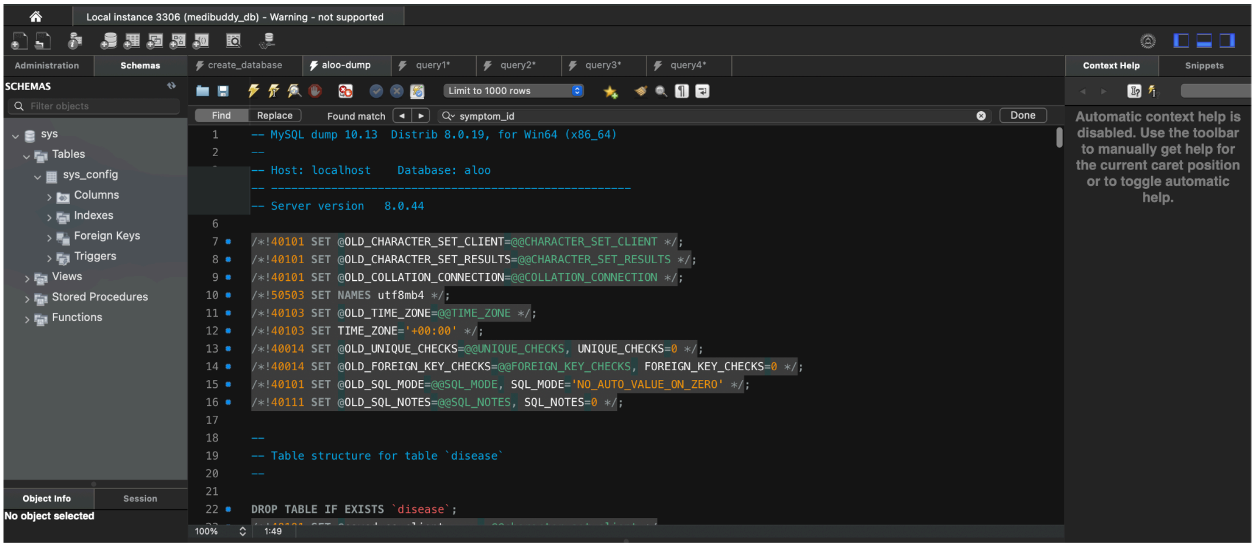
Proof of Proper Table Implementation:



```
-- MySQL dump 10.13  Distrib 8.0.19, for Win64 (x86_64)
--
-- Host: localhost    Database: aloo
-- ------------------------------------------------------
-- Server version	8.0.44

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `disease`
--

DROP TABLE IF EXISTS `disease`;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

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.

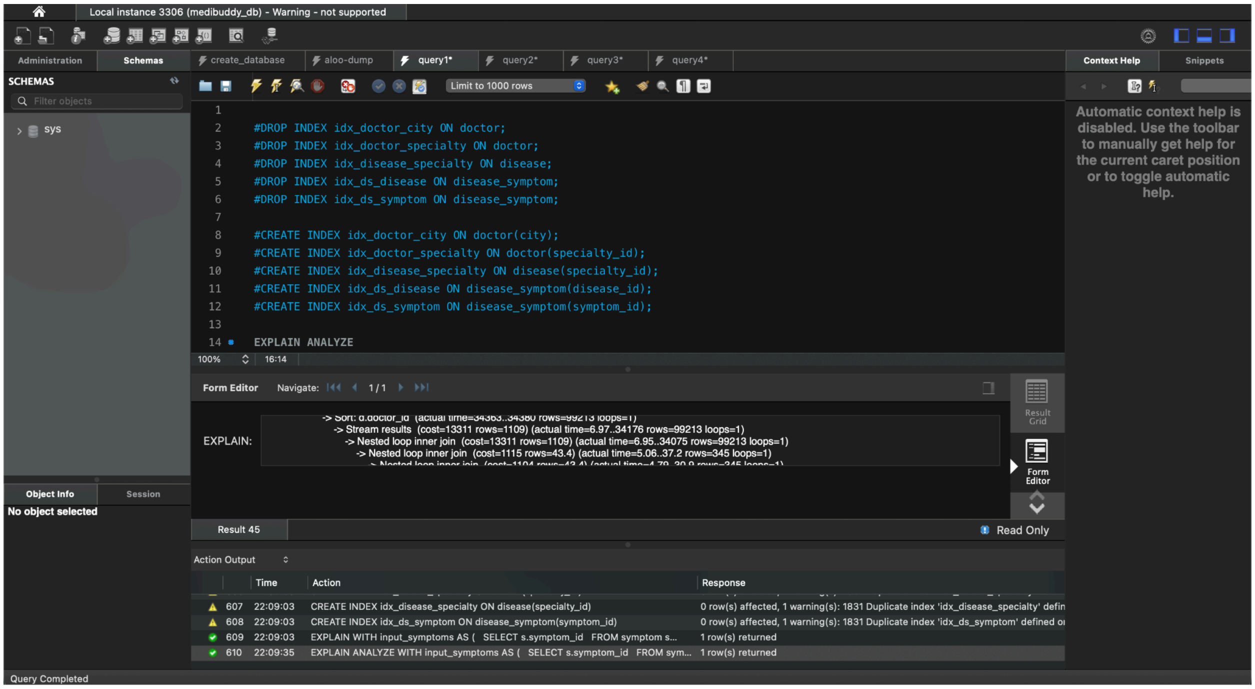
Query 1

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

doctor(specialty_id) + disease(specialty_id) + disease_symptom(symptom_id)	13,311



```
#DROP INDEX idx_doctor_city ON doctor;
#DROP INDEX idx_doctor_specialty ON doctor;
#DROP INDEX idx_disease_specialty ON disease;
#DROP INDEX idx_ds_disease ON disease_symptom;
#DROP INDEX idx_ds_symptom ON disease_symptom;

#CREATE INDEX idx_doctor_city ON doctor(city);
#CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
#CREATE INDEX idx_disease_specialty ON disease(specialty_id);
#CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
#CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);

EXPLAIN ANALYZE
```

EXPLAIN:

```
    -> Sort: d.doctor_id  (actual time=34363..34380 rows=99213 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=1104 rows=43.4) (actual time=4.79..30.9 rows=345 loops=1)
```

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' defin
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 o
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
```

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.

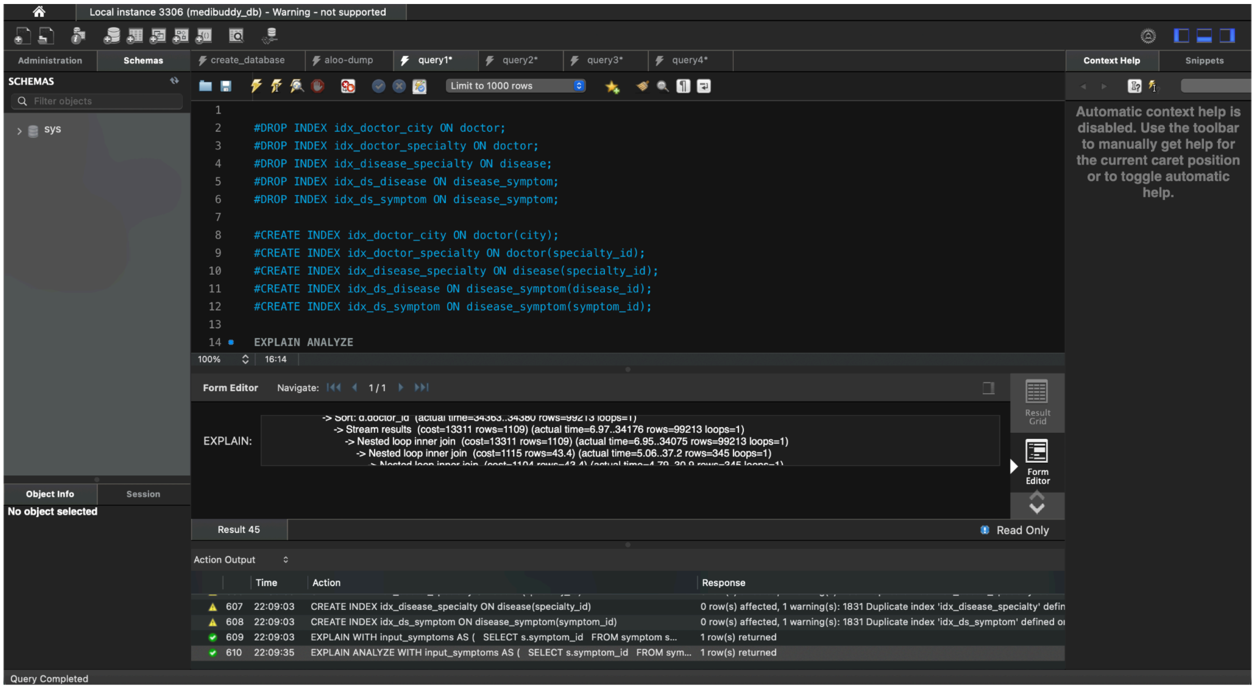
Query 2

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



```

#DROP INDEX idx_doctor_city ON doctor;
#DROP INDEX idx_doctor_specialty ON doctor;
#DROP INDEX idx_disease_specialty ON disease;
#DROP INDEX idx_ds_disease ON disease_symptom;
#DROP INDEX idx_ds_symptom ON disease_symptom;

#CREATE INDEX idx_doctor_city ON doctor(city);
#CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
#CREATE INDEX idx_disease_specialty ON disease(specialty_id);
#CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
#CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);

EXPLAIN ANALYZE
```

EXPLAIN:

```
-> Sort: d.doctor_id  (actual time=34363..34380 rows=99213 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=1104 rows=43.4) (actual time=4.79..30.9 rows=345 loops=1)
```

```
     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' defin
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 o
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
```

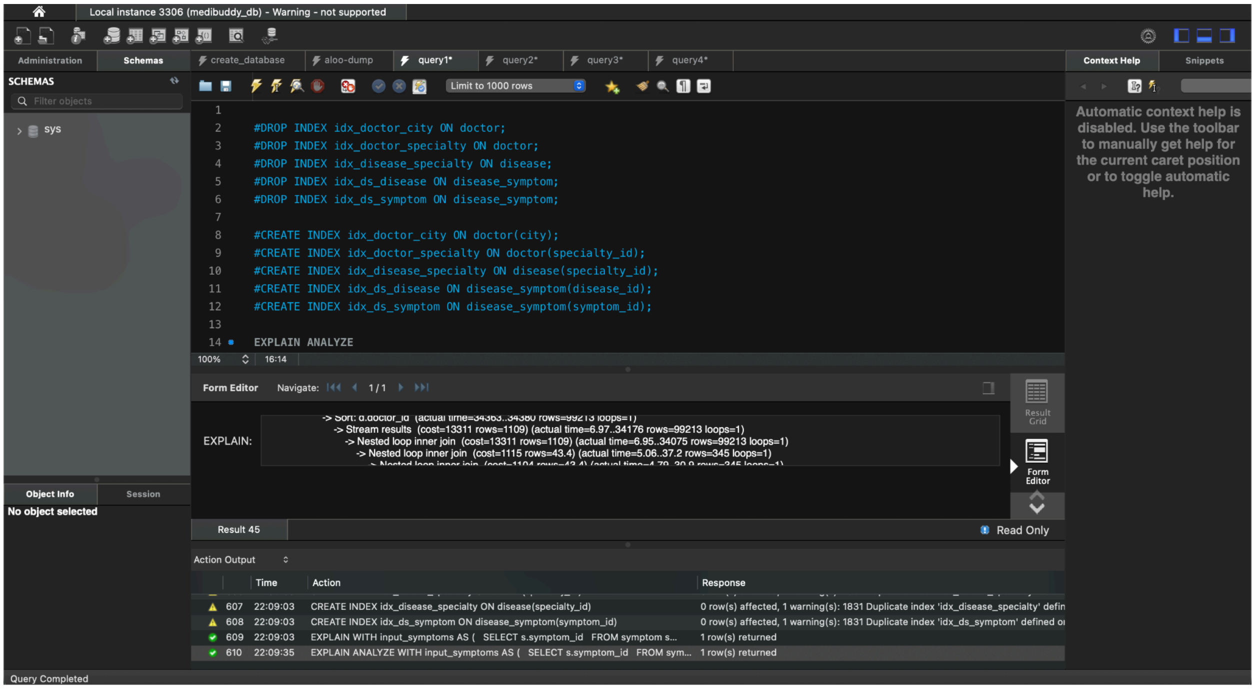
Conclusion: Through forward selection, doctor(city) should be the only attribute w/ an index associated w/ it to minimize estimated total costs.

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



```

#DROP INDEX idx_doctor_city ON doctor;
#DROP INDEX idx_doctor_specialty ON doctor;
#DROP INDEX idx_disease_specialty ON disease;
#DROP INDEX idx_ds_disease ON disease_symptom;
#DROP INDEX idx_ds_symptom ON disease_symptom;

#CREATE INDEX idx_doctor_city ON doctor(city);
#CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
#CREATE INDEX idx_disease_specialty ON disease(specialty_id);
#CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
#CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);

EXPLAIN ANALYZE
```

EXPLAIN:

```
-> Sort: d.doctor_id  (actual time=34363..34380 rows=99213 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=1104 rows=43.4) (actual time=4.79..30.9 rows=345 loops=1)
```

```
     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' defin
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 o
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
```

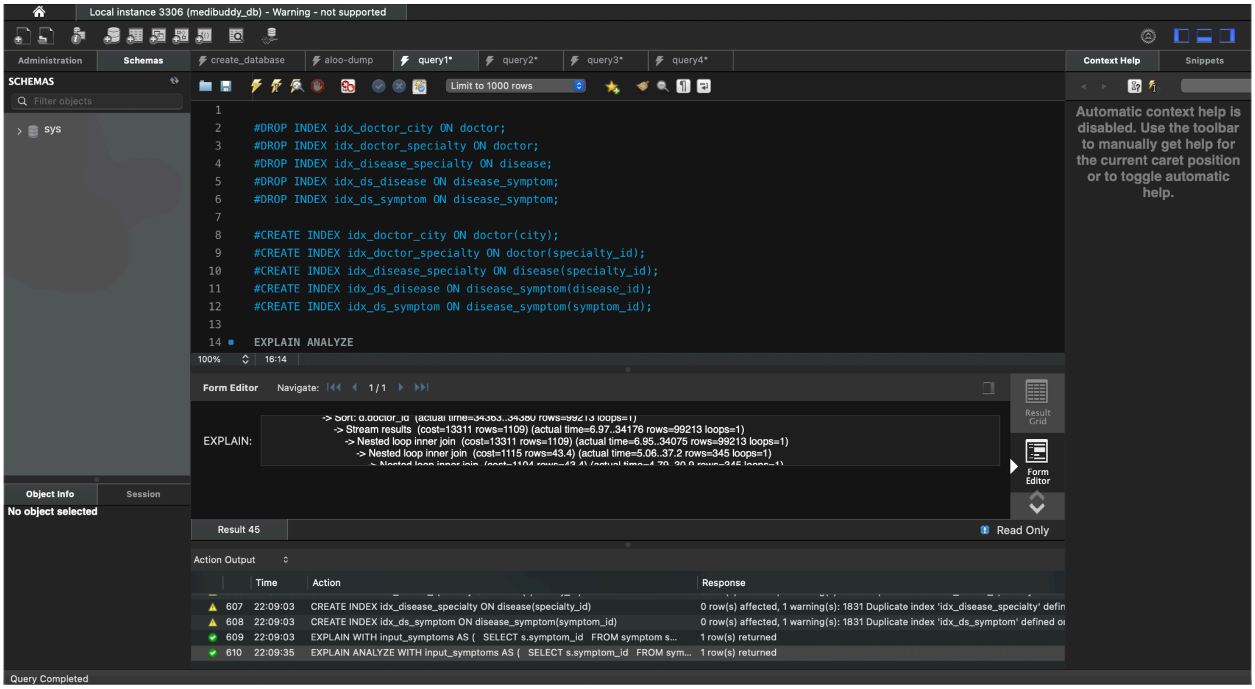
Conclusion: Through forward selection, no attributes should have an index associated w/ it to minimize estimated total costs. This could be because of factors such as doctor(state) not having many unique values, which increases lookup time.

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



Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

#DROP INDEX idx_doctor_city ON doctor;
#DROP INDEX idx_doctor_specialty ON doctor;
#DROP INDEX idx_disease_specialty ON disease;
#DROP INDEX idx_ds_disease ON disease_symptom;
#DROP INDEX idx_ds_symptom ON disease_symptom;

#CREATE INDEX idx_doctor_city ON doctor(city);
#CREATE INDEX idx_doctor_specialty ON doctor(specialty_id);
#CREATE INDEX idx_disease_specialty ON disease(specialty_id);
#CREATE INDEX idx_ds_disease ON disease_symptom(disease_id);
#CREATE INDEX idx_ds_symptom ON disease_symptom(symptom_id);

EXPLAIN ANALYZE
```

EXPLAIN:

```
-> Sort: d.doctor_id  (actual time=34363..34380 rows=99213 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=1104 rows=43.4) (actual time=4.79..30.9 rows=345 loops=1)
```

```
        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' defin
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
```

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