1. Bi-temporal Data

Bi-Temporal Databases store "history" of data:
- **system time**: when a fact is recorded in a DBMS
- **application time**: when a fact is valid in real life
- not necessarily correlated
- often open-ended lifetime

Different Update Semantics
- **system time**: append-only
- **application time**: arbitrary
- bi-temporal: application time updates only with system time updates

Not supported well by existing data structures

2. Bi-temporal Analytics

A) Temporal Aggregation
- Aggregation grouped by time

B) Time Slicing
- Consistent snapshot at a particular time

C) Temporal Join
- Correlating tables on temporal dimension(s)

Workloads require both multi-dimensional and time-ordered access, but often there is a dominant dimension:
- slice one dimension, evaluate other

3. Bi-temporal Timeline Index

**Bi-temporal Table**

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Balance</th>
<th>StartApp</th>
<th>EndApp</th>
<th>StartSys</th>
<th>EndSys</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Smallville</td>
<td>50</td>
<td>10</td>
<td>∞</td>
<td>100</td>
<td>102</td>
</tr>
<tr>
<td>John</td>
<td>Smallville</td>
<td>40</td>
<td>10</td>
<td>11</td>
<td>102</td>
<td>∞</td>
</tr>
<tr>
<td>John</td>
<td>Largeville</td>
<td>30</td>
<td>10</td>
<td>13</td>
<td>105</td>
<td>110</td>
</tr>
<tr>
<td>John</td>
<td>Largeville</td>
<td>30</td>
<td>14</td>
<td>14</td>
<td>105</td>
<td>106</td>
</tr>
<tr>
<td>Max</td>
<td>Newtown</td>
<td>80</td>
<td>15</td>
<td>15</td>
<td>109</td>
<td>∞</td>
</tr>
</tbody>
</table>

- Lifetime intervals for each time dimension
- open intervals for undetermined end
- updates close system intervals or add a new row

**Logical Index Design**

- Single log-based index over system time
- Application time indexes at selected system times
- Checkpoints to limit log scan ranges

**Physical Index Design**

4. Query Processing

Depending on selectivity of each time dimension
and cost of temporal operations, perform a (combination of)
- index scan
- column scan
- partial application index rebuild

**Example**:

```sql
SELECT SUM(balance) FROM user
FOR SYSTEM_TIME AS OF 75 GROUP BY APPLICATION_TIME()
```

5. Measurements

A) Temporal Aggregation

```
Query: SELECT MAX(o_totalprice) 
FROM orders 
FOR SYSTEM_TIME AS OF TIMESTAMP "[SYS_TIME]",
WHERE o_orderstatus = 'O' 
GROUP BY ACTIVE_TIME() 
```

B) Time Slicing (both times)

```
Query: SELECT AVG(ps_supplycost) 
FROM partsupp 
FOR SYSTEM_TIME AS OF TIMESTAMP "[SYS_TIME]"
FOR APPLICATION_TIME AS OF TIMESTAMP "[APP_TIME]"
```

C) Temporal Join

```
Query: SELECT COUNT(*) 
FROM customer c 
TEMPORAL JOIN orders o 
ON c.SYSTEM_TIME OVERLAPS o.SYSTEM_TIME AND c.APPLICATION_TIME OVERLAPS o.APPLICATION_TIME 
WHERE c.custkey = o_custkey 
AND o_orderstatus = 'O' 
AND o_totalprice > 5000 AND 
c acctbal < 100
```