System Catalog Object

PostgreSQL:
```
postgres=# select * from pg_leon;
table_oid | leons
-----------|------
    100     | 1
    101     | 2
(2 rows)
```

Catalog Table “pg_leon”

```
class LeonCatalog {
  // Write API
  bool InsertLeon(oid_t table_oid, int leons);
  bool DeleteLeon(oid_t table_oid);
  // Read API
  int GetLeons(oid_t table_oid);
};
```
System Catalog Demo
DDL Operations Are Slow!

- Copying: DDL operations disrupt production systems and reduce throughput.
- Blocking: DDL operations block concurrent transactions on the same table.
- Some schema change can take hours!
Naive ALTER TABLE with 2PL (Postgres)

- Table-level explicit locking
- Use hierarchical 2PL with deadlock detection
- Can be examined through `pg_locks` system view

https://www.postgresql.org/docs/9.6/static/explicit-locking.html
Naive ALTER TABLE with 2PL (Postgres)

Access Exclusive Lock

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>l</td>
<td>m</td>
<td>n</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Table Foo (old schema)

Table Foo (new schema)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>z</td>
<td>0</td>
</tr>
</tbody>
</table>
Lazy & Online ALTER TABLE

- How to achieve lazy?
  Keep old data in place, but interpret with new schema (avoid copying data)

- How to achieve online?
  Make schema multi-versioned (avoid blocking on concurrent transactions)
ALTER TABLE

Physical Schema Object Caches

Schema Version N+1

Schema Version N

Schema Version 0

Logical table Foo

Data Table Version N+1

Data Table Version N

Data Table Version 0

Index

Txn 150: SELECT *
Txn 200: ALTER TABLE

<table>
<thead>
<tr>
<th>pg_table</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Begin-TS</code></td>
</tr>
<tr>
<td>~ End-TS</td>
</tr>
<tr>
<td>0~100</td>
</tr>
<tr>
<td>100~200</td>
</tr>
<tr>
<td>200~Inf</td>
</tr>
</tbody>
</table>
Schema Mapping

- Use **unique Column_id** per logical table for schema mapping
Tuple Migration

- When to update old data tuples with new schema?
  - On tuple access
  - On tuple update
  - Use a background worker
Future Work

- Online ALTER TABLE working in progress
- ALTER TABLE benchmarking
- Support transactional DDL