|  34.0542 | -118.2566 |
|  34.0583 | -118.2378 |
|  34.2075 | -118.5068 |
|  34.1946 | -118.3835 |
|  34.0317 | -118.2626 |
|  34.0221 | -118.4166 |
|  34.0736 | -118.2156 |
+----------+-----------+
15 rows in set (0.01 sec)
```

Indexing:

Created indexing on time in the Crime table since the query will be searching through a range of time. The nested loop inner join for the unindexed version has a higher cost than the indexed version (248.27 and 233.06). The table scan on c costs more than the Index range scan on c which is the expected behavior.

Not indexed:

```
mysql> EXPLAIN ANALYZE SELECT loc.Latitude, loc.Longitude FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID WHERE c.Time BETWEEN 0000 AND 0600;
+----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
| EXPLAIN

+----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
| -> Nested loop inner join  (cost=248.27 rows=197) (actual time=0.072..1.103 rows=291 loops=1)
    -> Filter: ((c.`Time` between 0 and 600) and (c.LocationID is not null))  (cost=179.25 rows=197) (actual time=0.054..0.603 rows=291 loops=1)
        -> Table scan on c  (cost=179.25 rows=1775) (actual time=0.049..0.475 rows=1775 loops=1)
    -> Single-row index lookup on loc using PRIMARY (LocationID=c.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=291)
 |
+----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

Indexed:

```
mysql> CREATE INDEX time_asc ON Crime(time ASC);
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> EXPLAIN ANALYZE SELECT loc.Latitude, loc.Longitude
    -> FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID
    -> WHERE c.Time BETWEEN 0000 AND 0600;
+----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------+
| EXPLAIN

                                  |
+----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------+
| -> Nested loop inner join  (cost=233.06 rows=291) (actual time=0.243..1.075 rows=291 loops=1)
    -> Filter: (c.LocationID is not null)  (cost=131.21 rows=291) (actual time=0.230..0.557 rows=291 loops=1)
        -> Index range scan on c using time_asc, with index condition: (c.`Time` between 0 and 600)  (cost=131.21 rows=291) (actual time=0.228..0.527 rows=291 loops=1)
    -> Single-row index lookup on loc using PRIMARY (LocationID=c.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=291)
 |
+----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------+
1 row in set (0.00 sec)
```

Second Query: GROUP BY and JOIN

SELECT loc.AreaName, COUNT(loc.LocationID)
FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID
WHERE c.Time < 0600
GROUP BY loc.AreaName;

Top 15 rows:

```
mysql> SELECT loc.AreaName, COUNT(loc.LocationID)
    -> FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID
    -> WHERE c.Time < 0600
    -> GROUP BY loc.AreaName
    -> LIMIT 15;
+-------------+----------------------+
| AreaName    | COUNT(loc.LocationID) |
+-------------+----------------------+
| Central     |                  172 |
| Mission     |                    3 |
| Southeast   |                    2 |
| Pacific     |                    4 |
| West Valley |                    2 |
| N Hollywood |                    5 |
| Hollenbeck  |                    7 |
| Olympic     |                    5 |
| Newton      |                    2 |
| Topanga     |                    3 |
| Northeast   |                    2 |
| Southwest   |                    4 |
| Wilshire    |                    3 |
| Hollywood   |                    3 |
| 77th Street |                    3 |
+-------------+----------------------+
15 rows in set (0.01 sec)
```

Indexing:

Created indexing on Time in the Crime table since the query will be searching through a range of time. The nested loop inner join for the unindexed version has a higher cost than the indexed version. The table scan on c costs more than Index range scan on c which is the expected behavior. The indexing with Crime.Time seems to cost the least.

Not indexed:

```
mysql> EXPLAIN ANALYZE SELECT loc.AreaName, COUNT(loc.LocationID)
    -> FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID
    -> WHERE c.Time < 0600
    -> GROUP BY loc.AreaName;
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
| EXPLAIN


+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
| -> Table scan on <temporary>  (actual time=0.001..0.002 rows=20 loops=1)
    -> Aggregate using temporary table  (actual time=1.430..1.433 rows=20 loops=1)
        -> Nested loop inner join  (cost=386.31 rows=592) (actual time=0.070..1.228 rows=275 loops=1)
            -> Filter: ((c.`Time` < 600) and (c.LocationID is not null))  (cost=179.25 rows=592) (actual time=0.053..0.687 rows=275 loops=1)
                -> Table scan on c  (cost=179.25 rows=1775) (actual time=0.048..0.545 rows=1775 loops=1)
            -> Single-row index lookup on loc using PRIMARY (LocationID=c.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=275)
 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

Indexing with Location.LocationID and Crime.LocationID and Crime.Time:

```
mysql> CREATE INDEX loc1 ON Location(LocationID);
ERROR 1061 (42000): Duplicate key name 'loc1'
mysql> CREATE INDEX loc ON Crime(LocationID);
ERROR 1061 (42000): Duplicate key name 'loc'
mysql> CREATE INDEX time ON Crime(time);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> EXPLAIN ANALYZE SELECT loc.AreaName, COUNT(loc.LocationID) FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID WHERE c.Time < 0600 GROUP BY loc.AreaName;
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------+
| EXPLAIN


                                                                                                 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------+
| -> Table scan on <temporary>  (actual time=0.001..0.003 rows=20 loops=1)
    -> Aggregate using temporary table  (actual time=1.381..1.384 rows=20 loops=1)
        -> Nested loop inner join  (cost=220.26 rows=275) (actual time=0.209..1.187 rows=275 loops=1)
            -> Filter: (c.LocationID is not null)  (cost=124.01 rows=275) (actual time=0.197..0.694 rows=275 loops=1)
                -> Index range scan on c using time, with index condition: (c.`Time` < 600)  (cost=124.01 rows=275) (actual time=0.195..0.671 rows=275 loops=1)
            -> Single-row index lookup on loc using PRIMARY (LocationID=c.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=275)
 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

Indexing with Location.LocationID and Crime.LocationID:

```
mysql> CREATE INDEX loc1 ON Location(LocationID);
ERROR 1061 (42000): Duplicate key name 'loc1'
mysql> CREATE INDEX loc ON Crime(LocationID);
ERROR 1061 (42000): Duplicate key name 'loc'
mysql> EXPLAIN ANALYZE SELECT loc.AreaName, COUNT(loc.LocationID) FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID WHERE c.Time < 0600 GROUP BY loc.AreaName;
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------+
| EXPLAIN                                                                                              |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------+
| -> Table scan on <temporary>  (actual time=0.001..0.002 rows=20 loops=1)
    -> Aggregate using temporary table  (actual time=1.328..1.331 rows=20 loops=1)
        -> Nested loop inner join  (cost=386.31 rows=592) (actual time=0.057..1.143 rows=275 loops=1)
            -> Filter: ((c.`Time` < 600) and (c.LocationID is not null))  (cost=179.25 rows=592) (actual time=0.043..0.680 rows=275 loops=1)
                -> Table scan on c  (cost=179.25 rows=1775) (actual time=0.040..0.530 rows=1775 loops=1)
            -> Single-row index lookup on loc using PRIMARY (LocationID=c.LocationID)  (cost=0.25 rows=1) (actual time=0.001..0.001 rows=1 loops=275)
 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------+
1 row in set (0.01 sec)
```

Indexing with Location.AreaName and Crime.Time:

```
mysql> CREATE INDEX area ON Location(AreaName);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX time ON Crime(Time);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> EXPLAIN ANALYZE SELECT loc.AreaName, COUNT(loc.LocationID)
    -> FROM Crime c JOIN Location loc ON c.LocationID = loc.locationID
    -> WHERE c.Time < 0600
    -> GROUP BY loc.AreaName;
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
| EXPLAIN


+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
| -> Table scan on <temporary>  (actual time=0.002..0.004 rows=20 loops=1)
    -> Aggregate using temporary table  (actual time=1.597..1.601 rows=20 loops=1)
        -> Nested loop inner join  (cost=220.26 rows=275) (actual time=0.239..1.353 rows=275 loops=1)
            -> Filter: (c.LocationID is not null)  (cost=124.01 rows=275) (actual time=0.228..0.759 rows=275 loops=1)
                -> Index range scan on c using time, with index condition: (c.`Time` < 600)  (cost=124.01 rows=275) (actual time=0.226..0.660 rows=275 loops=1)
            -> Single-row index lookup on loc using PRIMARY (LocationID=c.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.002 rows=1 loops=275)
 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
1 row in set (0.01 sec)
```

Third query: Set operations, Subqueries, JOIN.

SELECT * FROM

(SELECT CrimeID, Time

FROM Crime c NATURAL JOIN Location loc

WHERE AreaName = "Central"

```
UNION
SELECT CrimeID, Time
FROM Crime c NATURAL JOIN Victim v
WHERE Sex = "F" ) as tab
ORDER BY CrimeID;
```

Top 15 rows:

```
mysql> SELECT * FROM
    -> (SELECT CrimeID, Time
    -> FROM Crime c NATURAL JOIN Location loc
    -> WHERE AreaName = "Central"
    -> UNION
    -> SELECT CrimeID, Time
    -> FROM Crime c NATURAL JOIN Victim v
    -> WHERE Sex = "F" ) as tab
    -> ORDER BY CrimeID
    -> LIMIT 15;
+---------+------+
| CrimeID | Time |
+---------+------+
|       1 | 2230 |
|       2 |  330 |
|       4 | 1730 |
|       6 |   30 |
|       7 | 1315 |
|       8 |   40 |
|       9 |  200 |
|      10 |   30 |
|      11 | 2200 |
|      12 |  955 |
|      13 | 1355 |
|      14 | 1638 |
|      15 | 1805 |
|      17 | 1320 |
|      18 | 1900 |
+---------+------+
15 rows in set (0.02 sec)
```

Indexing:

Created indexing on Location.AreaName since the query will be searching through the area names to find names equal to "Central". Created indexing on Victim.Sex since the query will be searching through the victim's sex to find all the female victims.

Not indexed:

```
mysql> EXPLAIN ANALYZE SELECT * FROM (SELECT CrimeID, Time FROM Crime c NATURAL JOIN Location loc WHERE AreaName = "Central" UNION SELECT CrimeID, Time FROM Crime c NATURAL JOIN Victim v WHERE Sex = "F" ) as tab ORDER BY CrimeID;
+------------------------------------------------------------------------------------------------------------------------
------------------------------+
| EXPLAIN
                                                          |
+------------------------------------------------------------------------------------------------------------------------
------------------------------+
| -> Sort: tab.CrimeID  (actual time=0.255..0.337 rows=1320 loops=1)
    -> Table scan on tab  (cost=45.48 rows=382) (actual time=0.001..0.071 rows=1320 loops=1)
        -> Union materialize with deduplication  (cost=559.02..559.02 rows=383) (actual time=8.533..8.688 rows=1320 loops=1)
            -> Nested loop inner join  (cost=242.38 rows=178) (actual time=0.082..4.632 rows=1097 loops=1)
                -> Filter: (loc.AreaName = 'Central')  (cost=180.25 rows=178) (actual time=0.057..0.944 rows=1097 loops=1)
                    -> Table scan on loc  (cost=180.25 rows=1775) (actual time=0.050..0.574 rows=1775 loops=1)
                -> Index lookup on c using LocationID (LocationID=loc.LocationID)  (cost=0.25 rows=1) (actual time=0.003..0.003 rows=1 loops=1097)
            -> Nested loop inner join  (cost=278.39 rows=205) (actual time=0.043..2.868 rows=579 loops=1)
                -> Filter: (v.Sex = 'F')  (cost=206.60 rows=205) (actual time=0.030..0.822 rows=684 loops=1)
                    -> Table scan on v  (cost=206.60 rows=2051) (actual time=0.029..0.577 rows=2052 loops=1)
                -> Index lookup on c using VictimID (VictimID=v.VictimID)  (cost=0.25 rows=1) (actual time=0.002..0.003 rows=1 loops=684)
 |
+------------------------------------------------------------------------------------------------------------------------
------------------------------+
1 row in set (0.01 sec)
```

Indexed:

Indexing with Victim.Sex:

```
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------+
| -> Sort: tab.CrimeID  (actual time=0.247..0.318 rows=1320 loops=1)
    -> Table scan on tab  (cost=99.36 rows=861) (actual time=0.002..0.066 rows=1320 loops=1)
        -> Union materialize with deduplication  (cost=639.39..639.39 rows=862) (actual time=8.264..8.415 rows=1320 loops=1)
            -> Nested loop inner join  (cost=242.38 rows=178) (actual time=0.046..4.079 rows=1097 loops=1)
                -> Filter: (loc.AreaName = 'Central')  (cost=180.25 rows=178) (actual time=0.031..0.900 rows=1097 loops=1)
                    -> Table scan on loc  (cost=180.25 rows=1775) (actual time=0.028..0.530 rows=1775 loops=1)
                -> Index lookup on c using loc (LocationID=loc.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.003 rows=1 loops=1097)
            -> Nested loop inner join  (cost=310.86 rows=684) (actual time=0.047..3.267 rows=579 loops=1)
                -> Index lookup on v using sex (Sex='F')  (cost=71.46 rows=684) (actual time=0.034..0.309 rows=684 loops=1)
                -> Index lookup on c using VictimID (VictimID=v.VictimID)  (cost=0.25 rows=1) (actual time=0.004..0.004 rows=1 loops=684)
 |
+------------------------------------------------------------------------------------------------------------------------
```

Indexing with Location.AreaName:

```
-----------------------------------------------------------------+
| EXPLAIN                                                                                                                   |
+------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------+
| -> Sort: tab.CrimeID  (actual time=0.227..0.293 rows=1320 loops=1)
    -> Table scan on tab  (cost=148.98 rows=1302) (actual time=0.001..0.065 rows=1320 loops=1)
        -> Union materialize with deduplication  (cost=1005.75..1005.75 rows=1302) (actual time=7.451..7.603 rows=1320 loops=1)
            -> Nested loop inner join  (cost=597.15 rows=1097) (actual time=0.068..3.811 rows=1097 loops=1)
                -> Index lookup on loc using area (AreaName='Central')  (cost=213.20 rows=1097) (actual time=0.053..0.480 rows=1097 loops=1)
                -> Index lookup on c using loc (LocationID=loc.LocationID)  (cost=0.25 rows=1) (actual time=0.002..0.003 rows=1 loops=1097)
            -> Nested loop inner join  (cost=278.39 rows=205) (actual time=0.040..2.874 rows=579 loops=1)
                -> Filter: (v.Sex = 'F')  (cost=206.60 rows=205) (actual time=0.029..0.803 rows=684 loops=1)
                    -> Table scan on v  (cost=206.60 rows=2051) (actual time=0.026..0.537 rows=2052 loops=1)
                -> Index lookup on c using VictimID (VictimID=v.VictimID)  (cost=0.25 rows=1) (actual time=0.002..0.003 rows=1 loops=684)
 |
```

Indexing with Location.AreaName and Victim.Sex:

```
mysql> CREATE INDEX areaname ON Location(AreaName ASC);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> CREATE INDEX sex ON Victim(Sex ASC);
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> EXPLAIN ANALYZE SELECT * FROM
    -> (SELECT CrimeID, Time
    -> FROM Crime c NATURAL JOIN Location loc
    -> WHERE AreaName = "Central"
    -> UNION
    -> SELECT CrimeID, Time
    -> FROM Crime c NATURAL JOIN Victim v
    -> WHERE Sex = "F" ) as tab
    -> ORDER BY CrimeID;
+------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------+
| EXPLAIN



                                                              |
+------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------+
| -> Sort: tab.CrimeID  (actual time=0.252..0.320 rows=1320 loops=1)
    -> Table scan on tab  (cost=202.86 rows=1781) (actual time=0.002..0.068 rows=1320 loops=1)
        -> Union materialize with deduplication  (cost=1086.11..1086.11 rows=1781) (actual time=7.034..7.177 rows=1320 loops=1)
            -> Nested loop inner join  (cost=597.15 rows=1097) (actual time=0.081..4.074 rows=1097 loops=1)
                -> Index lookup on loc using areaname (AreaName='Central')  (cost=213.20 rows=1097) (actual time=0.064..0.500 rows=1097 loops=1)
                -> Index lookup on c using LocationID (LocationID=loc.LocationID)  (cost=0.25 rows=1) (actual time=0.003..0.003 rows=1 loops=1097)
            -> Nested loop inner join  (cost=310.86 rows=684) (actual time=0.069..2.165 rows=579 loops=1)
                -> Index lookup on v using sex (Sex='F')  (cost=71.46 rows=684) (actual time=0.048..0.230 rows=684 loops=1)
                -> Index lookup on c using VictimID (VictimID=v.VictimID)  (cost=0.25 rows=1) (actual time=0.002..0.003 rows=1 loops=684)
 |
+------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------+
1 row in set (0.01 sec)
```