



Preserving the future

Database preservation

DPC training course

Practical session (advanced)

Resolution



Activities on DBPTK Desktop



Create

This option allows you to create a SIARD file from a supported DBMS.

CREATE 

Open

This option allows you to open a SIARD file.

OPEN 

Manage

This option allows you to open, edit, validate, migrate, or visualize the information about SIARD file previously ingested.

MANAGE 

Click here to start the process of create a SIARD file

DBMS

- JDBC
- Microsoft Access
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- Progress Openedge
- Sybase

General SSH Tunnel

| | |
|--------------------|---|
| Hostname * | dpc.database-preservation.com |
| | The name of the database server host (e.g. localhost) |
| Port number | 3306 |
| | The server port number |
| Username * | mgulmaraes |
| | The name of the user to use in connection |
| Password * | ***** |
| | The password of the user to use in connection |
| Database * | sakila |
| | The name of the database to connect |

 Disable Encryption

Use to turn off encryption in the connection

TEST CONNECTION ↕

1. Select the DBMS on the left sidebar panel and fill up the connection form

2. Test the connection to ensure that you have the right information

3. Click **Next** to continue the process

CANCEL ✖

< BACK

NEXT >

Database

sakila

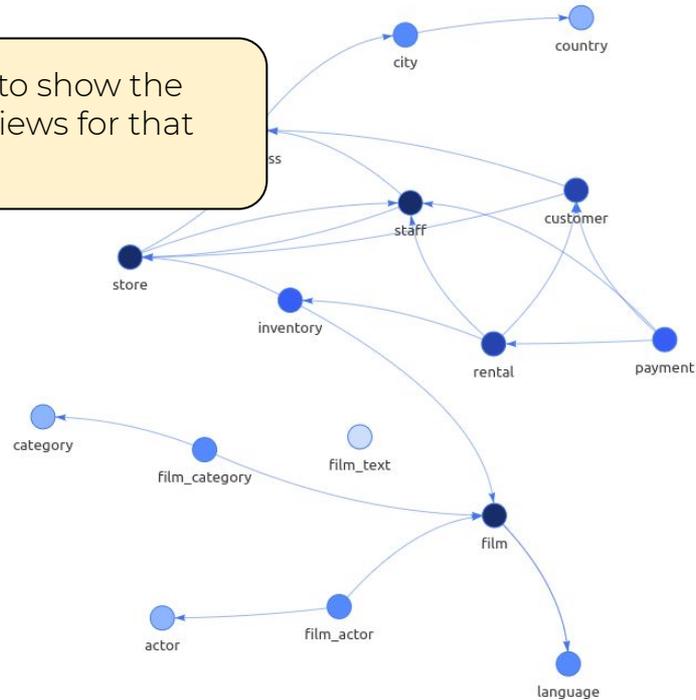
Tables

- actor
- address
- category
- city
- country
- customer
- film
- film_actor
- film_category
- film_text
- inventory
- language
- payment
- rental
- staff
- store

Views

- actor_info
- customer_list
- film_list
- nicer_but_slower_film_list
- sales_by_film_category
- sales_by_store
- staff_list

Click **sakila** to show the tables and views for that schema



Database

sakila

Tables

- actor
 - address
 - category
 - city
 - country
 - customer
 - film
 - film_actor
 - film_category
 - film_text
 - inventory
 - language
 - payment
 - rental
 - staff
 - store
- Views
- actor_info
 - customer_list
 - film_list
 - nicer_but_slower_film_list
 - sales_by_film_category
 - sales_by_store
 - staff_list

| Select | Table Name | Description |
|-------------------------------------|---------------|---|
| <input checked="" type="checkbox"/> | actor | The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table. |
| <input type="checkbox"/> | address | 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. |
| <input checked="" type="checkbox"/> | category | The category table lists the categories that can be assigned to a film. The category table is joined to the film table by means of the film_category table. |
| <input type="checkbox"/> | city | The city table contains a list of cities. The city table is referred to by a foreign key in the address table and refers to the country table using a foreign key. |
| <input type="checkbox"/> | country | The country table contains a list of countries. The country table is referred to by a foreign key in the city table. |
| <input type="checkbox"/> | customer | The customer table contains a list of all customers. The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign keys. |
| <input checked="" type="checkbox"/> | film | The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the language table and is referred to by the film_category, film_actor, and inventory tables. |
| <input checked="" type="checkbox"/> | film_actor | The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film. The film_actor table refers to the film and actor tables using foreign keys. |
| <input checked="" type="checkbox"/> | film_category | The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film_category table listing the category and film. The film_category table refers to the film and category tables using foreign keys. |
| <input type="checkbox"/> | film_text | The film_text table contains the film_id, title and description columns of the film table, with the contents of the table kept in synchrony with the film table by means of triggers on the film table's INSERT, UPDATE and DELETE operations. The contents of the film_text table should never be modified directly. All changes should be made to the film table instead. |
| <input type="checkbox"/> | inventory | The inventory table contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by the rental table. |
| <input checked="" type="checkbox"/> | language | The language table lists the possible languages that films can have for their language and original language. The language table is referred to by the film table. |
| <input type="checkbox"/> | payment | The payment table lists each payment made by a customer, with information such as the amount and the rental being paid. The payment table refers to the customer, rental, and staff tables. |

Select the tables: actor, category, film, film_actor, film_category and language

SELECT ALL

CANCEL

< BACK

NEXT >

Filter sidebar

Database

sakila

Tables

- actor
- address
- category
- city
- country
- customer
- film
- film_actor
- film_category
- film_text
- inventory
- language
- payment
- rental
- staff
- store

Views

- actor_info
- customer_list
- film_list
- nicer_but_slower_film_list
- sales_by_film_category
- sales_by_store
- staff_list

Tables Views

| Select | Materialize | View Name | Description |
|-------------------------------------|-------------------------------------|----------------------------|-------------|
| <input type="checkbox"/> | <input type="checkbox"/> | actor_info | |
| <input type="checkbox"/> | <input type="checkbox"/> | customer_list | |
| <input checked="" type="checkbox"/> | <input type="checkbox"/> | film_list | |
| <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | nicer_but_slower_film_list | |
| <input type="checkbox"/> | <input type="checkbox"/> | sales_by_film_category | |
| <input type="checkbox"/> | <input type="checkbox"/> | sales_by_store | |
| <input type="checkbox"/> | <input type="checkbox"/> | staff_list | |

Select the views: film_list and nicer_but_slower_film_list

Materialize the nicer_but_slower_film_list

SELECT ALL

CANCEL

< BACK

NEXT >

Filter sidebar

- Database
- sakila
 - Tables
 - actor
 - address
 - category
 - city
 - country
 - customer
 - film
 - film_actor
 - film_category
 - film_text
 - inventory
 - language
 - payment
 - rental
 - staff
 - store
 - Views
 - actor_info
 - customer_list
 - film_list
 - nicer_but_slower_film_list
 - sales_by_film_category
 - sales_by_store
 - staff_list

| Select | Column Name | Original Type | Description | Column Filters | Options | Merkle |
|-------------------------------------|-------------|-------------------|--|----------------|-----------|-------------------------------------|
| <input checked="" type="checkbox"/> | actor_id | SMALLINT UNSIGNED | A surrogate primary key used to uniquely identify each actor in the table. | | CONFIGURE | <input checked="" type="checkbox"/> |
| <input checked="" type="checkbox"/> | first_name | VARCHAR | The actor's first name. | | CONFIGURE | <input checked="" type="checkbox"/> |
| <input checked="" type="checkbox"/> | last_name | VARCHAR | The actor's last name. | | CONFIGURE | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> | last_update | TIMESTAMP | The time that the row was created or most recently updated. | | CONFIGURE | <input checked="" type="checkbox"/> |

1. Remove the last_update column from each previous selected table

2. Click **Next** to continue the process

SELECT NONE □

CANCEL ✖

< BACK

NEXT >

Custom Views

Custom Views

Custom views option allows you to aggregate data with SQL queries.

Schema Name *

sakila

Schema where the custom view should be integrated

Custom View Name *

address

Name of the custom view

Description

Addresses

Description of the custom view

Query *

```
select address.address, address.district, city.city, country.country from sakila.address address, sakila.city city, sakila.country country where address.city_id = city.city_id and city.country_id = country.country_id
```

SQL query to obtain the data to populate the custom view

SAVE

TEST

2. **Save** the query

1. Fill up the query text area and **test** the query

CANCEL

< BACK

SKIP >

Custom Views

address



Custom Views

Custom views option allows you to aggregate data with SQL queries.

Schema Name *

sakila

Schema where the custom view should be integrated

Custom View Name *

Name of the custom view

Description

Description of the custom view

Query *

SQL query to obtain the data to populate the custom view

SAVE

TEST

Click **Next** to continue the process

CANCEL

< BACK

NEXT >

Merkle Tree Filter

This filter module is used to generate a hash which represents the content of large data structures. This process aims to fulfill the requirement of completeness and correctness of the archival process, i.e. to ensure that no message is lost (not archived or not correctly archived).

Apply Merkle Tree Filter

Destination Folder * [BROWSE](#)

Path to save the merkle tree

Digest

SHA-256

Define the type of integrity information. Supported types are: MD5, SHA-1 and SHA-256.

Explain

Flag to show the hashes of the whole tree (tables, rows, cells). Mainly for debug purpose.

Font Case

Lower case

Define the type of font case for the message digest. Supported font case are: upper case and lower case.

Click **Skip** to continue the process

CANCEL

< BACK

SKIP >

SIARD Export Options

Resumes the different options available to create a SIARD file.

SIARD Version

Destination Folder * /home/mguimaraes/Desktop/practical_session_day_2_resolution.siard
Path to SIARD file

Compress
Compress the SIARD file with deflate method

Format XML
Write human-readable XML

Save LOBs outside SIARD file

External LOBs per Folder Files
The maximum number of files present in an external LOB folder. Default: 1000 files

External LOBs Folder Size Megabytes
Divide LOBs across multiple external folders with (approximately) the specified maximum size (in Megabytes). Default: do not divide

Digest
Define the type of integrity information. Supported types are: MD5, SHA-1 and SHA-256.

Font Case
Define the type of font case for the message digest. Supported font case are: upper case and lower case.

1. Select the destination folder

2. Choose compress checkbox
(this will reduce the size of the
SIARD file)

3. Choose to save LOBs outside
the SIARD file

4. Hit **Next** to continue the
process

CANCEL

< BACK

NEXT >

Metadata Export Options

Allows to specify the different available metadata values for a SIARD file.

| | |
|-----------------------------|--|
| Description | <input type="text" value="DPC Workshop practical session activities resolution"/> <small>Description of database meaning and content as a whole</small> |
| Archiver | <input type="text" value="mgulmaraes"/> <small>Name of the person who carried out the archiving of the database</small> |
| Archiver Contact | <input type="text" value="mgulmaraes@keep.pt"/> <small>Contact details (telephone, email) of the person who carried out the archiving of the database</small> |
| Data Owner | <input type="text"/> <small>Owner of the data in the database. The person or institution that, at the time of archiving, has the right to grant usage rights for the data and is responsible for compliance with legal obligations such as data protection guidelines</small> |
| Data Origin Timespan | <input type="text"/> <small>Origination period of the data in the database (approximate indication in text form)</small> |
| Client Machine | <input type="text"/> <small>DNS name of the (client) computer on which the archiving was carried out</small> |

1. Fill up metadata information about the SIARD file

2. Click **Create** to start the migration process

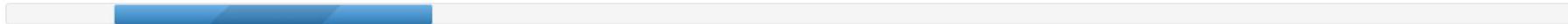
CANCEL

< BACK

CREATE >

SIARD File Creation

This process may take a while. Please wait for it to finish



Retrieving database structure

Wait for the process to finish, this may take a while, depending on the machine specs and total size of the database

CANCEL 

< BACK

CREATE >

SIARD File Creation

This process may take a while. Please wait for it to finish

Retrieving database structure

Tables: 13 of 16 (81%)

Current Table: rental

Numer of rows processed: 32823

Create SIARD

SIARD file successfully created, do you want to import?

CANCEL

IMPORT ▶

Click on **Cancel**

The background consists of numerous vertical lines of varying thickness and color, ranging from dark grey to bright red. The lines are slightly blurred and have a 3D effect, appearing to recede into the distance. The overall color palette is monochromatic, dominated by red and black tones.

Activities on DBPTK Enterprise

Do an advanced search and save it

Welcome to Database Preservation Toolkit Enterprise!

A web application to browse and search the content of multiple large archived databases.

The Database Preservation Toolkit Enterprise is a Lightweight web viewer for relational databases, specially if preserved in SIARD 2, that uses SOLR as a backend, and allows browsing, search, and export.

Click on **Login**



Enterprise Architecture

For large institutions with many databases and users

A web application that can be horizontally scaled to support many large-sized databases, accessed by hundreds of users simultaneously. Need help? [Get support](#).



Browse and search

Allow users to access database content on the Web

Allow them to search on a prepared, de-normalized and anonymized database content



SIARD validation

Validate archived database

Validate SIARD against specification plus many additional checks for a thorough validation.



Advanced data transformation

Transform content to answer useful questions

De-normalization and table/column hiding, to simplify browsing/search and allow anonymization of content.



Manage multiple databases

Single system, multiple databases

Search through the databases, manage their status, enrich their metadata, validate them, make them ready for your end users.



Export features

Export data into tabular data

Allow users to save search results in Microsoft Excel or other spreadsheet software format for easy analytics and diagrams.



Edit SIARD metadata

Enrich archived database with descriptions

Add descriptions to database, tables and columns to better understand its contents.



Single sign-on

Support for multiple protocols

LDAP, Active Directory, Database, SAML, ADFS, OAuth2, OpenID, Google, Facebook, Twitter, FIDO U2F, YubiKey, Google Authenticator, Authy, etc. Supports internal authorization definition or configurable external authorization.

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Databases

Databases available on the application

Search...



Choose the database

| Database name | Database name | Data Owner | Archival date | Producer application | Size | Version | Valid | Status | Actions |
|-----------------------|---|-----------------|---------------|-----------------------|----------|---------|-------|--------|---------|
| ahayesbrady | Database Test | Ali Hayes Brady | 2020-07-29 | MySQL 5.6.49 | | | Valid | Ready | OPEN |
| User21 | Sakila | user21 | 2020-07-29 | MySQL 5.6.49 | 788 KB | 2.1 | Valid | Ready | OPEN |
| gemmawestbrook_sample | gemmawestbrook_sample | Gemma Westbrook | 2020-07-29 | MySQL 5.6.49 | 790.6 KB | 2.1 | Valid | Ready | OPEN |
| User26 | unspecified | unspecified | 2020-07-29 | MySQL 5.6.49 | 6.4 MB | 2.1 | Valid | Ready | OPEN |
| evewright | evewright | unspecified | 2020-07-29 | MySQL 5.6.49 | 6.4 MB | 2.1 | Valid | Ready | OPEN |
| David | unspecified | unspecified | 2020-07-29 | MySQL 5.6.49 | 6.4 MB | 2.1 | Valid | Ready | OPEN |
| marty | sakila database for DPC workshop | KEEP SOLUTIONS | 2020-07-29 | MySQL 5.6.49 | 9.4 MB | 2.1 | Valid | Ready | OPEN |
| lizkata | A test MySQL database to save to the SIARD format | KEEP | 2020-07-29 | MySQL 5.6.49 | 789.5 KB | 2.1 | Valid | Ready | OPEN |
| JordanSaunders | Rental | Jordan Saunders | 2020-07-29 | MySQL 5.6.49 | 790.6 KB | 2.1 | Valid | Ready | OPEN |
| BWard Sakila | DVD Rental Database | Bradley Ward | 2020-07-29 | MySQL 5.6.49 | 790.5 KB | 2.1 | Valid | Ready | OPEN |
| Ivan | Ivan Test | unspecified | 2020-07-29 | MySQL 5.6.49 | 6.4 MB | 2.1 | Valid | Ready | OPEN |
| admin27 | Test DPC | Tanguy | 2020-07-29 | MySQL 5.6.49 | 6.4 MB | 2.1 | Valid | Ready | OPEN |
| Warwick Pease | Sakila | Keep | 2020-07-29 | MySQL 5.6.49 | 790.7 KB | 2.1 | Valid | Ready | OPEN |
| North Wind Traders | North Wind Traders | Microsoft | 2020-07-30 | Microsoft Access 2010 | 334.2 KB | 2.1 | Valid | Ready | OPEN |
| mgumaraes | DPC Workshop demonstration | KEEP SOLUTIONS | 2020-07-29 | MySQL 5.6.49 | 789.7 KB | 2.1 | Valid | Ready | OPEN |

1-15 of 30

Show More

LOAD SIARD FILE

DOWNLOAD DBPTK DESKTOP

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#)

[DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#)

[CONFIGURATION](#)

[DELETE](#)

[BACK](#)

[DELETE](#)

Click **Browse**

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

- Information
- Search all records
- Saved searches
- Tables
 - actor
 - address**
 - category
 - city
 - country
- inventory
- language
- payment
- rental
- staff
- store
- actor_info
- customer_list
- film_list
- nicer_but_slower_film_list
- sales_by_film_category
- sales_by_store
- staff_list

1. Choose a table

address

CONFIGURATION OPTIONS

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.

Search... advanced

ADDRESS

address_id to

address

address2

district

city_id to

postal_code

phone

last_update at to at

2. Click on **advanced** and fill up the search criteria

| address_id | address | address2 | district | city_id | postal_code | phone | last_update |
|------------|------------------------------|----------|----------|---------|-------------|--------------|---------------------|
| 10 | 1795 Santiago de Compostel | | Texas | 295 | 18743 | 860452626434 | 2014-09-25 22:33:49 |
| 122 | 333 Goinia Way | | Texas | 185 | 78625 | 909029256431 | |
| 310 | 913 Coacalco de Berriozbal L | | Texas | 33 | 42141 | 262088367001 | |
| 405 | 530 Lausanne Lane | | Texas | 135 | 11067 | 775235029633 | |
| 567 | 1894 Boa Vista Way | | Texas | 178 | 77464 | 239357986667 | 2014-09-25 22:33:51 |

1-5 of 5

3. Click on **save search**

- Information
- Search all records
- Saved searches**
- Tables
 - actor
 - address
 - category
 - city
 - country
 - customer
 - film
 - film_actor
 - film_category
 - film_text
 - inventory
 - language
 - payment
 - rental
 - staff
 - store
 - actor_info
 - customer_list
 - film_list
 - nicer_but_slower_film_list
 - sales_by_film_category
 - sales_by_store
 - staff_list
- Technical Information

Search on address

Name

Description

SAVE **CANCEL**

Click on **save** search

Hide the table `film_text` and the store table

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#)

[DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#)

[CONFIGURATION](#)

[DELETE](#)

[BACK](#)

[DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

Click on **Manage tables**

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Table management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

| Show | Name | Label | Description |
|-------------------------------------|---------------|---------------|--|
| <input checked="" type="checkbox"/> | actor | actor | The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table. |
| <input checked="" type="checkbox"/> | address | address | 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. |
| <input checked="" type="checkbox"/> | category | category | The category table lists the categories that can be assigned to a film. The category table is joined to the film table by means of the film_category table. |
| <input checked="" type="checkbox"/> | city | city | The city table contains a list of cities. The city table is referred to by a foreign key in the address table and refers to the country table using a foreign key. |
| <input checked="" type="checkbox"/> | country | country | The country table contains a list of countries. The country table is referred to by a foreign key in the city table. |
| <input checked="" type="checkbox"/> | customer | customer | The customer table contains a list of all customers. The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign |
| <input checked="" type="checkbox"/> | film | film | The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the |
| <input checked="" type="checkbox"/> | film_actor | film_actor | The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing |
| <input checked="" type="checkbox"/> | film_category | film_category | The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the |
| <input type="checkbox"/> | film_text | film_text | The film_text table contains the film_id, title and description columns of the film table, with the contents of the table kept in synchrony with the film table by means of triggers |
| <input checked="" type="checkbox"/> | inventory | inventory | The inventory table contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by |
| <input checked="" type="checkbox"/> | language | language | The language table is a lookup table listing the possible languages that films can have for their language and original language values. The language table is referred to by the |
| <input checked="" type="checkbox"/> | payment | payment | The payment table records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable). The payment table refers to |
| <input checked="" type="checkbox"/> | rental | rental | The rental table contains one row for each rental of each inventory item with information about who rented what item, when it was rented, and when it was returned. The rental |
| <input checked="" type="checkbox"/> | staff | staff | The staff table lists all staff members, including information on email address, login information, and picture. The staff table refers to the store and address tables using foreign |
| <input type="checkbox"/> | store | store | The store table lists all stores in the system. All inventory is assigned to specific stores, and staff and customers are assigned a "home store". The store table refers to the staff |

1. Remove the checkbox tick from film_text and store

2. Click **save**

CANCEL SAVE

The background consists of numerous vertical lines of varying thicknesses, creating a dense, textured effect. The lines are primarily in shades of dark red and black, with some lighter red tones interspersed. The lines are slightly blurred, giving a sense of depth and movement. The overall color palette is monochromatic, focusing on red and black.

Edit film table name and description

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#)

[DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#)

[CONFIGURATION](#)

[DELETE](#)

[BACK](#)

[DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Click on **Manage tables**

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

Table management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

| Show | Name | Label | Description |
|-------------------------------------|---------------|---------------|--|
| <input checked="" type="checkbox"/> | actor | actor | The actor table lists information for all actors. The actor table is joined to the film table by means of the film_actor table. |
| <input checked="" type="checkbox"/> | address | address | 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. |
| <input checked="" type="checkbox"/> | category | category | The category table lists the categories that can be assigned to a film. The category table is joined to the film table by means of the film_category table. |
| <input checked="" type="checkbox"/> | city | city | The city table contains a list of cities. The city table is referred to by a foreign key in the address table and refers to the country table using a foreign key. |
| <input checked="" type="checkbox"/> | country | country | The country table contains a list of countries. The country table is referred to by a foreign key in the city table. |
| <input checked="" type="checkbox"/> | customer | customer | The customer table contains a list of all customers. The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign |
| <input checked="" type="checkbox"/> | film | Film | language table and is referred to by the film_category, film_actor, and inventory tables. More info. |
| <input checked="" type="checkbox"/> | film_actor | film_actor | The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing |
| <input checked="" type="checkbox"/> | film_category | film_category | The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the |
| <input type="checkbox"/> | film_text | film_text | contains the film_id, title and description columns of the film table, with the contents of the table kept in synchrony with the film table by means of triggers |
| <input checked="" type="checkbox"/> | inventory | inventor | contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by |
| <input checked="" type="checkbox"/> | language | language | a lookup table listing the possible languages that films can have for their language and original language values. The language table is referred to by the |
| <input checked="" type="checkbox"/> | payment | payment | records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable). The payment table refers to |
| <input checked="" type="checkbox"/> | rental | rental | contains one row for each rental of each inventory item with information about who rented what item, when it was rented, and when it was returned. The rental |
| <input checked="" type="checkbox"/> | staff | staff | The staff table lists all staff members, including information on email address, login information, and picture. The staff table refers to the store and address tables using foreign |
| <input type="checkbox"/> | store | store | The store table lists all stores in the system. All inventory is assigned to specific stores, and staff and customers are assigned a "home store". The store table refers to the staff |

1. Edit the label and description for film

CANCEL SAVE

2. Click **save**

**Configure the table customer to show only
first_name, last_name and email from the
overall view**

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#) [DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#) [CONFIGURATION](#) [DELETE](#)

[BACK](#) [DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Click on **Manage columns**

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

- actor
- address
- category
- city
- country
- customer**
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

customer

Column management allows to configure several aspects over a table, such as the order which the columns appear, the label name, and the description text. The column visibility can be personalized in three different levels. Select which columns should appear in the overall view of the table data, select which columns should appear in the detailed view of a record, and select which columns should appear in the advanced search option. The option to configure a template for displaying the column is available for denormalized columns.

| Order | Name | Label | Description | | | | |
|-------|-------------|-------------|--|--|-------------------------------------|-------------------------------------|-------------------------------------|
| | first_name | First name | The customer's first name. | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | last_name | Last name | The customer's last name. | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | email | Email | The customer's email address. | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | customer_id | customer_id | A surrogate primary key used to uniquely identify each customer in | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | store_id | store_id | A foreign key identifying the custom | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | address_id | address_id | A foreign key identifying the custom | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | active | active | Indicates whether the customer is a | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | create_date | create_date | The date the customer was added to | | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| | last_update | last_update | The time that the row was created or most recently updated. | | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

2. Tick only the first_name, last_name and email columns

1. Use the arrows to reorder the columns

CANCEL SAVE

3. Click on **Save**

Perform a data transformation

Actor table

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#)

[DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#)

[CONFIGURATION](#)

[DELETE](#)

[BACK](#)

[DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

Click on
Transform data

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

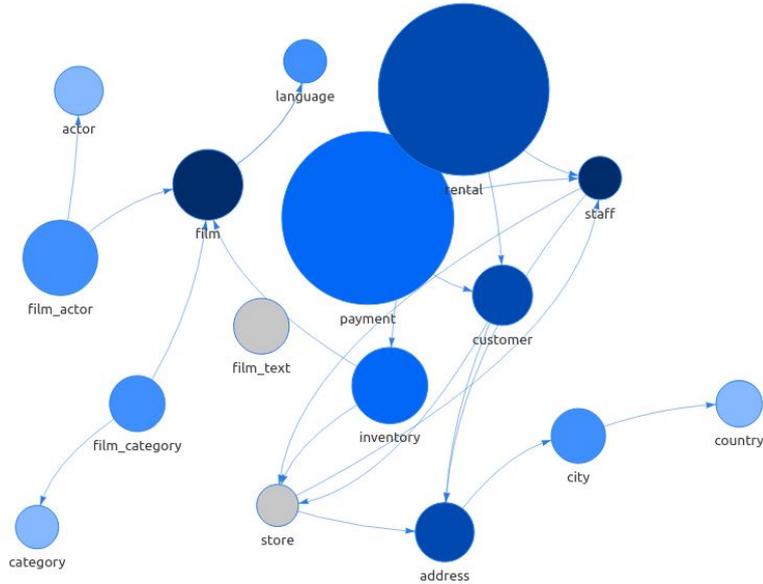
Contact us

Information & Commercial support

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

Select actor table



About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

TRANSFORM ALL ▶ CANCEL ○

- Information
- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

After adding new columns, you can change the columns that are shown, labels and their order at column management

sakila.actor

TRANSFORM ▶ BROWSE

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 | Description |
|---------------|--|
| @ actor_id | A surrogate primary key used to uniquely identify each actor in the table. |
| @ first_name | The actor's first name. |
| @ last_name | The actor's last name. |
| @ last_update | The time that the row was created or most recently updated. |

film_actor

Enable

The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film. The film_actor table refers to the film and actor tables using foreign keys.

Contains reference to sakila.actor on column actor_id

Enable the bridge table

sakila.actor

TRANSFORM ▶

BROWSE 🗃

TRANSFORM ALL ▶

CANCEL ○

Information

actor

address

category

city

country

customer

Film

film_actor

film_category

inventory

language

payment

rental

staff

film_actor

Enable

The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film. The film_actor table refers to the film and actor tables using foreign keys.

🔗 Contains reference to sakila.actor on column actor_id

| | Column name | Description |
|--------------------------|-------------|---|
| <input type="checkbox"/> | actor_id | A foreign key identifying the actor. |
| <input type="checkbox"/> | film_id | A foreign key identifying the film. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

film

Enable

The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the language table and is referred to by the film_category, film_actor, and inventory tables.

🔗 Is referenced by sakila.film_actor on column film_id

Enable the film table

[About DBPTK Enterprise](#)

What is DBPTK Enterprise?
License
Sponsors

[Download](#)

Binary
Source code

[Development](#)

Bug reporting

[Contact us](#)

Information & Commercial support

TRANSFORM ALL CANCEL

- Information
- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

2. Click on Transform

film

The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table is referred to by the film_category, film_actor, and inventory tables.

Is referenced by sakila.film_actor on column film_jd

| | Column name | Description |
|-------------------------------------|----------------------|---|
| <input type="checkbox"/> | film_jd | A surrogate primary key used to uniquely identify each film in the table. |
| <input checked="" type="checkbox"/> | title | The title of the film. |
| <input checked="" type="checkbox"/> | description | A short description or plot summary of the film. |
| <input checked="" type="checkbox"/> | release_year | The year in which the movie was released. |
| <input type="checkbox"/> | language_id | A foreign key pointing at the language table; identifies the language of the film. |
| <input type="checkbox"/> | original_language_id | A foreign key pointing at the language table; identifies the original language of the film. Used when a film has been dubbed into a new language. |
| <input type="checkbox"/> | rental_duration | The length of the rental period, in days. |
| <input type="checkbox"/> | rental_rate | The cost to rent the film for the period specified in the rental_duration column. |
| <input type="checkbox"/> | length | The duration of the film, in minutes. |
| <input type="checkbox"/> | replacement_cost | The amount charged to the customer if the film is not returned or is returned in a damaged state. |
| <input checked="" type="checkbox"/> | rating | The rating assigned to the film. Can be one of: G, PG, PG-13, R, or NC-17. |
| <input type="checkbox"/> | special_features | Lists which common special features are included on the DVD. Can be zero or more of: Trailers, Commentaries, Deleted Scenes, Behind the Scenes. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

1. Select the columns

language

The language table is a lookup table listing the possible languages that films can have for their language and original language values. The language table is referred to by the film table.

Enable

Jobs

Jobs are processes that run in the background and require less user interaction. They are used to process high volumes of data that would normally consume more memory if run in the foreground. For example data transformation is a job that require read all data of database in order to perform denormalization actions.

mguimaraes



| ID | Database | Table | Name | Create time | Start time | End time | Progress | Status |
|-----|------------|------------|----------------|---------------------|---------------------|---------------------|-----------------------|-----------|
| 159 | mguimaraes | actor | denormalizeJob | 2020-08-03 11:44:43 | 2020-08-03 11:44:43 | 2020-08-03 11:44:58 | 100% (200 of 200) | Completed |
| 143 | mguimaraes | film | denormalizeJob | 2020-07-31 16:53:48 | 2020-07-31 16:53:48 | 2020-07-31 16:54:47 | 100% (1000 of 1000) | Completed |
| 101 | mguimaraes | address | denormalizeJob | 2020-07-31 16:33:46 | 2020-07-31 16:33:46 | 2020-07-31 16:34:37 | 100% (603 of 603) | Completed |
| 100 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:46 | 2020-07-31 16:33:46 | 2020-07-31 16:34:19 | 100% (600 of 600) | Completed |
| 99 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:34:18 | 100% (599 of 599) | Completed |
| 98 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:33:47 | 100% (6 of 6) | Completed |
| 97 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:33:46 | 100% (16 of 16) | Completed |
| 96 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:38:47 | 100% (5462 of 5462) | Completed |
| 95 | mguimaraes | rental | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:53:48 | 100% (16044 of 16044) | Completed |
| 93 | mguimaraes | rental | denormalizeJob | 2020-07-31 16:12:07 | 2020-07-31 16:12:07 | 2020-07-31 16:19:41 | 100% (16044 of 16044) | Completed |
| 92 | mguimaraes | rental | denormalizeJob | 2020-07-31 15:17:02 | 2020-07-31 15:17:02 | 2020-07-31 15:24:19 | 100% (16044 of 16044) | Completed |
| 91 | mguimaraes | film_actor | denormalizeJob | 2020-07-31 15:17:02 | 2020-07-31 15:17:02 | 2020-07-31 15:18:42 | 100% (5462 of 5462) | Completed |
| 90 | mguimaraes | category | denormalizeJob | 2020-07-31 15:16:39 | 2020-07-31 15:16:39 | 2020-07-31 15:16:40 | 100% (16 of 16) | Completed |
| 89 | mguimaraes | language | denormalizeJob | 2020-07-31 15:16:38 | 2020-07-31 15:16:38 | 2020-07-31 15:16:39 | 100% (6 of 6) | Completed |
| 88 | mguimaraes | customer | denormalizeJob | 2020-07-31 15:16:38 | 2020-07-31 15:16:38 | 2020-07-31 15:16:56 | 100% (599 of 599) | Completed |

Click on the job to see the result when completed

1-15 of 25

Show More

Perform a data transformation

Film table

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#)

[DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#)

[CONFIGURATION](#)

[DELETE](#)

[BACK](#)

[DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

Click on
Transform data

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

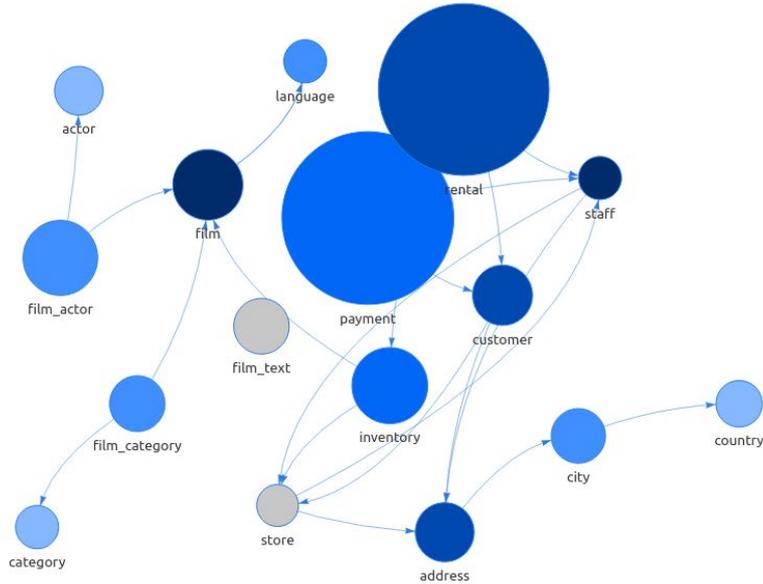
Contact us

Information & Commercial support

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

Select film table



About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

sakila.film

TRANSFORM ▶

BROWSE 🗃

TRANSFORM ALL ▶

CANCEL ✕

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

| | | |
|---|------------------|---|
| @ | replacement_cost | The amount charged to the customer if the film is not returned or is returned in a damaged state. |
| @ | rating | The rating assigned to the film. Can be one of: G, PG, PG-13, R, or NC-17. |
| @ | special_features | Lists which common special features are included on the DVD. Can be zero or more of: Trailers, Commentaries, Deleted Scenes, Behind the Scenes. |
| @ | last_update | The time that the row was created or most recently updated. |

language

Enable

The language table is a lookup table listing the possible languages that films can have for their language and original language values. The language table is referred to by the film table.

↔ Is referenced by sakila.film on column `language_id`

language

Enable

The language table is a lookup table listing the possible languages that films can have for their language and original language values. The language table is referred to by the film table.

↔ Is referenced by sakila.film on column `original_language_id`

film_actor

Enable

The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film. The film_actor table refers to the film and actor tables using foreign keys.

↔ Contains reference to sakila.film on column `film_id`

Enable the bridge table

film_category

Enable

The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film_category table listing the category and film. The film_category table refers to the film and category tables using foreign keys.

↔ Contains reference to sakila.film on column `film_id`

TRANSFORM ALL ▶ CANCEL ✖️

- Information
- actor
- address
- category
- city
- country
- customer
- Film**
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

film_actor

Enable

The film_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film_actor table listing the actor and film. The film_actor table refers to the film and actor tables using foreign keys.

Contains reference to sakila.film on column film_id

| | Column name | Description |
|--------------------------|-------------|---|
| <input type="checkbox"/> | actor_id | A foreign key identifying the actor. |
| <input type="checkbox"/> | film_id | A foreign key identifying the film. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

actor

Enable

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

Is referenced by sakila.film_actor on column actor_id

Enable the actor table

film_category

The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film_category table listing the category and film. The film_category table refers to the film and category tables using foreign keys.

Contains reference to sakila.film on column film_id

inventory

Enable

The inventory table contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by the rental table.

Contains reference to sakila.film on column film_id

sakila.film

TRANSFORM ▶

BROWSE 🗪

TRANSFORM ALL ▶

CANCEL ✕

Information

- actor
- address
- category
- city
- country
- customer
- Film**
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

| | Column name | Description |
|-------------------------------------|-------------|---|
| <input type="checkbox"/> | actor_id | A foreign key identifying the actor. |
| <input checked="" type="checkbox"/> | film_id | A foreign key identifying the film. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

2. Click on
Transform

actor

Enable

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

🔗 Is referenced by sakila.film_actor on column actor_id

| | Column name | Description |
|-------------------------------------|-------------|--|
| <input type="checkbox"/> | actor_id | A surrogate primary key used to uniquely identify each actor in the table. |
| <input checked="" type="checkbox"/> | first_name | The actor's first name. |
| <input checked="" type="checkbox"/> | last_name | The actor's last name. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

1. Select the
columns

film_category

Enable

The film_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film_category table listing the category and film. The film_category table refers to the film and category tables using foreign keys.

🔗 Contains reference to sakila.film on column film_id

inventory

Enable

The inventory table contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by the rental table.

Jobs

Jobs are processes that run in the background and require less user interaction. They are used to process high volumes of data that would normally consume more memory if run in the foreground. For example data transformation is a job that require read all data of database in order to perform denormalization actions.

mguimaraes



| ID | Database | Table | Name | Create time | Start time | End time | Progress | Status |
|-----|------------|------------|----------------|---------------------|---------------------|---------------------|-----------------------|-----------|
| 160 | mguimaraes | film | denormalizeJob | 2020-08-03 11:49:30 | 2020-08-03 11:49:30 | unknown | 27% (274 of 1000) | Started |
| 159 | mguimaraes | actor | denormalizeJob | 2020-08-03 11:44:43 | 2020-08-03 11:44:43 | 2020-08-03 11:44:58 | 100% (200 of 200) | Completed |
| 143 | mguimaraes | film | denormalizeJob | 2020-07-31 16:53:48 | 2020-07-31 16:53:48 | 2020-07-31 16:54:47 | 100% (1000 of 1000) | Completed |
| 101 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:46 | 2020-07-31 16:33:46 | 2020-07-31 16:34:37 | 100% (603 of 603) | Completed |
| 100 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:46 | 2020-07-31 16:33:46 | 2020-07-31 16:34:19 | 100% (600 of 600) | Completed |
| 99 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:34:18 | 100% (599 of 599) | Completed |
| 98 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:33:47 | 100% (6 of 6) | Completed |
| 97 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:33:46 | 100% (16 of 16) | Completed |
| 96 | mguimaraes | film_actor | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:38:47 | 100% (5462 of 5462) | Completed |
| 95 | mguimaraes | rental | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:53:48 | 100% (16044 of 16044) | Completed |
| 93 | mguimaraes | rental | denormalizeJob | 2020-07-31 16:12:07 | 2020-07-31 16:12:07 | 2020-07-31 16:19:41 | 100% (16044 of 16044) | Completed |
| 92 | mguimaraes | rental | denormalizeJob | 2020-07-31 15:17:02 | 2020-07-31 15:17:02 | 2020-07-31 15:24:19 | 100% (16044 of 16044) | Completed |
| 91 | mguimaraes | film_actor | denormalizeJob | 2020-07-31 15:17:02 | 2020-07-31 15:17:02 | 2020-07-31 15:18:42 | 100% (5462 of 5462) | Completed |
| 90 | mguimaraes | category | denormalizeJob | 2020-07-31 15:16:39 | 2020-07-31 15:16:39 | 2020-07-31 15:16:40 | 100% (16 of 16) | Completed |
| 89 | mguimaraes | language | denormalizeJob | 2020-07-31 15:16:38 | 2020-07-31 15:16:38 | 2020-07-31 15:16:39 | 100% (6 of 6) | Completed |

Click on the job to see the result when completed

1-15 of 26

Show More

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

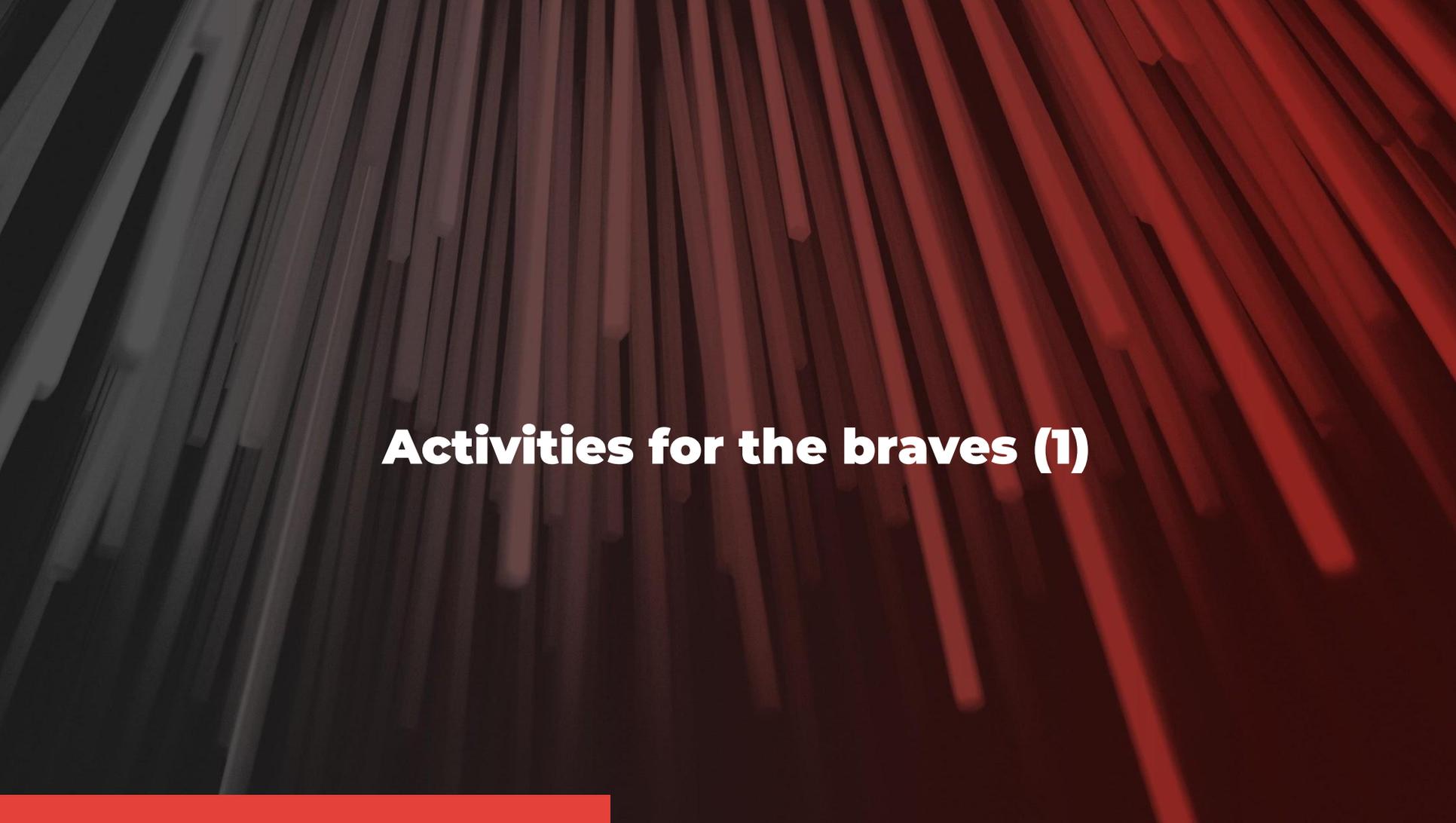
Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

The background consists of numerous vertical lines of varying thickness and color, ranging from dark grey to bright red. The lines are slightly blurred and have a sense of depth, creating a dynamic, textured effect. The overall color palette is monochromatic, focusing on red and black tones.

Activities for the braves (1)

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#)

[DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#)

[CONFIGURATION](#)

[DELETE](#)

[BACK](#)

[DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Click on **Manage columns**

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_actor_info
- film_actor_info_info
- film_actor_info_info_info
- payment
- rental
- staff

1. Add a space between the two fields

2. Add a
 tag

3. Add a space between the two fields

4. Click **save**

Template that will be rendered in the table panel.

Template: `{{first_name}} {{last_name}}`

Possible fields: `first_name` `last_name`

Separator that will be used if this column has multiple values.

Separator: `
`

Template that will be rendered in the table panel when table or columns are exported.

Template: `{{first_name}} {{last_name}}`

Possible fields: `first_name` `last_name`

Quantity of records that will show in the table panel.

10

CANCEL **SAVE**

| Column Name | Description | Visible | Exportable |
|------------------|--|--------------------------|-------------------------------------|
| length | The duration of the film, in minutes. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| replacement_cost | The amount charged to the customer if the film is not returned or is damaged. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| special_features | Lists which common special features are included on the DVD. Can be used to identify the original release of the film. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| last_update | The time that the row was created or most recently updated. | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

- actor
- address
- category
- city
- country
- customer
- Film**
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

Film

BROWSE

Column management allows to configure several aspects over a table, such as the order which the columns appear, the label name, and the description text. The column visibility can be personalized in three different levels. Select which columns should appear in the overall view of the table data, select which columns should appear in the detailed view of a record, and select which columns should appear in the advanced search option. The option to configure a template for displaying the data is available for denormalized columns.

| Order | Name | Label | Description | | | | |
|-------|--|----------------------|---|---|-------------------------------------|-------------------------------------|-------------------------------------|
| ↑ ↓ | title | Title | The title of the film. | | | | |
| ↑ ↓ | @film_id > film_actor @actor_id > actor @first_name, last_name | List of actors | List of actors | | | | |
| ↑ ↓ | description | Description | A short description or plot summary of the film. | ⚙ | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | release_year | Release Year | The year in which the movie was released. | ⚙ | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | rating | Rating | The rating assigned to the film. Can be one of: G, PG, PG-13, R, or | ⚙ | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | film_id | film_id | A surrogate primary key used to uniquely identify each film in the | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | language_id | language_id | A foreign key pointing at the language table; identifies the language | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | original_language_id | original_language_id | A foreign key pointing at the language table; identifies the original | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | rental_duration | rental_duration | The length of the rental period, in days. | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | rental_rate | rental_rate | The cost to rent the film for the period specified in the | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | length | length | The duration of the film, in minutes. | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | replacement_cost | replacement_cost | The amount charged to the customer if | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | special_features | special_features | Lists which common special features are | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| ↑ ↓ | last_update | last_update | The time that the row was created or modified. | ⚙ | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

1. Click **browse**1. Click **save**

CANCEL

SAVE

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

- Information**
- Search all records
- Saved searches
- Tables
 - actor
 - address
 - category
 - city
 - country
 - customer
 - Film**
 - film_actor
 - film_category
 - inventory
 - language
 - payment
 - rental
 - staff
 - actor_info
 - customer_list
 - film_list
 - nicer_but_slower_film_list
 - sales_by_film_category
 - sales_by_store
 - staff_list
- Technical Information

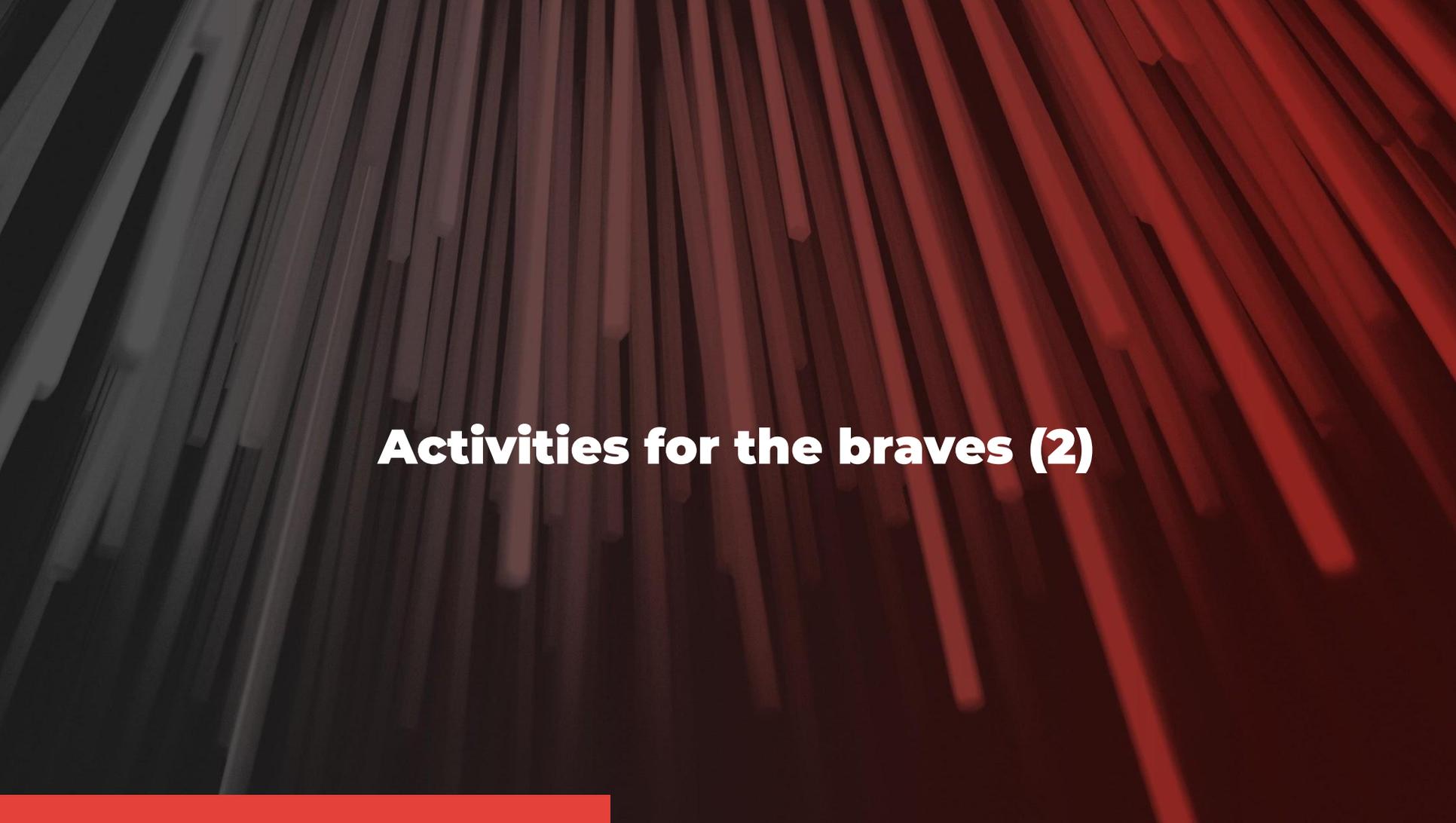
Film

[CONFIGURATION](#) [OPTIONS](#)

The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the language table and is referred to by the film_category, film_actor, and inventory tables.

Search... advanced

| Title | List of actors | Description | Release Year | Rating |
|---------------------------------------|--|---|--|---|
| <small>The title of the film.</small> | <small>List of actors</small> | <small>A short description or plot summary of the film.</small> | <small>The year in which the movie was released.</small> | <small>The rating assigned to the film. Can be one of: G, PG, PG...</small> |
| ACADEMY DINOSAUR | PENELOPE GUINESS CHRISTIAN GABLE LUCILLE TRACY SANDRA PECK JOHNNY CAGE MENA TEMPLE WARREN NOLTE OPRAH KILMER ROCK DUKAKIS MARY KEITEL | A Epic Drama of a Feminist And a Mad Scientist v | 2006 | PG |
| ACE GOLDFINGER | BOB FAWCETT MINNIE ZELLWEGER SEAN GUINESS CHRIS DEPP | A Astounding Epistle of a Database Administratr | 2006 | G |
| ADAPTATION HOLES | NICK WAHLBERG BOB FAWCETT CAMERON STREEP RAY JOHANSSON JULIANNE DENCH | A Astounding Reflection of a Lumberjack And a C | 2006 | NC-17 |
| AFFAIR PREJUDICE | JODIE DEGENERES SCARLETT DAMON KENNETH PESCI FAY WINSLET OPRAH KILMER | A Fanciful Documentary of a Frisbee And a Lumb | 2006 | G |
| AFRICAN EGG | GARY PHOENIX DUSTIN TAUTOU MATTHEW LEIGH MATTHEW CARREY THORA TEMPLE | A Fast-Paced Documentary of a Pastry Chef And | 2006 | G |

The background features a series of vertical, slightly blurred lines in various shades of red and black, creating a sense of depth and movement. The lines are most prominent in the center and fade towards the edges.

Activities for the braves (2)

mguimaraes

DPC Workshop practical session activities resolution

Archival date: 2020-08-03 01:00 (UTC+1)
Archiver: mguimaraes
Archiver contact: mguimaraes@keep.pt
Client machine: unspecified

Database product: MySQL 5.6.49
Data origin timespan: unspecified
Data owner: unspecified
Producer Application: Database Preservation Toolkit (version 2.9.2)

SIARD

See details about the archived format and manage the metadata of tables, columns and other technical elements of the database.

Version: SIARD standard version 2.1 ([SEE SPECIFICATION](#))
Path: PRACTICAL_SESSION_1.SIARD
Size: 789.7 KB

[EDIT METADATA](#) [DELETE](#)

Validation

See details about the validation process.

Status: Not Validated

[VALIDATE](#)

Browsing

See details about the database status. Manage the database metadata by configuring table, columns and data.

Status: Ready

[BROWSE](#) [CONFIGURATION](#) [DELETE](#)

[BACK](#) [DELETE](#)

Click on
Configuration

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

Configuration

Databases often does not have the best naming strategies applied as well as the data internal organization. Configuration panel allows to change or improve certain aspects of the database, such as tables, columns or the data. Table management acts at table level and allows to change the table name, table description, and its visibility to the users. Columns management acts at column level and enables to change column name, column description, two different levels of column visibility, and search option. Data transformation acts at data level and allows to denormalized the database.

Table Management

Select which tables you want to show to the user, you can also modify and override the original table name and the table description.

MANAGE TABLES

Columns Management

Configure several aspects about the table columns, such as the order which the columns appear, the column name, and the description text. Select different columns for table data visualization, detailed record visualization, and advanced search option. Template configuration for displaying the column is available for denormalized columns.

MANAGE COLUMNS

Data Transformation

Improve database readability by transform the data. Join data from one or more tables into another table by performing a denormalization process.

TRANSFORM DATA

Click on
Transform data

About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

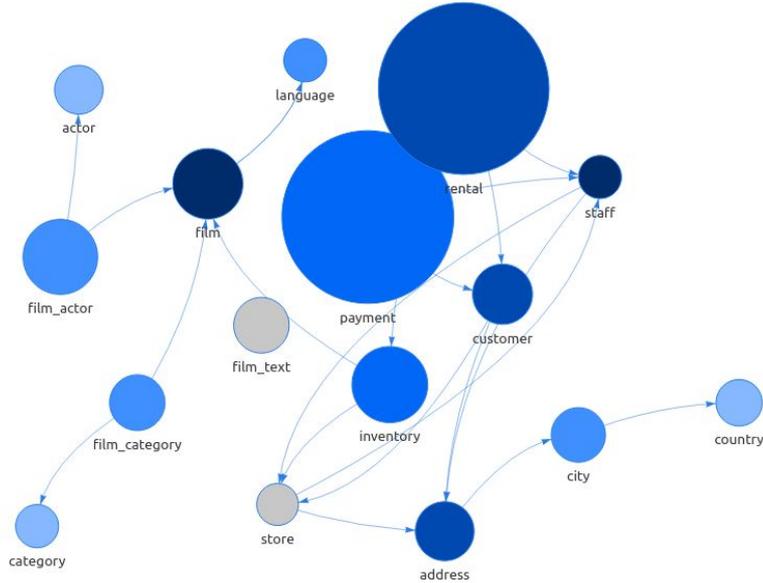
Contact us

Information & Commercial support

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

Select payment table



About DBPTK Enterprise

What is DBPTK Enterprise?
License
Sponsors

Download

Binary
Source code

Development

Bug reporting

Contact us

Information & Commercial support

sakila.payment

TRANSFORM ▶

BROWSE 🗃

TRANSFORM ALL ▶

CANCEL ✕

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

| | | |
|---|--------------|---|
| @ | amount | The amount of the payment. |
| @ | payment_date | The date the payment was processed. |
| @ | last_update | The time that the row was created or most recently updated. |

customer

Enable

The customer table contains a list of all customers. The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign keys.

↳ Is referenced by sakila.payment on column `customer_id`

rental

Enable

The rental table contains one row for each rental of each inventory item with information about who rented what item, when it was rented, and when it was returned. The rental table refers to the inventory, customer, and staff tables and is referred to by the payment table.

↳ Is referenced by sakila.payment on column `rental_id`

Enable rental table

staff

The staff table lists all staff members, including information on email address, login information, and picture. The staff table refers to the store and address tables using foreign keys, and is referred to by the rental, payment, and store tables.

↳ Is referenced by sakila.payment on column `staff_id`

[About DBPTK Enterprise](#)

What is DBPTK Enterprise?
License

[Download](#)

Binary
Source code

[Development](#)

Bug reporting

[Contact us](#)

Information & Commercial support

sakila.payment

TRANSFORM ▶

BROWSE 🗪

TRANSFORM ALL ▶

CANCEL ✕

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff

| | | |
|--------------------------|--------------|--|
| <input type="checkbox"/> | rental_id | A surrogate primary key that uniquely identifies the rental. |
| <input type="checkbox"/> | rental_date | The date and time that the item was rented. |
| <input type="checkbox"/> | inventory_id | The item being rented. |
| <input type="checkbox"/> | customer_id | The customer renting the item. |
| <input type="checkbox"/> | return_date | The date and time the item was returned. |
| <input type="checkbox"/> | staff_id | The staff member who processed the rental. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

customer

Enable

The customer table contains a list of all customers. The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign keys.

↔ Is referenced by sakila.rental on column `customer_id`

inventory

Enable

The inventory table contains one row for each copy of a given film in a given store. The inventory table refers to the film and store tables using foreign keys and is referred to by the rental table.

↔ Is referenced by sakila.rental on column `inventory_id`

staff

The staff table lists all staff members, including information on email address, login information, and picture. The staff table refers to the store and address tables using foreign keys, and is referred to by the payment, and store tables.

↔ Is referenced by sakila.rental on column `staff_id`

Enable inventory table

sakila.payment

TRANSFORM ▶

BROWSE 🗪

TRANSFORM ALL ▶

CANCEL ✕

Information

- 🗪 actor
- 🗪 address
- 🗪 category
- 🗪 city
- 🗪 country
- 🗪 customer
- 🗪 Film
- 🗪 film_actor
- 🗪 film_category
- 🗪 inventory
- 🗪 language
- 🗪 payment
- 🗪 rental
- 🗪 staff

Is referenced by sakila.rental on column inventory_id

| | Column name | Description |
|--------------------------|--------------|---|
| <input type="checkbox"/> | inventory_id | A surrogate primary key used to uniquely identify each item in inventory. |
| <input type="checkbox"/> | film_id | A foreign key pointing to the film this item represents. |
| <input type="checkbox"/> | store_id | A foreign key pointing to the store stocking this item. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

🗪 film

Enable

The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the language table and is referred to by the film_category, film_actor, and inventory tables.

↔ Is referenced by sakila.inventory on column **film_id**

Enable film table

🗪 store

The store table lists all stores in the system. All inventory is assigned to specific stores, and staff and customers are assigned a "home store". The store table refers to the staff and address tables using foreign keys and is referred to by the staff, customer, and inventory tables.

↔ Is referenced by sakila.inventory on column **store_id**

🗪 staff

Enable

The staff table lists all staff members, including information on email address, login information, and picture. The staff table refers to the store and address tables using foreign keys, and is referred to by the rental, payment, and store tables.

↔ Is referenced by sakila.rental on column **staff_id**

TRANSFORM ALL ▶

CANCEL ✕

Information

- actor
- address
- category
- city
- country
- customer
- Film
- film_actor
- film_category
- inventory
- language
- payment
- rental
- staff



last_update

The time that the row was created or most recently updated.

2. Click on
Transform

Enable

film

The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table. The film table refers to the language table and is referred to by the film_category, film_actor, and inventory tables.

↔ Is referenced by sakila.inventory on column film_id

| | Column name | Description |
|-------------------------------------|----------------------|---|
| <input type="checkbox"/> | film_id | A surrogate primary key used to uniquely identify each film in the table. |
| <input checked="" type="checkbox"/> | title | The title of the film. |
| <input type="checkbox"/> | description | A short description or plot summary of the film. |
| | release_year | The year in which the movie was released. |
| | language_id | A foreign key pointing at the language table; identifies the language of the film. |
| | original_language_id | A foreign key pointing at the language table; identifies the original language of the film. Used when a film has been dubbed into a new language. |
| | rental_duration | The length of the rental period, in days. |
| | rental_rate | The cost to rent the film for the period specified in the rental_duration column. |
| <input type="checkbox"/> | length | The duration of the film, in minutes. |
| <input type="checkbox"/> | replacement_cost | The amount charged to the customer if the film is not returned or is returned in a damaged state. |
| <input type="checkbox"/> | rating | The rating assigned to the film. Can be one of: G, PG, PG-13, R, or NC-17. |
| <input type="checkbox"/> | special_features | Lists which common special features are included on the DVD. Can be zero or more of: Trailers, Commentaries, Deleted Scenes, Behind the Scenes. |
| <input type="checkbox"/> | last_update | The time that the row was created or most recently updated. |

1. Select title
column

language

Enable

Jobs

Jobs are processes that run in the background and require less user interaction. They are used to process high volumes of data that would normally consume more memory if run in the foreground. For example data transformation is a job that require read all data of database in order to perform denormalization actions.

mguimaraes



| ID | Database | Table | Name | Create time | Start time | End time | Progress | Status |
|-----|------------|------------|----------------|---------------------|---------------------|---------------------|-----------------------|-----------|
| 161 | mguimaraes | payment | denormalizeJob | 2020-08-03 12:18:13 | 2020-08-03 12:18:13 | 2020-08-03 12:23:28 | 100% (16049 of 16049) | Completed |
| 160 | mguimaraes | film | denormalizeJob | 2020-08-03 11:49:30 | 2020-08-03 11:49:30 | 2020-08-03 11:49:58 | 100% (1000 of 1000) | Completed |
| 159 | mguimaraes | | denormalizeJob | 2020-08-03 11:44:43 | 2020-08-03 11:44:43 | 2020-08-03 11:44:58 | 100% (200 of 200) | Completed |
| 143 | mguimaraes | | denormalizeJob | 2020-07-31 16:53:48 | 2020-07-31 16:53:48 | 2020-07-31 16:54:47 | 100% (1000 of 1000) | Completed |
| 101 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:46 | 2020-07-31 16:33:46 | 2020-07-31 16:34:37 | 100% (603 of 603) | Completed |
| 100 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:46 | 2020-07-31 16:33:46 | 2020-07-31 16:34:19 | 100% (600 of 600) | Completed |
| 99 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:34:18 | 100% (599 of 599) | Completed |
| 98 | mguimaraes | | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:33:47 | 100% (6 of 6) | Completed |
| 97 | mguimaraes | category | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:33:46 | 100% (16 of 16) | Completed |
| 96 | mguimaraes | film_actor | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:38:47 | 100% (5462 of 5462) | Completed |
| 95 | mguimaraes | rental | denormalizeJob | 2020-07-31 16:33:45 | 2020-07-31 16:33:45 | 2020-07-31 16:53:48 | 100% (16044 of 16044) | Completed |
| 93 | mguimaraes | rental | denormalizeJob | 2020-07-31 16:12:07 | 2020-07-31 16:12:07 | 2020-07-31 16:19:41 | 100% (16044 of 16044) | Completed |
| 92 | mguimaraes | rental | denormalizeJob | 2020-07-31 15:17:02 | 2020-07-31 15:17:02 | 2020-07-31 15:24:19 | 100% (16044 of 16044) | Completed |
| 91 | mguimaraes | film_actor | denormalizeJob | 2020-07-31 15:17:02 | 2020-07-31 15:17:02 | 2020-07-31 15:18:42 | 100% (5462 of 5462) | Completed |
| 90 | mguimaraes | category | denormalizeJob | 2020-07-31 15:16:39 | 2020-07-31 15:16:39 | 2020-07-31 15:16:40 | 100% (16 of 16) | Completed |

Click on the job to see the result when completed

1-15 of 27

Show More

- Information
- Search all records
- Saved searches
- Tables
- Technical Information

payment

CONFIGURATION OPTIONS

The payment table records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable). The payment table refers to the customer, rental, and staff tables.

Search... advanced

| payment_id | customer_id | staff_id | rental_id | amount | payment_date | last_update | payment>rental>inventory |
|--|---|--|---|---|---|--|----------------------------|
| <small>A surrogate primary key used L...</small> | <small>The customer whose balance th...</small> | <small>The staff member who process...</small> | <small>The rental that the payment is ...</small> | <small>The amount of the payment.</small> | <small>The date the payment was proc...</small> | <small>The time that the row was crea...</small> | <small>Please EDIT</small> |
| 1 | 1 | 1 | 76 | 2.99 | 2005-05-25 11:30:37 | 2006-02-15 22:12:30 | PATIENT SISTER |
| 2 | 1 | 1 | 573 | 0.99 | 2005-05-28 10:35:23 | 2006-02-15 22:12:30 | TALENTED HOMICIDE |
| 3 | 1 | 1 | 1185 | 5.99 | 2005-06-15 00:54:12 | 2006-02-15 22:12:30 | MUSKETEERS WAIT |
| 4 | 1 | 2 | 1422 | 0.99 | 2005-06-15 18:02:53 | 2006-02-15 22:12:30 | DETECTIVE VISION |
| 5 | 1 | 2 | 1476 | 9.99 | 2005-06-15 21:08:46 | 2006-02-15 22:12:30 | FERRIS MOTHER |
| 6 | 1 | 1 | 1725 | 4.99 | 2005-06-16 15:18:57 | 2006-02-15 22:12:30 | CLOSER BANG |
| 7 | 1 | 1 | 2308 | 4.99 | 2005-06-18 08:41:48 | 2006-02-15 22:12:30 | ATTACKS HATE |
| 8 | 1 | 2 | 2363 | 0.99 | 2005-06-18 13:33:59 | 2006-02-15 22:12:30 | SAVANNAH TOWN |
| 9 | 1 | 1 | 3284 | 3.99 | 2005-06-21 06:24:45 | 2006-02-15 22:12:30 | YOUTH KICK |
| 10 | 1 | 2 | 4526 | 5.99 | 2005-07-08 03:17:05 | 2006-02-15 22:12:30 | FIRE WOLVES |
| 11 | 1 | 1 | 4611 | 5.99 | 2005-07-08 07:33:56 | 2006-02-15 22:12:30 | SATURDAY LAMBS |
| 12 | 1 | 1 | 5244 | 4.99 | 2005-07-09 13:24:07 | 2006-02-15 22:12:30 | SNATCH SLIPPER |
| 13 | 1 | 1 | 5326 | 4.99 | 2005-07-09 16:38:01 | 2006-02-15 22:12:30 | CONFIDENTIAL INTERVIEW |
| 14 | 1 | 1 | 6163 | 7.99 | 2005-07-11 10:13:46 | 2006-02-15 22:12:30 | EXPECTATIONS NATURAL |
| 15 | 1 | 2 | 7273 | 2.99 | 2005-07-27 11:31:22 | 2006-02-15 22:12:30 | LUCK OPUS |
| 16 | 1 | 1 | 7841 | 4.99 | 2005-07-28 09:04:45 | 2006-02-15 22:12:30 | DOORS PRESIDENT |
| 17 | 1 | 2 | 8033 | 4.99 | 2005-07-28 16:18:23 | 2006-02-15 22:12:30 | USUAL UNTOUCHABLES |
| 18 | 1 | 1 | 8074 | 0.99 | 2005-07-28 17:33:39 | 2006-02-15 22:12:30 | FROST HEAD |
| 19 | 1 | 2 | 8116 | 0.99 | 2005-07-28 19:20:07 | 2006-02-15 22:12:30 | WOMEN DORADO |
| 20 | 1 | 2 | 8326 | 2.99 | 2005-07-29 03:58:49 | 2006-02-15 22:12:30 | AMISTAD MIDSUMMER |

1-20 of 16,049

Show More Export

Click **Export**

- Information
- Search all records
- Saved searches
- Tables
- Technical Information

payment

CONFIGURATION OPTIONS

The payment table records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable). The payment table refers to the customer, rental, and staff tables.

Search... advanced

| payment_id | customer_id | staff_id | rental_id | amount | payment_date | last_update | payment>rental>inventory |
|--|---|--|---|---|---|--|----------------------------|
| <small>A surrogate primary key used t...</small> | <small>The customer whose balance th...</small> | <small>The staff member who process...</small> | <small>The rental that the payment is ...</small> | <small>The amount of the payment.</small> | <small>The date the payment was proc...</small> | <small>The time that the row was crea...</small> | <small>Please EDIT</small> |
| 1 | 1 | 1 | 76 | 2.99 | 2005-05-25 11:30:37 | 2006-02-15 22:12:30 | PATIENT SISTER |
| 2 | 1 | 1 | | 4.99 | 2005-05-28 10:35:23 | 2006-02-15 22:12:30 | TALENTED HOMICIDE |
| 3 | 1 | 1 | | 4.99 | 2005-06-15 00:54:12 | 2006-02-15 22:12:30 | MUSKETEERS WAIT |
| 4 | 1 | 2 | | 4.99 | 2005-06-15 18:02:53 | 2006-02-15 22:12:30 | DETECTIVE VISION |
| 5 | 1 | 2 | | 4.99 | 2005-06-15 21:08:46 | 2006-02-15 22:12:30 | FERRIS MOTHER |
| 6 | 1 | 1 | | 4.99 | 2005-06-16 15:18:57 | 2006-02-15 22:12:30 | CLOSER BANG |
| 7 | 1 | 1 | | 4.99 | 2005-06-18 08:41:48 | 2006-02-15 22:12:30 | ATTACKS HATE |
| 8 | 1 | 2 | | 4.99 | 2005-06-18 13:33:59 | 2006-02-15 22:12:30 | SAVANNAH TOWN |
| 9 | 1 | 1 | | 4.99 | 2005-06-21 06:24:45 | 2006-02-15 22:12:30 | YOUTH KICK |
| 10 | 1 | 2 | | 4.99 | 2005-07-08 03:17:05 | 2006-02-15 22:12:30 | FIRE WOLVES |
| 11 | 1 | 1 | 4611 | 5.99 | 2005-07-08 07:33:56 | 2006-02-15 22:12:30 | SATURDAY LAMBS |
| 12 | 1 | 1 | | 4.99 | 2005-07-09 13:24:07 | 2006-02-15 22:12:30 | SNATCH SLIPPER |
| 13 | 1 | 1 | | 4.99 | 2005-07-09 16:38:01 | 2006-02-15 22:12:30 | CONFIDENTIAL INTERVIEW |
| 14 | 1 | 1 | | 7.99 | 2005-07-11 10:13:46 | 2006-02-15 22:12:30 | EXPECATIONS NATURAL |
| 15 | 1 | 2 | 7273 | 2.99 | 2005-07-27 11:31:22 | 2006-02-15 22:12:30 | LUCK OPUS |
| 16 | 1 | 1 | 7841 | 4.99 | 2005-07-28 09:04:45 | 2006-02-15 22:12:30 | DOORS PRESIDENT |
| 17 | 1 | 2 | 8033 | 4.99 | 2005-07-28 16:18:23 | 2006-02-15 22:12:30 | USUAL UNTOUCHABLES |
| 18 | 1 | 1 | 8074 | 0.99 | 2005-07-28 17:33:39 | 2006-02-15 22:12:30 | FROST HEAD |
| 19 | 1 | 2 | 8116 | 0.99 | 2005-07-28 19:20:07 | 2006-02-15 22:12:30 | WOMEN DORADO |
| 20 | 1 | 2 | 8326 | 2.99 | 2005-07-29 03:58:49 | 2006-02-15 22:12:30 | AMISTAD MIDSUMMER |

Export to CSV

Filename
name of the file

Export Rows All Visible
choose between exporting all rows or only the visibles

Include descriptions
select to include the description in the header

CANCEL
CONFIRM

Click **Confirm**

Import the resulting CSV to a spreadsheet software (For this activity it was used google sheets software)

fx | payment>rental>inventory>film>[title]

| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|----|-----------------------------------|---------------|---|---|---|---|---|---|---|---|---|---|---|
| 1 | payment>rental>inventory>film>[t] | SUM of amount | | | | | | | | | | | |
| 2 | TELEGRAPH VOYAGE | 231.73 | | | | | | | | | | | |
| 3 | WIFE TURN | 223.69 | | | | | | | | | | | |
| 4 | ZORRO ARK | 214.69 | | | | | | | | | | | |
| 5 | GOODFELLAS SALUTE | 209.69 | | | | | | | | | | | |
| 6 | SATURDAY LAMBS | 204.72 | | | | | | | | | | | |
| 7 | TITANS JERK | 201.71 | | | | | | | | | | | |
| 8 | TORQUE BOUND | 198.72 | | | | | | | | | | | |
| 9 | HARRY IDAHO | 195.7 | | | | | | | | | | | |
| 10 | INNOCENT USUAL | 191.74 | | | | | | | | | | | |
| 11 | HUSTLER PARTY | 190.78 | | | | | | | | | | | |
| 12 | PELICAN COMFORTS | 188.74 | | | | | | | | | | | |
| 13 | CAT CONEHEADS | 181.7 | | | | | | | | | | | |
| 14 | ENEMY ODDS | 180.71 | | | | | | | | | | | |
| 15 | BUCKET BROTHERHOOD | 180.66 | | | | | | | | | | | |
| 16 | RANGE MOONWALKER | 179.73 | | | | | | | | | | | |
| 17 | MASSACRE USUAL | 179.7 | | | | | | | | | | | |
| 18 | VIDEOTAPE ARSENIC | 178.71 | | | | | | | | | | | |
| 19 | DOGMA FAMILY | 178.7 | | | | | | | | | | | |
| 20 | APACHE DIVINE | 178.69 | | | | | | | | | | | |
| 21 | VELVET TERMINATOR | 177.74 | | | | | | | | | | | |
| 22 | DORADO NOTTING | 176.73 | | | | | | | | | | | |
| 23 | FOOL MOCKINGBIRD | 175.77 | | | | | | | | | | | |
| 24 | WITCHES PANIC | 173.7 | | | | | | | | | | | |
| 25 | CLOSER BANG | 172.72 | | | | | | | | | | | |
| 26 | SCALAWAG DUCK | 172.68 | | | | | | | | | | | |
| 27 | ROSES TREASURE | 171.72 | | | | | | | | | | | |
| 28 | SUNRISE LEAGUE | 170.76 | | | | | | | | | | | |
| 29 | SHOW LORD | 170.75 | | | | | | | | | | | |
| 30 | NIGHTMARE CHILL | 169.75 | | | | | | | | | | | |
| 31 | SCORPION APOLLO | 168.77 | | | | | | | | | | | |
| 32 | AMERICAN CIRCUS | 167.78 | | | | | | | | | | | |
| 33 | CONFIDENTIAL INTERVIEW | 164.71 | | | | | | | | | | | |
| 34 | MAIDEN HOME | 163.76 | | | | | | | | | | | |
| 35 | WORKING MICROCOSMOS | 163.75 | | | | | | | | | | | |
| 36 | SEATTLE EXPECTATIONS | 163.75 | | | | | | | | | | | |
| 37 | BOOGIE AMELIE | 163.7 | | | | | | | | | | | |

Configure as such

Pivot table editor

sakila_payment!A1:H16050

Suggested

Rows Add

payment>rental>inventory>... ×

Order: Descending ▼ Sort by: SUM of amount ▼

Show totals

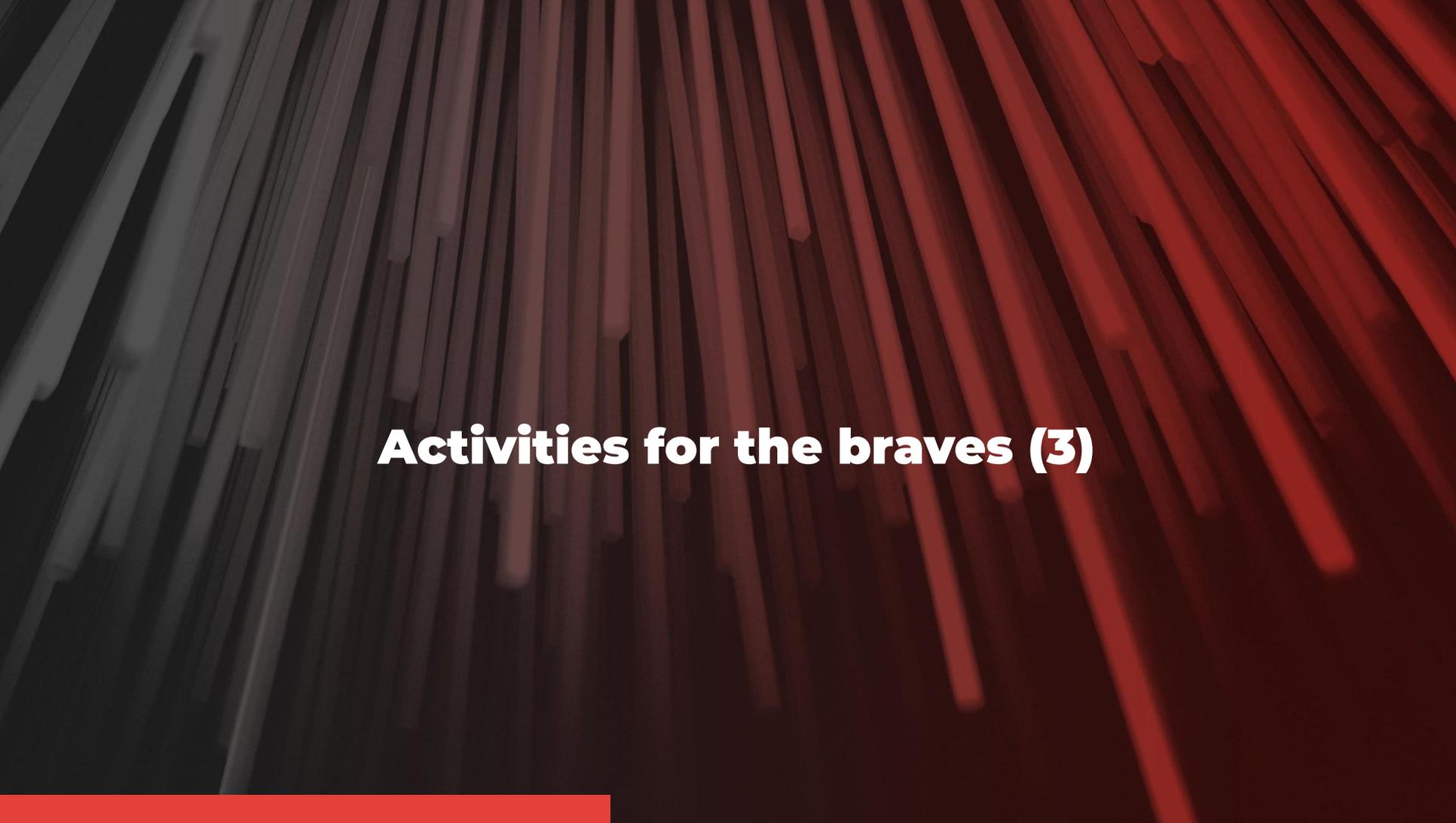
Columns Add

Values Add

amount ×

Summarize by: SUM ▼ Show as: Default ▼

Filters Add

The background consists of numerous vertical lines of varying thickness and color, ranging from dark grey to bright red. The lines are slightly blurred and have a sense of depth, creating a dynamic, textured effect. The overall color palette is monochromatic, focusing on shades of red and black.

Activities for the braves (3)

```
~  
→ java -jar dbptk-app-2.9.2.jar migrate -i mysql -ih dpc.database-preservation.com  
-iu <user> -ip <password> -idb sakila -e import-config -ef /home/mguimaraes/Desktop/dpc-workshop/import-config.yaml
```

Replace with
username and
password provided

import-config.yaml X

home > mguimaraes > Desktop > dpc-workshop > import-config.yaml > ignore > routines

```
348   - name: "manager"
349     merkle: false
350   - name: "total_sales"
351     merkle: false
352   where: ""
353   orderBy: ""
354   - name: "staff list"
355     materialized: false
356     columns:
357       - name: "ID"
358         merkle: false
359       - name: "name"
360         merkle: false
361       - name: "address"
362         merkle: false
363       - name: "zip code"
364         merkle: false
365       - name: "phone"
366         merkle: false
367       - name: "city"
368         merkle: false
369       - name: "country"
370         merkle: false
371       - name: "SID"
372         merkle: false
373     where: ""
374     orderBy: ""
375 ignore:
376   users: false
377   roles: false
378   privileges: false
379   routines: true
380   triggers: true
381   primaryKeys: false
382   candidateKeys: false
383   foreignKeys: false
384   checkConstraints: true
385   views: true
386
```

Open the file in a text editor

Change the **routines**, **triggers**, **check constraints** and **views** value to **true**

```
~  
→ java -jar dbptk-app-2.9.2.jar migrate -i import-config -if /home/mguimaraes/Desktop/dpc-workshop/import-config.yaml -e siard-2 -ef /home/mguimaraes/Desktop/dpc-workshop/import-config-example.siard_
```

Run the command to create the SIARD from an import-config file