



Large and complex databases are the norm in all areas of today's business world.

Businesses want rapid implementation of accurate and effective system enhancements and/or bug fixes.

Development and support teams need test databases that reflect the live production system.

Full size copies of the production database are expensive to create, store and maintain.

An ideal solution is a small size replica of the original database that maintains referential integrity – a subset database.

Manual population of a subset database is difficult and very time consuming.

SelExL automates the process of populating a subset database.

You determine the exact data you want for the population by simply specifying a list of the required items e.g. account numbers. Alternatively you can request a "random" data selection with the use of wildcards.

## Reducing Costs and Improving Application Quality

One of the keys in developing and maintaining a high quality application is rigorous and thorough testing. This requires the right test environment with the right test database. Often the best test databases are those that have genuine production data in them - nothing's better than the real world. The simple answer is to use a copy of the production database for testing.

However it's not practical for all testers and developers to share the same test database. In addition consideration must be given to the business's need for training, business acceptance testing and other requirements.

Clearly, creating multiple copies of a production database to overcome these problems is too expensive both in terms of storage space and man-hours to create and maintain the copies. Using SelExL to create small, referentially correct, subset databases alleviates the expense in both these areas. The table below provides an illustrative comparison between a small, subset test database, populated using SelExL, and a full size production copy of a 500 GB database:

	Subset Database	Full Size Copy
Disk space	1-2 GB	500 GB
Refresh time	< 30 minutes	Upto 12-24 hours <sup>(1)</sup>
Realistic refresh frequency	Daily	At most weekly
Staff effort	Approx. 30 minutes	Approx. 2-8 hours

<sup>(1)</sup> Actual time varies greatly depending on the hardware and technology involved.

In addition, the subset databases that SelExL populates can be tailored to the needs of the developer/trainer/tester, thereby making even more efficient use of time. For example:

### Bug fixing

Extract 'problem' sets of data (e.g. energy accounts that aren't billing correctly) into a small subset database. Analyse the problem, develop and test a fix against the subset database. Safely implement the fix into the production database. This is much more efficient and reliable than a programmer working on a test system that doesn't contain the specific data causing the problem.

### User training

Create appropriately sized training databases with the data targeted at specific training scenarios.

### System test

Generate a 'random' sample of data for system test by extracting a small (say 1%) amount of data from the production database e.g. all account numbers ending in 01.

## Populating Subset Test Databases using SelExL

The preliminary step in populating a subset database is to “tell” SelExL which tables you wish to extract data from and how these tables are related. This is accomplished by filling in the required details for three small configuration files. These “rules” help ensure the referential correctness of the subset database. This configuration work is generally carried out only once for each source database e.g. the billing database.

Each time a subset database is populated you need to decide what data is required. Normally this is simply a case of providing a list of identifiers in a file e.g. account numbers, customer ids, or whatever is appropriate for the database in question.

SelExL is now ready to run and can operate in one of two modes:

- File based – Data is first extracted to files (with the option to perform “on the fly” compression) ready for loading into the subset database at a later time. This helps minimise the time SelExL spends accessing the production system.
- Extract and Load – Data is extracted and loaded as a single process. This is the quick way to populate a subset database when only a relatively small amount of data is required.

There are times when a good size test database is needed for large-scale system test or performance and volume testing, 10% - 20% of production is often a good size for these purposes. However performing these large extracts from a production database is not normally feasible due to issues with elapsed time and identifying the required data sample. SelExL provides a number of features that make this feasible.

- Wildcard data extraction – You don’t need to provide a list of each item to extract e.g. each bank account number. Instead you can use wildcards to simplify the process e.g. for a 20% sample use %1, %2 to extract all accounts ending in 1 or 2.
- Multi-streaming – SelExL is set up to run all extraction and data loading as a multi-streamed process to improve performance. The number of streams is any value up to a maximum of 100, user defined at runtime.
- SQL design – All SelExL’s standard extract SQL involves at most two tables per statement. This simplifies the SQL and aids overall performance by eliminating the possibility of complex inefficient multi-table joins.
- Server installation – As well as installing SelExL on a client PC you can elect to install SelExL on the database server itself. This gives SelExL access to the full processing power of the server and thus performance isn’t restricted by the power of the client PC.

**For more information, visit our website at [www.christallize.com](http://www.christallize.com).**

### *System Requirements*

- Oracle client software
  - Python<sup>(1)</sup> 2.3 or above
  - cx\_Oracle<sup>(1)</sup> 4.0 or above
- <sup>(1)</sup> Free open source products

### *Supported Environments*

- Windows, UNIX and Linux platforms that support an Oracle client installation
- Oracle versions 8i, 9i, 10g.
- SelExL accesses any Oracle database on any platform to which it can make an Oracle Net connection.

### *30 Day Free Trial*

A trial version of SelExL is available for download from [www.christallize.com](http://www.christallize.com).

### *Support and Maintenance*

Christallize Ltd provides basic support for SelExL as standard. An optional support and maintenance package is also available.

### *Licensing and Prices*

SelExL uses site based licensing rather than machine or “per seat” based licensing. Our current price list is available from our website at [www.christallize.com](http://www.christallize.com)

SelExL™ is a trademark of Christallize Ltd.  
© 2004 Christallize Ltd.

2 Carvel Way | Littlehampton  
West Sussex | BN17 6RJ | UK

[www.christallize.com](http://www.christallize.com)