

Team 042 – 411Go

Database Design & Implementation

Part 1:

1. Database implementation:

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to safewalkla.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
gcloud sql connect db=safewalkla --user=root --quietziyeherry@cloudshell:- (safewalkla) $ gcloud sql connect db=safewalkla --user=root --quiet
Allowlisting your IP for incoming connection for 5 minutes...done.
Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 953
Server version: 8.0.31-google (Google)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use safewalk_la;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables in safewalk_la |
+-----+
| Area                  |
| Cases                 |
| Crime                 |
| Premis                |
| Status                |
| User                  |
| Weapon                |
| crime_data_raw        |
| crime_data_raw last five years |
+-----+
9 rows in set (0.00 sec)

mysql> 
```

2. DDL used to create tables:

```
CREATE TABLE User (
    UserId CHAR(15) PRIMARY KEY,
    UserName VARCHAR(25),
    Email VARCHAR(50),
    PhoneNumber VARCHAR(20),
    Password VARCHAR(50)
);
```

```
CREATE TABLE Crime (
    CrimeCode INT PRIMARY KEY,
    CrimeCodeDesc VARCHAR(255),
    CrimeCode1 INT,
    CrimeCode2 INT,
    CrimeCode3 INT
);
```

```
CREATE TABLE Area (
    AreaCode INT PRIMARY KEY,
    AreaName VARCHAR(50)
```

);

```
CREATE TABLE Premis (  
    PremisCode INT PRIMARY KEY,  
    PremisDesc VARCHAR(255)  
);
```

```
CREATE TABLE Weapon (  
    WeaponUsedCode INT PRIMARY KEY,  
    WeaponDesc VARCHAR(255)  
);
```

```
CREATE TABLE Status (  
    StatusCode VARCHAR(255) PRIMARY KEY,  
    StatusDesc VARCHAR(255)  
);
```

```
CREATE TABLE Cases (  
    DR_NO INT PRIMARY KEY,  
    DateReported VARCHAR(22),  
    DateOccurred VARCHAR(22),  
    TimeOccurred INT,  
    RptDistNo INT,  
    CrimeCode INT,  
    CrimeCodeDesc VARCHAR(255),  
    MOCode VARCHAR(255),  
    VictimSex CHAR,  
    VictimAge INT,  
    VictimDescent VARCHAR(2),  
    Status VARCHAR(255),  
    WeaponUsed INT,  
    Location VARCHAR(255),  
    LAT DOUBLE,  
    LON DOUBLE,  
    AreaCode INT,  
    FOREIGN KEY (CrimeCode) REFERENCES Crime (CrimeCode),  
    FOREIGN KEY (Status) REFERENCES Status (StatusCode),  
    FOREIGN KEY (WeaponUsed) REFERENCES Weapon  
(WeaponUsedCode),  
    FOREIGN KEY (AreaCode) REFERENCES Area (AreaCode)  
);
```

3. Count of rows for three tables:

of rows for Cases:

```
mysql> select count(*) from Cases;
+-----+
| count(*) |
+-----+
|    317854 |
+-----+
1 row in set (0.05 sec)
```

of rows for Premis:

```
mysql> select count(*) from Premis;
+-----+
| count(*) |
+-----+
|    1121 |
+-----+
1 row in set (0.01 sec)
```

of rows for Area:

```
mysql> select count(*) from Area;
+-----+
| count(*) |
+-----+
|    1045 |
+-----+
1 row in set (0.00 sec)
```

4. Advanced Queries:

Query 1:

```
mysql> select CrimeCodeDesc, count(1) CrimeTypeCnt
-> from (
-> select c.DR NO, w.CrimeCode, w.CrimeCodeDesc
-> from safewalk_la.Cases c
-> left join safewalk_la.Crime w
-> on c.CrimeCode = w.CrimeCode
-> ) it
-> group by CrimeCode
-> LIMIT 15;
+-----+-----+
| CrimeCodeDesc | CrimeTypeCnt |
+-----+-----+
| CRIMINAL HOMICIDE | 591 |
| MANSLAUGHTER, NEGLIGENT | 4 |
| RAPE, FORCIBLE | 1297 |
| RAPE, ATTEMPTED | 112 |
| ROBBERY | 11158 |
| ATTEMPTED ROBBERY | 1799 |
| ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT | 19341 |
| ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER | 531 |
| CHILD ABUSE (PHYSICAL) - AGGRAVATED ASSAULT | 222 |
| INTIMATE PARTNER - AGGRAVATED ASSAULT | 4704 |
| CHILD NEGLECT (SEE 300 W.I.C.) | 424 |
| SHOTS FIRED AT MOVING VEHICLE, TRAIN OR AIRCRAFT | 185 |
| SHOTS FIRED AT INHABITED DWELLING | 665 |
| BURGLARY | 19954 |
| BURGLARY, ATTEMPTED | 1416 |
+-----+-----+
15 rows in set (2.09 sec)

mysql>
```

Query 2:

```
mysql> SELECT WeaponDesc WeaponName, count(1) WeaponUsedCnt from( se  
lect c.DR_NO, w.WeaponUsedCode WeaponID, WeaponDesc from safewalk_la.  
Cases c left join safewalk_la.Weapon w on c.WeaponUsed=w.WeaponUsedCo  
de ) it group by WeaponID LIMIT 15;
```

WeaponName	WeaponUsedCnt
	201377
REVOLVER	520
HAND GUN	6910
RIFLE	178
SHOTGUN	140
SAWED OFF RIFLE/SHOTGUN	14
UNKNOWN FIREARM	2397
OTHER FIREARM	280
AUTOMATIC WEAPON/SUB-MACHINE GUN	8
SEMI-AUTOMATIC PISTOL	2553
SEMI-AUTOMATIC RIFLE	15
STARTER PISTOL/REVOLVER	11
TOY GUN	65
SIMULATED GUN	418
AIR PISTOL/REVOLVER/RIFLE/BB GUN	701

15 rows in set (0.78 sec)

Part 2: Indexing

1.0

Code:

EXPLAIN ANALYZE

Find most frequent crime type in all cases

select CrimeCodeDesc, count(1) CrimeTypeCnt

from (

select c.DR_NO, w.CrimeCode, w.CrimeCodeDesc

from safewalk_la.Cases c

left join safewalk_la.Crime w

on c.CrimeCode = w.CrimeCode

) it

group by CrimeCode

LIMIT 15;

```
| EXPLAIN
+-----+
| -> Limit: 15 row(s) (actual time=706.142..706.148 rows=15 loops=1)
|   -> Table scan on <temporary> (actual time=706.140..706.144 rows=15 loops=1)
|     -> Aggregate using temporary table (actual time=706.133..706.133 rows=133 loops=1)
|       -> Nested loop left join (cost=146922.80 rows=317024) (actual time=0.081..186.626 rows=317854 loops=1)
|         -> Covering index scan on c using CrimeCode (cost=35964.40 rows=317024) (actual time=0.063..77.963 rows=317854 loops=1)
|           -> Single-row index lookup on w using PRIMARY (CrimeCode=c.CrimeCode) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|
+-----+
1 row in set (0.71 sec)
```

1.1

CREATE INDEX CrimeDesc_idx ON Crime (CrimeCodeDesc);

SHOW INDEX FROM Crime;

EXPLAIN ANALYZE ...;

DROP INDEX CrimeDesc_idx ON Crime;

```
| EXPLAIN
+-----+
| -> Limit: 15 row(s) (actual time=702.920..702.927 rows=15 loops=1)
|   -> Table scan on <temporary> (actual time=702.916..702.922 rows=15 loops=1)
|     -> Aggregate using temporary table (actual time=702.902..702.902 rows=133 loops=1)
|       -> Nested loop left join (cost=146922.80 rows=317024) (actual time=0.061..193.117 rows=317854 loops=1)
|         -> Covering index scan on c using CrimeCode (cost=35964.40 rows=317024) (actual time=0.048..80.896 rows=317854 loops=1)
|           -> Single-row index lookup on w using PRIMARY (CrimeCode=c.CrimeCode) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|
+-----+
1 row in set (0.70 sec)
```

1.2

CREATE INDEX DateOccurred_idx ON Cases (DateOccurred);

SHOW INDEX FROM Cases;

EXPLAIN ANALYZE ...;

DROP INDEX DateOccurred_idx ON Cases;

```

| EXPLAIN
+-----+
|
| -> Limit: 15 row(s) (actual time=703.901..703.907 rows=15 loops=1)
|   -> Table scan on <temporary> (actual time=703.898..703.903 rows=15 loops=1)
|     -> Aggregate using temporary table (actual time=703.889..703.889 rows=133 loops=1)
|       -> Nested loop left join (cost=143726.30 rows=317024) (actual time=0.094..185.002 rows=317854 loops=1)
|         -> Covering index scan on c using CrimeCode (cost=32767.90 rows=317024) (actual time=0.075..77.198 rows=317854 loops=1)
|         -> Single-row index lookup on w using PRIMARY (CrimeCode=c.CrimeCode) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|       |
|     +-----+
|   +-----+
| 1 row in set (0.70 sec)

```

1.3

CREATE INDEX CrimeDesc_idx ON Crime (CrimeCodeDesc);

SHOW INDEX FROM Crime;

EXPLAIN ANALYZE ...;

DROP INDEX CrimeDesc_idx ON Crime;

DROP INDEX DateOccurred_idx ON Cases;

```

| EXPLAIN
+-----+
|
| -> Limit: 15 row(s) (actual time=693.254..693.259 rows=15 loops=1)
|   -> Table scan on <temporary> (actual time=693.251..693.256 rows=15 loops=1)
|     -> Aggregate using temporary table (actual time=693.245..693.245 rows=133 loops=1)
|       -> Nested loop left join (cost=143726.30 rows=317024) (actual time=0.067..184.943 rows=317854 loops=1)
|         -> Covering index scan on c using CrimeCode (cost=32767.90 rows=317024) (actual time=0.052..77.929 rows=317854 loops=1)
|         -> Single-row index lookup on w using PRIMARY (CrimeCode=c.CrimeCode) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|       |
|     +-----+
|   +-----+
| 1 row in set (0.69 sec)

```

Analyze:

There's no significant difference improvement between those different ways of indexing. The reason may be that the selecting process only use CrimeCode which is the primary key of Crime table, so using Crime.CrimeCodeDesc and Cases.DateOccurred can't help with the speed of sorting results.

2.0

Code:

EXPLAIN ANALYZE

Find most frequent weapon used in all cases

select WeaponDesc WeaponName, count(1) WeaponUsedCnt

from (

select c.DR_NO, w.WeaponUsedCode WeaponID, WeaponDesc

from safewalk_1a.Cases c

left join safewalk_1a.Weapon w

on c.WeaponUsed = w.WeaponUsedCode

) it

group by WeaponID

LIMIT 15;

---> 0.53 sec

```
mysql> EXPLAIN ANALYZE
-> # Find most frequent weapon used in all cases
-> select WeaponDesc WeaponName, count(1) WeaponUsedCnt
-> from t
-> select c.DR NO, w.WeaponUsedCode WeaponID, WeaponDesc
-> from safewalk_la.Cases c
-> left join safewalk_la.Weapon w
-> on c.WeaponUsed = w.WeaponUsedCode
-> ) as t
-> group by WeaponID
-> LIMIT 15;

+-----+
| EXPLAIN
+-----+

| -> Limit: 15 row(s) (actual time=524.534..524.542 rows=15 loops=1)
| -> Table scan on <temporary> (actual time=524.531..524.538 rows=15 loops=1)
| -> Aggregate using temporary table (actual time=524.528..524.528 rows=79 loops=1)
| -> Nested loop left join (cost=143726.30 rows=317024) (actual time=0.062..258.319 rows=317854 loops=1)
| -> Covering index scan on c using WeaponUsed (cost=32767.90 rows=317024) (actual time=0.047..101.489 rows=317854 loops=1)
| -> Single-row index lookup on w using PRIMARY (WeaponUsedCode=c.WeaponUsed) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|
+-----+
1 row in set (0.53 sec)
```

2.1

First add index to WeaponUsedCode

CREATE INDEX WeaponID_idx ON Weapon (WeaponUsedCode);

SHOW INDEX FROM Weapon;

EXPLAIN ANALYZE ...;

DROP INDEX WeaponID_idx ON Weapon;

EXPLAIN ANALYZE ...;

---> 0.53 sec

```
| EXPLAIN
+-----+

| -> Limit: 15 row(s) (actual time=524.534..524.542 rows=15 loops=1)
| -> Table scan on <temporary> (actual time=524.531..524.538 rows=15 loops=1)
| -> Aggregate using temporary table (actual time=524.528..524.528 rows=79 loops=1)
| -> Nested loop left join (cost=143726.30 rows=317024) (actual time=0.062..258.319 rows=317854 loops=1)
| -> Covering index scan on c using WeaponUsed (cost=32767.90 rows=317024) (actual time=0.047..101.489 rows=317854 loops=1)
| -> Single-row index lookup on w using PRIMARY (WeaponUsedCode=c.WeaponUsed) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|
+-----+
1 row in set (0.53 sec)
```

2.2

Then add index to Cases.DateReported, now two idx in total

CREATE INDEX DateReported_idx ON Cases (DateReported);

SHOW INDEX FROM Cases;

EXPLAIN ANALYZE ...;

---> 0.47 sec

```
| EXPLAIN
+-----+

| -> Limit: 15 row(s) (actual time=458.722..458.730 rows=15 loops=1)
| -> Table scan on <temporary> (actual time=458.719..458.727 rows=15 loops=1)
| -> Aggregate using temporary table (actual time=458.715..458.715 rows=79 loops=1)
| -> Nested loop left join (cost=143726.30 rows=317024) (actual time=0.131..192.673 rows=317854 loops=1)
| -> Covering index scan on c using Case weapon idx (cost=32767.90 rows=317024) (actual time=0.107..85.598 rows=317854 loops=1)
| -> Single-row index lookup on w using PRIMARY (WeaponUsedCode=c.WeaponUsed) (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|
+-----+
1 row in set (0.47 sec)
```

DROP INDEX DateReported_idx ON Cases;

2.3

Use another Case idx to replace the DateReported_idx, try it

CREATE INDEX Case_weapon_idx ON Cases (WeaponUsed);

SHOW INDEX FROM Cases;

---> 0.43 sec

DROP INDEX Case_weapon_idx ON Cases;

```
| EXPLAIN
+-----+
| -> Limit: 15 row(s)  (actual time=422.460..422.470 rows=15 loops=1)
|   -> Table scan on <temporary>  (actual time=422.458..422.466 rows=15 loops=1)
|     -> Aggregate using temporary table  (actual time=422.453..422.453 rows=79 loops=1)
|       -> Nested loop left join  (cost=143726.30 rows=317024) (actual time=0.082..178.400 rows=317854 loops=1)
|         -> Covering index scan on c using Case_weapon_idx  (cost=32767.90 rows=317024) (actual time=0.065..177.826 rows=317854 loops=1)
|           -> Single-row index lookup on w using PRIMARY (WeaponUsedCode=c.WeaponUsed)  (cost=0.25 rows=1) (actual time=0.000..0.000 rows=1 loops=317854)
|
+-----+
1 row in set (0.43 sec)
```

Analyze:

2.1 use the same parameter as primary so it won't affect the processing speed. 2.2 used Cases.DateOccurr but its not quite relevant to the parameters used in query process so the improvement is limited. The main improvement appears as using Case_weapon_idx, which is exactly the parameter used in advance query.