European Archival Records and Knowledge Preservation

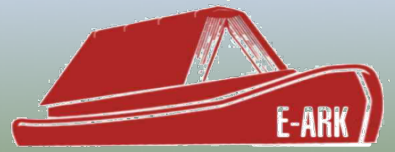
#earkproject www.eark-project.eu @EARKProject

Database Preservation Toolkit

A relational database conversion and normalization tool

Bruno Ferreira

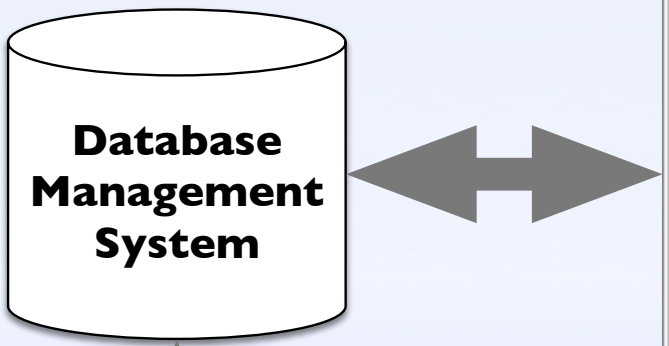
2016-10-04, Bern, Switzerland

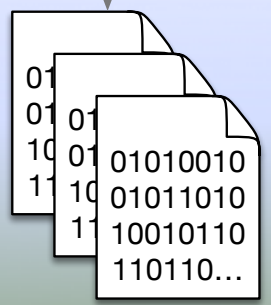


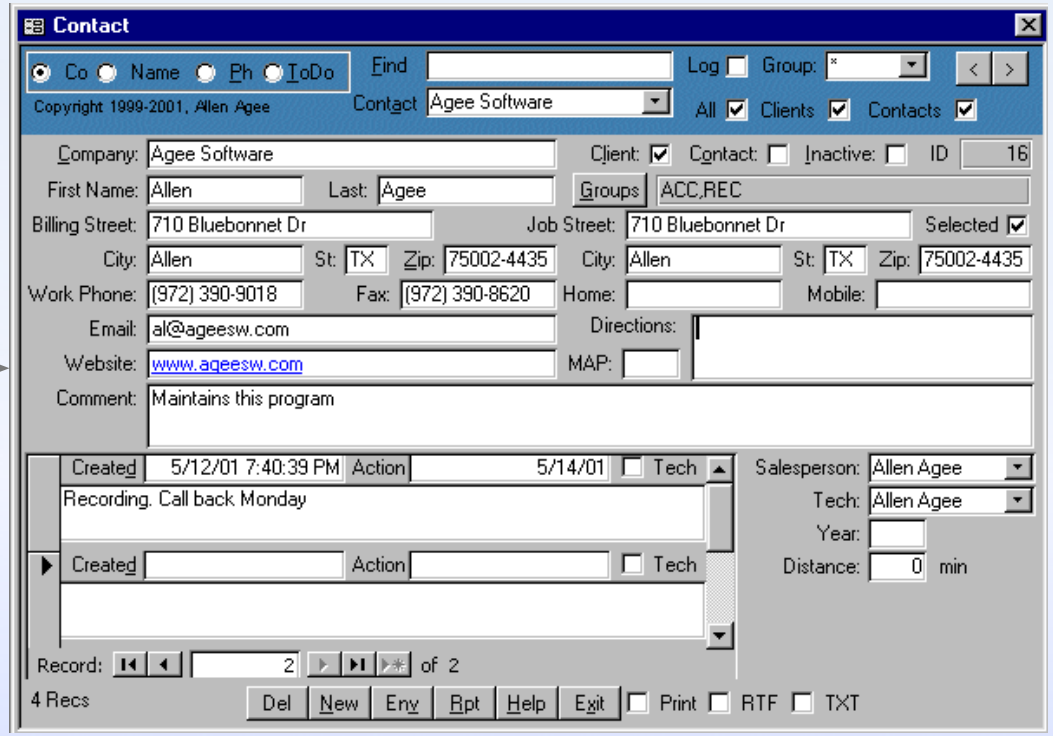
About the work

- KEEP SOLUTIONS
- E-ARK

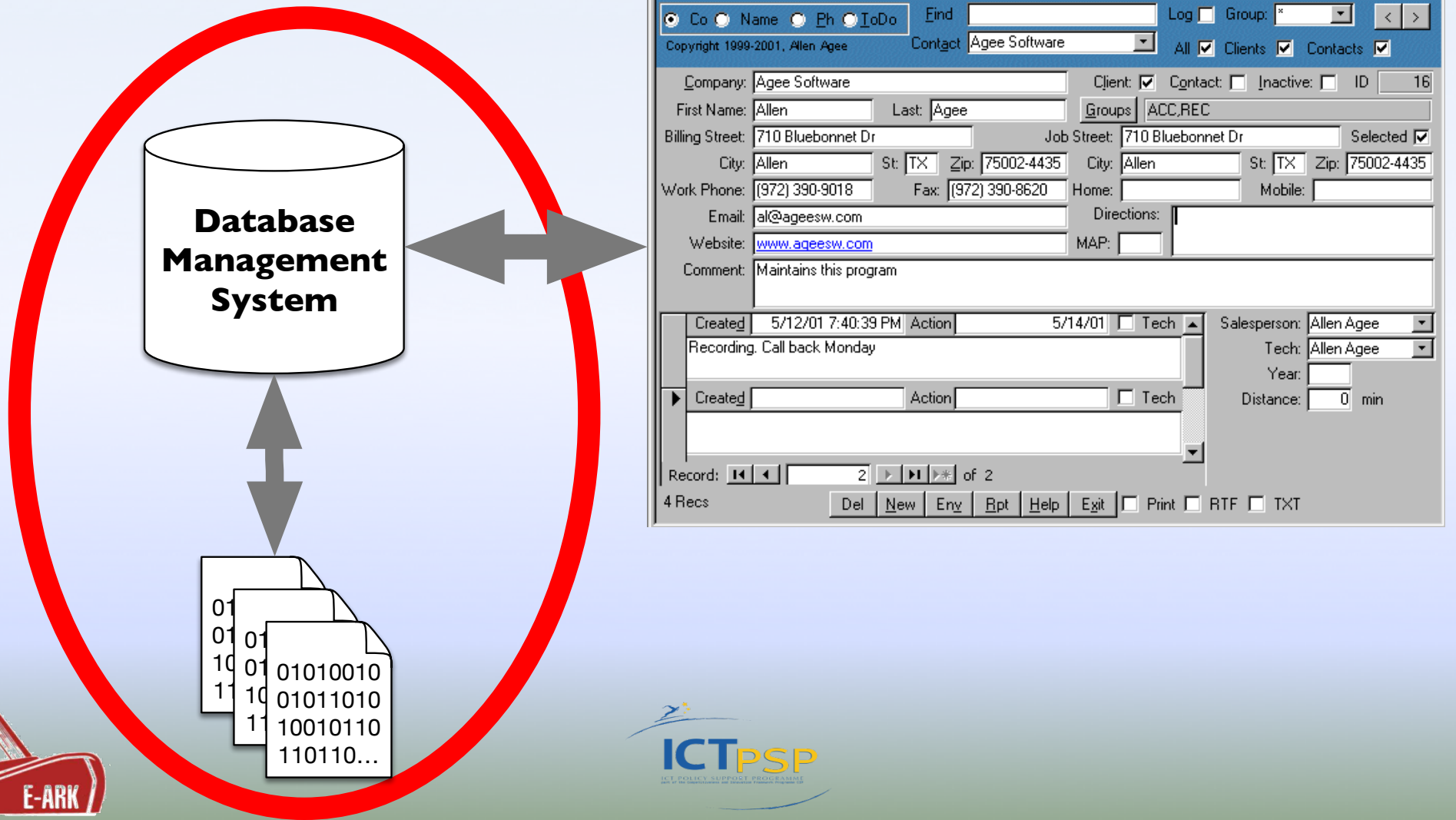
What are relational databases?





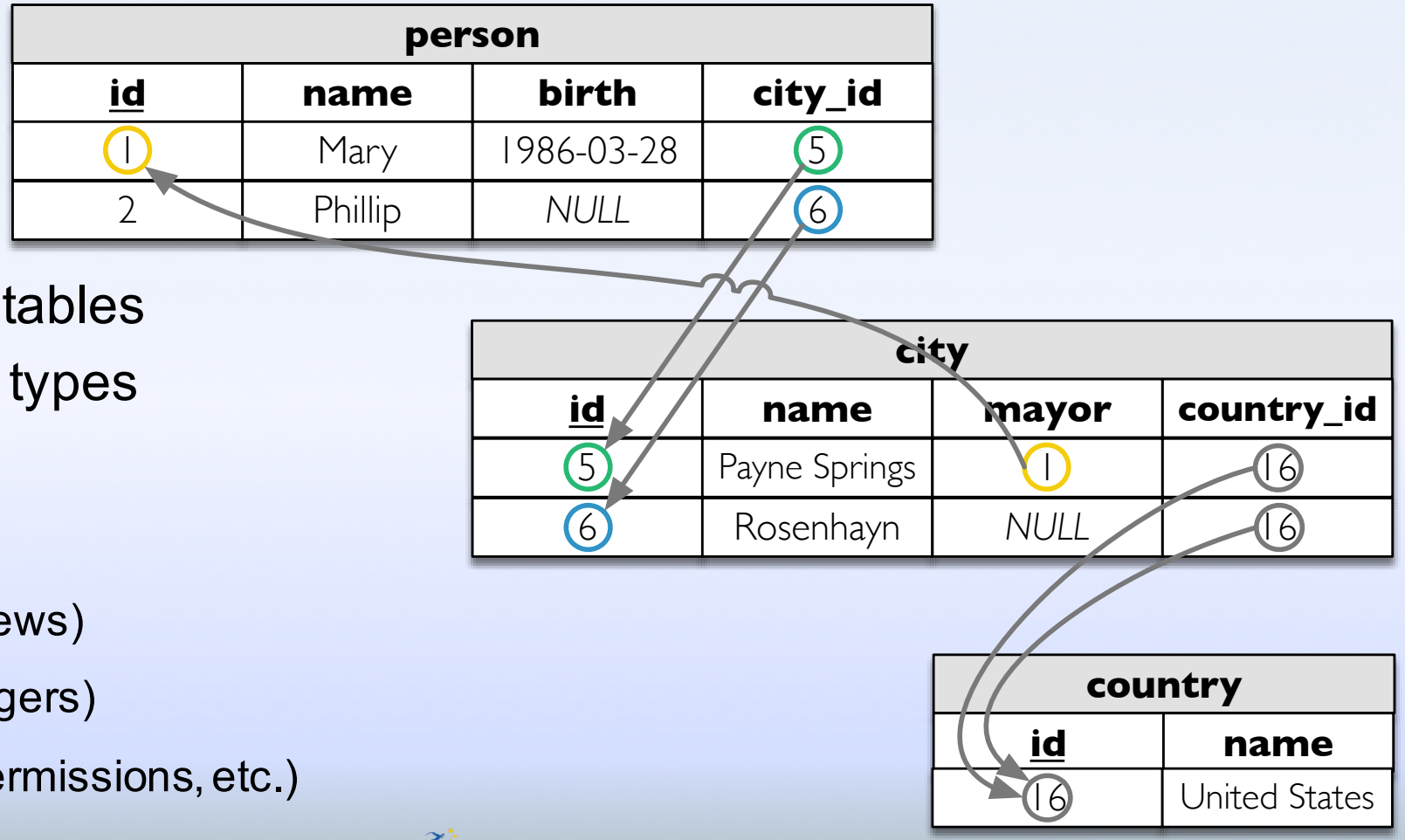


What are relational databases?



What are relational databases?

- Information in tables
- Columns data types
- Relations
- Constraints
- Projections (views)
- Behaviour (triggers)
- Other (users, permissions, etc.)



Where are databases used?

- Finances
- Social health
- Hospitals
- Banks
- Insurance records
- Science
- and many more…

The problem

- Every vendor has his data types and export formats
- It is rare that information exported from one vendor's system works on another
- Sometimes doesn't work on different versions of the same product
- Needs a vendor-agnostic format based on standards

SIARD 1

- Database preservation format
- Based on international standards
- Database data, structure and behaviour
- Swiss standard eCH-0165

SIARD 2

- Better support for binary and large objects
- Updated standards
- Support for user defined types and arrays
- Improved validation constraints

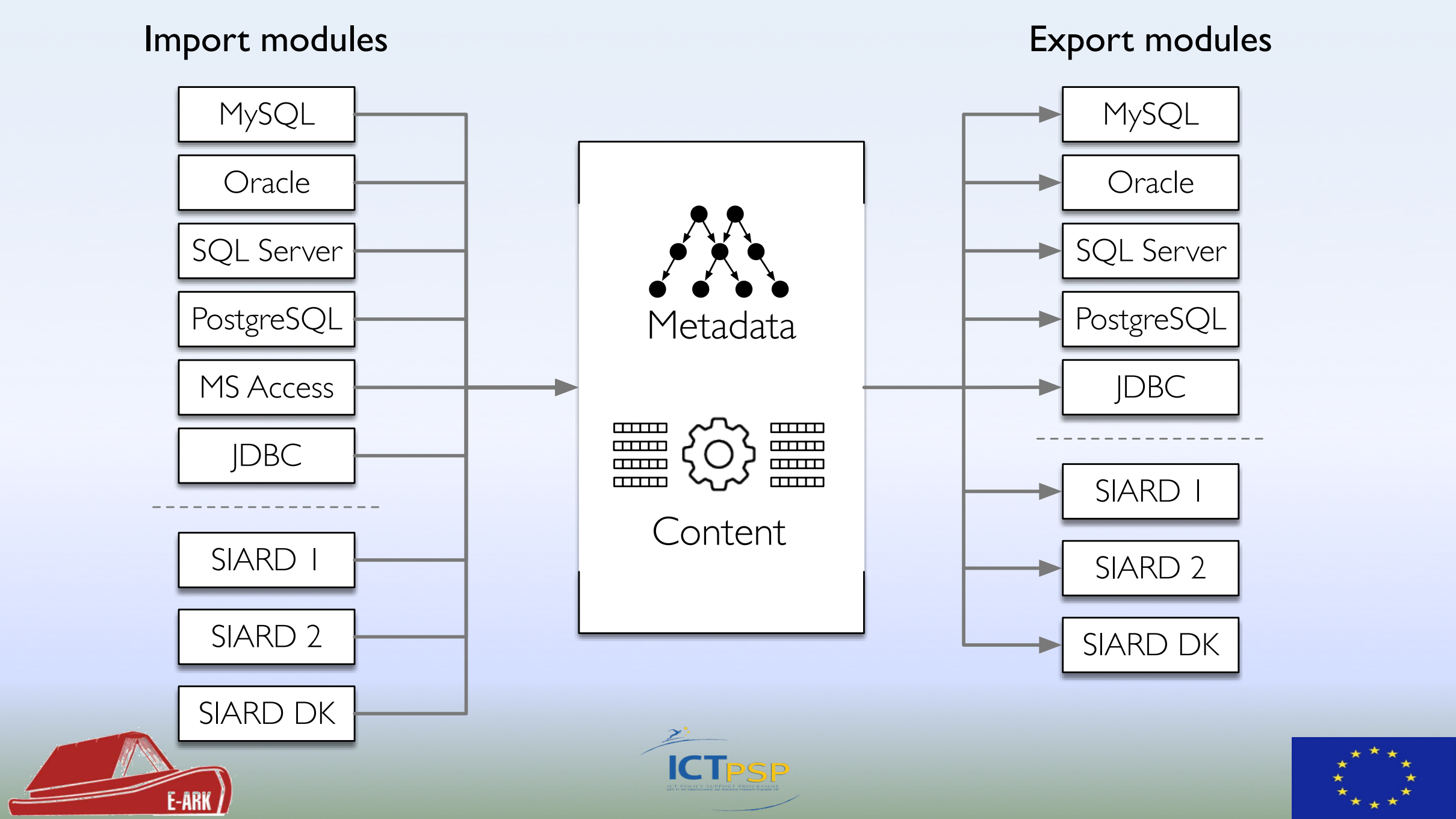
database preservation toolkit

View on GitHub | Download v2-beta5

Database Preservation

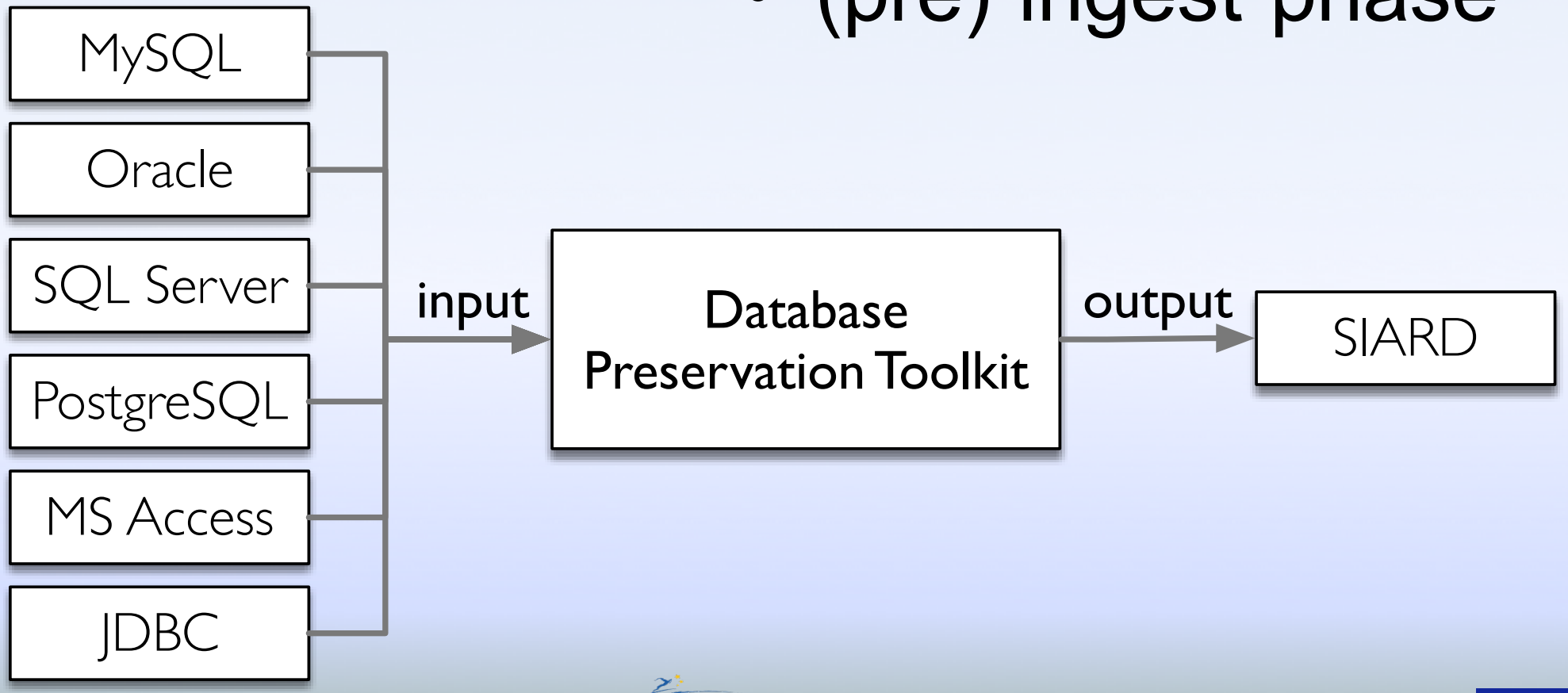
Relational databases are one of the most important technologies supporting today's information management activities. They are designed to store, organize and explore digital records that not only support but also document day-to-day business operations. Very often, these records are irreplaceable or prohibitively expensive to reacquire by other means rendering the preservation of databases a serious concern.

This page focus on workflows, tools and standards to allow information managers to extract, archive and preserve records of information currently managed by relational databases.


Import modules
MySQL
Oracle
SQL Server
PostgreSQL
MS Access
JDBC
SIARD 1
SIARD 2
SIARD DK
Metadata
Content
Export modules
MySQL
Oracle
SQL Server
PostgreSQL
JDBC
SIARD 1
SIARD 2
SIARD DK

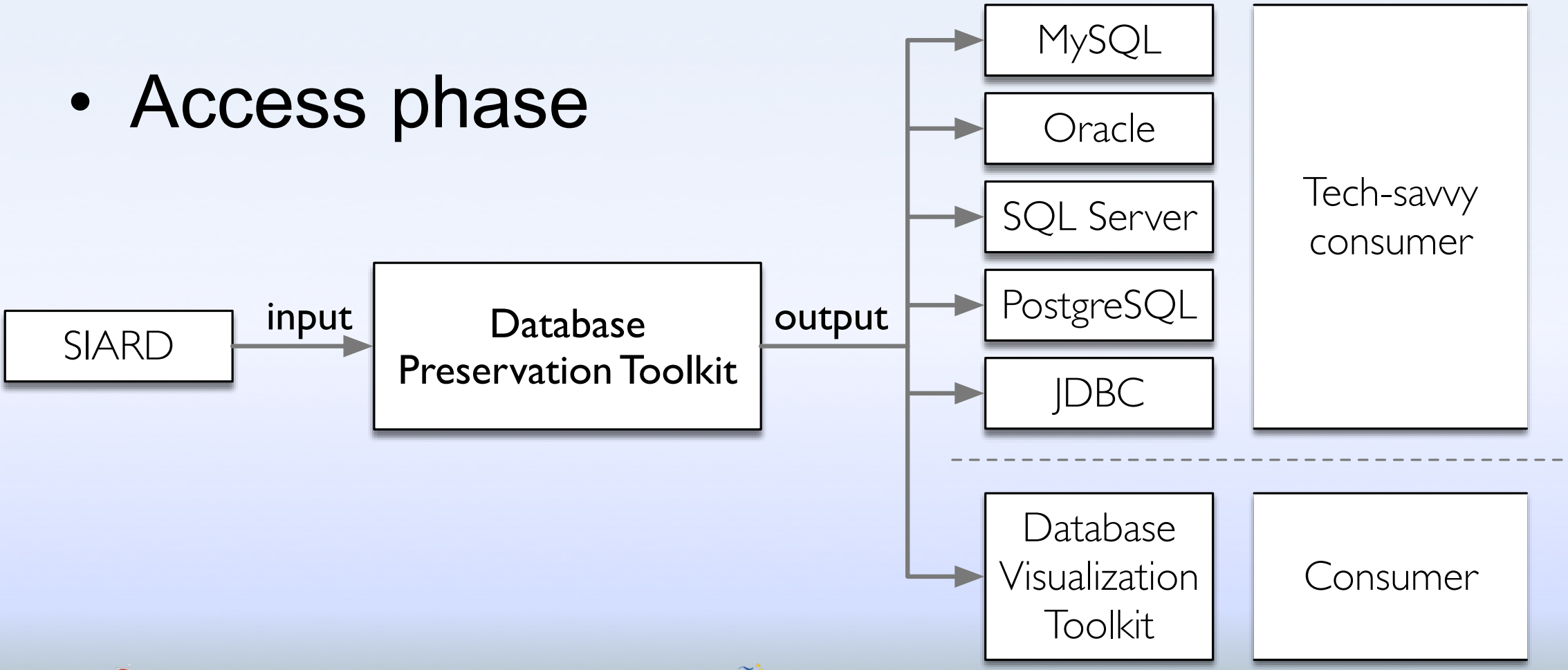
From DBMS to SIARD

- (pre) ingest phase



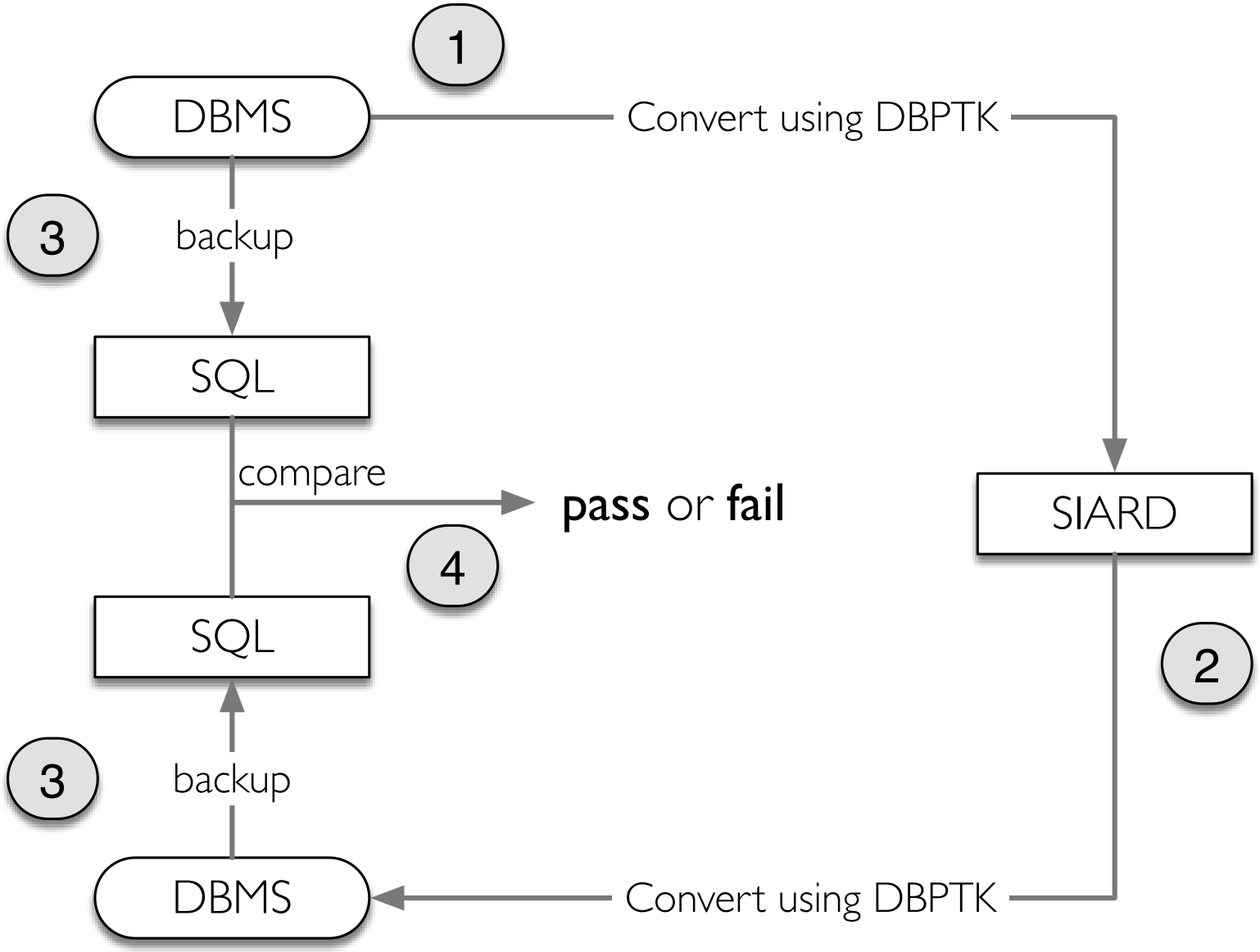
From SIARD to DBMS

- Access phase



Ensure no information is lost

- SIARD built-in validation
- Roundtrip tests


1
DBMS
Convert using DBPTK
3
backup
SQL
compare
pass or fail
4
SQL
SIARD
2
3
backup
DBMS
Convert using DBPTK

Prioritization and Flexibility

- Priority: 1) data, 2) structure, 3) behaviour
- Precision (dates, floating points, etc.)
- Types may differ
- Fallback, because constraints are not always checked in old databases

Database Visualization Toolkit

Relational Database Viewer for databases based on SIARD 2

View on GitHub | Download for Windows | Download for Mac OS X | Download for Linux

Database Visualization Toolkit

Lightweight web viewer for relational databases, specially if preserved in SIARD 2, that uses SOLR as a backend, and allows browsing, search, and export. It uses the Database Preservation Toolkit to process new relational databases that are in the SIARD2 format or on the original live DBMS.

A compreensive list of features, screenshots and other documentation will be produced soon and available here.

Setting up the Database Visualization Toolkit

The Database Visualization Toolkit is a web application. To run it you need to download the appropriate ZIP package file for your operating system at https://github.com/keeps/db-visualization-

Filter sidebar

Databases / sakila

- Database
  - Information
  - Users & Roles
  - Saved searches
  - Search all records
- sakila
  - Structure
  - Routines
  - Triggers
  - Check constraints
  - Views
  - Data
    - actor
    - address
    - category
    - city
    - country
    - customer
    - film
    - film_actor
    - film_category
    - film_text
    - inventory
    - language
    - payment

Database information

Database Name
sakila

Archival Date
2016-09-15T01:00:00.000+0100

Archivist
Bruno Ferreira

Archivist contact
email: bferreira@keep.pt

Client machine
young (fetched automatically)

Database product
MySQL 5.5.5-10.1.11-MariaDB-1~trusty

Data origin time span
Early 2005 to March 2006

Data owner
MySQL team

Description
The Sakila sample database was initially developed by Mike Hillyer, a former member of the MySQL AB documentation team, and is intended to provide a standard schema that can be used for examples in books, tutorials, articles, samples, and so forth. Sakila sample database also serves to highlight the latest features of MySQL such as Views, Stored Procedures, and Triggers. The Sakila sample database is designed to represent a DVD rental store.

Producer application

Filter sidebar

Databases / sakila

- Database
  - Information
  - Users & Roles
  - Saved searches
  - Search all records
- sakila
  - Structure
  - Routines
  - Triggers
  - Check constraints
  - Views
  - Data
    - actor
    - address
    - category
    - city
    - country
    - customer
    - film
    - film_actor
    - film_category
    - film_text
    - inventory
    - language
    - payment

Search all records

allen

sakila › actor

actor_id	first_name	last_name	last_update
118	CUBA	ALLEN	2006-02-15 04:34:33
145	KIM	ALLEN	2006-02-15 04:34:33
194	MERYL	ALLEN	2006-02-15 04:34:33

1-3 of 3

sakila › customer

customer_id	store_id	first_name	last_name	email	address_id	active	creat
27	2	SHIRLEY	ALLEN	SHIRLEY.ALLEN@sak	31	true	2006-
412	2	ALLEN	BUTTERFIELD	ALLEN.BUTTERFIELD	417	true	2006-

1-2 of 2

Database
- Information
- Users & Roles
- Saved searches
- Search all records

sakila
- Structure
- Routines
- Triggers
- Check constraints
- Views
- Data
  - actor
  - address
  - category
  - city
  - country
  - customer
  - film
  - film_actor
  - film_category
  - film_text
  - inventory
  - language
  - payment

sakila › actor

Description

The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table.

Search...

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINESS	2006-02-15 04:34:33
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
11	ZERO	CAGE	2006-02-15 04:34:33
12	KARL	BERRY	2006-02-15 04:34:33
13	UMA	WOOD	2006-02-15 04:34:33
14	VIVIEN	BERGEN	2006-02-15 04:34:33
15	CUBA	OLIVIER	2006-02-15 04:34:33

Filter sidebar

Databases / sakila / sakila / actor

- Database
  - Information
  - Users & Roles
  - Saved searches
  - Search all records
- sakila
  - Structure
  - Routines
  - Triggers
  - Check constraints
  - Views
  - Data
    - actor
    - address
    - category
    - city
    - country
    - customer
    - film
    - film_actor
    - film_category
    - film_text
    - inventory
    - language
    - payment

sakila › actor

Description

The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table.

Search...

☑ actor_id	42	108		
☑ first_name	christian			
☑ last_name				
☑ last_update	2008-04-01	16:30:00	2016-06-20	18:10:30

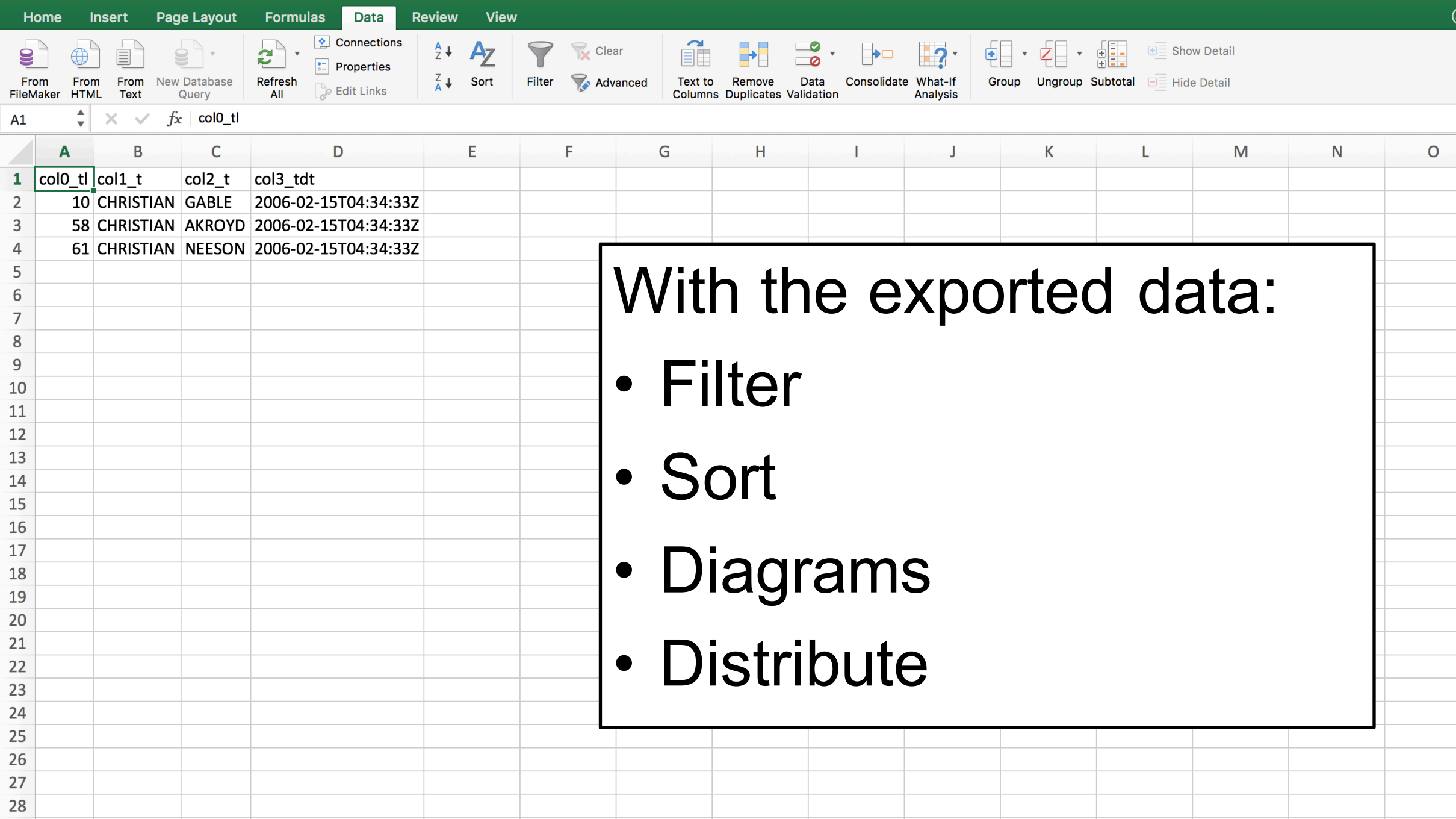
ADD SEARCH FIELD

SAVE SEARCH SEARCH

actor_id	first_name	last_name	last_update
10	CHRISTIAN	GABLE	2006-02-15 04:34:33
58	CHRISTIAN	AKROYD	2006-02-15 04:34:33
61	CHRISTIAN	NEESON	2006-02-15 04:34:33

1-3 of 3

Export visible Export all


Home
Insert
Page Layout
Formulas
Data
Review
View
From FileMaker
From HTML
From Text
New Database Query
Refresh All
Connections
Properties
Edit Links
Sort
Filter
Clear
Advanced
Text to Columns
Remove Duplicates
Data Validation
Consolidate
What-If Analysis
Group
Ungroup
Subtotal
Show Detail
Hide Detail
A1
col0_tl
col0_tl col1_t col2_t col3_tdt
10 CHRISTIAN GABLE 2006-02-15T04:34:33Z
58 CHRISTIAN AKROYD 2006-02-15T04:34:33Z
61 CHRISTIAN NEESON 2006-02-15T04:34:33Z
With the exported data:
• Filter
• Sort
• Diagrams
• Distribute

Structure

Schema name

sakila

Schema description

Schema enclosing the whole Sakila database.

sakila › actor

Description: The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table.

	column name	Type name	Original type name	Nullable	Description
🔑	actor_id	SMALLINT	SMALLINT UNSIGNED	No	A surrogate primary key used to uniquely id
	first_name	CHARACTER VARYING(45)	VARCHAR	No	The actor's first name.
	last_name	CHARACTER VARYING(45)	VARCHAR	No	The actor's last name.
	last_update	TIMESTAMP	TIMESTAMP	No	The time that the row was created or most

sakila › address

Description: The address table contains address information for customers, staff, and stores. The address table primary key appears as a foreign key in the customer, staff, and store tables.

	column name	Type name	Original type name	Nullable	Description
🔑	address_id	SMALLINT	SMALLINT UNSIGNED	No	A surrogate primary key used to uniquely id
	address	CHARACTER VARYING(50)	VARCHAR	No	The first line of an address.
	address2	CHARACTER VARYING(50)	VARCHAR	Yes	An optional second line of an address.

Views

Schema name

sakila

Schema description

Schema enclosing the whole Sakila database.

sakila › actor_info

Description

The actor_info view provides a list of all actors, including the films in which they have performed, broken down by category. The actor_info view incorporates data from the film, actor, category, film_actor, and film_category tables.

Original query

```
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`%` SQL SECURITY INVOKER VIEW `actor_info` AS select `a`.`actor_id` AS `actor_id`,`a`.`first_name` AS `first_name`,`a`.`last_name` AS `last_name`,group_concat(distinct concat(`c`.`name`,': ',(select group_concat(`f`.`title` order by `f`.`title` ASC separator ', ') from ((`film` `f` join `film_category` `fc` on((`f`.`film_id` = `fc`.`film_id`))) join `film_actor` `fa` on((`f`.`film_id` = `fa`.`film_id`))) where ((`fc`.`category_id` = `c`.`category_id`) and (`fa`.`actor_id` = `a`.`actor_id`)))) order by `c`.`name` ASC separator '; ') AS `film_info` from (((`actor` `a` left join `film_actor` `fa` on((`a`.`actor_id` = `fa`.`actor_id`))) left join `film_category` `fc` on((`fa`.`film_id` = `fc`.`film_id`))) left join `category` `c` on((`fc`.`category_id` = `c`.`category_id`))) group by `a`.`actor_id`,`a`.`first_name`,`a`.`last_name`
```

column name	Type name	Original type name	Nullable	Description
actor_id	SMALLINT	SMALLINT UNSIGNED	No	A surrogate primary key used to uniquely identify
first_name	CHARACTER VARYING(45)	VARCHAR	No	The actor's first name.
last_name	CHARACTER VARYING(45)	VARCHAR	No	The actor's last name.
film_info	CHARACTER LARGE OBJECT	TEXT	Yes	

Advanced use-case

Convert to DBMS and:

- Use SQL to query the database
- Hide sensitive data and convert to SIARD
- Use data analysis tools

Innovative options

- Easier to create modules in the future

Pilots

- 4 official pilots (other unofficial):
  - Denmark, Hungary, Estonia and Slovenia
- Testing in a real-world scenario

Workshop and tutorial

- Workshop 8 will provide a hands-on demonstration
- Tutorial 16 will provide insight on how institutions are using these tools
- Thursday morning and afternoon

Questions?

Make sure to check:

www.eark-project.com

https://github.com/eark-project

www.database-preservation.com

visualization.database-preservation.com

www.keep.pt

Bruno Ferreira
bferreira@keep.pt