

Index Suggestion In Peloton

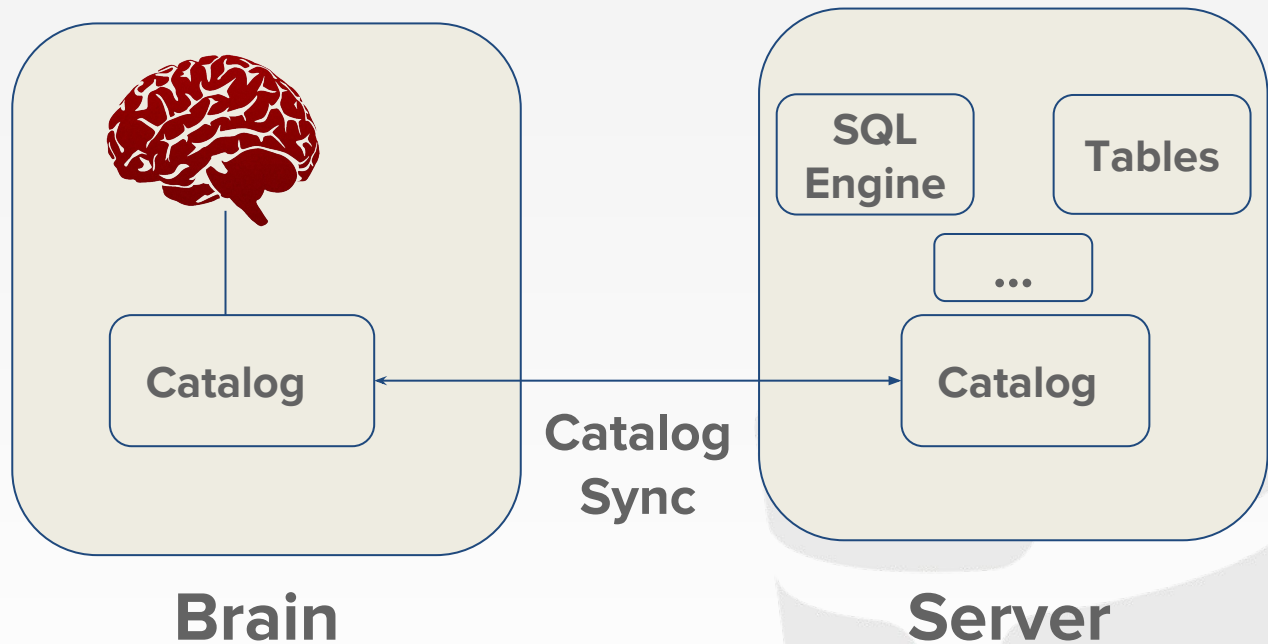
Priyatham Bollimpalli

Sivaprasad Sudhir

Vamshi Reddy Konagari

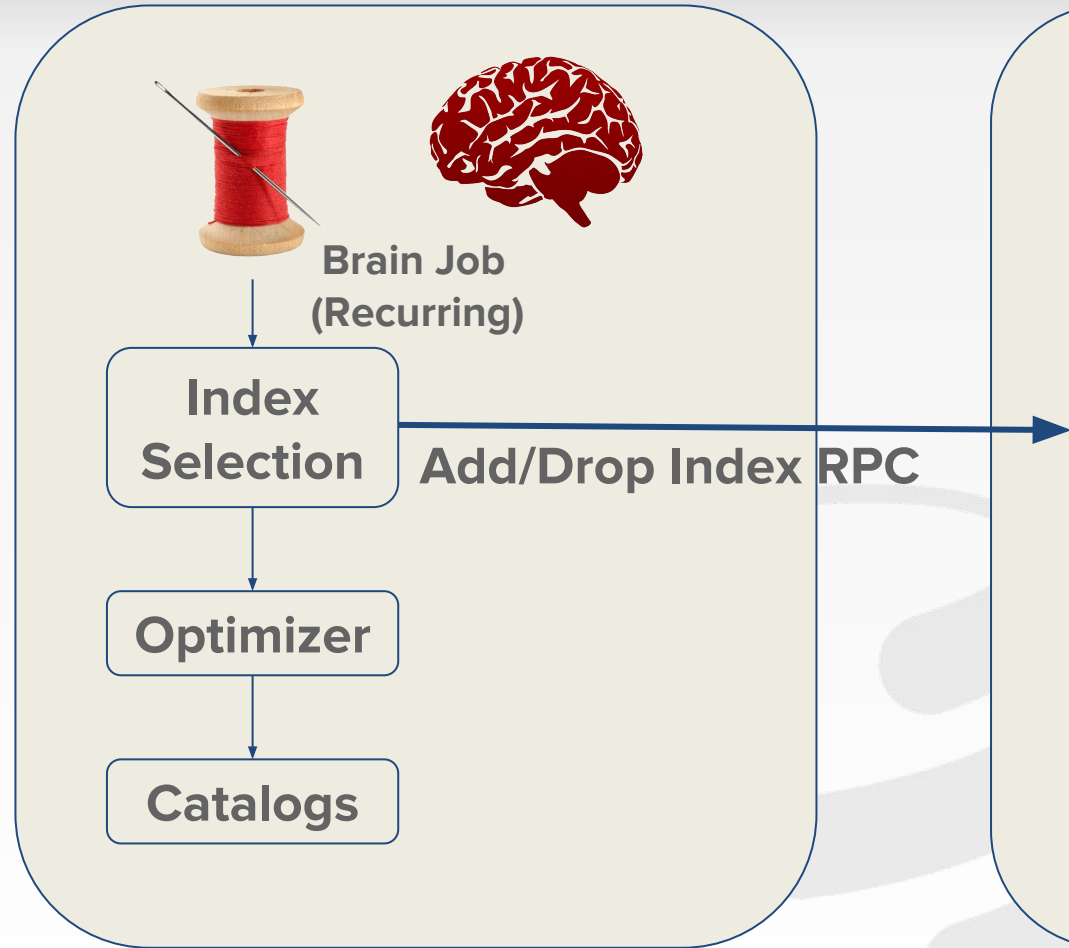
Infrastructure

- We have access to the query history table, stats, etc. at the brain end which can be used by the optimizer



What's in the brain??

- Brain job - scheduled every Δ seconds.
- Index selection Job
- Algorithm bases its cost on the optimizer cost model
- Suggested configuration installed through RPC



Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

Best Indexes?

{ {a}, {b, a}, {b, c} }

Parameters?

#indexes?

#columns in each index?

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

Best Indexes?

{ {a}, {b, a}, {b, c} }

Parameters?

#indexes $k = 2$

#columns in each index $n = 3$

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Some candidates to start with

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Some candidates to start with

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Some candidates to start with

```
{  
  {a}, {b}, {c},  
  {d}, {e}, {f}  
}
```


Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible
Index
Generation

```
{  
  {a}, {b}, {c},  
  {d}, {e}, {f}  
}
```

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

Need to choose k
indexes from these

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

Need to choose k
indexes from these

Naively enumerate
and choose the best
one from all subsets
of size $< k$?

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

Need to choose k

#indexes?

#columns in each index?

#naive enumeration threshold?

of size $< k$?

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

#indexes $k = 2$

#columns in each index $m = 3$

#naive enumeration threshold $n = 1$

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

Need to choose k

indices from these

of size $< k$?

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

Naively enumerate all
subsets of size $< n$
and choose the best
one from it

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {}, {a}, **{b}**, {c}, {d}, {e}, {f} } and choose the best one from it

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

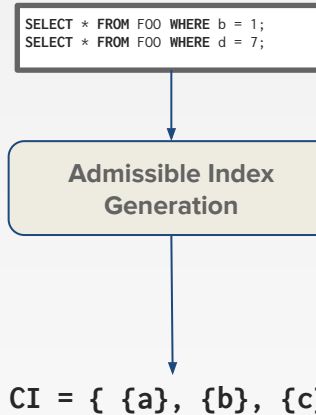
Naively enumerate all
subsets of size $< n$
and choose the best
one from it

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {b} }



We need 2 indexes

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {b} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

Greedily choose the
next best

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { **{b}** }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { **{a}**, {b}, **{c}**, {d}, **{e}**, **{f}** }

Greedily choose the
next best

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {b}, {a} }
 { {b}, {c} }
 { {b}, {d} }
 { {b}, {e} }
 { {b}, {f} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

CI = { {a}, {b}, {c}, {d}, {e}, {f} }

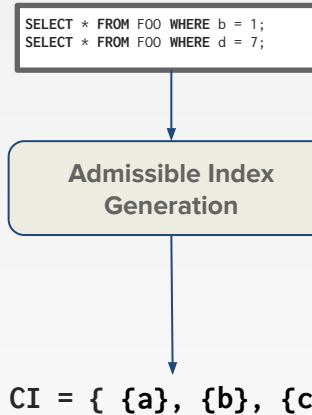
Greedily choose the
next best

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {b}, {a} }
 { {b}, {c} }
 { {b}, {d} }
 { {b}, {e} }
 { {b}, {f} }



Greedily choose the next best

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

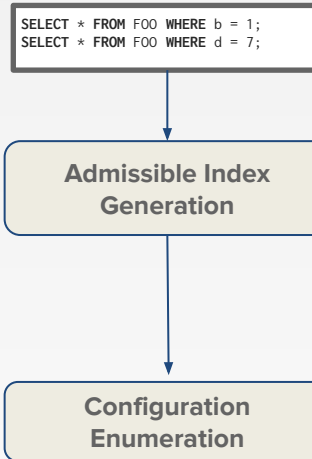
Configuration
Enumeration

CI = { {b}, {d} }

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }



CI = { {b}, {d} }

Multi-column
indexes?

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = CI U (CI × AI)

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

Configuration
Enumeration

CI = { {b}, {d} }

Multi-column
indexes?

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = CI U (CI × AI)

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

Configuration
Enumeration

CI = { {b}, {d} }

CI = {
 {b}, {d},
 {ba}, {bc}, {bd}, {be}, {bf},
 {da}, {db}, {dc}, {de}, {df},
}

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = CI U (CI × AI)

```
SELECT * FROM FOO WHERE b = 1;  
SELECT * FROM FOO WHERE d = 7;
```

Admissible Index
Generation

Configuration
Enumeration

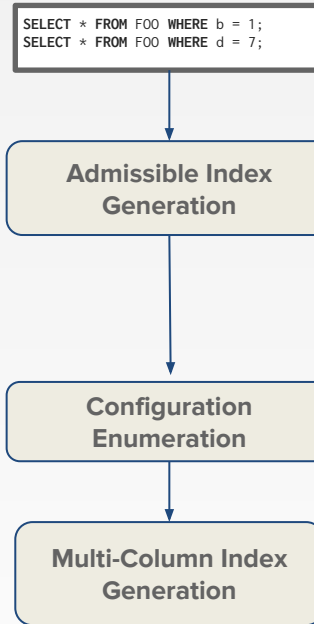
Multi-Column
Index Generation

CI = {
 {b}, {d},
 {ba}, {bc}, {bd}, {be}, {bf},
 {da}, {db}, {dc}, {de}, {df},
}

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

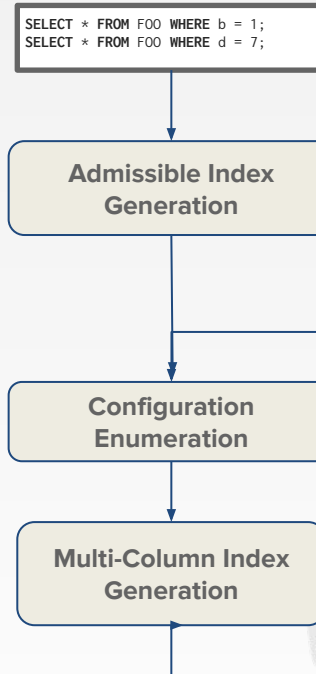


CI = {
 {b}, {d},
 {ba}, {bc}, {bd}, {be}, {bf},
 {da}, {db}, {dc}, {de}, {df},
}

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

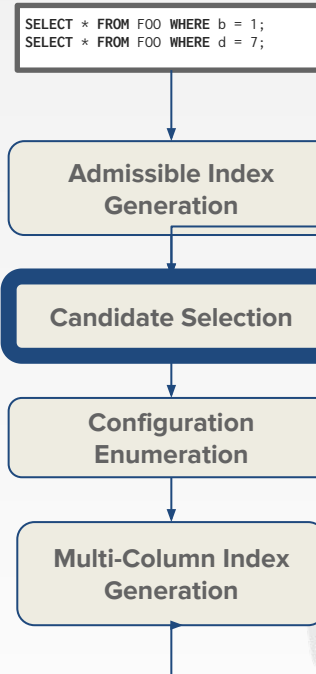


CI = {
 {b}, {d},
 {ba}, {bc}, {bd}, {be}, {bf},
 {da}, {db}, {dc}, {de}, {df},
}

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }



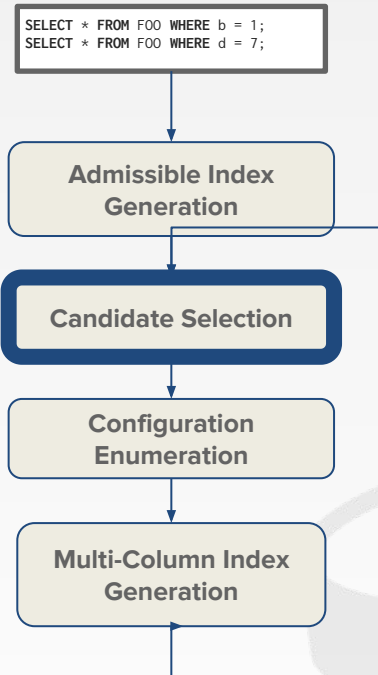
CI = {
 {b}, {d},
 {ba}, {bc}, ~~{bd}~~, ~~{be}~~, ~~{bf}~~,
 ~~{da}~~, ~~{db}~~, {dc}, {de}, {df},
}

Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {b}, {d}, {ba}, {bc}, {dc}, {de}, {df} }

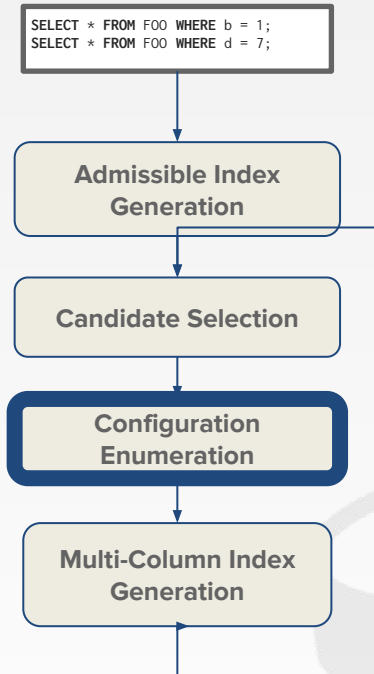


Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

CI = { {b}, {d}, {ba}, {bc}, {dc}, {de}, {df} }

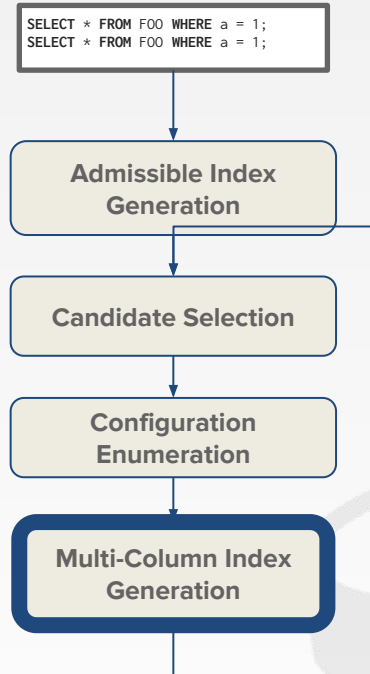


Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

• • •

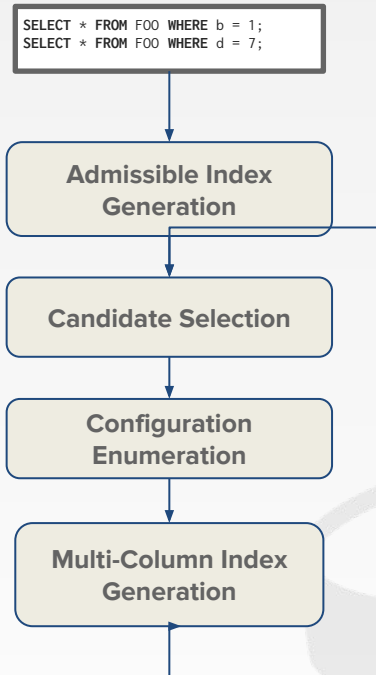


Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }

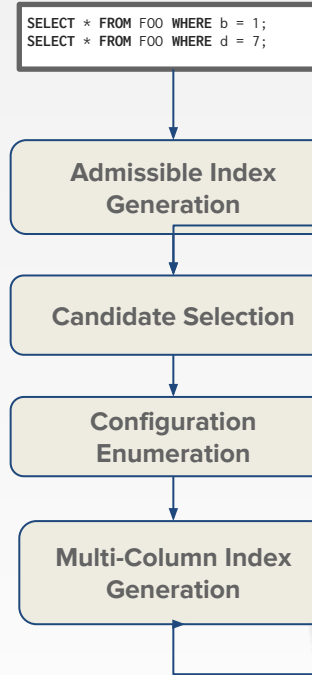
CI = { {bca}, {def} } Best k indexes



Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

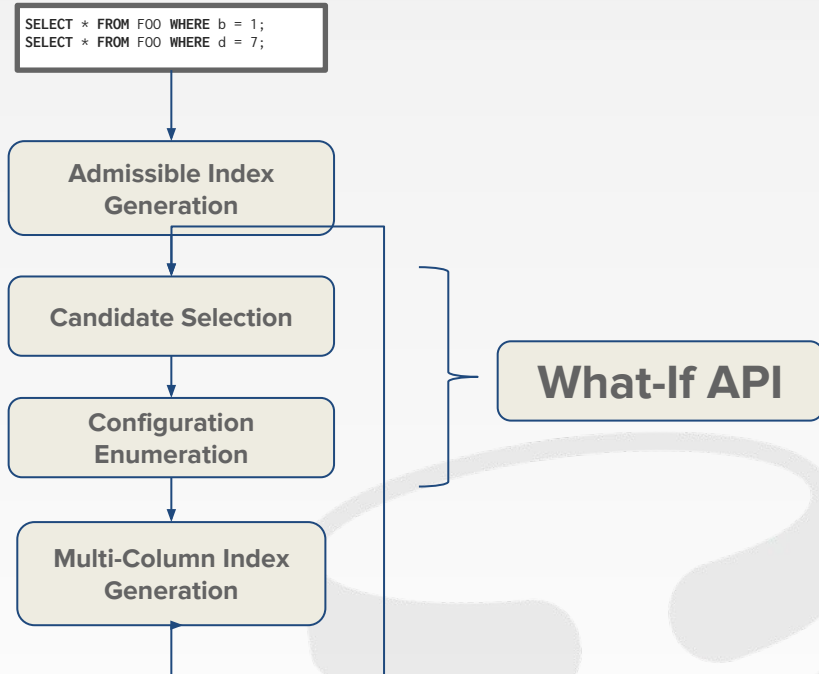
AI = { {a}, {b}, {c}, {d}, {e}, {f} }



Index Selection Algorithm

```
SELECT * FROM FOO WHERE b = 7;  
SELECT * FROM FOO WHERE b = 9 and c = 10;  
SELECT * FROM FOO WHERE a = 1 and b = 2 and c = 3;  
SELECT * FROM FOO WHERE c = 1 and b = 1 and a = 3;  
SELECT * FROM FOO WHERE d = 7;  
SELECT * FROM FOO WHERE d = 9 and e = 10;  
SELECT * FROM FOO WHERE d = 1 and e = 2 and f = 3;
```

AI = { {a}, {b}, {c}, {d}, {e}, {f} }



Testing Framework

- We wrote extensive tests to ensure the correctness of the index selection algorithm and what if index API.
- Found many bugs cost model/optimizer.
- Wrote a generic framework for creating workloads useful for Brain Index Suggestion algorithms.
- Then we thought of running the benchmarks...

Fixes, fixes and fixes...

- TPCC - throughput 9 queries/sec with or without indexes
- Statement cache not flushed on index creation
- Index creation takes ages (on string, floats columns)
- Analyze stats is not used by anyone - crashes on using it for the second time concurrently.
- Optimizer never considered multi-column indexes
- Optimizer - explicitly have to call Analyze stats - which segfaults if you call twice
- TPCC without primary keys segfaults.
- Prepared statements were not being logged

TPC-C Throughput

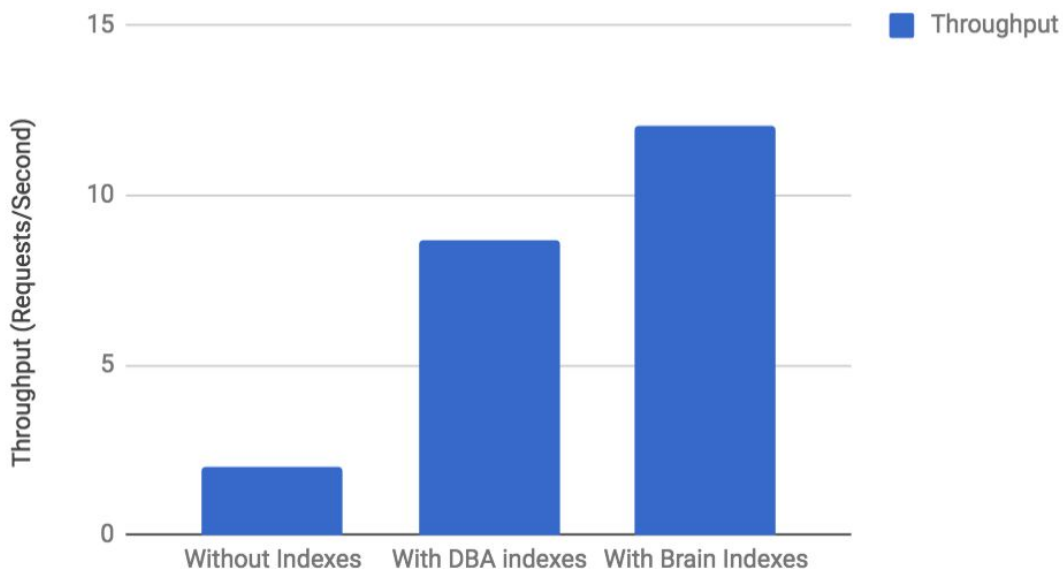
Workload: TPCC

Without Indexes: No primary or secondary indexes

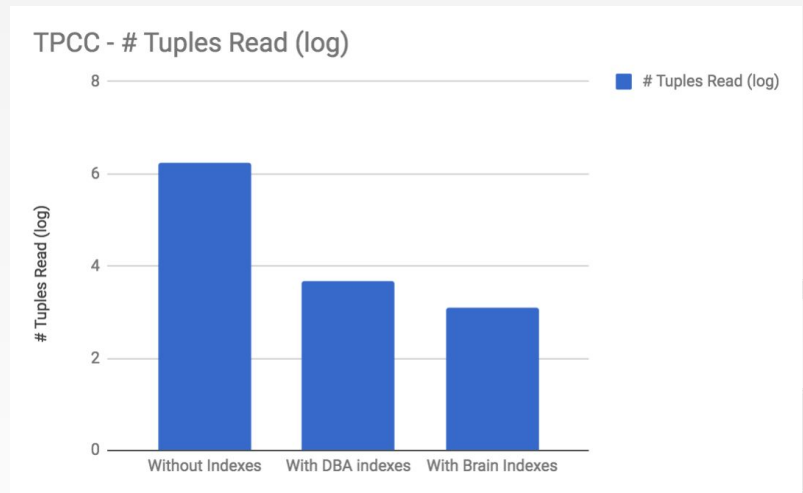
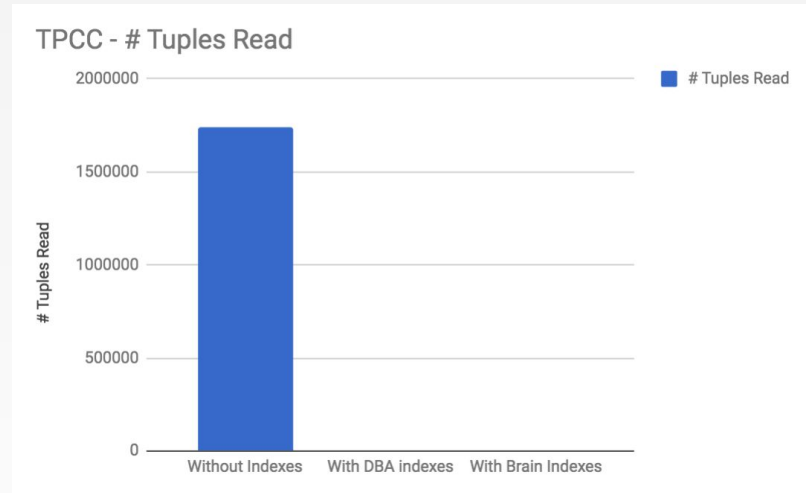
With DBA Indexes: Primary and secondary indexes in the workload

With Brain Indexes: Start with no indexes and use the indexes suggested by our algorithm

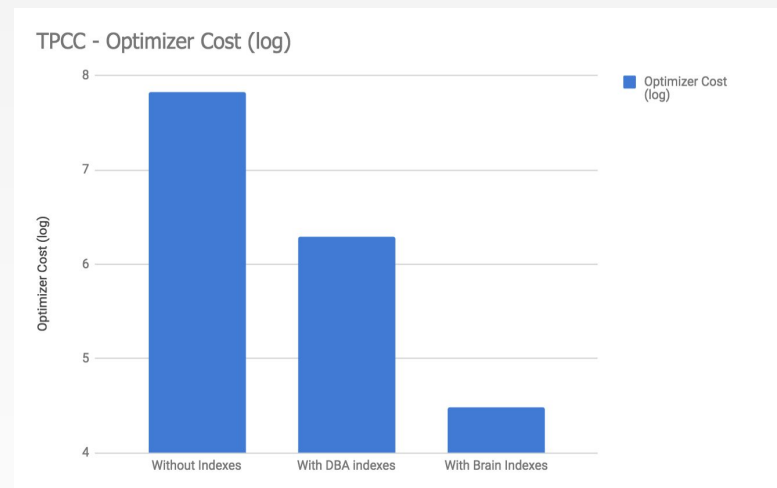
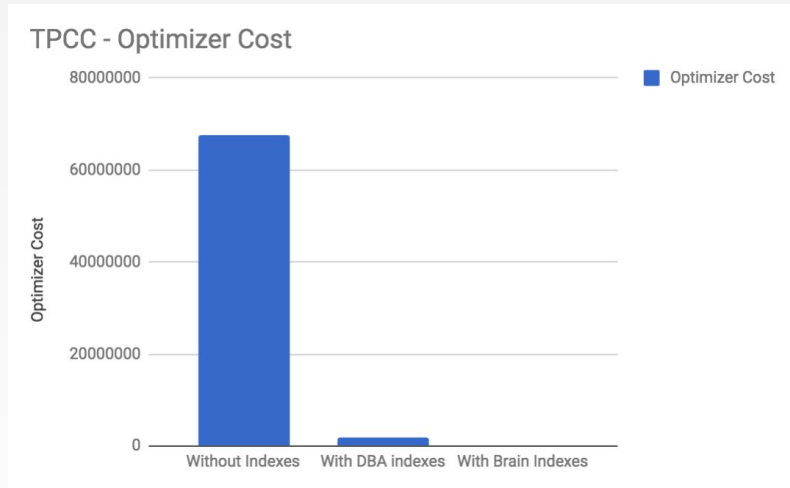
TPC-C Throughput



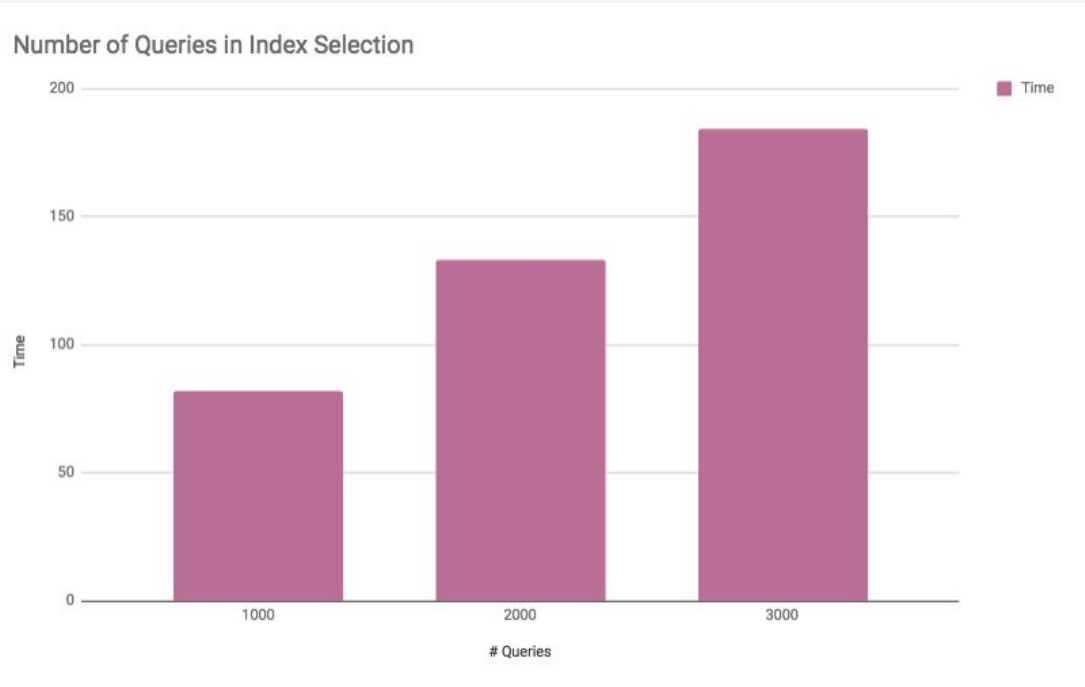
Number of tuples accessed



Optimizer Cost



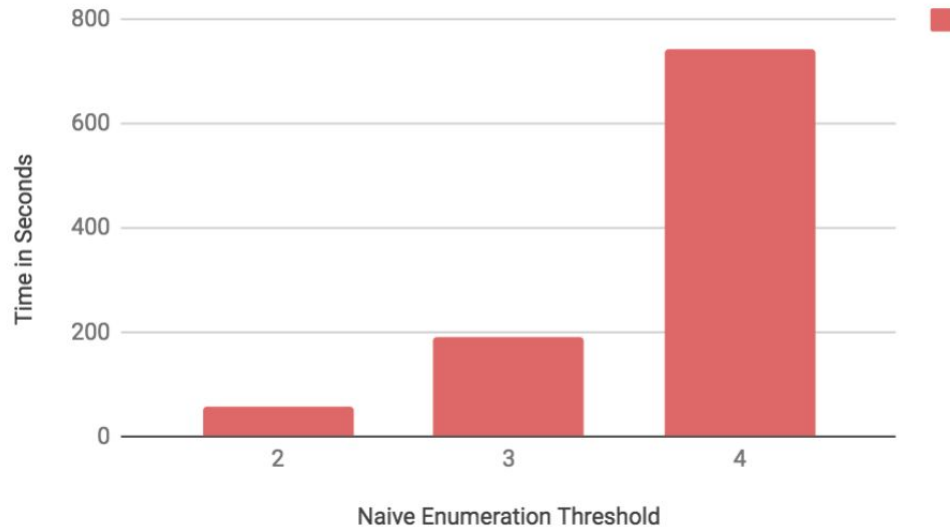
Algorithm Runtime - Number of queries



```
#indexes = 10  
#columns = 2  
#naive enumeration  
threshold = 2
```


Algorithm Runtime - Naive enumeration threshold

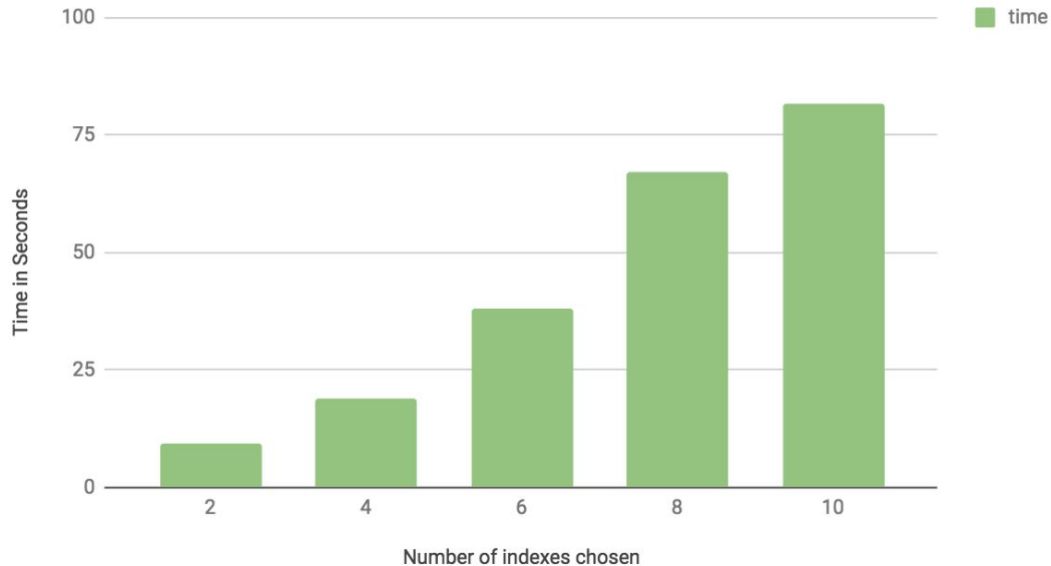
Naive Enumeration Threshold in Index Selection



```
#indexes = 10  
#columns = 3  
#queries = 2000
```

Algorithm Runtime - Number of indexes chosen

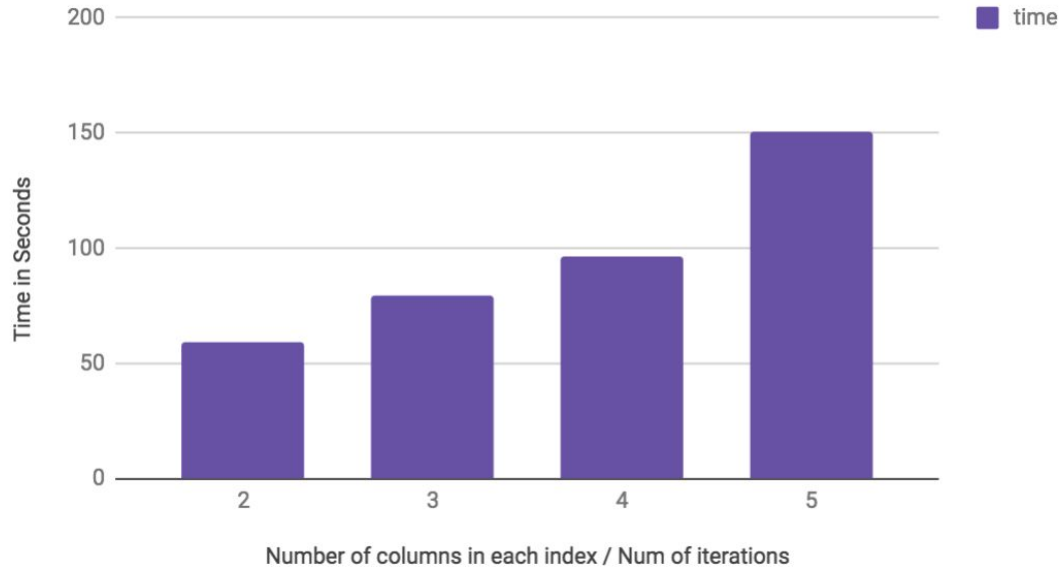
Number of Indexes Chosen in Index Selection



```
#columns= 3  
#naive enumeration  
threshold = 2  
#queries = 1000
```

Algorithm Runtime - Number of columns / iterations

Multi-column Index Generation in Index Selection



```
#indexes = 10  
#naive enumeration  
threshold = 2  
#queries = 1000
```

Future Work

- Tuning the knobs
- Incorporate memory usage of the indexes created
- Online version

Thanks :)

Questions?

