Tridex Db2 z/OS Virtual Meeting
September 23, 2021

Is your test data as agile as your developers?

Kai Stroh, UBS Hainer GmbH

Agenda

- New challenges from agile development
- Creating small test beds
- Making expert knowledge available
- Cloning very large databases
- Final thoughts

Development models are changing (1|3)

- In 2005, about 14% of all businesses used agile development methods
- In 2013, that number had grown to 84%
- In 2016, that number had grown to 95%
- Key aspects of agile development methods:
  - Continuous integration
  - Continuous delivery
  - Test driven development

Development models are changing (2|3)

- Waterfall model, V model, Rational unified process…
- Testing is done at well defined points in the process
- Prepare a small number of large test beds ahead of time

- Agile development, Scrum, Continuous integration…
- Testing is done daily, hourly, with every code change
- Prepare hundreds of small test beds on demand

Development models are changing (3|3)

- Performance tests, regression tests etc. still exist
- You don’t need to replace the large test beds, but have small test beds **in addition** to what you already have
- Small test beds are often created on demand and only exist for the duration of a test
- Creating small test beds can be more challenging than making a full copy of everything

Typical situation

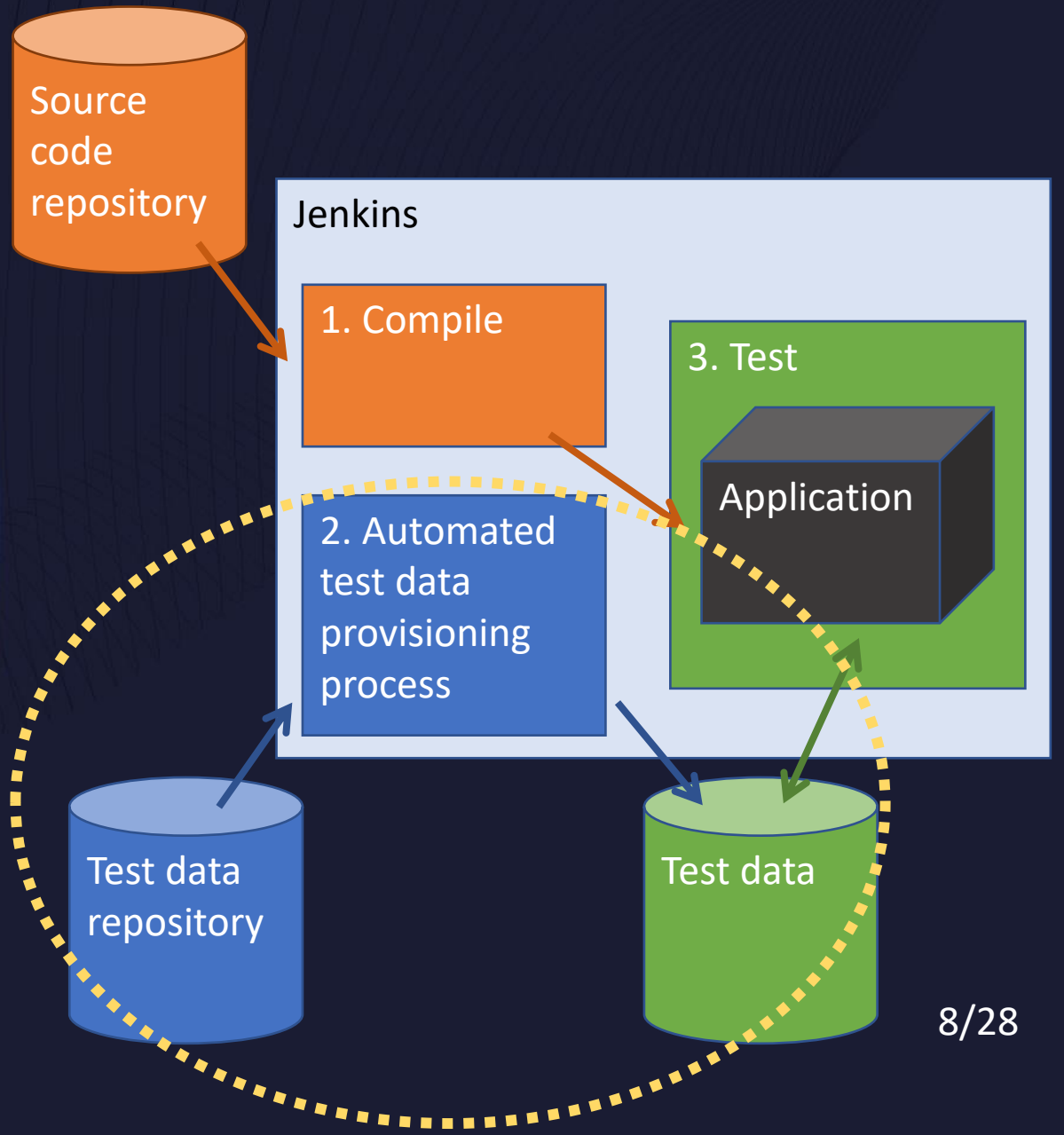
- Often based on Unload/Load
- Separate tools required for DDL generation
- Hundreds of jobs
- Data is taken directly from production
- Requires lots of temporary space (unless cross loader is used)
- Few environments
- Large environments (possibly full copies)

Usual complaints

- Refreshes are done too infrequently, data becomes stale
- Multiple tests run against the same tables
- Takes days or even weeks for a refresh, DBAs need to monitor jobs, check many return codes
- When problems occur, DBAs need to check what parts are missing, rewrite and rerun jobs
- Long running jobs with high CPU load have negative impact on four-hour rolling average MSU value
- LOB and XML data is difficult to copy

Build process

- Step 1: Check out branch from VCS and compile application
- Step 2: Create test database and populate with data
- Step 3: Run automated tests



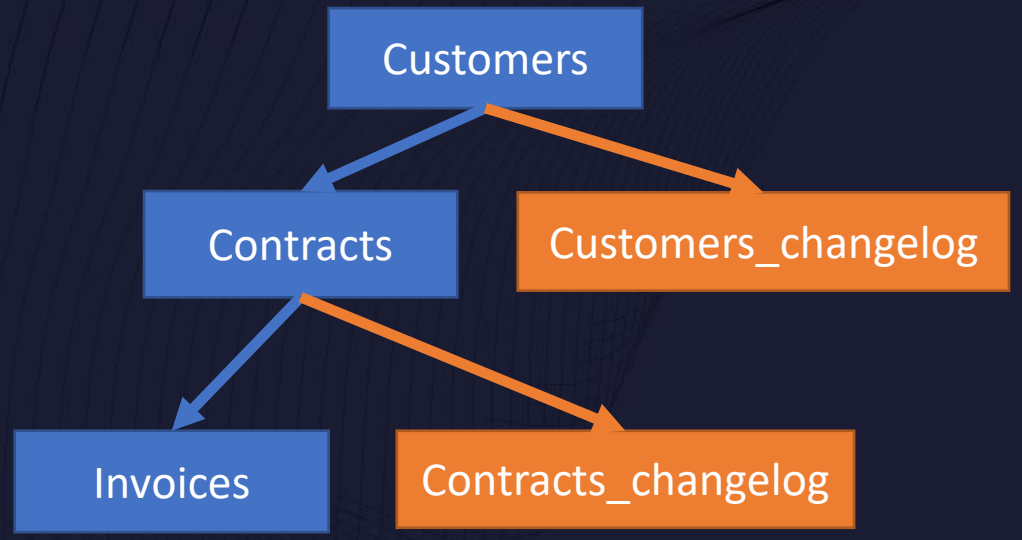
Test data for automated builds

- Typically small amount of data
  - Tests execute faster
  - Provisioning is done in seconds to minutes, not hours
  - Done via SQL (slow, but maximum flexibility)
- Larger environments are still needed for performance tests

Small test beds (1|2)

- Simple INSERT scripts are not sufficient. Program logic is required, for example to generate new primary keys
- Ability to use parameters in copy processes changes the WHERE condition and the number of rows that need to be copied
- Applications may use tables to store meta information about other tables (redundant, but common) - This cannot be handled by a simple INSERT script
- Logic is required to embed data into tables that had structural changes, for example to generate new data for extra target columns

Small test beds (2|2)

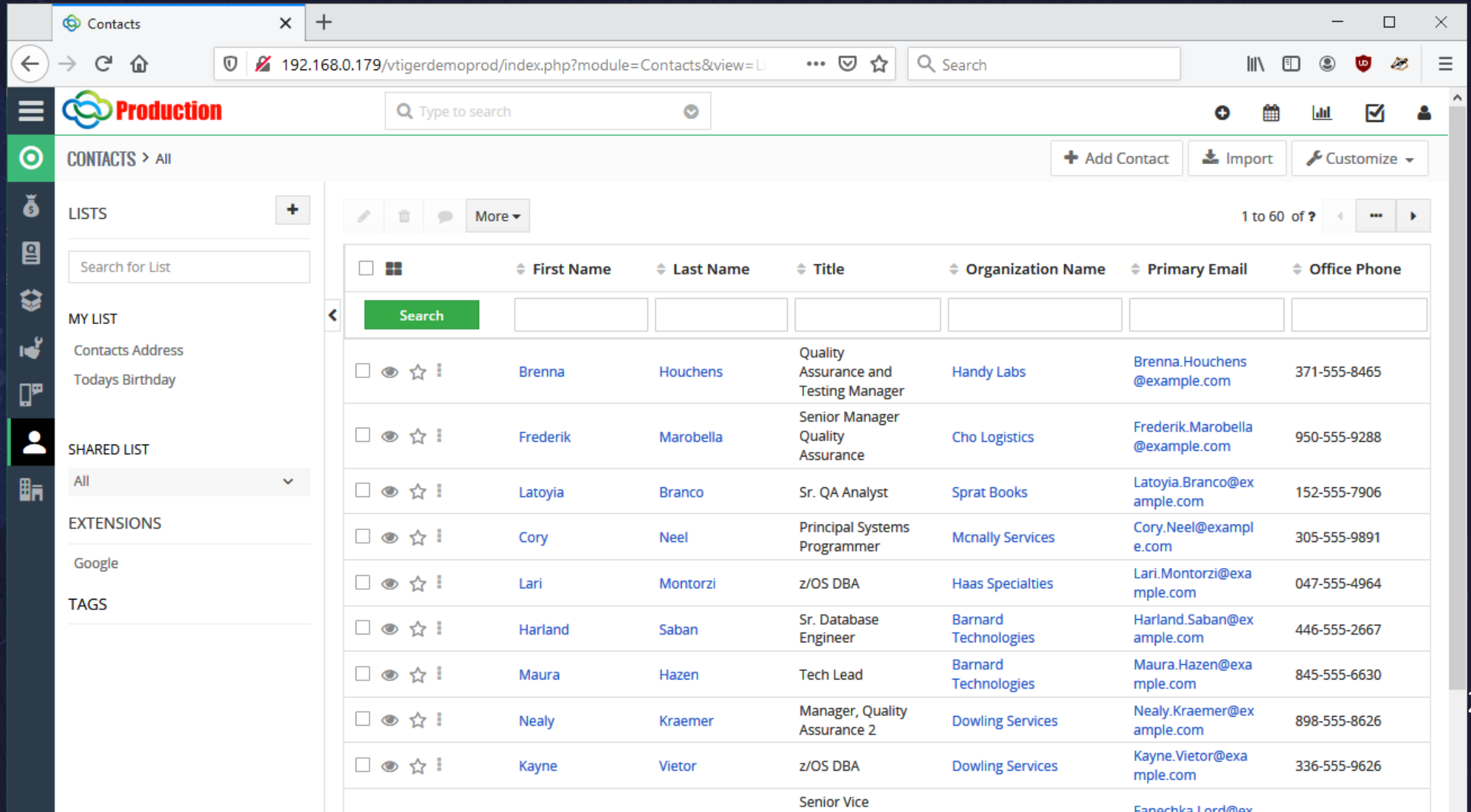


- Foreign keys may or may not exist
- Foreign keys
  - tell the database when data is consistent
  - dictate the order in which INSERTs can be done
- Foreign keys cannot help gauge when data makes sense for a test
- Need to define a “path” through the network of tables: After extracting rows from table X, which rows from which tables need to be extracted next?
- Maybe skip a very large history / protocol table despite foreign key

Example (1|4)

- Copy test case data for a CRM system
- Sample application: Vtiger 7.1.0
- Goals:
  - Easily copy organizations (accounts) that meet certain criteria
  - Including the contact names and sales opportunities

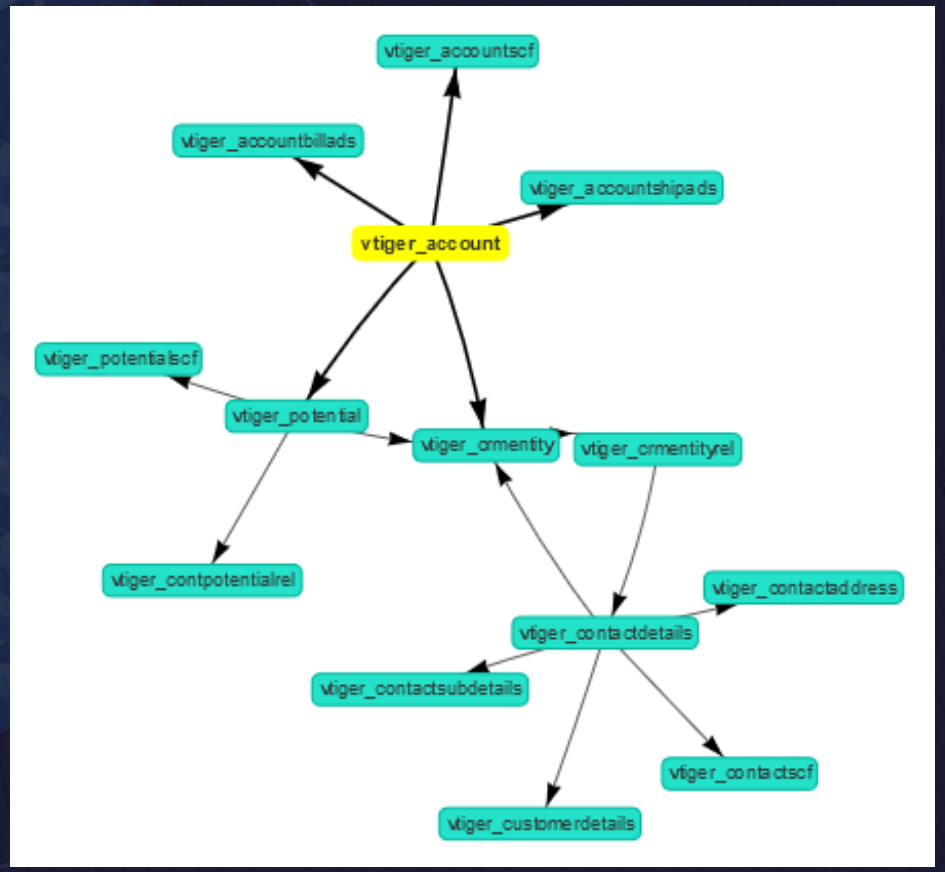
Example (2|4)


Contacts
192.168.0.179/vtigerdemoprod/index.php?module=Contacts&view=L
Search
Production
Type to search
CONTACTS > All
Add Contact
Import
Customize
LISTS
Search for List
MY LIST
Contacts Address
Todays Birthday
SHARED LIST
All
EXTENSIONS
Google
TAGS
More
1 to 60 of ?
First Name
Last Name
Title
Organization Name
Primary Email
Office Phone
Search
Brenna Houchens Quality Assurance and Testing Manager Handy Labs Brenna.Houchens@example.com 371-555-8465
Frederik Marobella Senior Manager Quality Assurance Cho Logistics Frederik.Marobella@example.com 950-555-9288
Latoyia Branco Sr. QA Analyst Sprat Books Latoyia.Branco@example.com 152-555-7906
Cory Neel Principal Systems Programmer Mcnally Services Cory.Neel@example.com 305-555-9891
Lari Montorzi z/OS DBA Haas Specialties Lari.Montorzi@example.com 047-555-4964
Harland Saban Sr. Database Engineer Barnard Technologies Harland.Saban@example.com 446-555-2667
Maura Hazen Tech Lead Barnard Technologies Maura.Hazen@example.com 845-555-6630
Nealy Kraemer Manager, Quality Assurance 2 Dowling Services Nealy.Kraemer@example.com 898-555-8626
Kayne Vietor z/OS DBA Dowling Services Kayne.Vietor@example.com 336-555-9626
Senior Vice

Example (3|4)

- 523 tables
- 97 tables connected directly or indirectly to the accounts table
- 14 of which are relevant for this example
- Central entity tables that assigns unique IDs to every object in the CRM
- Tables that store the highest ID of other tables (used like sequences)

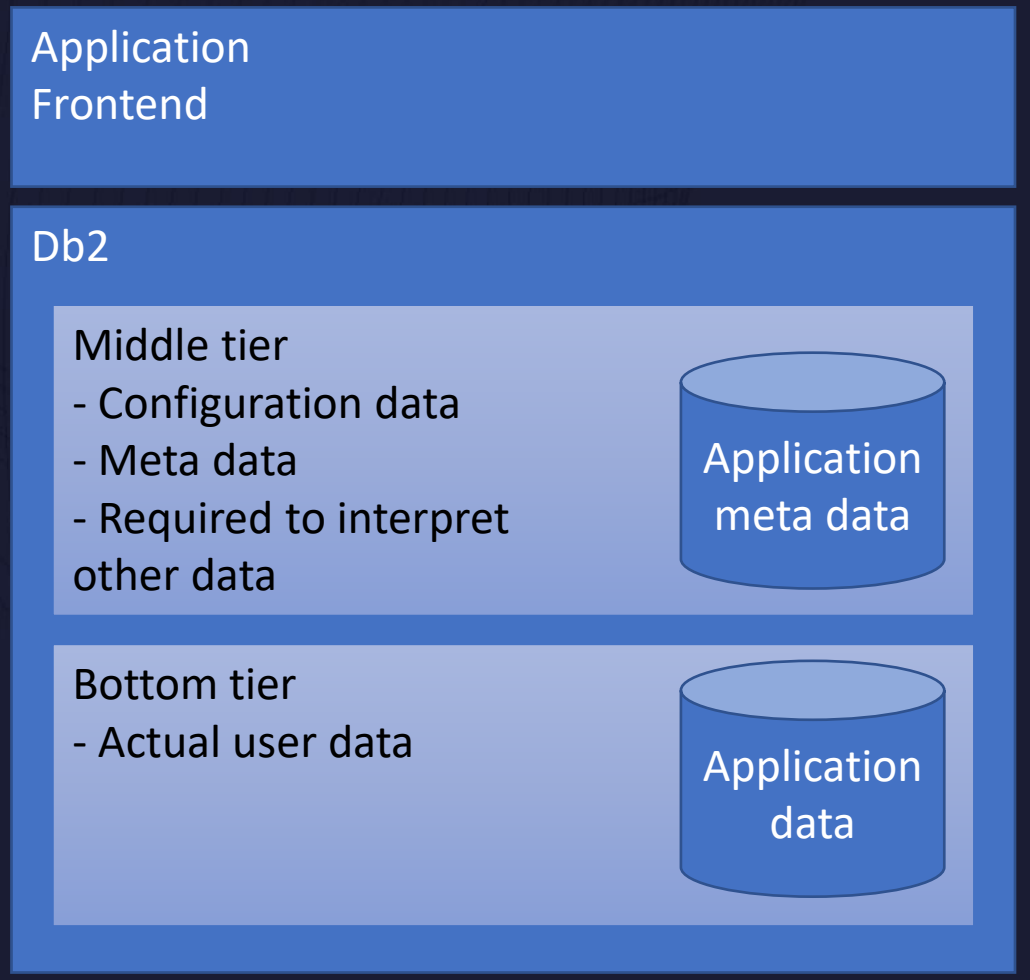
Example (4|4)



vtiger_account	Main accounts table
vtiger_accountscf	Accounts custom fields
vtiger_accountshipads	Accounts shipping addresses
vtiger_accountbillads	Accounts billing addresses
vtiger_crmentity	Central CRM table
vtiger_crmentityrel	Relationships between CRM objects
vtiger_contactdetails	Main contacts table
vtiger_contactsubdetails	Additional contact information
vtiger_contactscf	Contacts custom fields
vtiger_contactaddress	Contacts addresses
vtiger_customerdetails	Support contract durations
vtiger_potential	Main sales opportunities table
vtiger_potentialcf	Sales opportunities custom fields
vtiger_conpotentialrel	Relationships between sales, opportunities and contacts

Environment vs. data

- Data: Raw data in tables
- Environment: Meta information in tables - makes other data usable for application
  - Customized form fields
  - User settings
  - Information about active modules
- How much effort is required to create this “middle tier”?



Organizational aspects (1|2)

- People that need test data may not have the authorization to extract it from production, or the knowledge which of the 5000 tables are relevant
- A modeler translates requirements (“We need a customer with at least 5 active contracts”) into a queries and rule sets
- Specialized knowledge becomes usable for many testers
- This enables “exploratory testing”
- Use parameters to make a process into a blueprint where the exact specifications can be supplied at run time

Organizational aspects (2|2)

- Test data is requested by an automated process, e. g. Jenkins
  - Compile program
  - Create test bed
  - Run tests
  - Delete test bed
- Communication with other processes is key: REST API is de-facto standard to talk to other applications (potentially non-databases, e. g. for plausibility checking) that need to be aware of test data
- Keep a detailed log for auditing purposes

Where does the data come from?

- Taking test data directly from production is not a good idea:
  - SELECT statements to extract test cases are not typical production workload
  - Authorization problems
  - Production is a system that always changes – no two extracts are identical
- Ideally, test beds are extracted once and stored in a repository outside the database
- Can be restored into ANY system at ANY time

Golden Copies (1|5)

- For making test cases out of real production data
- Also for regression tests, acceptance tests
- Not a general test bed, but can be data source for other test environments
- Be careful about who has access to pre-production
- Can be central location where data masking is applied
  - Disadvantage: Large environment, you are potentially masking a lot of data that will never be used for tests
  - Advantage: Clear separation between unmasked and masked data

Golden Copies (2|5)

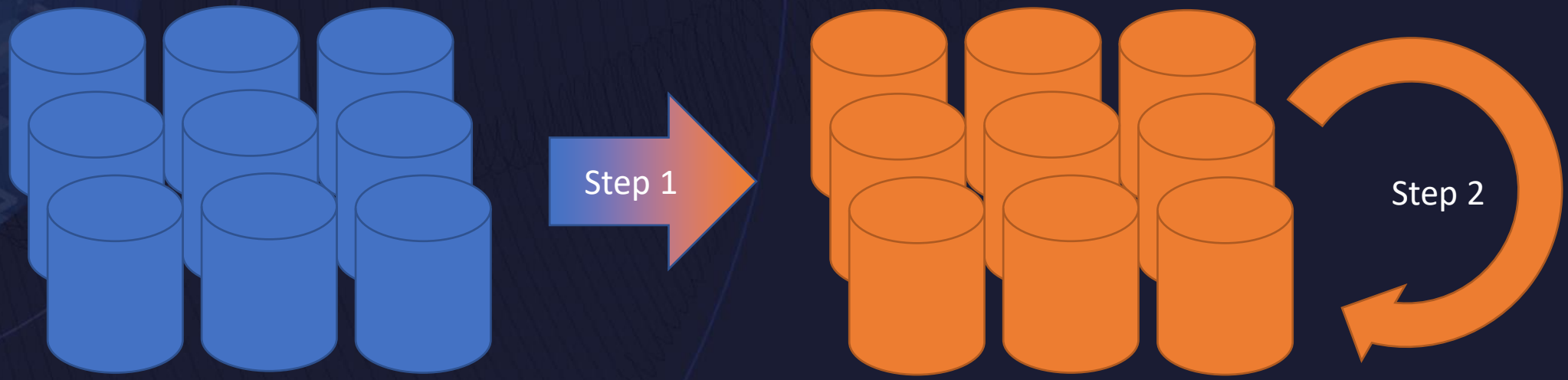
- Db2 for z/OS: Hardware-based clones on volume level
- Entire subsystem or data sharing group can be cloned without interrupting the source
- Making clone usable requires software
- Decoupling of production from everything else
- Full clones require very little maintenance
- DASD space may be an issue

Golden Copies (3|5)

- Initial setup
  - Add subsystem definitions for Db2 and IRLM in IEFSSNxx
  - Add DFSMS classes and adjust ACS routines
  - Configure VTAM and TCP/IP
  - Add RACF profiles
  - Assemble new ZPARMs (IRLMPRC, CATALOG, GRPNAME, ARCPFX*)

Golden Copies (4|5)

- Automated, repeated copies
  - Clone volumes using hardware assisted copy mechanisms
  - Rename and recatalog data sets, rename references in Db2 catalog, Db2 directory, BSDS, active logs



Golden Copies (5|5)

- Alternative: Partial clone on data set level
- Be more selective: Only copy those tables that are required for a certain application / business unit
- Flexible, can rename objects
- Can also be done without interrupting production, however log record processing outside of Db2 is required
  - Alternatively copy from existing image copies into new environment

Final thoughts (1|3)

- Ideally, you want both test cases taken from a real production system and hand-crafted test cases that cover edge cases
- In version control systems, multiple different branches can be built and tested in parallel
- Objects must not overlap
- Individual schemas for each branch
- Create objects as needed, drop after testing is done

Final thoughts (2|3)

- Meaningful test cases often involve multiple different applications
- Example:
  - Central address management
  - Insurance policy management (individual components for each type of insurance)
  - Claims settlement
  - Etc.
- There is a good chance you need to work with multiple different databases

Final thoughts (3|3)

- Creating small test cases goes beyond the capabilities of the DBA
- Team up!
  - Testers
  - Developers
  - Product managers
  - Database administrators
  - Test managers
  - System administrators
  - Data protection officers

Speaker: Kai Stroh
Company: UBS Hainer GmbH
Email Address: kai.stroh@ubs-hainer.com