

Proof of Proper Table Implementation:

```

1 -- MySQL dump 10.13 Distrib 8.0.19, for Win64 (x86_64)
2 --
3 -- Host: localhost Database: aloo
4 --
5 -- Server version 8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `disease`
20 --
21
22 DROP TABLE IF EXISTS `disease`;

```

Procedure:

1. Create the database on MySQL.
2. Create the relations.
3. Find the estimated total cost of each of the 4 queries without any indices added.
4. Through forward selection, add an index for each attribute, one at a time.
5. Keep attribute indices that reduce costs.

Query 1

List of non-primary key attributes used in JOIN, WHERE, GROUP BY, and HAVING clauses:

- doctor(city)
- doctor(specialty_id)
- disease(specialty_id)
- disease_symptom(disease_id)
- disease_symptom(symptom_id)

Idx Added	Estimated Total Cost
None	19,815
doctor(city)	21,166
doctor(specialty_id)	19,123
doctor(specialty_id) + disease(specialty_id)	19,123
doctor(specialty_id) + disease(specialty_id) + disease_symptom(disease_id)	19,123

The screenshot shows the MySQL Workbench interface with a query results grid and an Explain Analyze dialog.

Query Results Grid:

doctor(specialty_id) + disease(specialty_id) + disease_symptom(symptom_id)	13,311
Local instance 3306 (medibuddy_db) - Warning - not supported	

Explain Analyze Dialog:

```
1
2  #DROP INDEX idx_doctor_city ON doctor;
3  #DROP INDEX idx_doctor_specialty ON doctor;
4  #DROP INDEX idx_disease_specialty ON disease;
5  #DROP INDEX idx_ds_disease ON disease_symptom;
6  #DROP INDEX idx_ds_symptom ON disease_symptom;
7
8  #CREATE INDEX idx_doctor_city ON doctor(city);
9  #CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
10 #CREATE INDEX idx_disease_specialty ON disease(specialty_id);
11 #CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
12 #CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);
13
14 • EXPLAIN ANALYZE
```

EXPLAIN:

```
> Sort: o:doctor_id (actual time=54.953 .. 59.280 rows=9921 loops=1)
> Stream results. (cost=13311 rows=1109) (actual time=6.97 .. 34176 rows=99213 loops=1)
  > Nested loop inner join (cost=13311 rows=1109) (actual time=6.95 .. 34075 rows=99213 loops=1)
    > Nested loop inner join (cost=1115 rows=43.4) (actual time=5.06 .. 37.2 rows=345 loops=1)
      > Nested loop inner join (cost=1115 rows=43.4) (actual time=5.06 .. 37.2 rows=345 loops=1)
```

Object Info: No object selected

Action Output:

Time	Action	Response
607 22:09:03	CREATE INDEX idx_disease_specialty ON disease(specialty_id)	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_disease_specialty' defined
608 22:09:03	CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id)	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_ds_symptom' defined or
609 22:09:03	EXPLAIN WITH input_symptoms AS (SELECT s.symptom_id FROM symptom s...	1 row(s) returned
610 22:09:35	EXPLAIN ANALYZE WITH input_symptoms AS (SELECT s.symptom_id FROM sym...	1 row(s) returned

Result 45 | Read Only

Query Completed

Conclusion: Through forward selection, doctor(specialty_id), disease(specialty_id), and disease(symptom_id) should have indices associated with them to minimize estimated total cost.

Query 2

List of non-primary key attributes used in JOIN, WHERE, GROUP BY, and HAVING clauses:

- doctor(city)
 - doctor(specialty_id)
 - disease(specialty_id)
 - disease_symptom(disease_id)
 - disease_symptom(symptom_id)

Idx Added	Estimated Total Cost
None	660,382
doctor(city)	187,865
doctor(city) + doctor(specialty_id)	660,382
doctor(city) + disease(specialty_id)	202,553
doctor(city) + disease_symptom(disease_id)	202,553

doctor(city) + disease_symptom(symptom_id)	202,553
--	---------

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** sys
- Query Editor:**

```

1 #DROP INDEX idx_doctor_city ON doctor;
2 #DROP INDEX idx_doctor_specialty ON doctor;
3 #DROP INDEX idx_disease_specialty ON disease;
4 #DROP INDEX idx_ds_disease ON disease_symptom;
5 #DROP INDEX idx_ds_symptom ON disease_symptom;
6
7
8 #CREATE INDEX idx_doctor_city ON doctor(city);
9 #CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
10 #CREATE INDEX idx_disease_specialty ON disease(specialty_id);
11 #CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
12 #CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);
13
14 • EXPLAIN ANALYZE
    
```
- EXPLAIN:** Shows the execution plan for the EXPLAIN ANALYZE query. It includes a nested loop inner join with a cost of 13311 rows=1109, and a sort operation with a cost of 34553 rows=99213 loops=1.
- Result Grid:** Displays the results of the EXPLAIN ANALYZE command, showing actions like CREATE INDEX and EXPLAIN statements with their times and responses.
- Object Info:** Shows "No object selected".
- Session:** Shows "Query Completed".

Conclusion: Through forward selection, doctor(city) should be the only attribute w/ an index associated w/ it to minimize estimated total costs.

Query 3

List of non-primary key attributes used in JOIN, WHERE, GROUP BY, and HAVING clauses:

- doctor(state)
- doctor(specialty_id)
- disease(specialty_id)
- disease_symptom(disease_id)

Idx Added	Estimated Total Cost
None	9.07e+6
doctor(state)	400e+6
doctor(specialty_id)	9.07e+6
disease(specialty_id)	9.07e+6
disease_symptom(disease_id)	8.84e+6

```

1 #DROP INDEX idx_doctor_city ON doctor;
2 #DROP INDEX idx_doctor_specialty ON doctor;
3 #DROP INDEX idx_disease_specialty ON disease;
4 #DROP INDEX idx_ds_disease ON disease_symptom;
5 #DROP INDEX idx_ds_symptom ON disease_symptom;
6
7
8 #CREATE INDEX idx_doctor_city ON doctor(city);
9 #CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
10 #CREATE INDEX idx_disease_specialty ON disease(specialty_id);
11 #CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
12 #CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);
13
14 • EXPLAIN ANALYZE

```

EXPLAIN:

```

-> Sort:doctor_id (actual time=0.03, rows=109) (cost=15311 rows=109) (actual time=0.97,34178 rows=99213 loops=1)
  -> Stream results (cost=15311 rows=109) (actual time=0.97,34178 rows=99213 loops=1)
    -> Nested loop inner join (cost=15311 rows=109) (actual time=0.95,34075 rows=99213 loops=1)
      -> Nested loop inner join (cost=1115 rows=43.4) (actual time=0.06,37.2 rows=345 loops=1)
        -> Nested loop inner join (cost=1114 rows=43.4) (actual time=0.06,37.2 rows=345 loops=1)
          -> Nested loop inner join (cost=1114 rows=43.4) (actual time=0.06,37.2 rows=345 loops=1)

```

Result 45

Action	Time	Response
CREATE INDEX idx_disease_specialty ON disease(specialty_id)	22:09:03	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_disease_specialty' defined or used
CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id)	22:09:03	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_ds_symptom' defined or used
EXPLAIN WITH input_symptoms AS (SELECT s.symptom_id FROM symptom ...)	22:09:03	1 row(s) returned
EXPLAIN ANALYZE WITH input_symptoms AS (SELECT s.symptom_id FROM symptom ...)	22:09:35	1 row(s) returned

Query Completed

Conclusion: Through forward selection, no attributes should have an index associated w/ it to minimize estimated total costs. This could be because of factors such as doctor(state) not having many unique values, which increases lookup time.

Query 4

List of non-primary key attributes used in JOIN, WHERE, GROUP BY, and HAVING clauses:

- doctor(city)
- doctor(specialty_id)
- disease(specialty_id)

Idx Added	Estimated Total Cost
None	429,533
doctor(city)	10,345
doctor(city) + doctor(specialty_id)	10,345
doctor(city) + disease(specialty_id)	9.07e+6

The screenshot shows the MySQL Workbench interface. In the top-left pane, there's a tree view under 'SCHEMAS' for the 'sys' schema. The main area contains a query editor with the following SQL code:

```

1 #DROP INDEX idx_doctor_city ON doctor;
2 #DROP INDEX idx_doctor_specialty ON doctor;
3 #DROP INDEX idx_disease_specialty ON disease;
4 #DROP INDEX idx_ds_disease ON disease_symptom;
5 #DROP INDEX idx_ds_symptom ON disease_symptom;
6
7 #CREATE INDEX idx_doctor_city ON doctor(city);
8 #CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
9 #CREATE INDEX idx_disease_specialty ON disease(specialty_id);
10 #CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
11 #CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);
12
13
14 • EXPLAIN ANALYZE

```

The status bar at the bottom indicates '100%' completion and '16:14' time.

In the center, the 'EXPLAIN' section shows the execution plan:

```

EXPLAIN:
    > Sort:doctor_id (actual time=0.03, rows=109) rows_scanned=109 loops=1
    > Stream results (cost=15311 rows=109) (actual time=97.34178 rows=99213 loops=1)
        > Nested loop inner join (cost=15311 rows=109) (actual time=95.34075 rows=99213 loops=1)
            > Nested loop inner join (cost=1115 rows=434) (actual time=55.06.37.2 rows=345 loops=1)
                > Nested loop inner join (cost=1114 rows=434) (actual time=55.06.37.2 rows=345 loops=1)

```

The right side of the interface has a 'Result Grid' panel showing the results of the EXPLAIN command. The results table has columns 'Time' and 'Action' (with a dropdown for 'Response'). The log entries are:

Action	Time	Response
CREATE INDEX idx_disease_specialty ON disease(specialty_id)	22:09:03	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_disease_specialty' defined or used for this table
CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id)	22:09:03	0 row(s) affected, 1 warning(s): 1831 Duplicate index 'idx_ds_symptom' defined or used for this table
EXPLAIN WITH input_symptoms AS (SELECT s.symptom_id FROM symptom ...)	22:09:03	1 row(s) returned
EXPLAIN ANALYZE WITH input_symptoms AS (SELECT s.symptom_id FROM symptom ...)	22:09:35	1 row(s) returned

Conclusion: Through forward selection, doctor(city) should be the only attribute w/ an index associated w/ it to minimize estimated total costs.