1. **SELECT** *\** 3.1.1**FROM** employee  
     
   **CREATE INDEX** emp\_first **ON** employee (**first\_name**) 3.1.2  
     
   **DROP INDEX** emp\_first

|  |  |
| --- | --- |
| **3.1.1** | **3.1.2** |
| 148 ms | 90 ms |
| 143 ms | 213 ms |
| 144 ms | 103 ms |
| 127 ms | 88 ms |
| 129 ms | 92 ms |
| AVERAGE: 138 ms | AVERAGE: 117 ms |

3.1.3. The index is faster than no index. It appears that Index isn’t a whole lot faster because of the 213 ms outlier. Without it, index would be a whole lot faster. This is because no index is returning all the data in the table and index is returning just the first names.

1. **SELECT** *\** 3.2.1**FROM** city  
   **WHERE name** < **'City-300'**

**SELECT** *\** 3.2.2**FROM** city  
**WHERE name** = **'City-427'**

**SELECT** *\** 3.2.3**FROM** city  
**WHERE name** = **'City-9999'**

**DROP INDEX name**

|  |  |  |
| --- | --- | --- |
| **3.2.1** | **3.2.2** | **3.2.3** |
| 48 ms | 47 ms | 45 ms |
| 52 ms | 43 ms | 49 ms |
| 45 ms | 42 ms | 65 ms |
| 55 ms | 37 ms | 42 ms |
| 53 ms | 83 ms | 48 ms |
| AVERAGE: 51 ms | AVERAGE: 50 ms | 50 ms |

2.3.4

**CREATE INDEX name ON** city (**city\_id**, **name ASC**, **country**) 3.2.1  
 **WHERE name** < **'City-300'**

**CREATE INDEX name ON** city (**city\_id**, **name**, **country**) 3.2.2  
 **WHERE name** = **'City-427'**

**CREATE INDEX name ON** city (**city\_id**, **name**, **country**) 3.2.3  
 **WHERE name** = **'City-9999'**

**DROP INDEX name**

3.2.5

The timing for no index and index was pretty much the same for each example. I believe 3.2.3 for index is slower because it’s trying to index an item in the table that doesn’t exist. I think the times are about the same because it’s going through a table query and not a scan.

1. **SELECT** *\**  3.3.1**FROM** employee  
   **ORDER BY last\_name ASC**

**CREATE INDEX** lastname **ON** employee (**employee\_id**, **first\_name**, **last\_name ASC**) 3.3.2

**DROP INDEX** lastname

|  |  |
| --- | --- |
| **3.3.1** | **3.3.2** |
| 147 ms | 66 ms |
| 157 ms | 62 ms |
| 147 ms | 55 ms |
| 174 ms | 85 ms |
| 159 ms | 62 ms |
| AVERAGE: 157 ms | AVERAGE: 66 ms |

3.3.3

**CREATE INDEX** lastname **ON** employee (**employee\_id**, **first\_name**, **last\_name DESC**)

**DROP INDEX** lastname

|  |
| --- |
| **3.3.3** |
| 97 ms |
| 67 ms |
| 65 ms |
| 58 ms |
| 96 ms |
| AVERAGE: 77 ms |

3.3.4

The indexes are similar in that they are a lot faster than the non-indexed one. When it comes to sorting data, indexing is a whole lot faster. This is because the index gets the designated column and then uses a sorting algorithm to quickly sort the data.

|  |  |  |
| --- | --- | --- |
| **3.2.1** | **3.2.2** | **3.2.3** |
| 52 ms | 50 ms | 52 ms |
| 48 ms | 50 ms | 60 ms |
| 49 ms | 44 ms | 51 ms |
| 40 ms | 47 ms | 47 ms |
| 47 ms | 42 ms | 63 ms |
| AVERAGE: 47 | AVERAGE: 47 ms | AVERAGE: 54 ms |