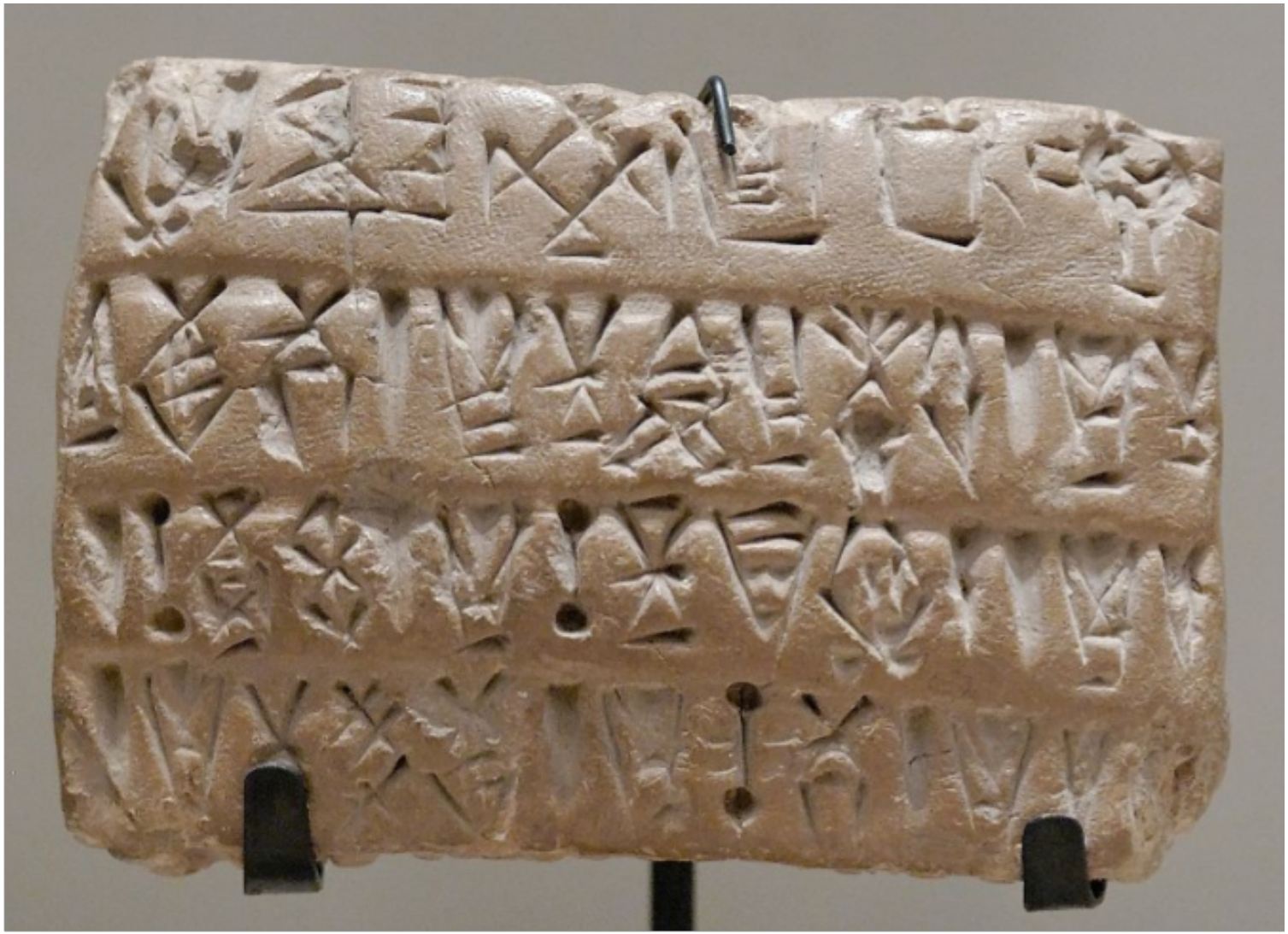
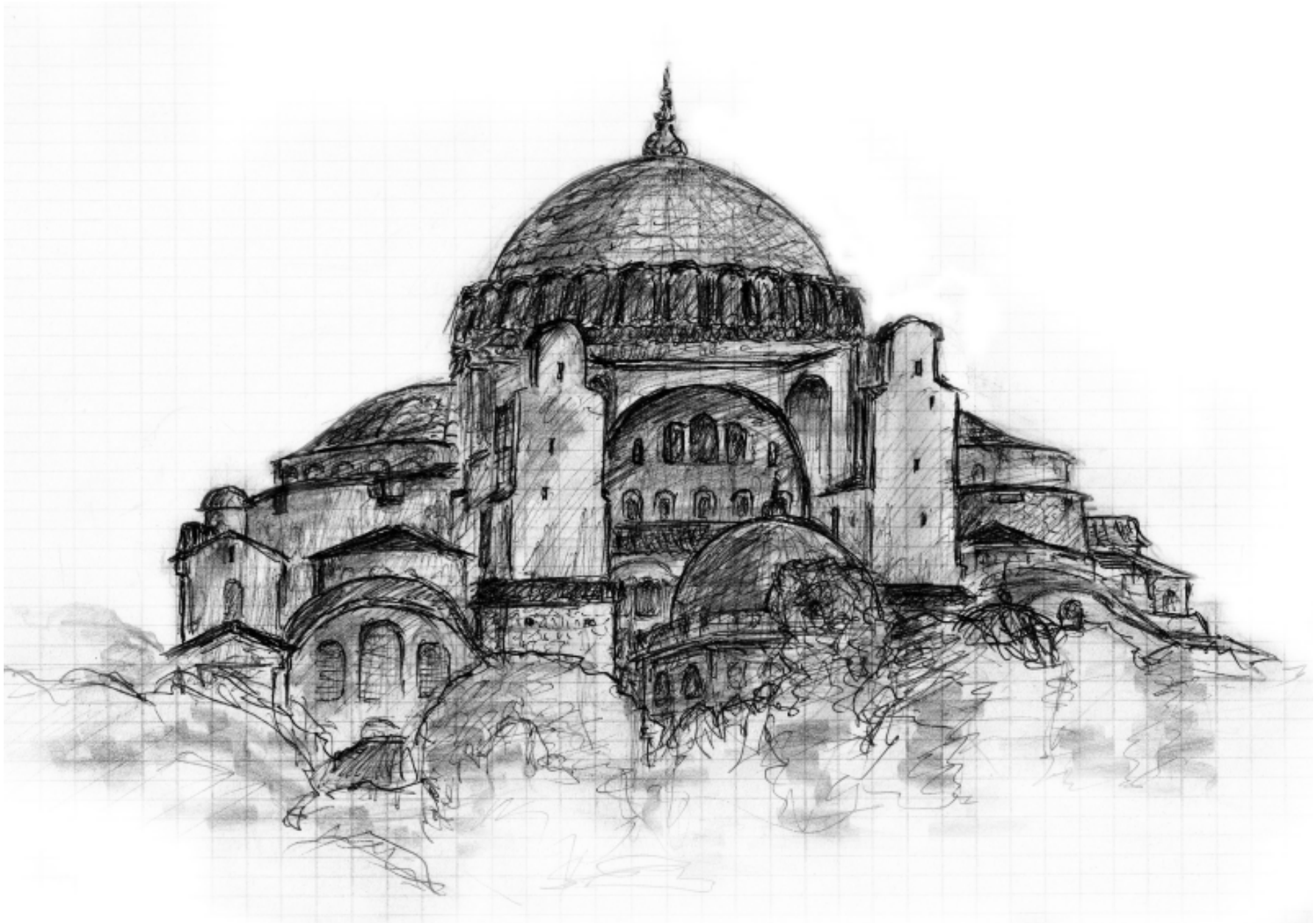
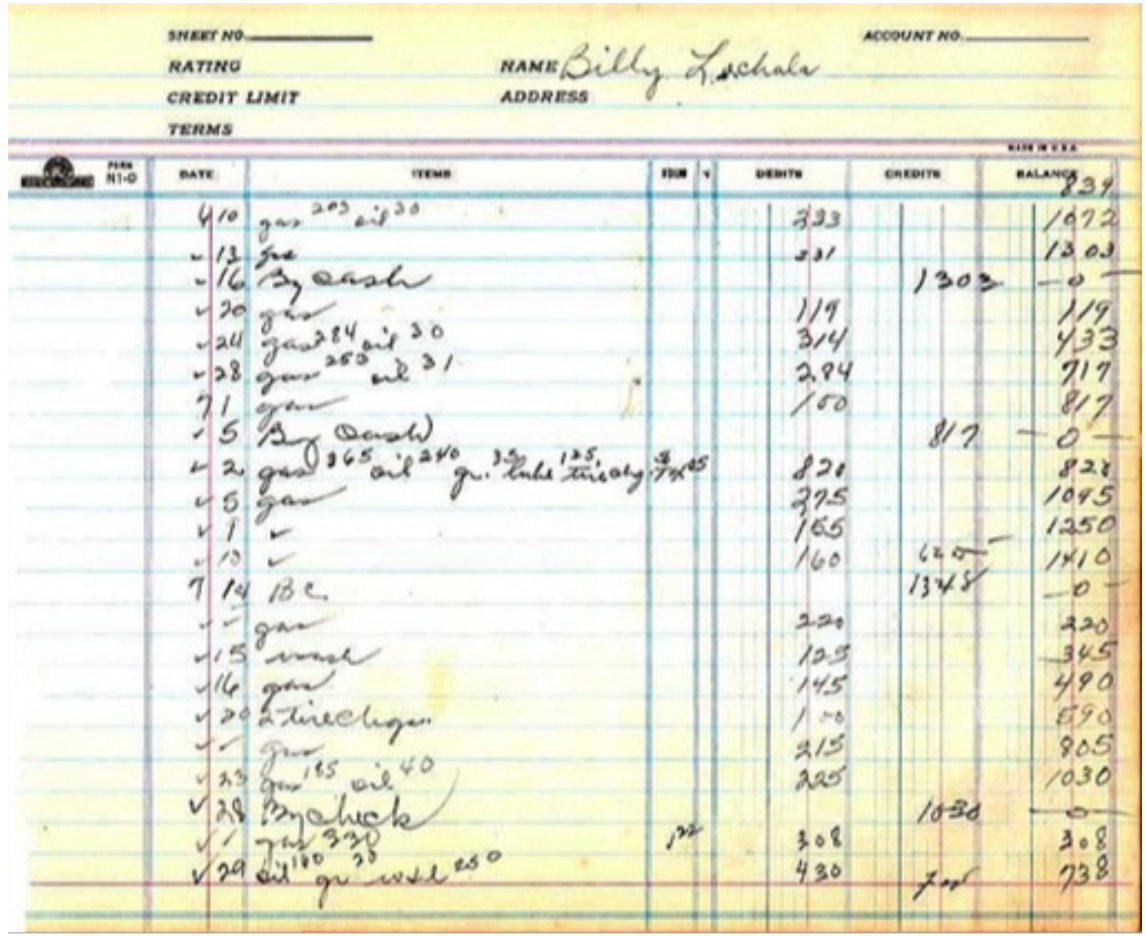
Better NoSQL

Economic transaction stored on Tablet from Mesopotamia





Economic transactions stored in relational “way” - e.g. table



SHEET NO. _____ ACCOUNT NO. _____

RATING NAME Billy Lachala

CREDIT LIMIT ADDRESS

TERMS

DATE	ITEMS	FOR	✓	DEBITS	CREDITS	BALANCE
						839
4 10	gas 205 oil 30			333		1072
✓ 13	Gas			231		1303
✓ 16	By cash				1303	— 0 —
✓ 20	gas			119		119
✓ 24	gas 284 oil 30			314		433
✓ 28	gas 250 oil 31			284		717
7 1	gas			100		817
✓ 5	By Cash				817	— 0 —
✓ 2	gas 365 oil 240 gr. 35 tube 125 tires 4 @ 7.45			820		820
✓ 5	gas			275		1095
✓ 1	✓			155		1250
✓ 10	✓			160	640 —	1410
7 14	Bc.				1348	— 0 —
✓ —	gas			220		220
✓ 15	mdse			125		345
✓ 16	gas			145		490
✓ 20	2 tire chgs			100		590
✓ —	gas			215		805
✓ 23	gas 185 oil 40			225		1030
✓ 28	By check				1030	— 0 —
✓ —	gas 330		122	308		308
✓ 29	oil 180 30 gr. mdse 250			430		738

Why NoSQL

"I think the primary lesson the Postgres community can learn from NoSQL is that not everyone needs all our features, and if it can be easily accomplished, we should allow users to trade features for some of the NoSQL desired behaviors.“

Bruce Momjian

(author of PostgreSQL)

Why NoSQL

Memory is the new disk, disk is the new tape.

Jim Gray

Storage comparison

	Speed	Price / GB
Memory	1	2000 EUR
SSD	1,000	2.00 EUR
HDD	100,000	0.20 EUR

Why NoSQL

If you're deployng memcache on top of your database, you're inventing your own ad-hoc, difficult to maintain NoSQL data store

Ian Eure

So what is Redis?

Key/value data store, similar to Memcached

Developed by

- Salvatore Sanfilippo and
- Pieter Noordhuis

More then any other NoSQL

- Persistance
- Replication
- More data types

More info at:

- http://redis.io/

So what is Redis?

] **SET USER1 Niki**

OK

] **GET USER1**

Niki

] **SADD ALL 1**

OK

] **SADD ALL 5**

OK

SMEMBERS ALL

1

5

So what is Redis?

] **INCR A**

1

] **INCRBY A 2**

3

] **GET A**

3

Client Libraries

ActionScript C C# C++

Clojure Lisp Erlang Go Haskell
Java Lua Node.js Objective-C

Perl PHP Pyhon Ruby Scala

Smalltalk TCL

Blazing fast

about 110,000 SETs per second,
about 81,000 GETs per second.

http://redis.io/topics/benchmarks

SQL cache

```
$r = new Redis();
$r->connect("127.0.0.1", "6379");

$id = (int) $_REQUEST["id"];
$sql = "select * from users where id = $id";

// Create a hash key
$key = "sql_cache:" . md5($sql);

// Check if data is in cache.
if ($data = @unseralize( $r->get($key) ) === false){
$data = exec_db_query($sql);
// Put data into cache for 1 hour
$r->set($key, serialize($data) );
$r->expires($key, 3600);
}

display_data($data);
```

IP restriction

```
// First of all, check if the IP address is member of trusted set...
if ( ! $r->sismember("trusted:all", $_SERVER["REMOTE_ADDR"]) ){
// Create a hash key
$key = "ip:" . $_SERVER["REMOTE_ADDR"];
// Increase the hit counter
$hits = $r->incr($key);
$r->expires($key, 3600);
// Check if this IP had more than 1000 visits for 1 hour...
if ($hits > 1000){
// Something fishy going on, redirect...
header("location: page_with_capcha.php");
exit;
}
}

// Web page continue here...
// ...
```

Random news

```
$newsroom = "sport";
$key = "news:" . $newsroom;

//Get random item...
$item = $r->lindex($key, rand(0, $r->llen
($key) - 1));

echo $item;
```

Processing queue

```
for($i = 0; $i < 1000; $i++){
$x = $r->incr("id:work");

$r->hmset(
"work:$x", array(
"id" => $i ,
"a" => rand() ,
"b" => rand()
)
);

$r->sadd("pool:work", $x);
}

echo "Work distributed\n";
```

Processing queue #2

```
while(true){
// Get random work id...
$x = $r->spop("pool:work");

if ($x === false){
echo "No more work to do\n";
sleep(10);
continue;
}

// Get work description...
$data = $r->hgetall("work:$x");

if (!count($data))
continue;

// Do the work...
$total = $data["a"] + $data["b"];

// Store back...
$r->hset("work:$x", "total", $total);

// And store id in finished pool...
$r->sadd("pool:finishedwork", $x);

echo "Work #$x is finished...\n";
// sleep(2);
}
```

Locks

```
$team_name = "Blue team";

$user = "john";

$key = "lock:" . $team_name;

// Try to get the lock...
// e.g. set the key only if not exists...
if ($r->setnx($key), $user){
// Because system can crash, give him 5 min.
$r->expire($key, 60 * 5);

update_the_info();

// Release the lock...
$r->del($key);
}else{
echo "Warning! User " . $r->get($key) . " updating the information at the moment. Try again later :)";
}
```

Database design

http://redis4you.com/articles.php?id=009

Try Redis 4 Free

http://redis4you.com/

Questions?

nmmm(at)nmmm.nu

skype: nmmmnu