Physical Logging

states (byte images) are logged

**before image** contains state **before** change was performed

**after image** contains state **after** change was performed

Page 42

- lens
- aperture
- Kamera
- depth of field

Page 42

- lens
- aperture
- camera
- depth of field

Log

Page 42: image at 367,2; before: ‘Ke’; after: ‘ca’
Logical Logging

high-level operations are logged

not necessarily limited to a single page

<table>
<thead>
<tr>
<th>CameraLingo</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>termID</strong></td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

Page 42

- lens
- aperture
- Kemera
- depth of field

Page 42

- lens
- aperture
- camera
- depth of field

Log

CameraLingo: update(0, 'Kemera' => 'camera')
Physiological Logging

like logical logging, but:

log entry **may only affect a single page**

```
Example for capital character
=> 3 log records
```
Main Performance Trade-Off (disk-based Systems)

- Logical (transactions)
- Logical (logical operations, e.g. insert/delete to table)
- Physiological
- Physical

$log\text{-size} \approx I/O\text{-time to read log}$

Log entry processing time [minutes]
Main Performance Trade-Off (Main Memory Systems)

- Logical (transactions)
- Logical (logical operations)
- Physiological
- Physical

Log entry processing time [ms]

Best deal

Log-size ≈ I/O-time to read log
Logical Logging Example in Main Memory

just store the invocation parameters of the stored procedure in the log

<table>
<thead>
<tr>
<th>queryID</th>
<th>SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>UPDATE CameraLingo SET term=$A WHERE term=$B</td>
</tr>
<tr>
<td>1</td>
<td>INSERT INTO CameraLingo VALUES ($A,$B)</td>
</tr>
<tr>
<td>2</td>
<td>DELETE FROM CameraLingo WHERE term=$A</td>
</tr>
</tbody>
</table>

Log
2:‘lens“
0:‘camera‘:‘Kamera‘
1:42:‘tripod‘