Analysis of Database Queries

by Nick Birnberg & Neil Bhateja

Summary

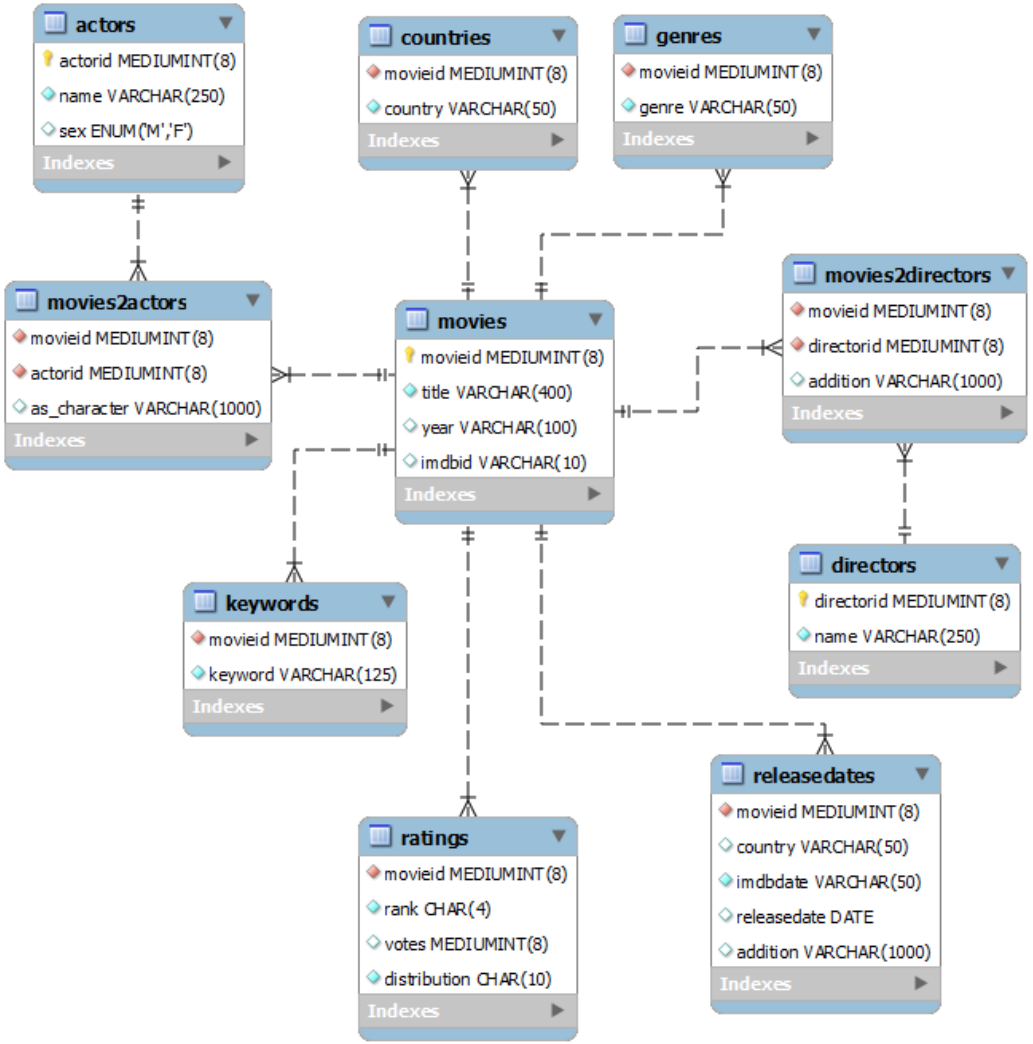
- Which inserts faster?
- Which responds faster?

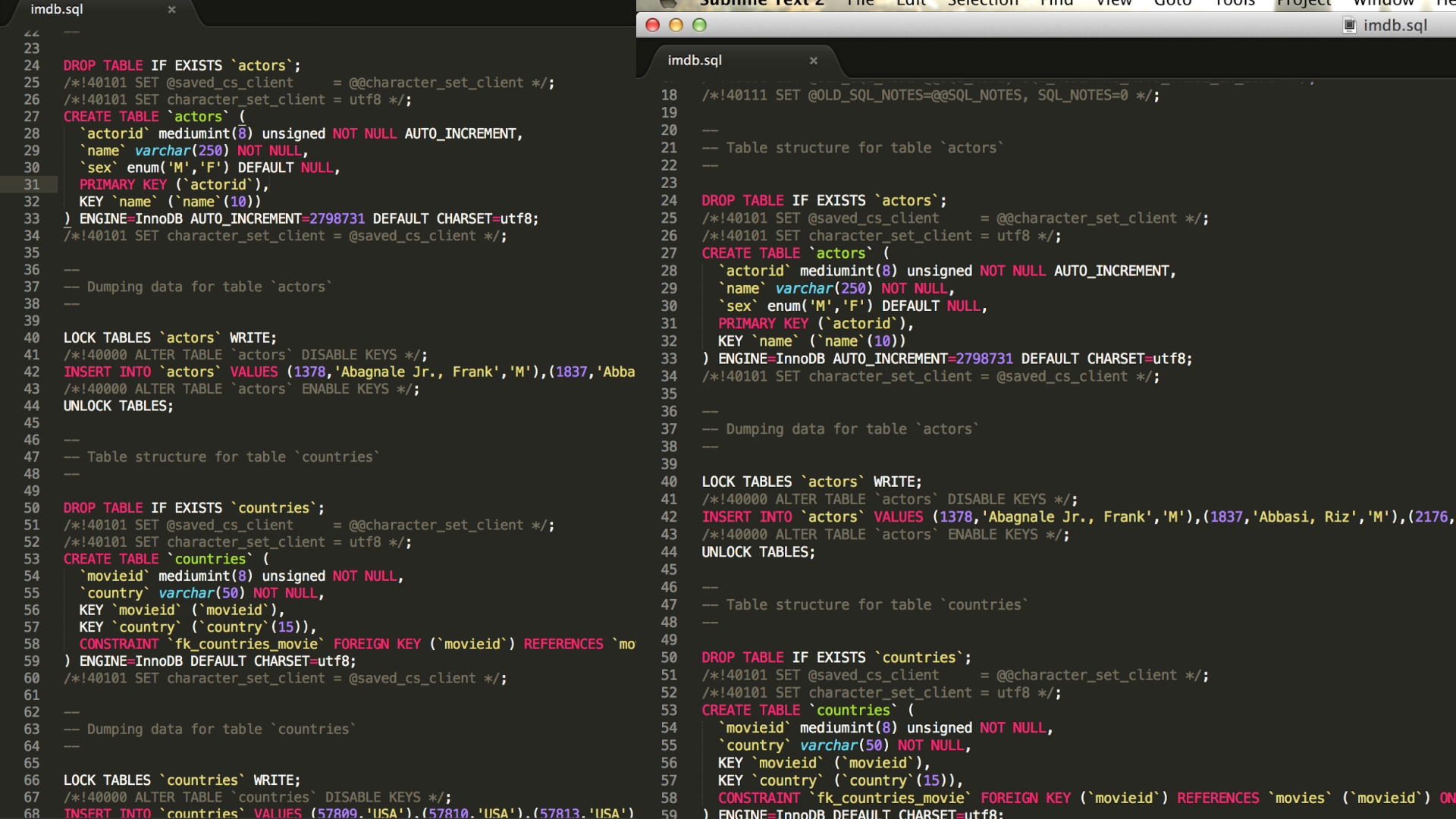
Query Size & Complexity

- Variances
- Covariance

Data

- Extracted from IMDB
- Converted to mySQL + mongoDB schema




imdb.sql
DROP TABLE IF EXISTS `actors`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `actors` (
`actorid` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
`name` varchar(250) NOT NULL,
`sex` enum('M','F') DEFAULT NULL,
PRIMARY KEY (`actorid`),
KEY `name` (`name`(10))
) ENGINE=InnoDB AUTO_INCREMENT=2798731 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
-- Dumping data for table `actors`
LOCK TABLES `actors` WRITE;
/*!40000 ALTER TABLE `actors` DISABLE KEYS */;
INSERT INTO `actors` VALUES (1378,'Abagnale Jr., Frank','M'),(1837,'Abba
/*!40000 ALTER TABLE `actors` ENABLE KEYS */;
UNLOCK TABLES;
-- Table structure for table `countries`
DROP TABLE IF EXISTS `countries`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `countries` (
`movieid` mediumint(8) unsigned NOT NULL,
`country` varchar(50) NOT NULL,
KEY `movieid` (`movieid`),
KEY `country` (`country`(15)),
CONSTRAINT `fk_countries_movie` FOREIGN KEY (`movieid`) REFERENCES `mo
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
-- Dumping data for table `countries`
LOCK TABLES `countries` WRITE;
/*!40000 ALTER TABLE `countries` DISABLE KEYS */;
INSERT INTO `countries` VALUES (57809,'USA'),(57810,'USA'),(57813,'USA')
imdb.sql
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
-- Table structure for table `actors`
DROP TABLE IF EXISTS `actors`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `actors` (
`actorid` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
`name` varchar(250) NOT NULL,
`sex` enum('M','F') DEFAULT NULL,
PRIMARY KEY (`actorid`),
KEY `name` (`name`(10))
) ENGINE=InnoDB AUTO_INCREMENT=2798731 DEFAULT CHARSET=utf8;
/*!40101 SET character_set_client = @saved_cs_client */;
-- Dumping data for table `actors`
LOCK TABLES `actors` WRITE;
/*!40000 ALTER TABLE `actors` DISABLE KEYS */;
INSERT INTO `actors` VALUES (1378,'Abagnale Jr., Frank','M'),(1837,'Abbasi, Riz','M'),(2176,
/*!40000 ALTER TABLE `actors` ENABLE KEYS */;
UNLOCK TABLES;
-- Table structure for table `countries`
DROP TABLE IF EXISTS `countries`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `countries` (
`movieid` mediumint(8) unsigned NOT NULL,
`country` varchar(50) NOT NULL,
KEY `movieid` (`movieid`),
KEY `country` (`country`(15)),
CONSTRAINT `fk_countries_movie` FOREIGN KEY (`movieid`) REFERENCES `movies` (`movieid`) ON

Sample Data

Data Size	Query Complexity	MySQL Insert(ms)	MySQL Select(ms)	MongoDB Insert(ms)	MongoDB Select(ms)
1	1	2510	498	864	1079
1	1	3215	452	782	1063
1	1	2534	429	754	985
1	1	2344	451	738	1001
2	1	2890	438	881	1048
2	1	2781	422	878	1004
2	1	2714	412	843	981

Analysis

- Variance of query size
- Variance of query complexity
- Covariance of query size and query complexity

Analysis

Hypothesis Testing:

- Training Data:

  Size Complexity

- Predict

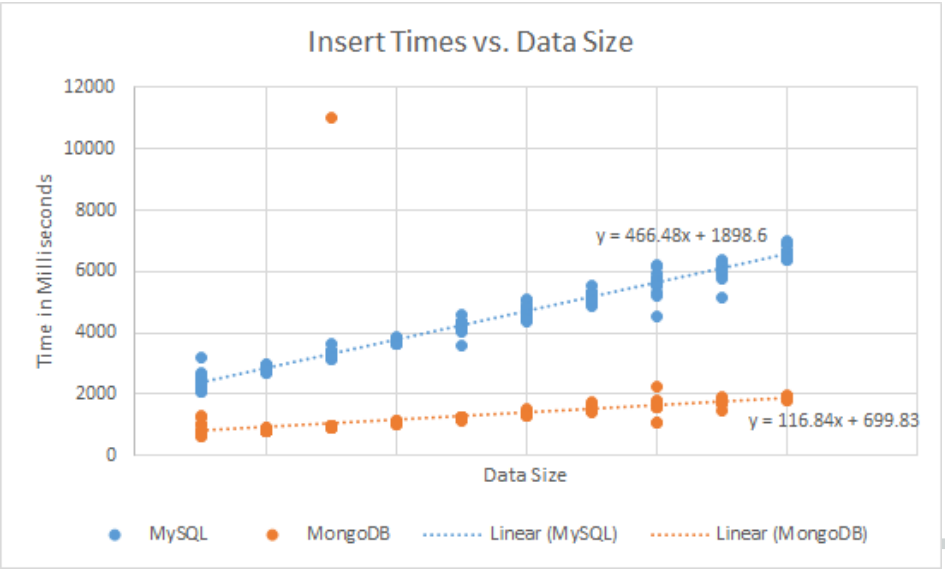
  Faster Insert Faster Response

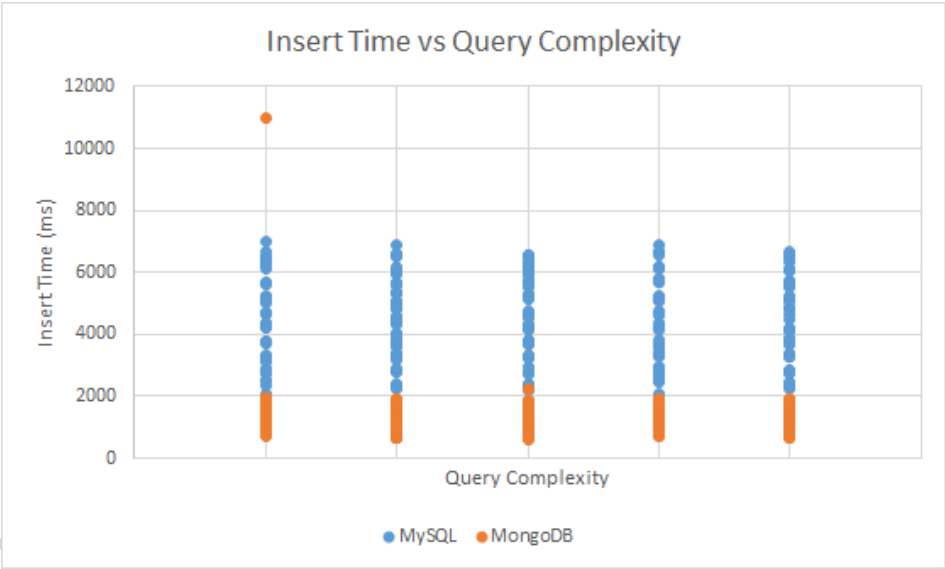
Results

Variance of Query Size:	8.28
Variance of Query Complexity:	2.01
Covariance:	0

Results

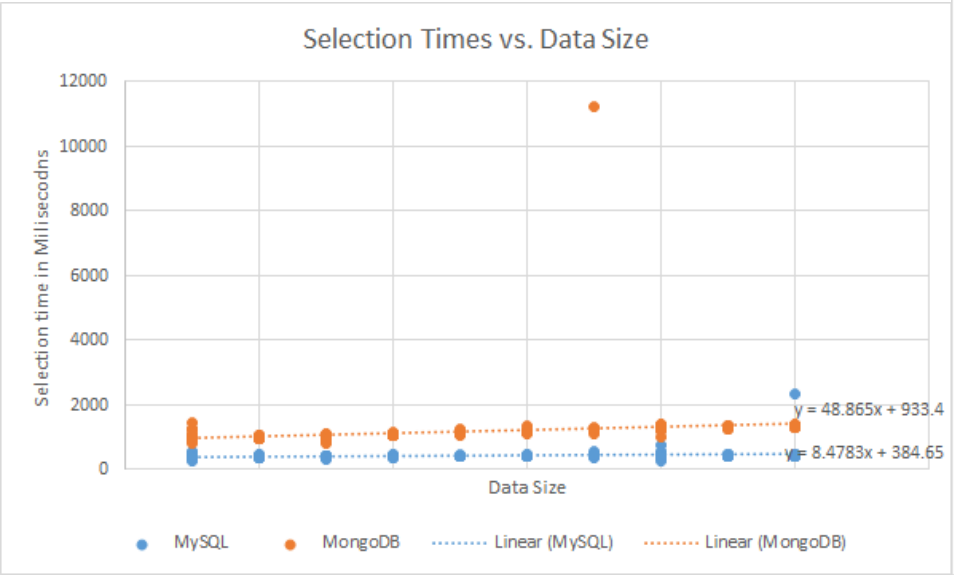
Inserting Into The Databases

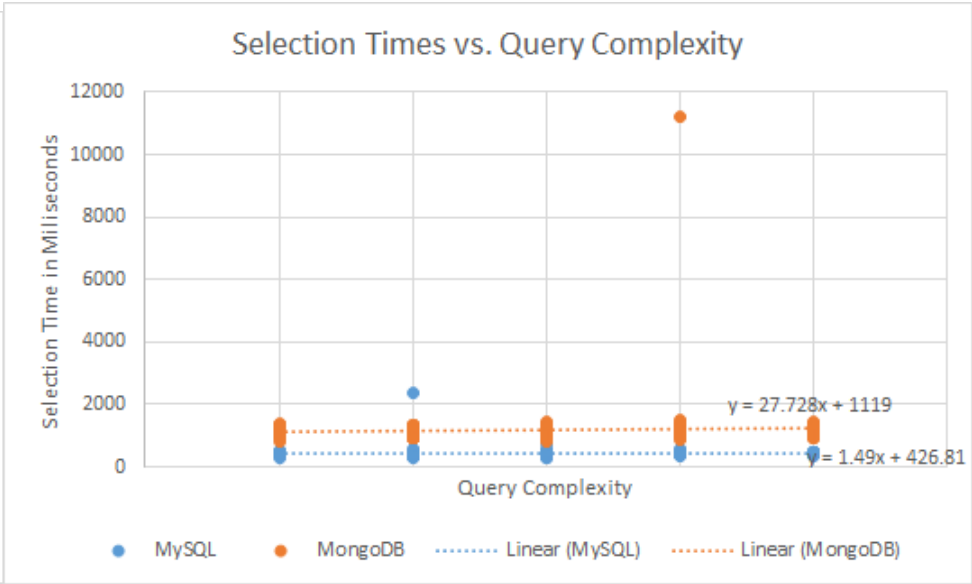




Results

Selection from the Database





Results

Insert database choice: 100% accurate

Response database choice:

Initially 20% accurate

100% accurate after tiebreaker change.

Conclusion

How are size and complexity related?

Can database insert and response times be predicted?

Why This Class Was Wonderful

- Emphasis on projects helps us apply knowledge in real life situations.
- More emphasis on examples.