

Bailey Thompson
bkthomps – 20653683
ECE 356
Project 2 – Part 3

- a. The primary keys define the unique element of the database. For every primary key combination, there is either 0 or 1 entry in the database. Primary keys should be minimal because the more primary keys, the longer lookup takes. To test this out, I ran queries on the Master and Salary table, with the following minimum number of primary keys:

Master: playerId

Salary: playerId, yearID, teamID

I then ran it with a maximum number of keys. For the Salary table, that means every key, but for the Master table I used the maximum number MySQL would let me use. The keys used were:

Master: playerId, birthYear, birthMonth, birthDay, birthCountry, birthState, birthCity

Salary: playerId, yearID, teamID, lgID, salary

Then, I recorded the fetch times of five trials and took the average:

Table 1. Fetch times of primary keys.

	Minimum (Master Table)	Maximum (Master Table)	Minimum (Salary Table)	Maximum (Salary Table)
Trial 1	0.18 ms	0.15 ms	0.16 ms	0.19 ms
Trial 2	0.18 ms	0.19 ms	0.14 ms	0.21 ms
Trial 3	0.15 ms	0.18 ms	0.17 ms	0.19 ms
Trial 4	0.15 ms	0.19 ms	0.11 ms	0.13 ms
Trial 5	0.20 ms	0.21 ms	0.16 ms	0.19 ms
Average	0.172 ms	0.184 ms	0.148 ms	0.182 ms

These observations make sense because the more primary keys there are, the longer it takes to query data.

- b. A foreign key will make sure the tables within the database stay valid, and make sure that there is no missing data. Other than that, foreign keys should have no impact on the efficiency. To test this out, I removed the foreign keys and added them back, and determined the fetch speed. The results are as follows:

Table 2. Fetch times of foreign keys.

	None (Master Table)	All (Master Table)	None (Salary Table)	All (Salary Table)
Trial 1	0.16 ms	0.15 ms	0.19 ms	0.19 ms
Trial 2	0.18 ms	0.14 ms	0.17 ms	0.17 ms
Trial 3	0.17 ms	0.18 ms	0.16 ms	0.19 ms
Trial 4	0.15 ms	0.15 ms	0.17 ms	0.15 ms
Trial 5	0.13 ms	0.16 ms	0.19 ms	0.19 ms
Average	0.158 ms	0.156 ms	0.176 ms	0.178 ms

The results between the select queries with and without foreign keys are fairly similar. This shows that foreign keys simply exist as a verification of data.

- c. An index will speed up the performance of select actions unless the attribute frequently is null but will take up space and reduce the performance of update, insert, and delete statements. For these reasons, it is important to only use useful indexes. The two statements have been run without indexes, and also with using all indexes related to the queries. The query with indexes has the birthYear, birthMonth, and birthDay indexes on the Master table, the salary index on the Salary table, the G_all index on the Appearances table, and the G index on the Managers table. The performance is as follows:

Table 3. Fetch time of explicit indexes.

	None (Master Table)	All (Master Table)	None (Salary Table)	All (Salary Table)
Trial 1	0.14 ms	0.23 ms	0.19 ms	0.12 ms
Trial 2	0.12 ms	0.13 ms	0.23 ms	0.17 ms
Trial 3	0.18 ms	0.17 ms	0.20 ms	0.13 ms
Trial 4	0.13 ms	0.14 ms	0.16 ms	0.15 ms
Trial 5	0.14 ms	0.20 ms	0.17 ms	0.14 ms
Average	0.142 ms	0.174 ms	0.19 ms	0.142 ms

Adding appropriate indexes actually slowed down the query for the master table because many of the values are null. This proves that adding indexes is only useful when most of the data is non-null. Adding the indexes for the salary sped up the query as predicted.