ECE356 Lab 3 Report

Part Three:

We focus on this query

```
select nameFirst,nameLast,max(RBI) from Batting
                inner join Master using (playerID)
                where HR = 0 limit 1;
```

Without any index:

Time = TIMER_END - TIMER_START

1st measurement: 168477451899000000 – 168394298164000000 = 83153735000000 ps = 83153735 µs
2nd measurement: 168843086711000000 – 168758774439000000 = 84312272000000 ps = 84312272 µs
3rd measurement: 169344248538000000 – 169259325750000000 = 84922788000000 ps = 84922788 µs
4th measurement: 170589663870000000 – 170505312999000000= 84350871000000 ps = 84350871 µs
5th measurement: 171170381180000000 – 171086841643000000 = 83539537000000 ps = 83539537 µs

(83153735 + 84312272 + 84922788 + 84350871 + 83539537) / 5 = 84055841 µs

The maximum execution time = 84922788 µs
The minimum execution time = 83153735 µs
The average execution time = 84055841 µs

With index:

1) **Only Add Primary and Foreign key:**

Since the query is joining Batting table and Master table, we try to add primary key on Batting and Master table, and foreign key on Batting table.

```
ALTER TABLE Batting ADD PRIMARY KEY (yearID, playerID, stint);

ALTER TABLE Master ADD PRIMARY KEY (playerID);

ALTER TABLE Batting
  ADD CONSTRAINT fk_Batting_Master FOREIGN KEY Batting(playerID) REFERENCES
Master(playerID) ON DELETE CASCADE;
```

After adding the index, we make five measurement of the execution time:
1st measurement: 246297444134000000 – 246297335952000000= 108182000000 ps = 108182 µs
2nd measurement: 246354749111000000 – 246354643827000000 = 105284000000 ps = 105284 µs
3rd measurement: 246461682902000000 – 246461577468000000 = 105434000000 ps = 105434 µs
4th measurement: 246490872711000000 – 246490766765000000 = 105946000000 ps = 105946 µs
5th measurement: 246514110120000000 – 246513999587000000 = 110533000000 ps = 110533 µs

(108182 + 105284 + 105434 + 105946 + 110533) / 5 = 107075.8 μs

The maximum execution time = 110533μs
The minimum execution time = 105284 μs
The average execution time = 107075.8 μs

Therefore, primary key and foreign key has improved the performance significantly.

2) Only Add Index on "HR":

Since "HR" has been used in the where clause, we try to add index on "HR".

```
CREATE INDEX HR_index ON Batting (HR) USING BTREE;
```

After adding the index, the query takes way too longer to execute (longer than 3 min…). Therefore, index on HR decrease the performance and should not be added.

3) Only Add Index on "RBI"

Since "RBI" has been used in aggregation Max (), we try to add index on "RBI"

```
CREATE INDEX RBI_index ON Batting (RBI) USING BTREE;
```

After adding the index, we make five measurement of the execution time:
1st measurement: 32313366570000000 – 32229062710000000 = 84303860000000 ps = 84303860 μs
2nd measurement: 32477823775000000 – 32393768305000000 = 84055470000000 ps = 84055470 μs
3rd measurement: 32628299511000000 – 32544419111000000 = 83880400000000 ps = 83880400 μs
4th measurement: 32776818397000000 – 32692715631000000 = 84102766000000 ps = 84102766 μs
5th measurement: 32934418101000000 – 32850174030000000 = 84244071000000 ps = 84244071 μs

(84303860 + 84055470 + 83880400 + 84102766 + 84244071) / 5 = 84117313.4 μs

The maximum execution time = 84303860 μs
The minimum execution time = 83880400 μs
The average execution time = 84117313.4 μs

Comparing to the "without any index" case, after adding index on "RBI", the execution time is still the same. Therefore, index on "RBI" does not affect the performance, which means it is optional.

4) Add Primary and Foreign key and index on "HR":

```
CREATE INDEX HR_index ON Batting (HR) USING BTREE;
```

After adding the index, we make five measurement of the execution time:
1st measurement: 248522442226000000 – 248522285781000000 = 156445000000 ps = 156445 μs
2nd measurement: 248562150530000000 – 248561990122000000 = 160408000000 ps = 160408 μs
3rd measurement: 248586600530000000 – 248586439530000000 = 161000000000 ps = 161000 μs
4th measurement: 248625797336000000 – 248625641269000000 = 156067000000 ps = 156067 μs
5th measurement: 248654082401000000 – 248653921984000000 = 160417000000 ps = 160417 μs

(156445 + 160408 + 161000 + 156067 + 160417) / 5 = 158867.4 μs

The maximum execution time = 161000 μs
The minimum execution time = 156067 μs
The average execution time = 158867.4 μs

Comparing to only adding primary and foreign key, adding index on "HR" has increase the execution time. Again, we should not add index on "HR".

5) **Add Primary and Foreign key and index on "RBI":**

```
CREATE INDEX RBI_index ON Batting (RBI) USING BTREE;
```

After adding the index, we make five measurement of the execution time:
1st measurement: 248990283486000000 – 248990179038000000 = 104448000000 ps = 104448 μs
2nd measurement: 249084720681000000 – 249084614995000000 = 105686000000 ps = 105686 μs
3rd measurement: 249113438604000000 – 249113328576000000 = 110028000000 ps = 110028 μs
4th measurement: 249140144370000000 – 249140037654000000 = 106716000000 ps = 106716 μs
5th measurement: 249164440852000000 – 249164333479000000 = 107373000000 ps = 107373 μs

(104448 + 105686 + 110028 + 106716 + 107373) / 5 = 106850.2 μs

The maximum execution time = 110028 μs
The minimum execution time = 104448 μs
The average execution time = 106850.2 μs

Comparing to only adding primary and foreign key, adding index on "RBI" does not affect the query performance. Therefore, index on "RBI" is optional.

In conclusion, here are the possible indexes to improve the query performance:

```
ALTER TABLE Batting ADD PRIMARY KEY (yearID, playerID, stint);

ALTER TABLE Master ADD PRIMARY KEY (playerID);

ALTER TABLE Batting
  ADD CONSTRAINT fk_Batting_Master FOREIGN KEY Batting(playerID) REFERENCES
Master(playerID) ON DELETE CASCADE;
```

Index on "RBI" is optional, since it does not affect the query performance.