



AT82.02

DATA MODELING AND MANAGEMENT

LAB9: DATA ENGINEERING



Co-funded by the
Erasmus+ Programme
of the European Union



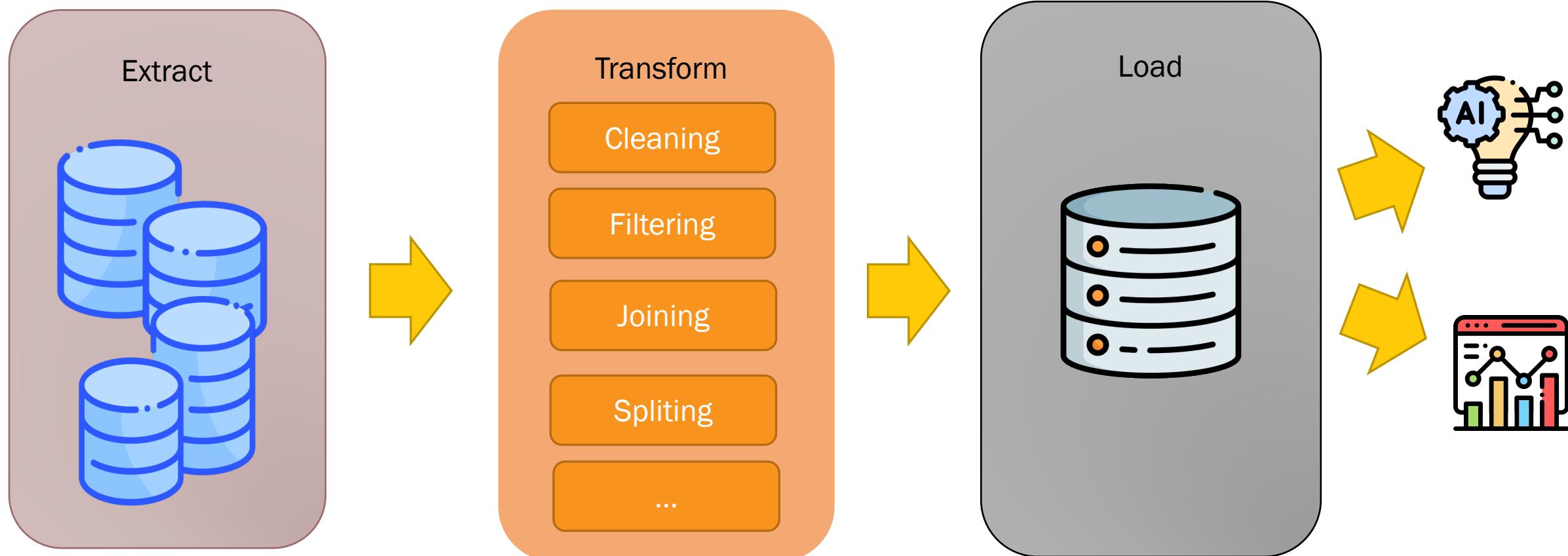
Outline

- Extract Transform Load (ETL)
- Talend Open Studio for Integration
- Tutorial: Movie Rating Prediction
 - Part 1 - Extract
 - Part 2 – Transform & Load

Extract Transform Load

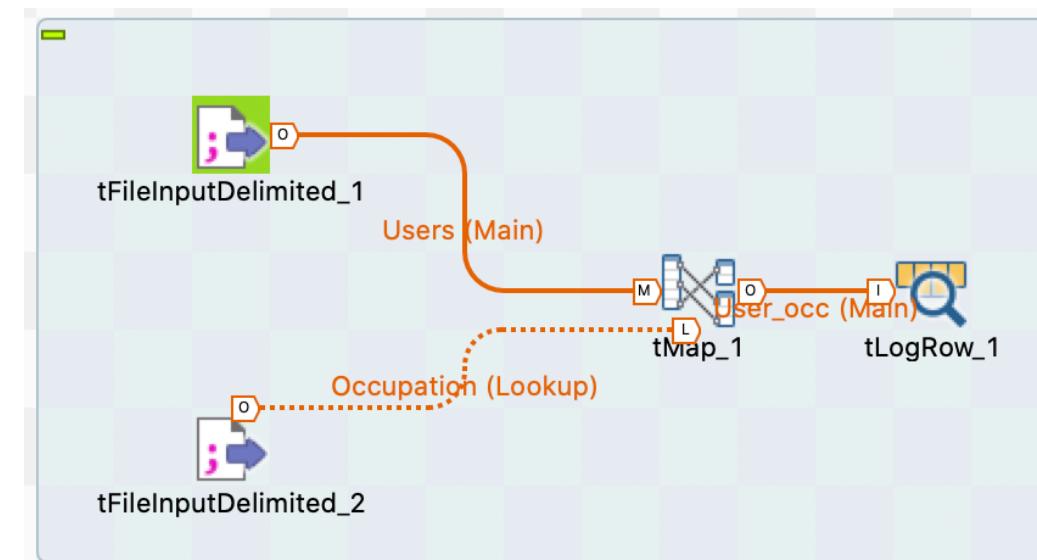
- Extract is the process of *reading* data from a database. In this stage, the data is collected, often from multiple and different types of sources.
- Transform is the *process of converting the extracted data* from its previous form into the form it needs to be in.
- Load is the process of *writing* the data into the target database.

ETL Process



Talend Open Studio for Data Integration

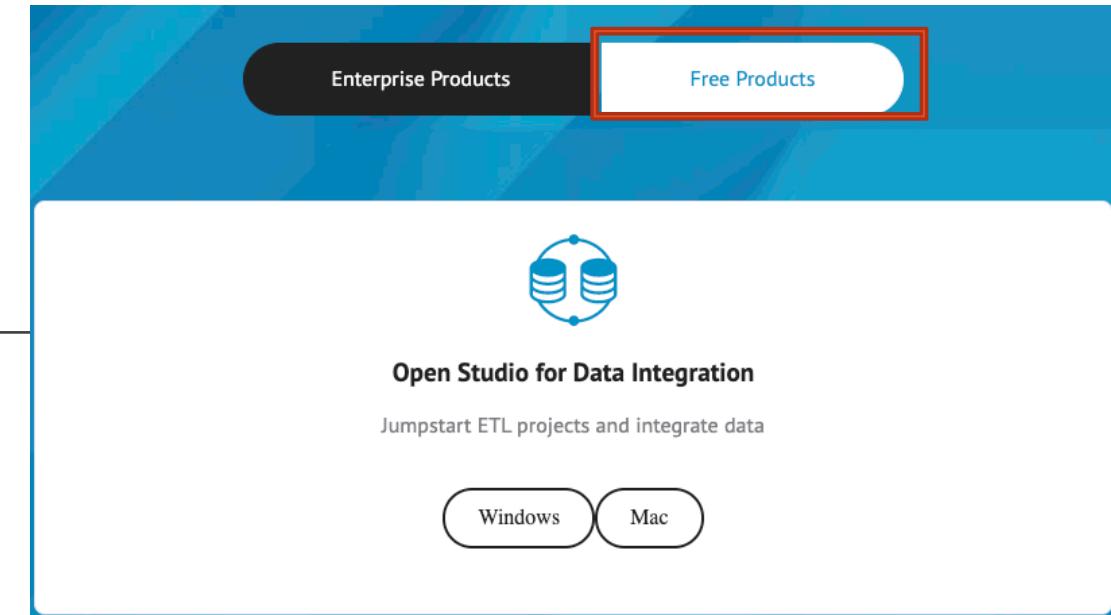
- Talend Open Studio is a free open source ETL tool for Data Integration
- Provide Drag and Drop components and connect them to create and run ETL Jobs.
- The tool will create the Java code for the job automatically
- There are multiple options to connect with Data Sources such as RDBMS, Excel, SaaS Big Data ecosystem



Talend Installation

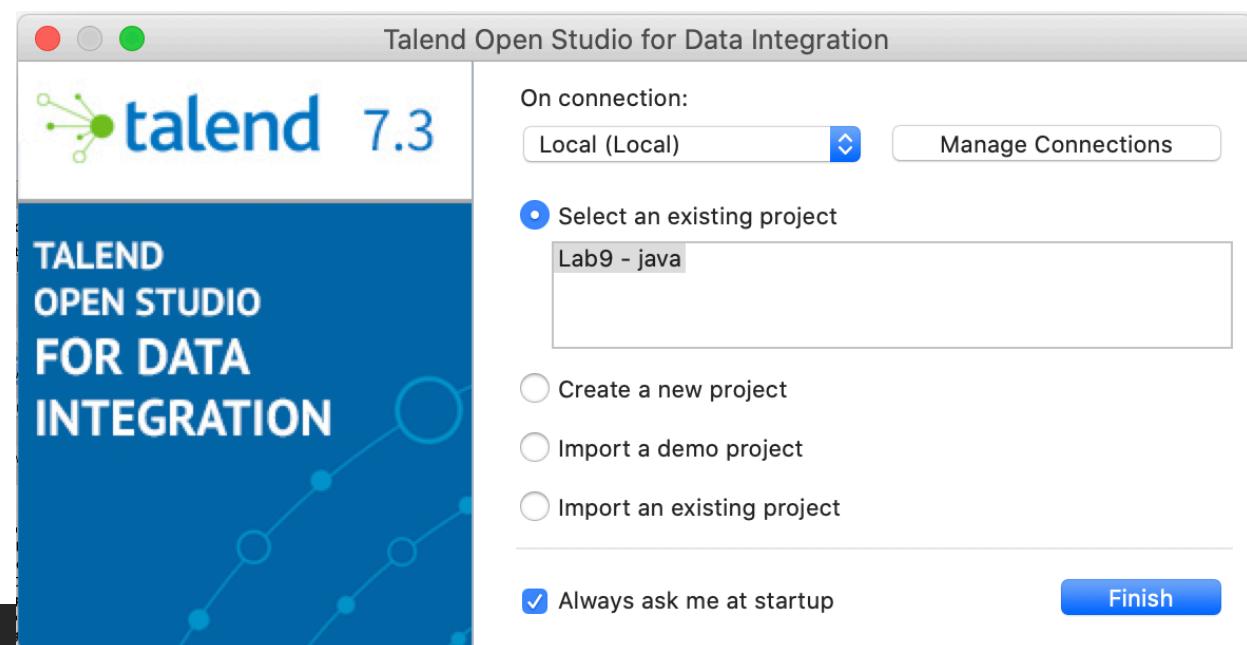
Open Studio for Data Integration

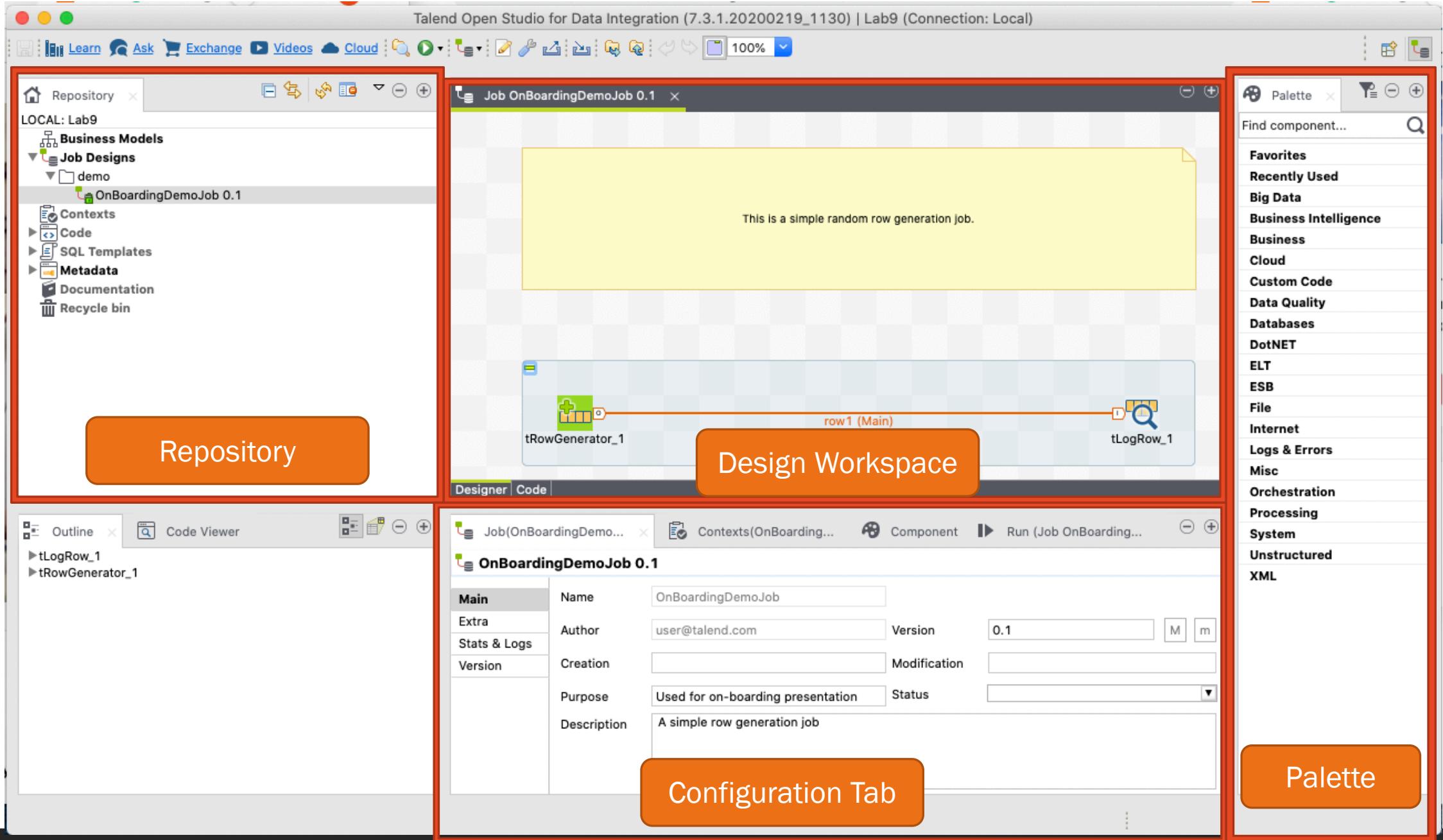
- <https://www.talend.com/products/data-integration/data-integration-open-studio/>



Cloud Integration (Free Trial):

- <https://www.talend.com/free-trial/>





Components for Data Integration

No	Component	Description
1	tMysqlConnection	Connects to MySQL database defined in the component.
2	tMysqlInput	Runs database query to read a database and extract fields (tables, views etc.) depending on the query.
3	tMysqlOutput	Used to write, update, modify data in a MySQL database.
4	tFileInputDelimited	Reads a delimited file row by row and divides them into separate fields and passes it to the next component.
5	tFileInputExcel	Reads an excel file row by row and divides them into separate fields and passes it to the next component.

Components for Data Integration

No	Component	Description
6	tFileDialog	Gets all the files and directories from a given file mask pattern.
7	tFileArchive	Compresses a set of files or folders in to zip, gzip or tar.gz archive file.
8	tRowGenerator	Provides an editor where you can write functions or choose expressions to generate your sample data.
9	tMsgBox	Returns a dialog box with the message specified and an OK button.
10	tLogRow	Monitors the data getting processed. It displays data/output in the run console
11	tPreJob	Defines the sub jobs that will run before your actual job starts

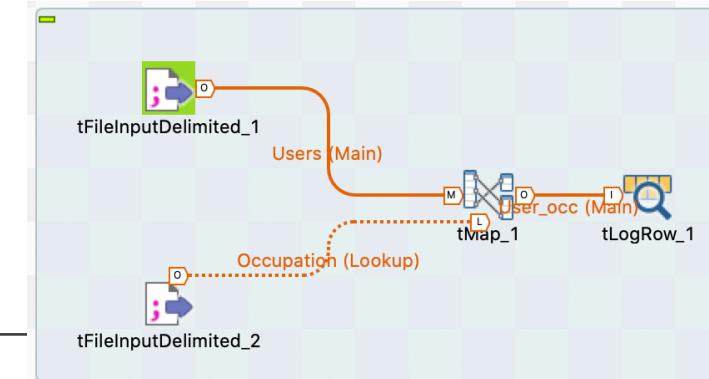
Components for Data Integration

No	Component	Description
12	tMap	Acts as a plugin in Talend studio. It takes data from one or more sources, transforms it, and then sends the transformed data to one or more destinations.
13	tJoin	Joins 2 tables by performing inner and outer joins between the main flow and the lookup flow.
14	tJava	Enables you to use personalized java code in the Talend program.
15	tRunJob	Manages complex job systems by running one Talend job after another.
16	tUniqRow	Compares entries and sorts out duplicate entries from the input flow.

Definition

Job Design

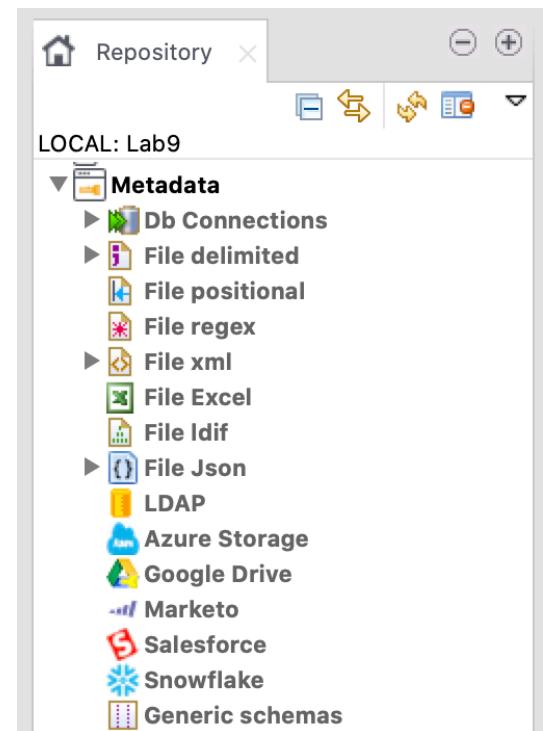
- This is the graphical representation of the business model.
- In this design, one or more components are connected with each other to run a data integration process.



Example Job

Metadata

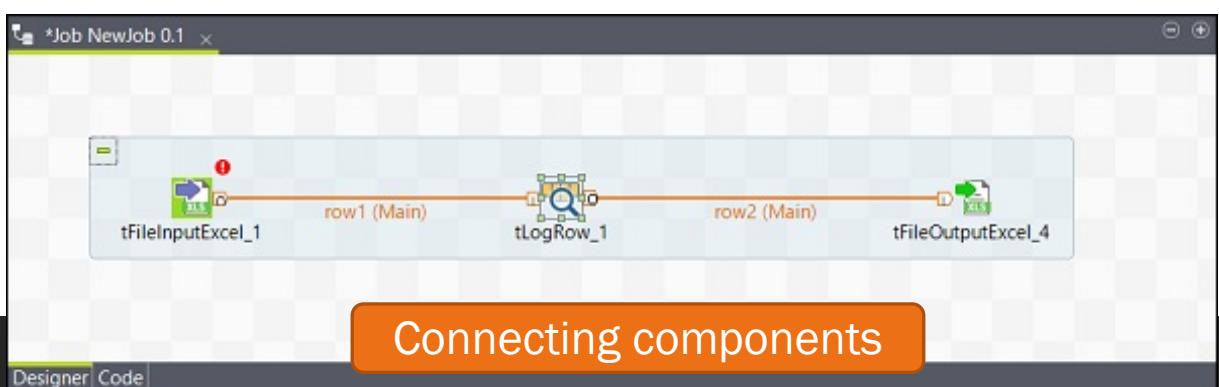
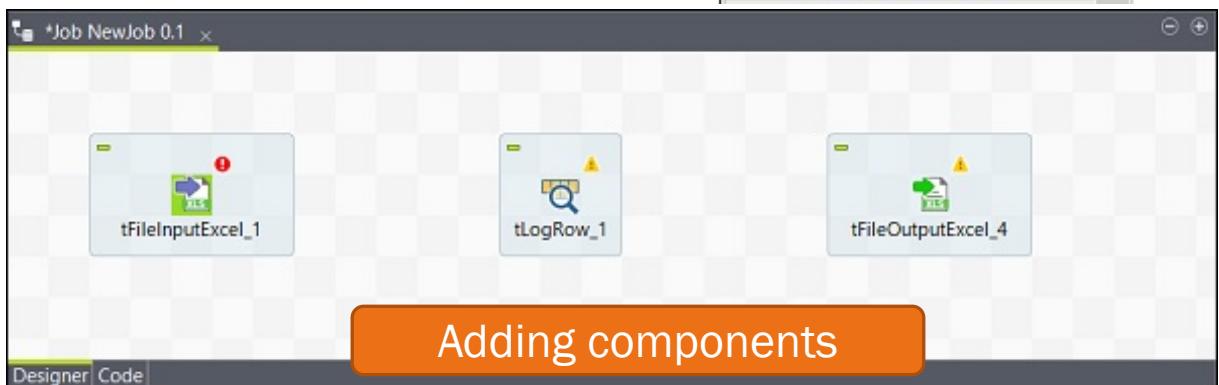
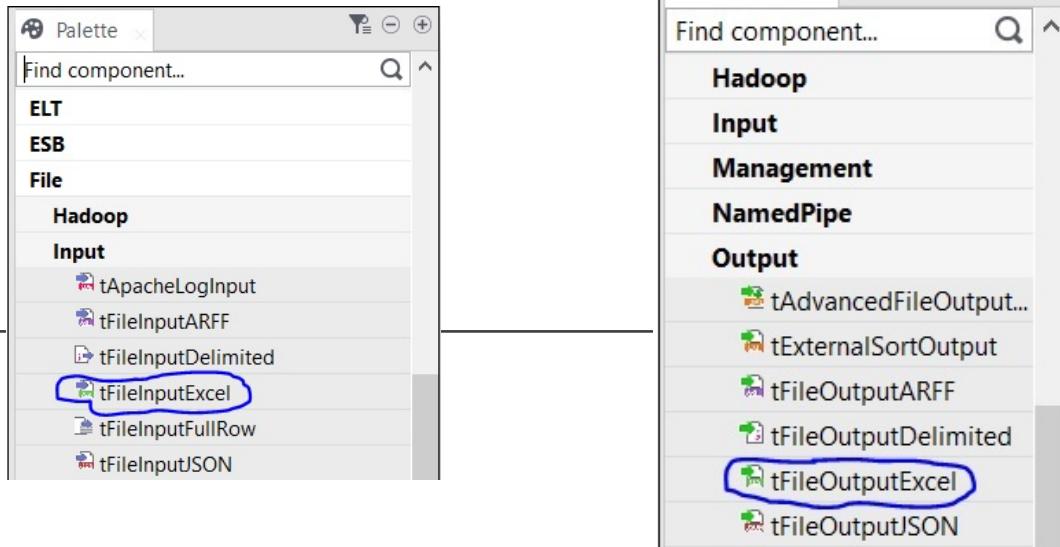
- Metadata basically means data about data.
- The main use of metadata in Talend Open Studio is that you can use these data sources in several jobs just by a simple drag and drop from the Metadata in repository panel.



Metadata

Job Design

- 1) Create a job
- 2) Adding Components to Job Design
- 3) Connecting the components
- 4) Configuring the components
- 5) Executing the Job



Metadata

- 1) Select type of Metadata
- 2) Set Configuration of the data

Example: File Delimited (CSV)

- Encoding
- Field Separator
- Schema

The screenshot illustrates the Talend Data Integration process for defining metadata. It is divided into three main sections:

- Step 1:** A screenshot of the Talend Repository interface. A red circle labeled "1" is positioned above the "Metadata" node under the "LOCAL: Lab9" section. The interface shows various connection types like Db Connections, File delimited, and Generic schemas.
- Step 2:** A screenshot of the "File - Step 3 of 4" dialog. A red circle labeled "2" is on the top left. The dialog title is "Add a Metadata File on repository". It contains sections for "File Settings" (Encoding: UTF-8, Field Separator: Comma, Row Separator: Standard EOL) and "Rows To Skip" (Header: 1). An orange arrow points from the "Set Schema" button in Step 3 to this dialog.
- Step 3:** A screenshot of the "File - Step 4 of 4" dialog. A red circle labeled "3" is at the top right. The title is "Add a Schema on repository Define the Schema". It includes fields for "Name" (metadata) and "Comment". The "Schema" section shows a table with columns: UserID (int), Gender (String), BirthDate (Date), Occupation (Integer), and Zipcode (String). A large orange button labeled "Set Schema" is located in the bottom right of this panel. Another orange arrow points from the "Set configuration" button in Step 2 to this dialog.

Set Schema

Set configuration

Column	Key	Type	Nullable	Date Pattern (Ctrl+Space Length)	Precision	Default	Comment
UserID	<input checked="" type="checkbox"/>	int	<input checked="" type="checkbox"/>	2	0		
Gender	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>	6	0		
BirthDate	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd"	10	0	
Occupation	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>	2	0		
Zipcode	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>	5	0		

Movie Rating Prediction

- Movie rating is an important element to decide movie quality.
- People prefer to use rating as reference to decide before deciding to watch a movie or not.
- We plan to use historical values of the movie as features (e.g. user profile, movie category, user rating) to predict movie rating before the movie released.



Movie Rating Prediction

Datasets: Movies, User Profile , Movie Rating by Users

The variables used for **input**:

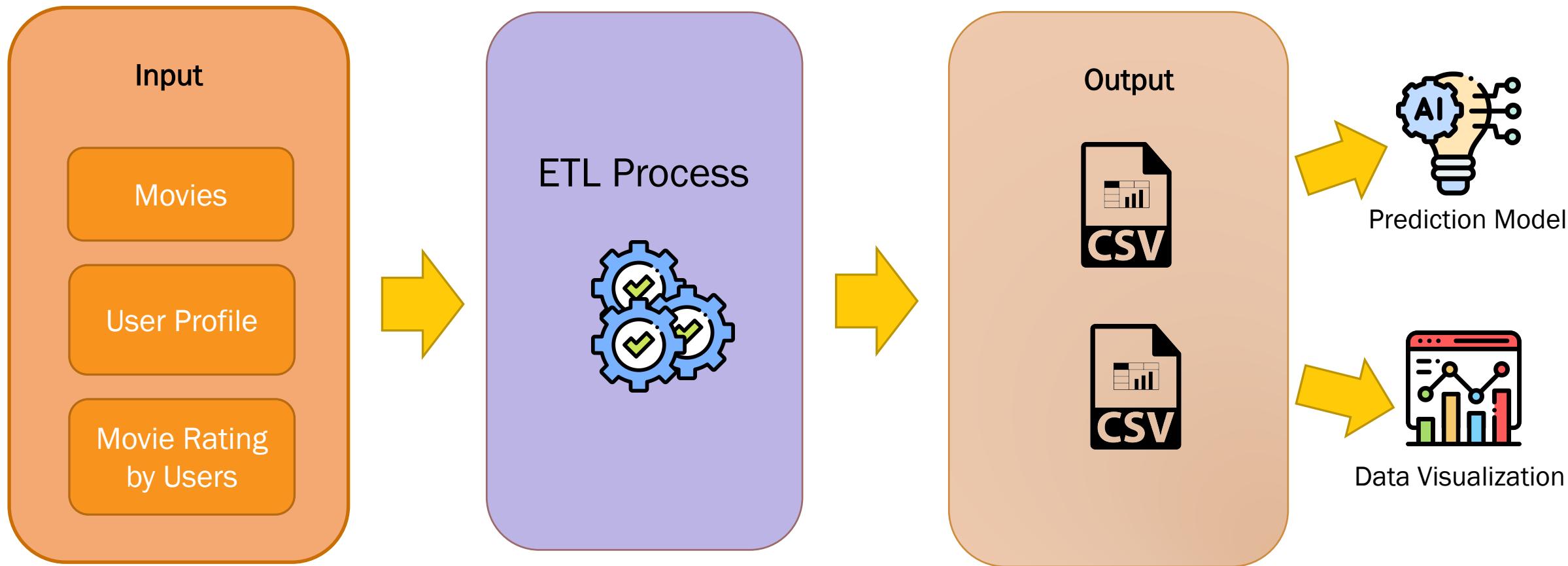
- Age , Gender , Occupation, Movie Category

The variables used for **model prediction**:

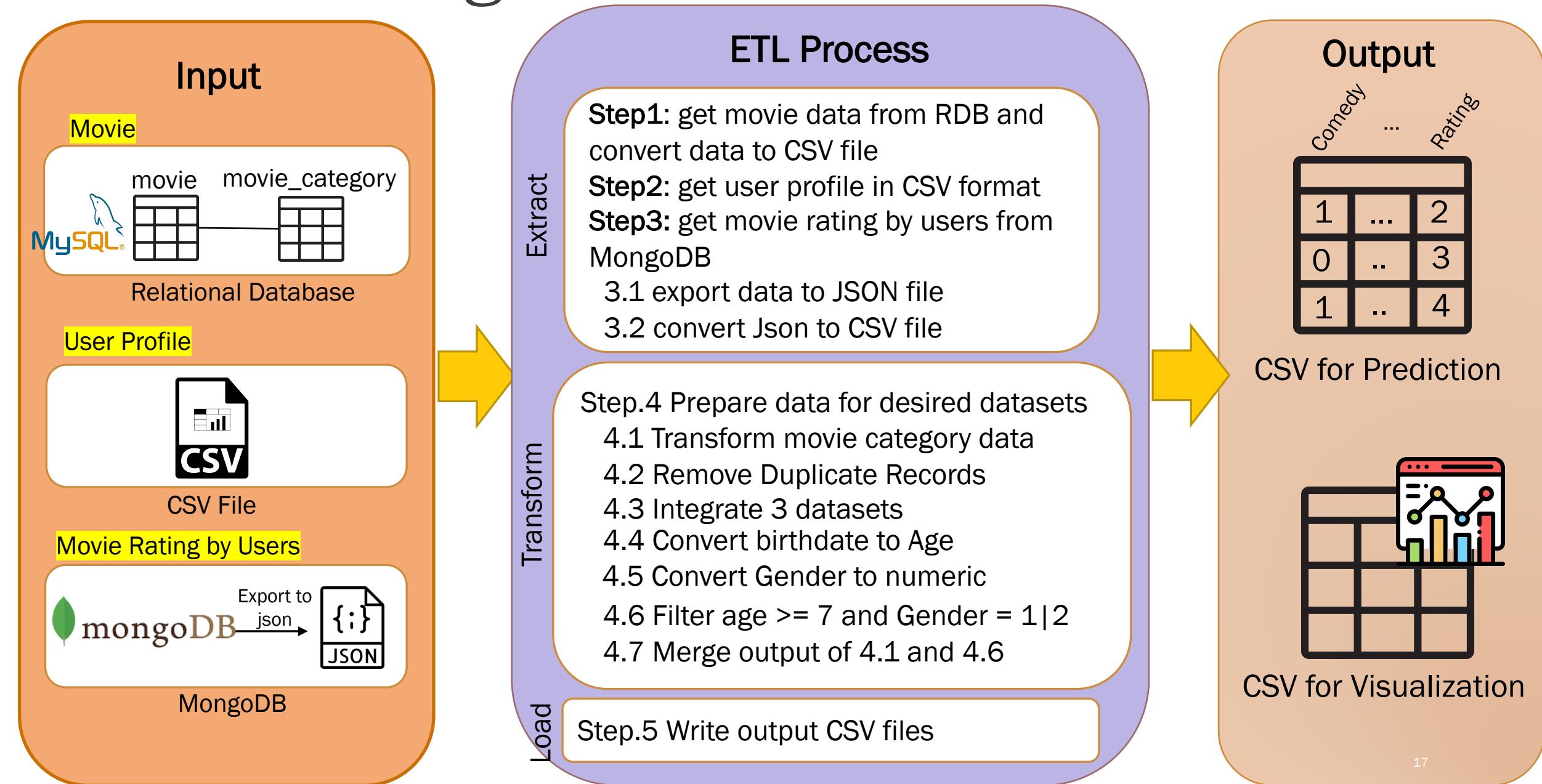
- User Rating

source: <https://www.kaggle.com/sherinclaudia/movie-rating-prediction>

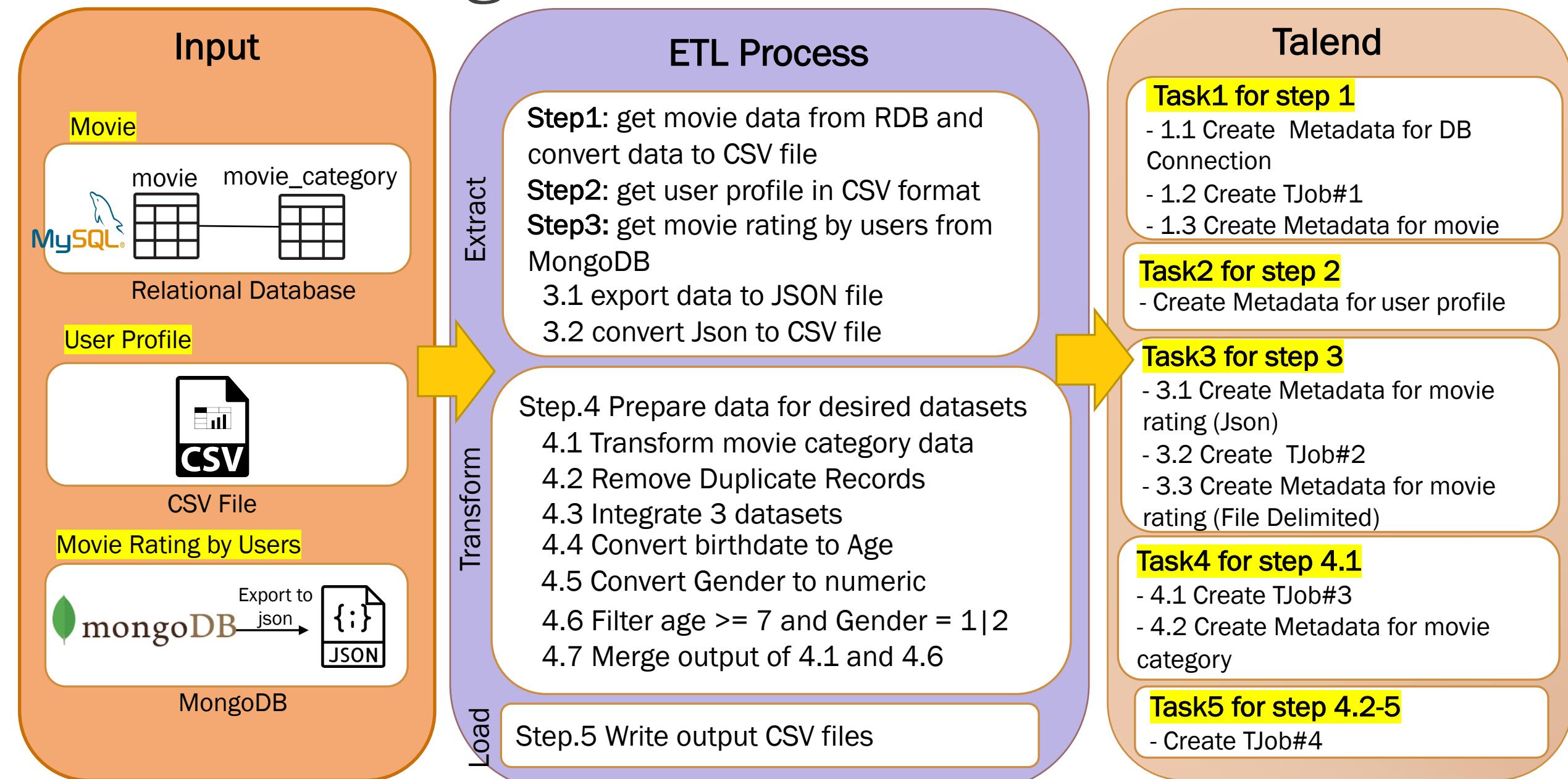
Movie Rating Prediction



Movie Rating Prediction



Movie Rating Prediction



Desired Output for Prediction

UserID	MovielD	Gender2	Age	Occupation	Animation	Children_s	Comedy	Adventure	Fantasy	Romance	Drama	Action	Crime	...	Rating
2	1357	1	50	16	0	0	0	0	0	1	1	0	0	...	5
2	3068	1	50	16	0	0	0	0	0	0	1	0	0	...	4
2	1537	1	50	16	0	0	1	0	0	0	0	0	0	...	4
2	647	1	49	16	0	0	0	0	0	0	1	0	0	...	3
2	2194	1	50	16	0	0	0	0	0	0	1	1	1	...	4
2	648	1	50	16	0	0	0	1	0	0	0	1	0	...	4
2	2268	1	49	16	0	0	0	0	0	0	1	0	1	...	5
2	2628	1	50	16	0	0	0	1	1	0	0	1	0	...	3
2	1103	1	50	16	0	0	0	0	0	0	1	0	0	...	3
2	2916	1	50	16	0	0	0	1	0	0	0	1	0	...	3
2	3468	1	50	16	0	0	0	0	0	0	1	0	0	...	3
2	1210	1	49	16	0	0	0	1	0	1	0	1	0	...	5
2	1792	1	50	16	0	0	0	0	0	0	0	1	0	...	4
2	1687	1	49	16	0	0	0	0	0	0	0	1	0	...	3
2	1213	1	50	16	0	0	0	0	0	0	1	0	1	...	3
2	3578	1	50	16	0	0	0	0	0	0	1	1	0	...	2
2	2881	1	50	16	0	0	0	0	0	0	0	1	0	...	5
2	3030	1	49	16	0	0	1	0	0	0	1	0	0	...	3
2	1217	1	49	16	0	0	0	0	0	0	1	0	0	...	4
2	3105	1	50	16	0	0	0	0	0	0	1	0	0	...	3
2	434	1	50	16	0	0	0	1	0	0	0	1	1	...	4



Gender Age Occupation

Movie category

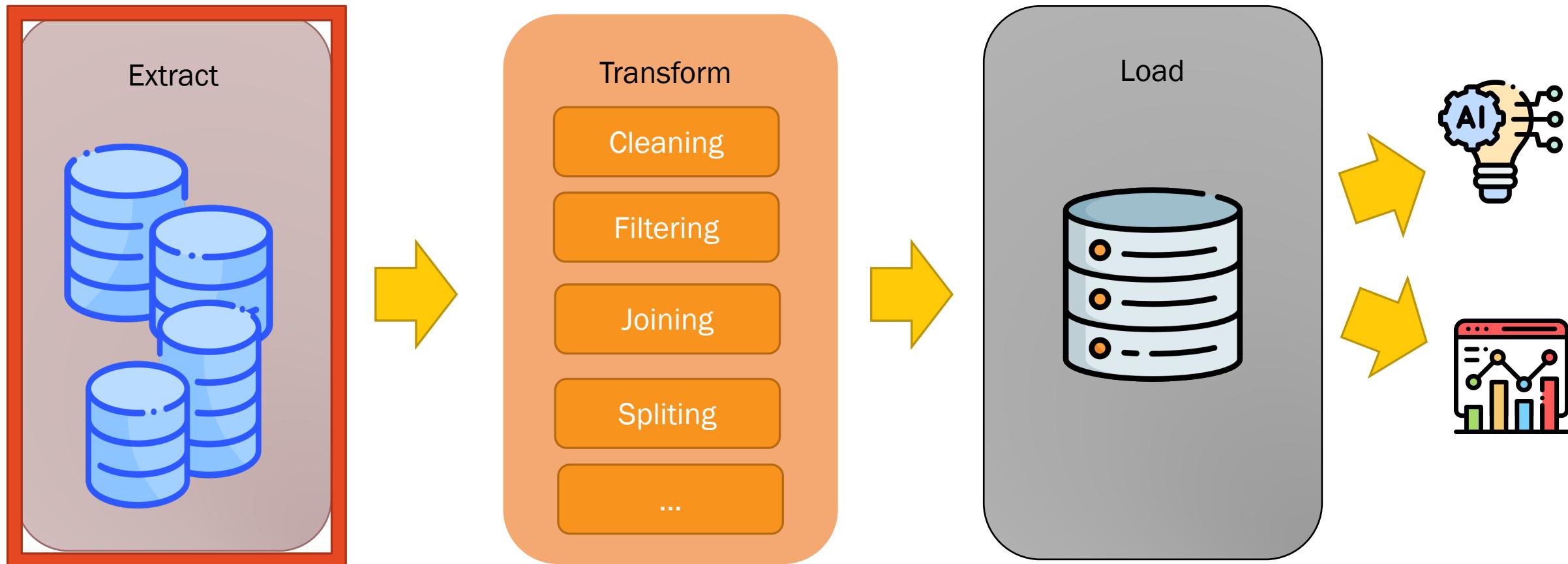
Rating₁₉

Desired Output for Visualization

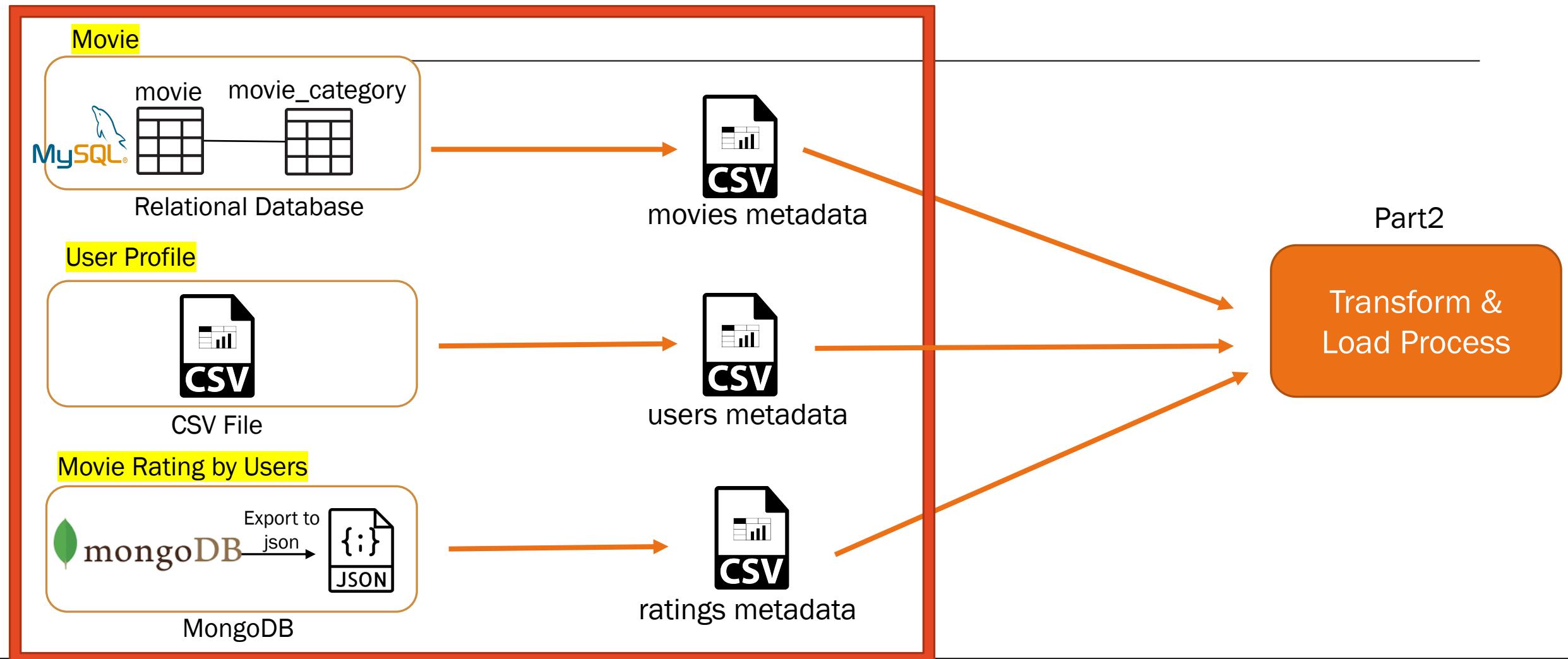
A	B	C	D	E	F	G	H	I
UserID	MovielID	Title	Year	Gender	Gender2	Age	Timestamp	Rating
2	1357	Shine (1996)	1996	Male	1	50	2015-10-18 15:24:38	5
2	3068	Verdict, The (1982)	1982	Male	1	50	2015-01-14 15:31:09	4
2	1537	Shall We Dance? (Shall We Dansu?) (1996)	1996	Male	1	50	2015-05-22 09:03:30	4
2	647	Courage Under Fire (1996)	1996	Male	1	49	2014-02-21 07:02:49	3
2	2194	Untouchables, The (1987)	1987	Male	1	50	2015-10-25 03:23:12	4
2	648	Mission: Impossible (1996)	1996	Male	1	50	2015-01-07 10:58:19	4
2	2268	Few Good Men, A (1992)	1992	Male	1	49	2014-05-27 15:12:42	5
2	2628	Star Wars: Episode I - The Phantom Menace (1999)	1999	Male	1	50	2015-11-03 14:11:03	3
2	1103	Rebel Without a Cause (1955)	1955	Male	1	50	2015-06-07 05:36:56	3
2	2916	Total Recall (1990)	1990	Male	1	50	2015-09-19 18:07:05	3
2	3468	Hustler, The (1961)	1961	Male	1	50	2015-08-17 09:42:45	5
2	1210	Star Wars: Episode VI - Return of the Jedi (1983)	1983	Male	1	49	2014-02-09 14:01:03	4
2	1792	U.S. Marshalls (1998)	1998	Male	1	50	2015-07-02 00:47:44	3
2	1687	Jackal, The (1997)	1997	Male	1	49	2014-05-11 10:57:57	3
2	1213	GoodFellas (1990)	1990	Male	1	50	2014-11-28 21:19:25	2
2	3578	Gladiator (2000)	2000	Male	1	50	2015-06-24 00:51:45	5
2	2881	Double Jeopardy (1999)	1999	Male	1	50	2014-12-29 21:27:23	3
2	3030	Yojimbo (1961)	1961	Male	1	49	2014-10-09 16:27:37	4
2	1217	Ran (1985)	1985	Male	1	49	2014-03-12 16:41:31	3
2	3105	Awakenings (1990)	1990	Male	1	50	2015-10-17 05:59:39	4
2	434	Cliffhanger (1993)	1993	Male	1	50	2015-10-03 23:12:06	2
2	2126	Snake Eyes (1998)	1998	Male	1	49	2014-02-04 15:38:02	3
2	2427	Rocky IV (1985)	1985	Male	1	50	2014-10-22 00:11:11	2

Part 1

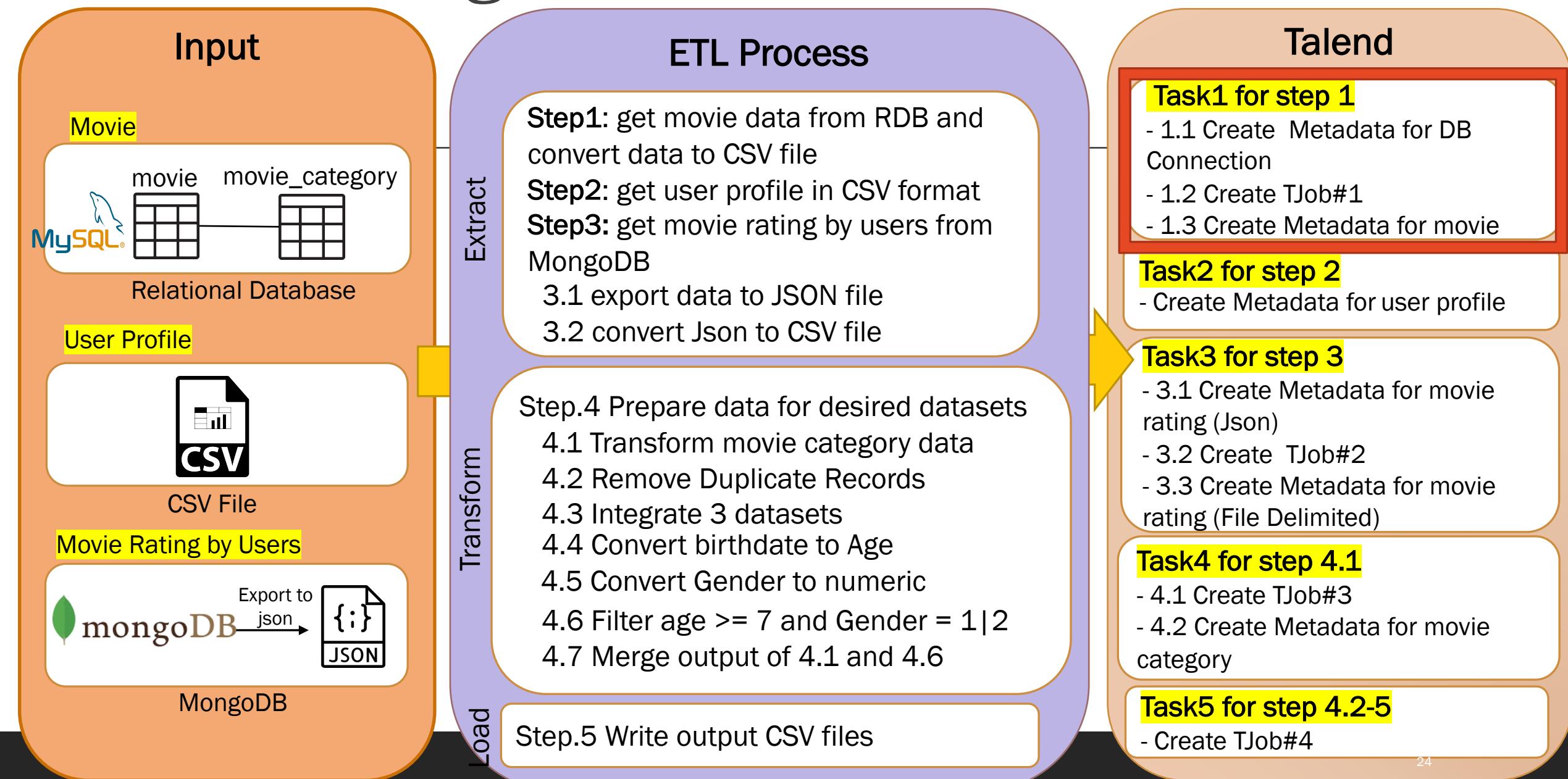
ETL Process



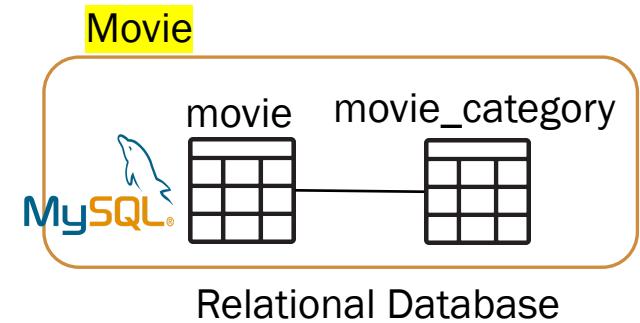
Goal of Part1



Movie Rating Prediction



Task1 for step 1



Step1: get movie data from RDB and convert data to CSV file

Task1:

- 1.1 Create Metadata for DB Connection
- 1.2 Create TJob#1 => Convert data in MySQL to CSV file named “movie_data.csv”
- 1.3 Create Metadata for movie data (File Delimited)

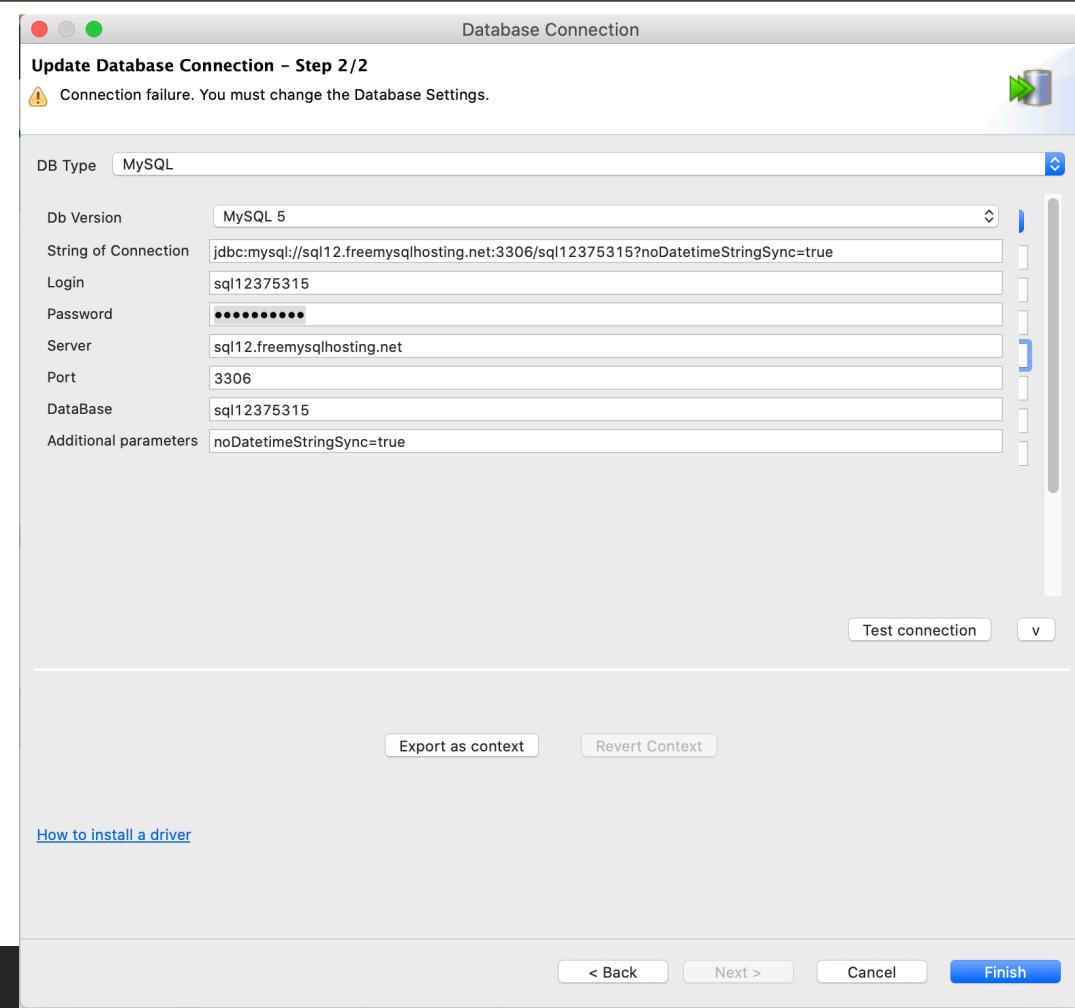
Task1

1.1 Create Metadata for DB Connection (1)

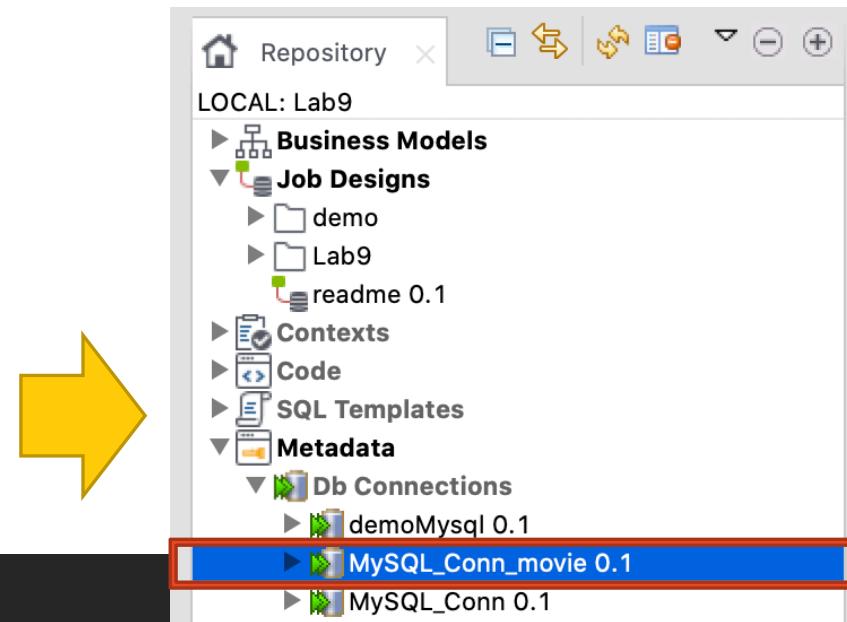
The screenshot shows the Talend Studio interface. On the left, the repository tree is visible under the 'LOCAL: Lab9' section, showing various components like PivotMovieCategory 0.1, RangeLookup 0.1, ReadCSVFile 0.1, SortCSVFile 0.1, TransformBasic 0.1, and readme 0.1. Below this, there are sections for Contexts, Code, SQL Templates, and Metadata. Under Metadata, the 'Db Connections' item is selected and highlighted in blue. A context menu is open over this item, with 'Create connection' being the top option, also highlighted in blue. Other options in the context menu include 'Create folder', 'Expand/Collapse', 'Export items', and 'Import items'. To the right of the repository tree, a 'Database Connection' dialog box is open. The title bar says 'Database Connection' and the sub-title is 'New Database Connection on repository - Step 1/2'. A warning message at the top states 'It is inadvisable to leave the purpose blank.' The 'Name' field is filled with 'MySQL_Conn_movie'. The 'Purpose' field is empty. Other fields include 'Description', 'Author' (set to 'user@talend.com'), 'Locker', 'Version' (set to '0.1'), 'Status', and 'Path'. At the bottom of the dialog are buttons for '< Back', 'Next >', 'Cancel', and 'Finish'. A red annotation text 'DB Connection name = "MySQL_Conn_movie"' is overlaid on the 'Name' field area.

Task1

1.1 Create Metadata for DB Connection (2)

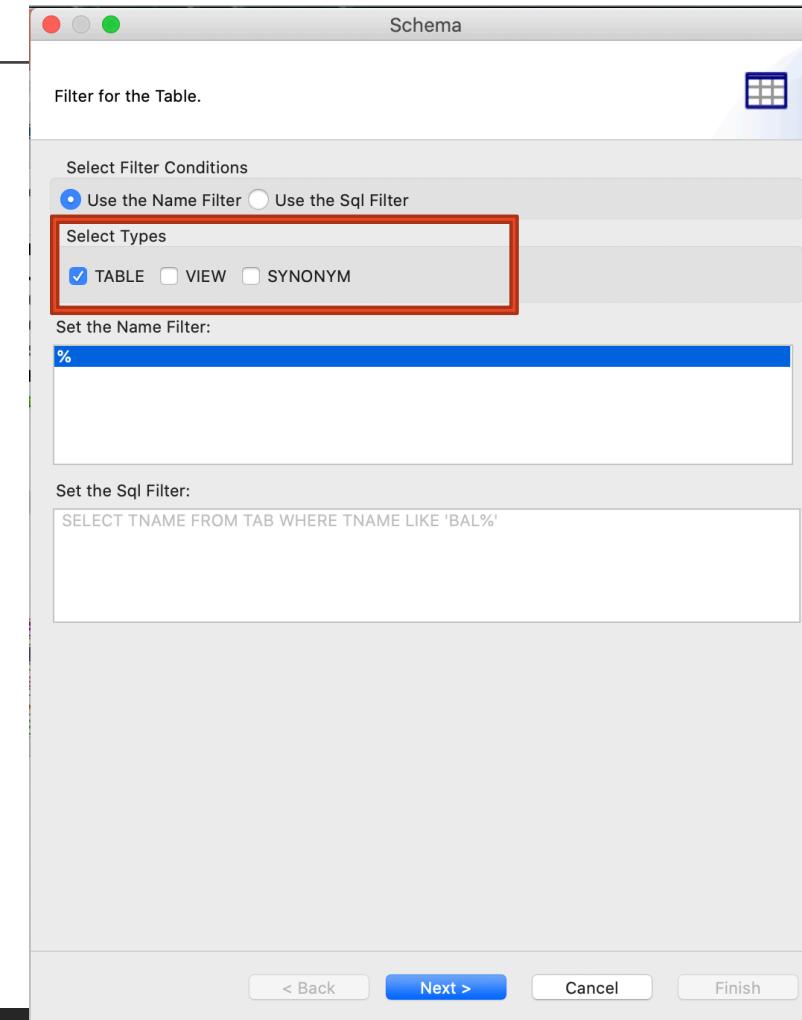
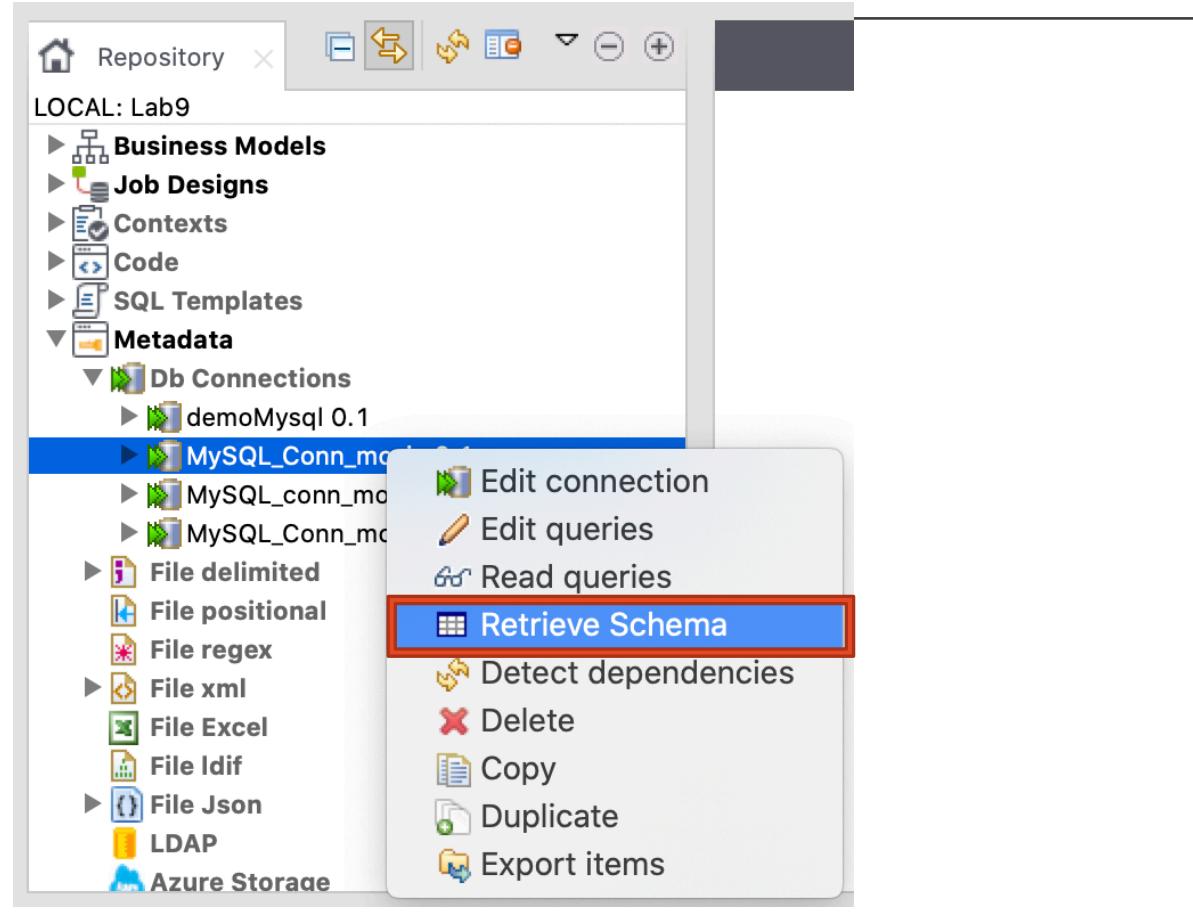


Server: sql12.freemysqlhosting.net
DB Name: sql12375315
Username: sql12375315
Password: mlejQH4MPw
Port number: 3306



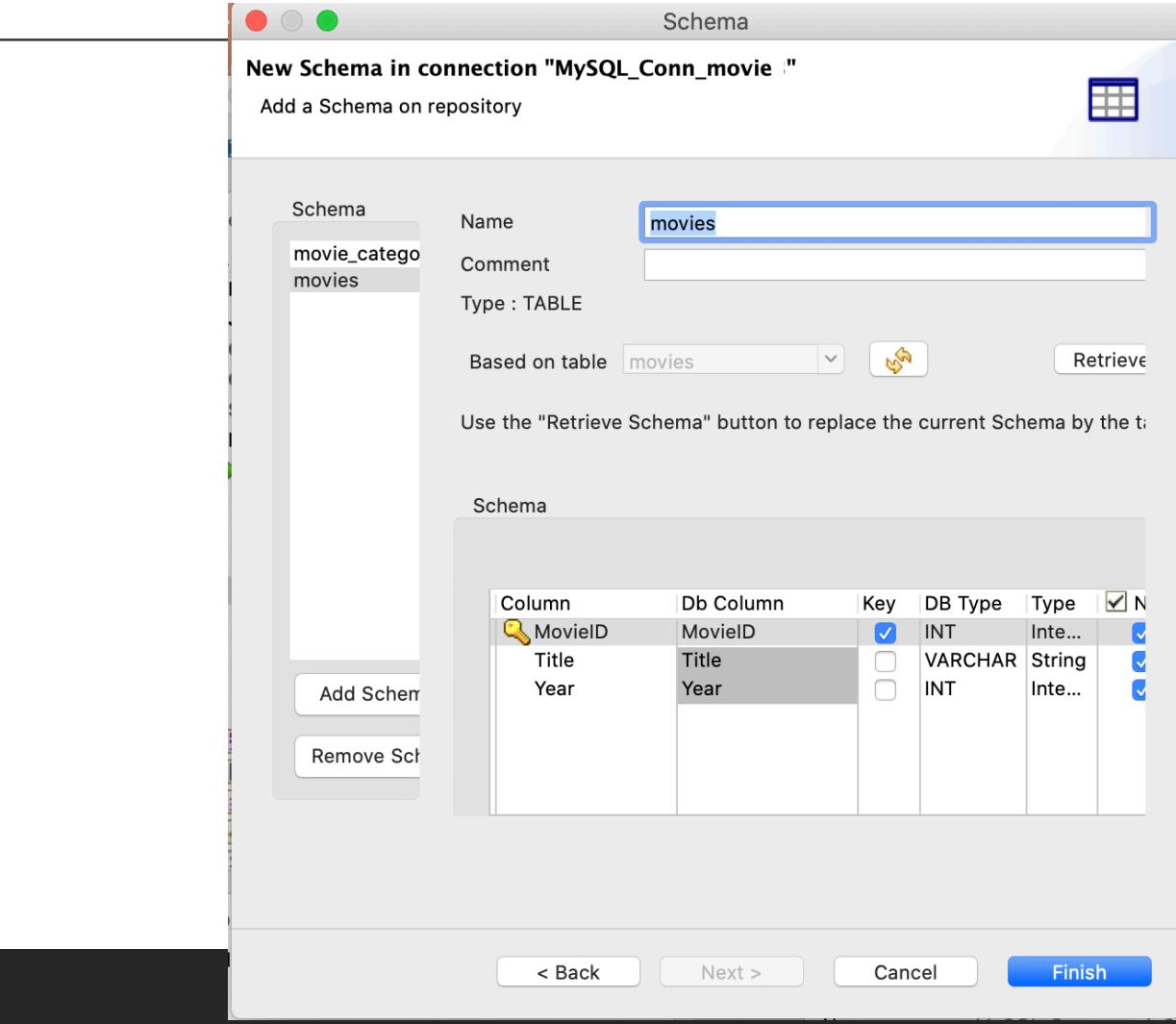
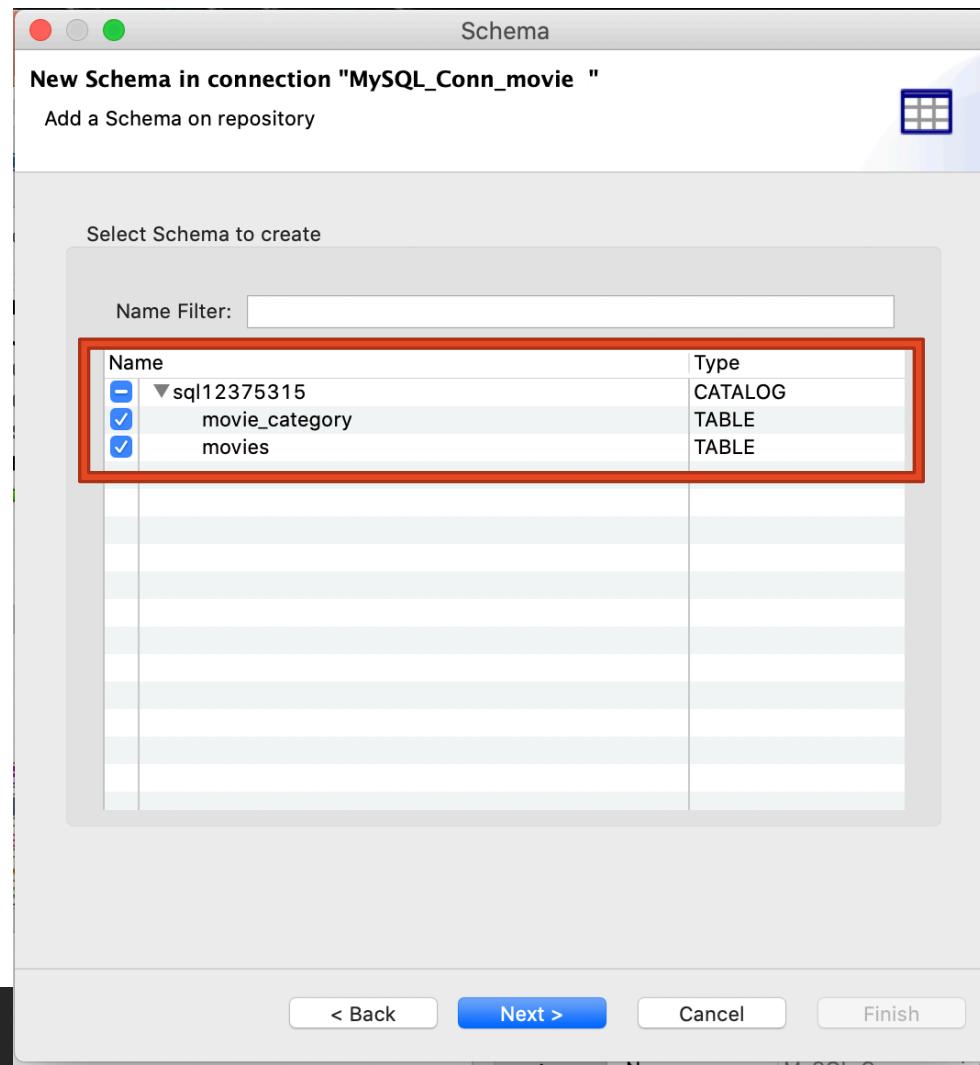
Task1

1.1 Create Metadata for DB Connection (3)



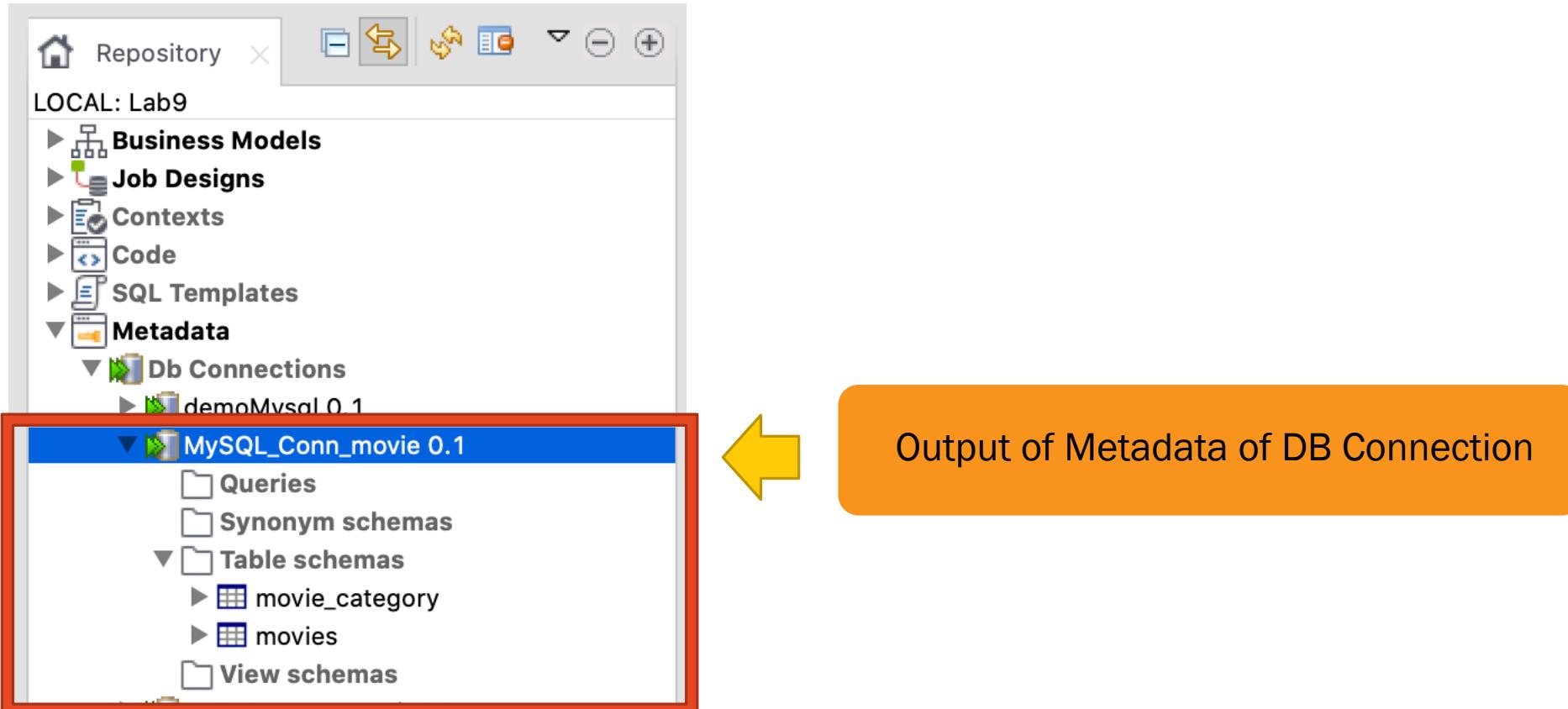
Task1

1.1 Create Metadata for DB Connection (4)



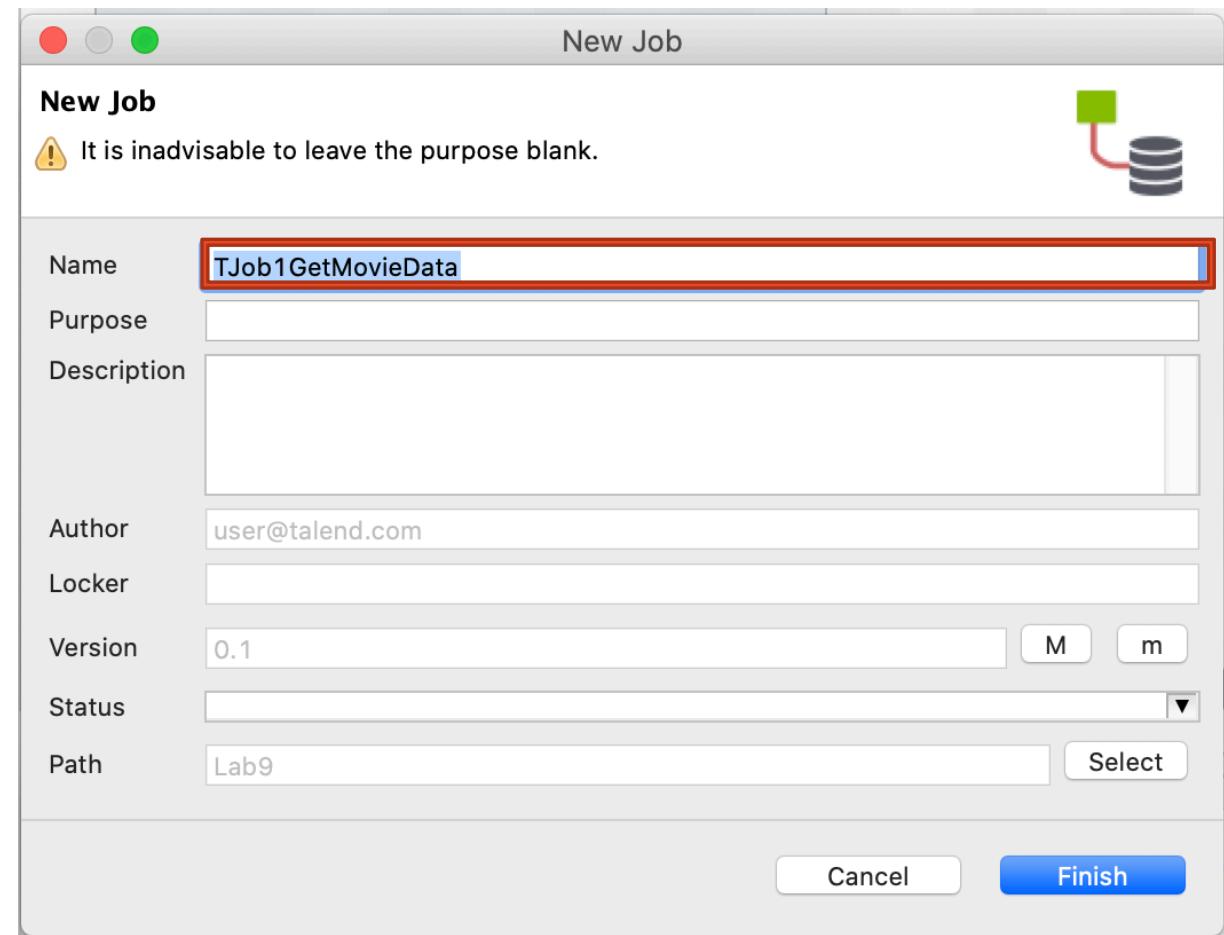
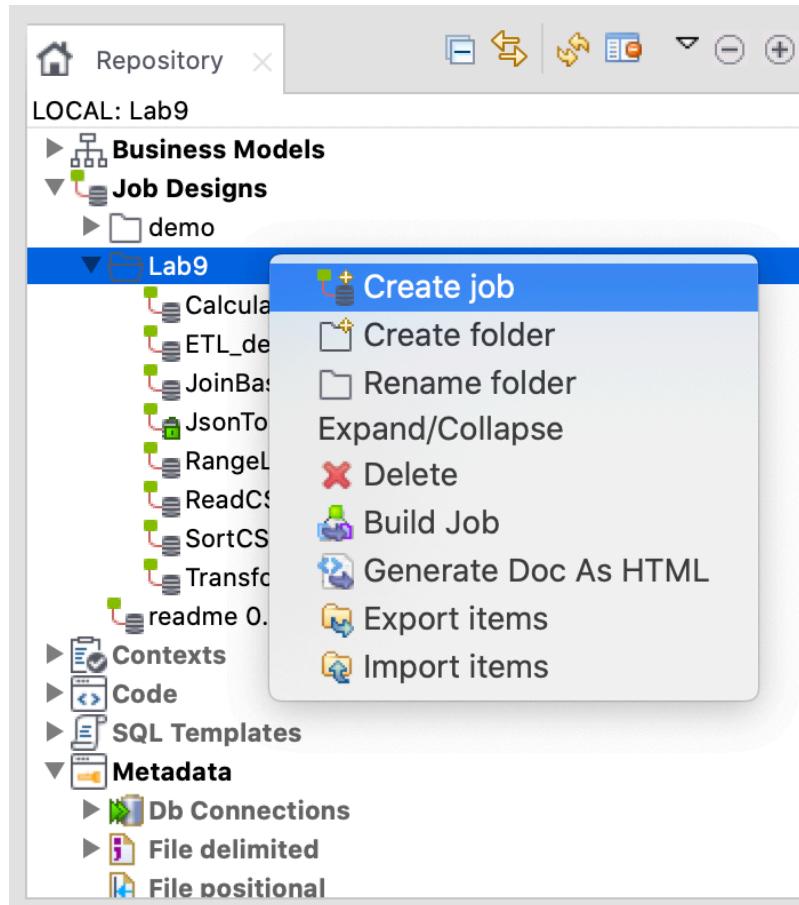
Task1

1.1 Create Metadata for DB Connection (5)



Task1

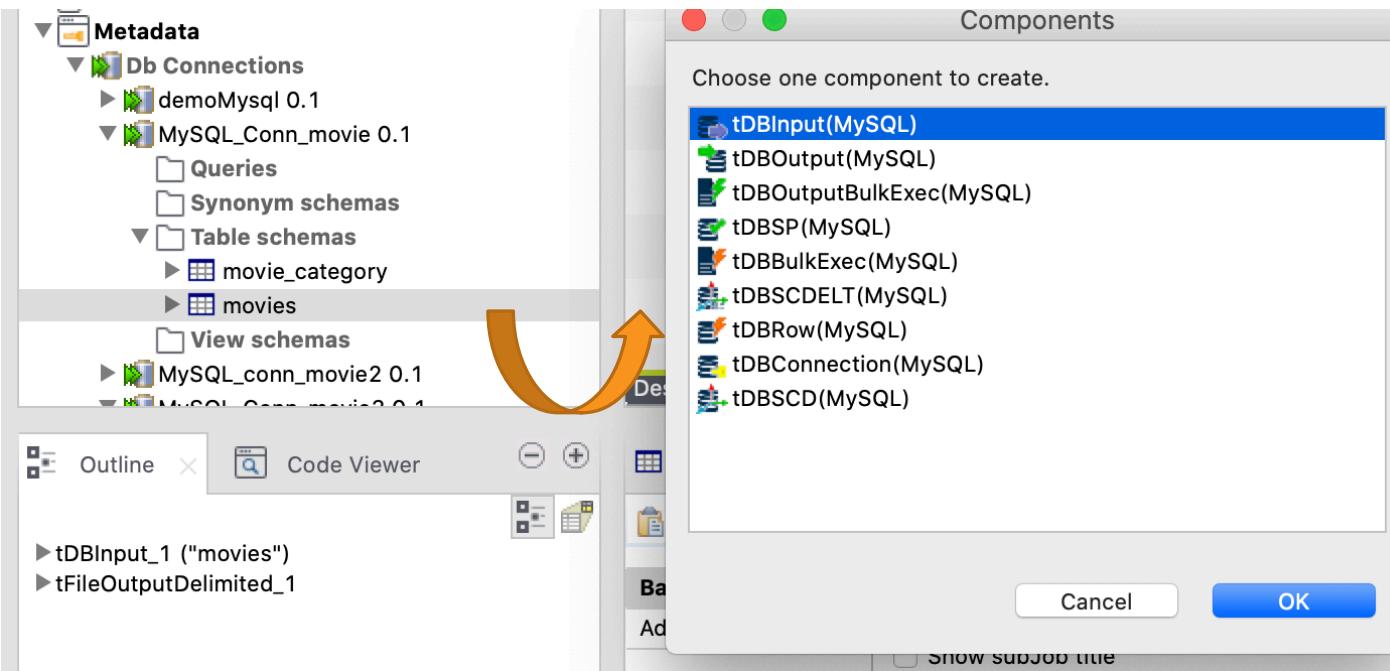
1.2 Create TJob#1: get movie data from RDB and convert data to CSV file (1)



Task1

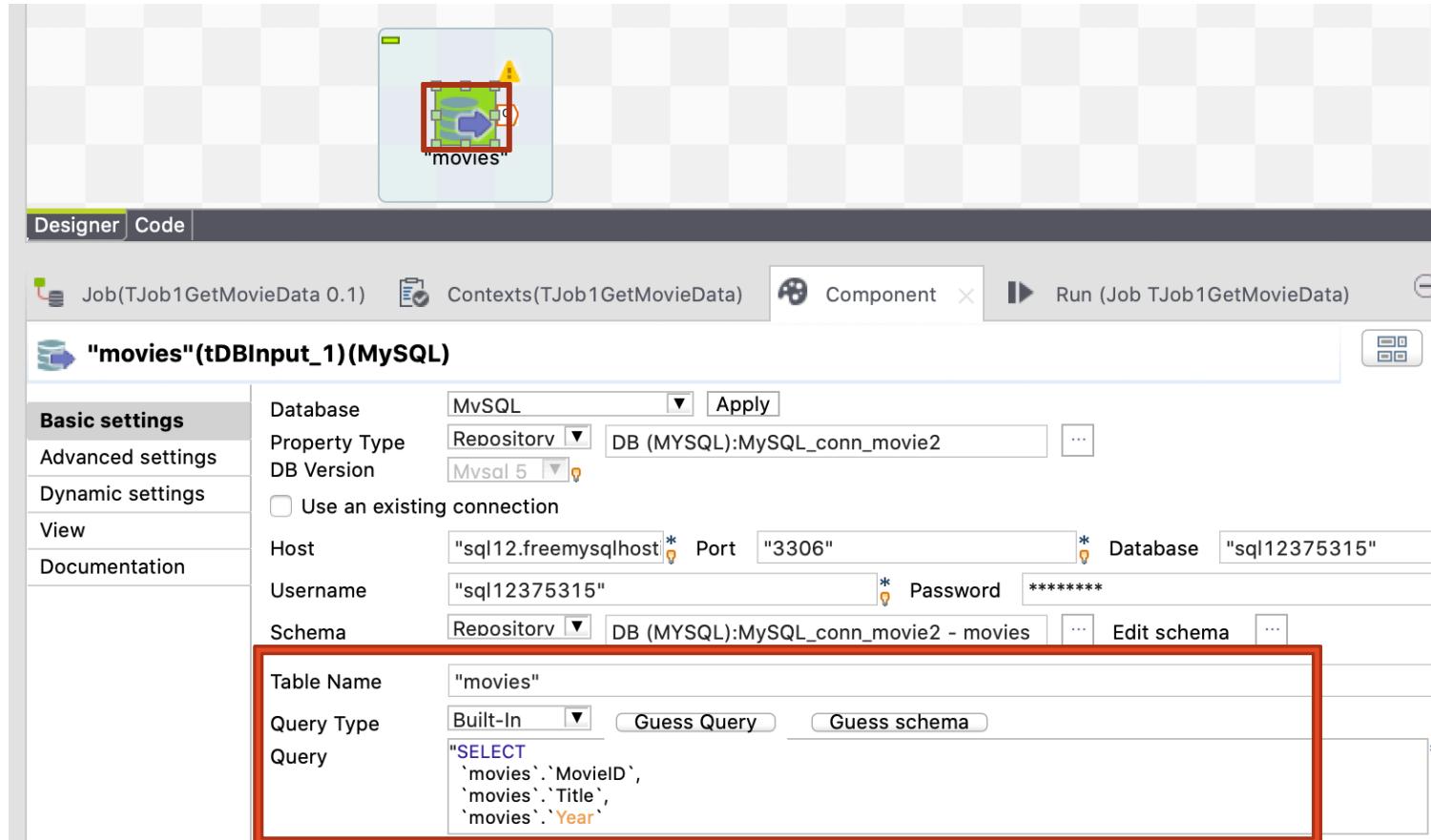
1.2 Create TJob#1: get movie data from RDB and convert data to CSV file (2)

- Drag Metadata of table “movies” under “MySQL_Conn_movie” to Design Workspace



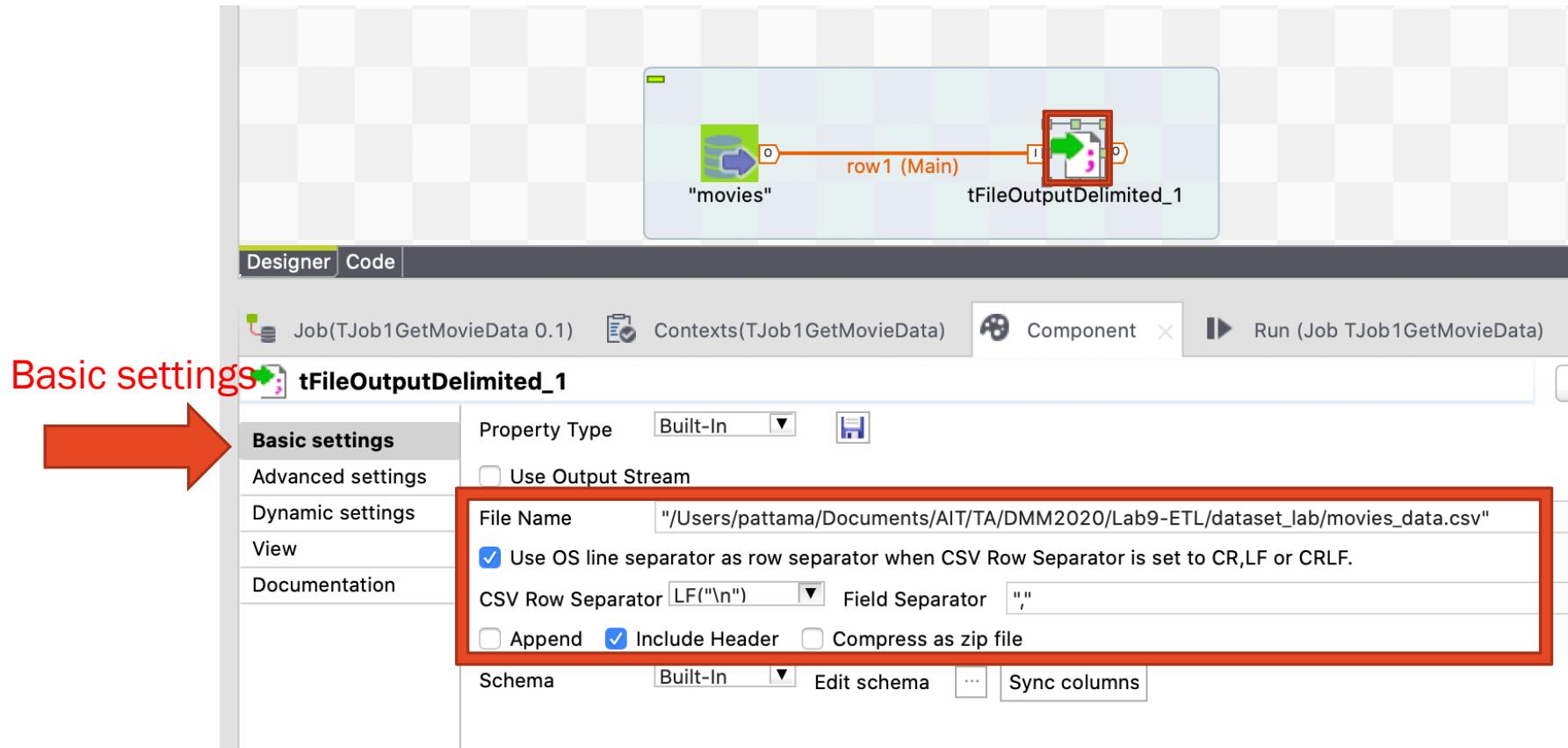
Task1

1.2 Create TJob#1: get movie data from RDB and convert data to CSV file (2)



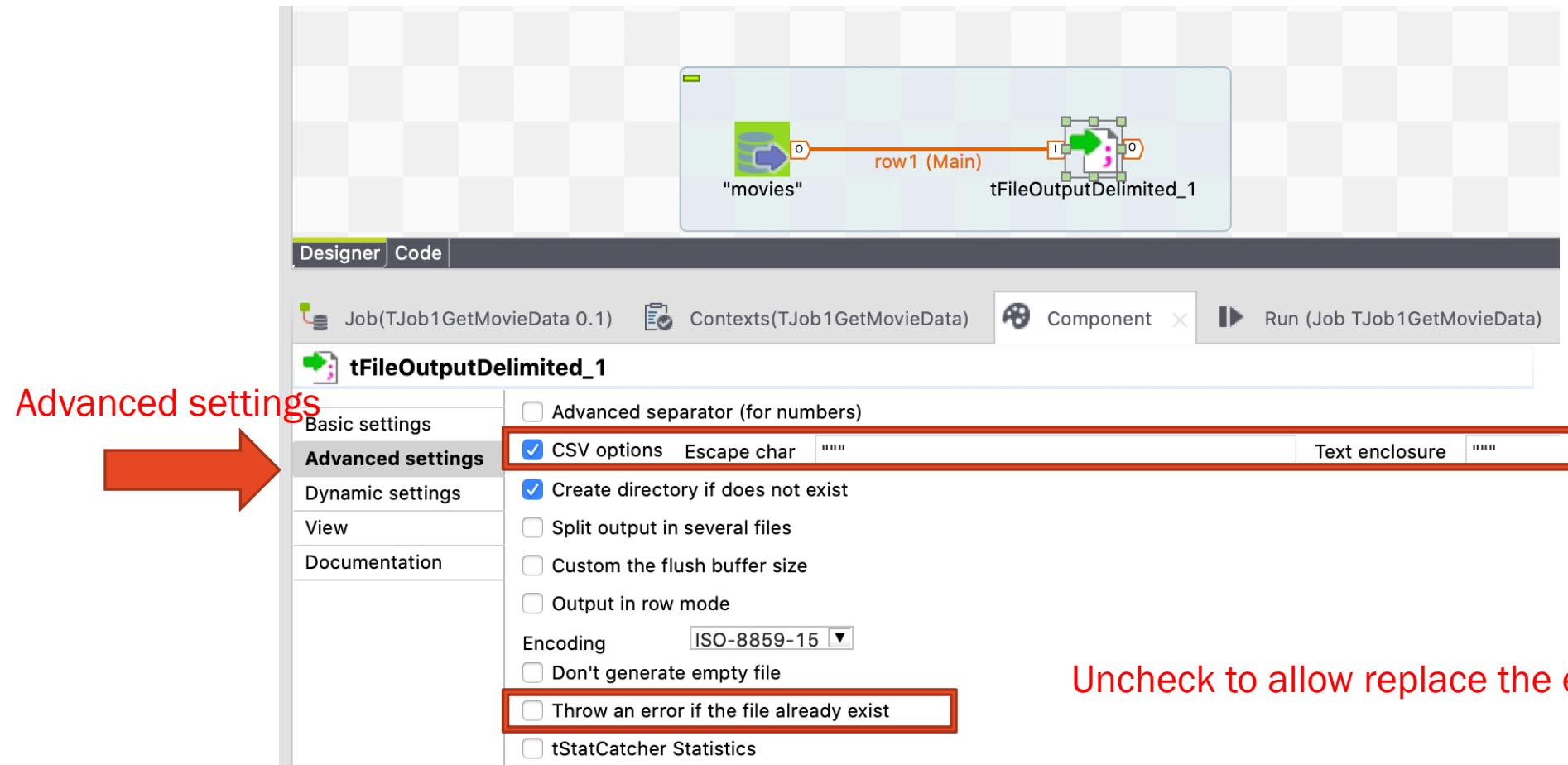
Task1

1.2 Create TJob#1: get movie data from RDB and convert data to CSV file (3)



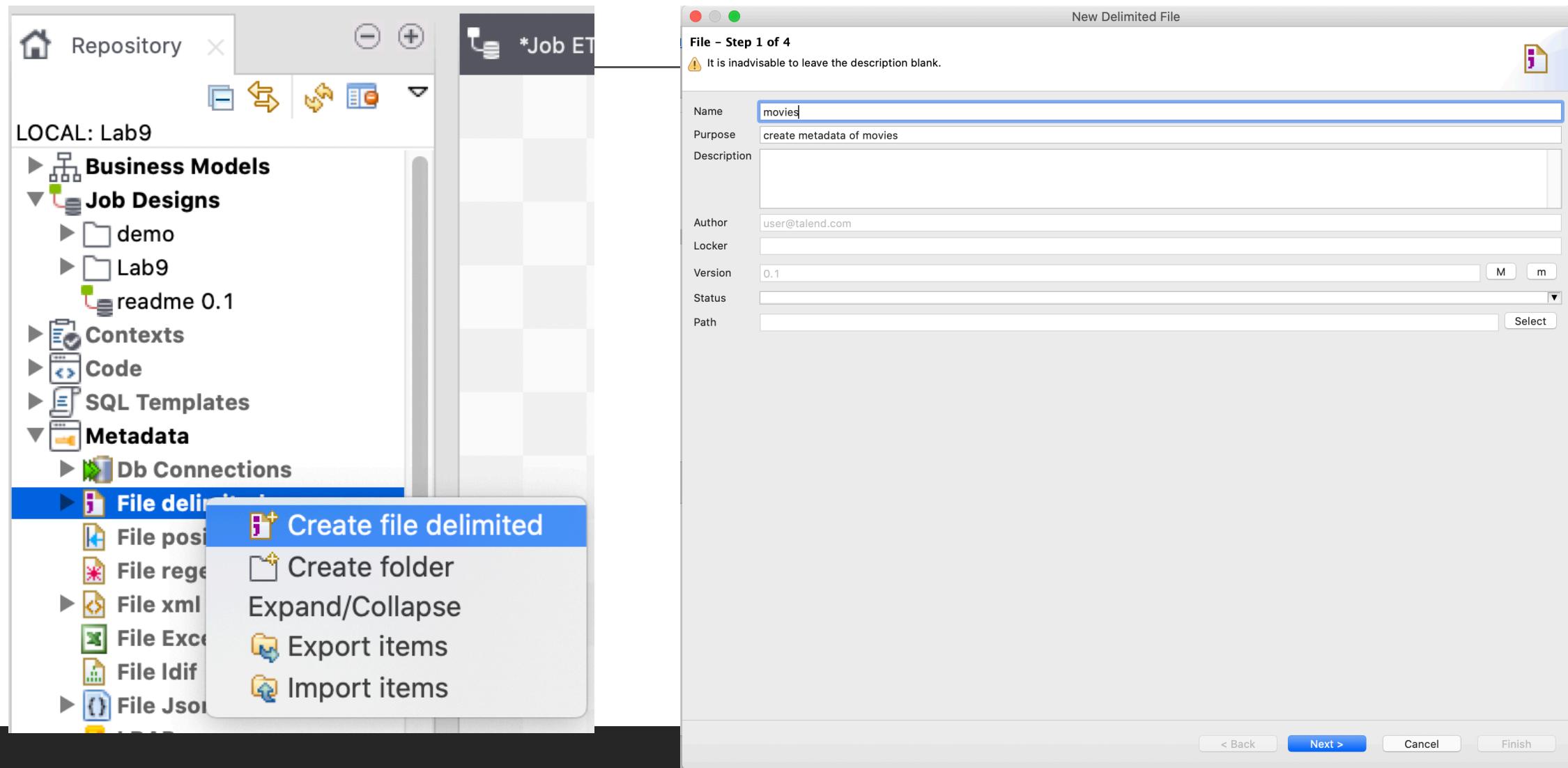
Task1

1.2 Create TJob#1: get movie data from RDB and convert data to CSV file (4)



Task1

1.3 Create CSV Metadata for Movies (1)



Task1

1.3 Create CSV Metadata for Movies (2)

Edit an existing Delimited File

File – Step 2 of 3

Edit an existing Metadata File on repository
Update the path of the file and the format settings

File Settings

Server: Localhost 127.0.0.1

File: /Users/pattama/Documents/AIT/TA/DMM2020/Lab9-ETL/dataset/movies_data.csv

Format: UNIX

File Viewer

```
MovielID,Title,Year
"1","Toy Story (1995)","1995"
"2","Jumanji (1995)","1995"
"3","Grumpier Old Men (1995)","1995"
"4","Waiting to Exhale (1995)","1995"
"5","Father of the Bride Part II (1995)","1995"
"6","Heat (1995)","1995"
"7","Sabrina (1995)","1995"
"8","Tom and Huck (1995)","1995"
"9","Sudden Death (1995)","1995"
"10","GoldenEye (1995)","1995"
"11","American President, The (1995)","1995"
"12","Dracula: Dead and Loving It (1995)","1995"
"13","Balto (1995)","1995"
"14","Nixon (1995)","1995"
"15","Cutthroat Island (1995)","1995"
"16","Casino (1995)","1995"
"17","Sense and Sensibility (1995)","1995"
"18","Four Rooms (1995)","1995"
"19","Ace Ventura: When Nature Calls (1995)","1995"
```

< Back Next > Cancel Finish

Edit an existing Delimited File

File – Step 3 of 3

Update an existing Metadata File on repository
Define the setting of the parse job

File Settings

Encoding: UTF-8

Field Separator: Comma Corresponding Character: " "

Row Separator: Standard EOL Corresponding Character: "\n"

Rows To Skip

If any rows must be ignored, specify the following parameters

Header: 1

Footer:

Skip empty row

Limit Of Rows

If the number of lines must be limited, specify this number

Limit:

Escape Char Settings

CSV Delimited

Escape Char: Empty

Text Enclosure: ""

Split row before field

Preview Output

Set heading row as column names Refresh Preview

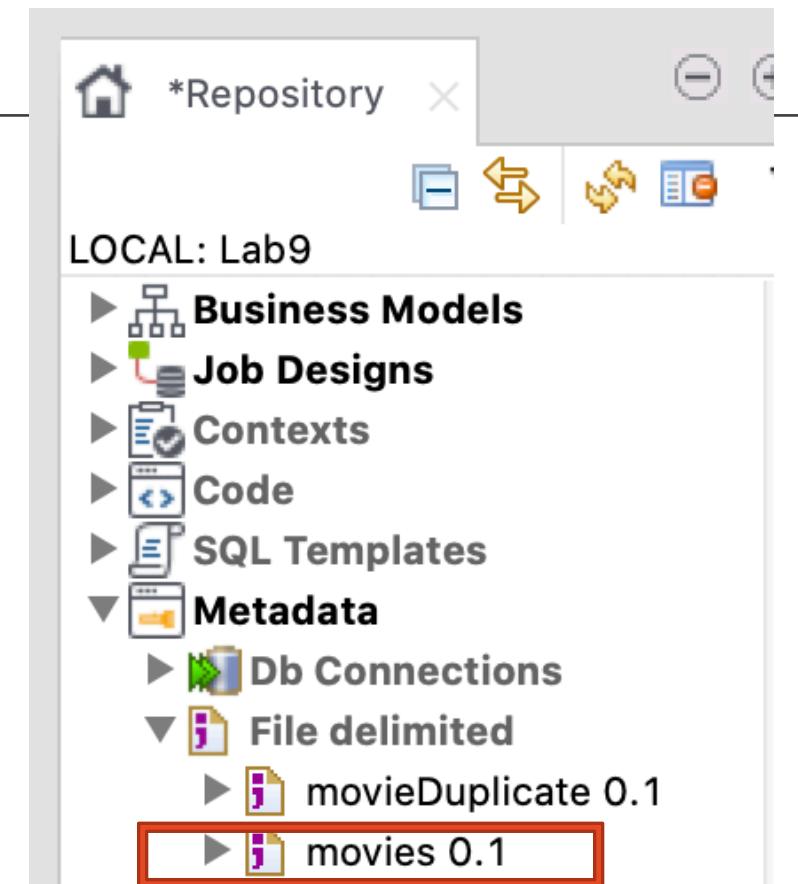
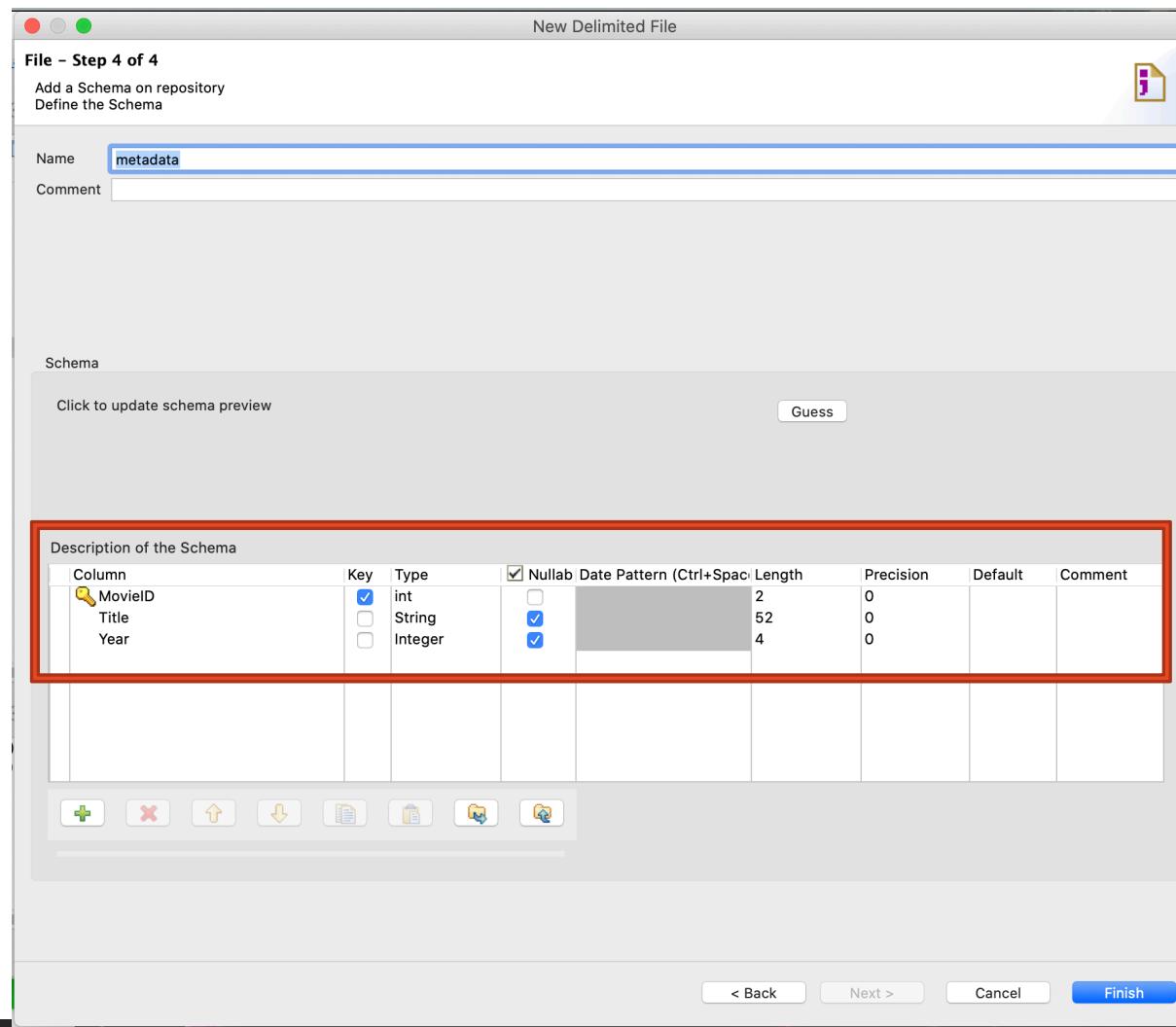
MovielID	Title	Year
1	Toy Story (1995)	1995
2	Jumanji (1995)	1995
3	Grumpier Old Men (1995)	1995
4	Waiting to Exhale (1995)	1995
5	Father of the Bride Part II (1995)	1995
6	Heat (1995)	1995
7	Sabrina (1995)	1995
8	Tom and Huck (1995)	1995

Export as context Revert Context

< Back Next > Cancel Finish

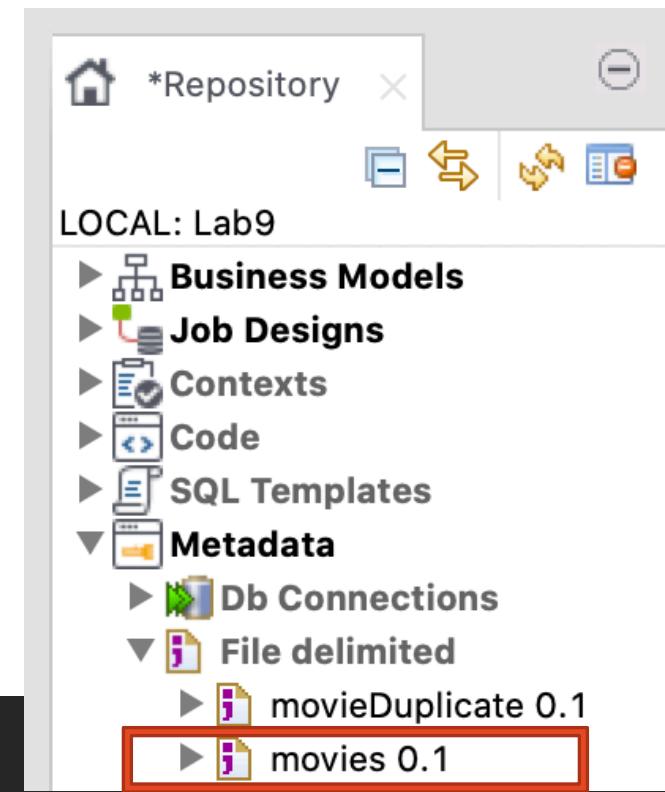
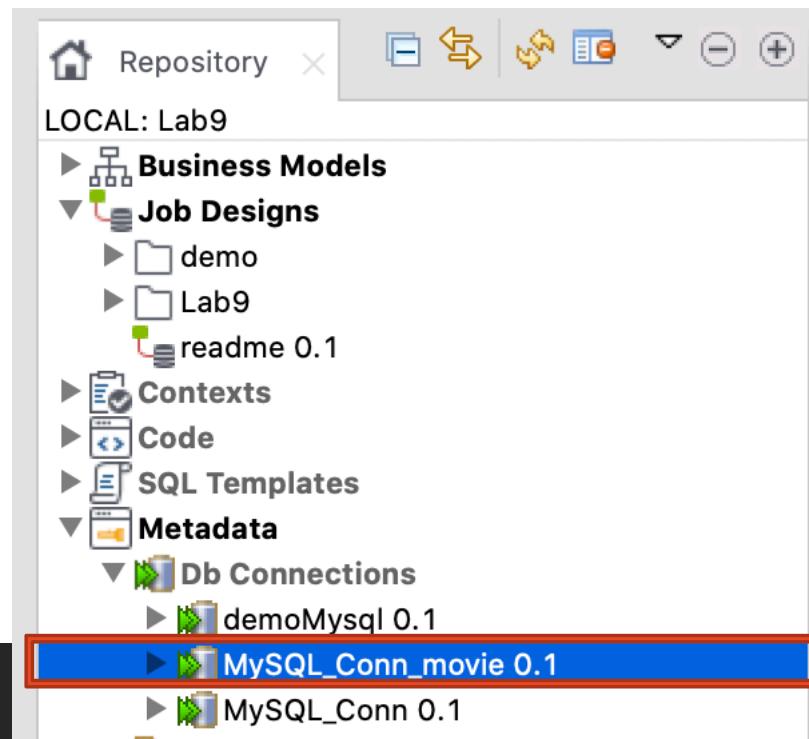
Task1

1.3 Create CSV Metadata for Movies (3)

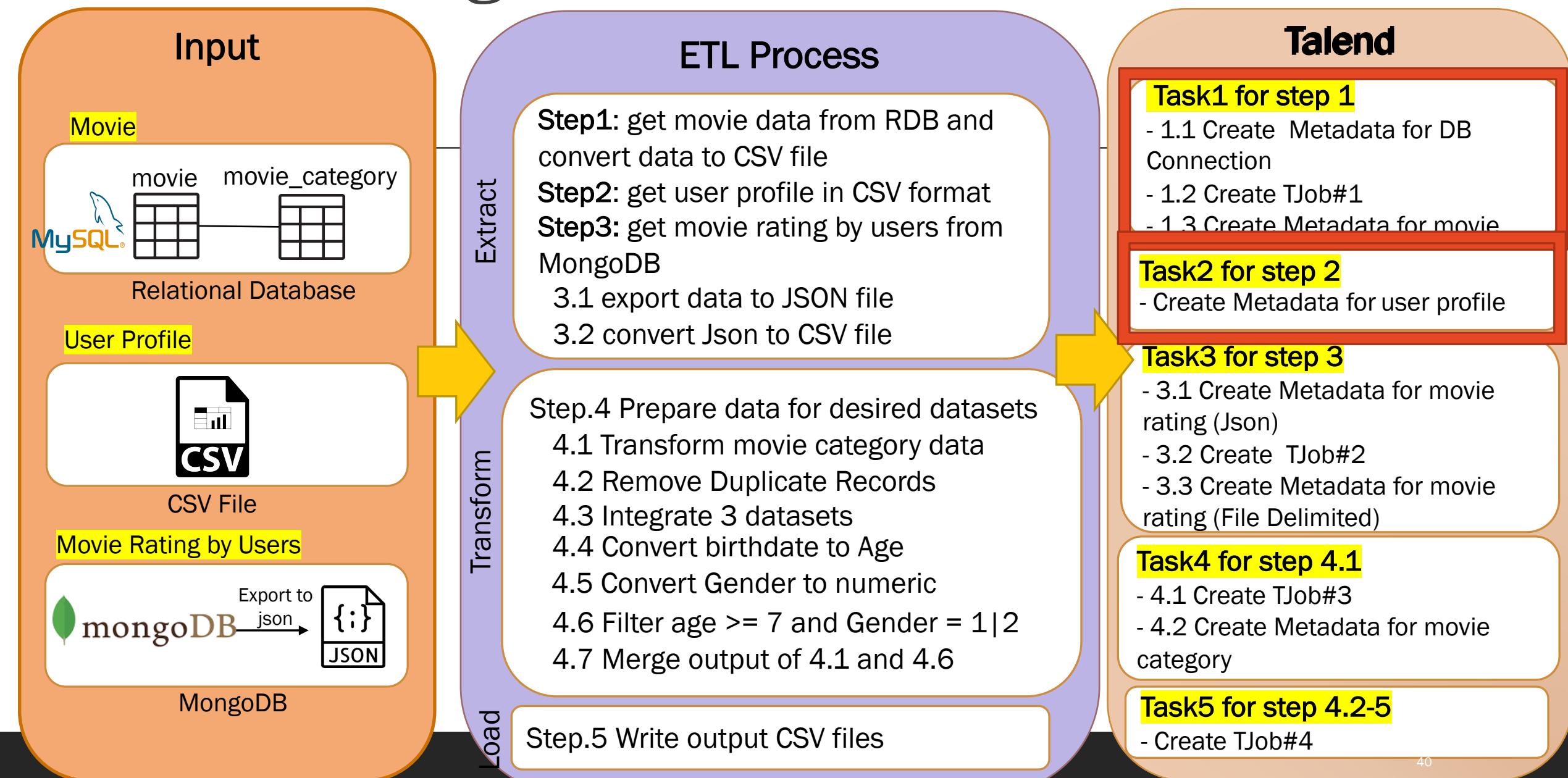


Output of Task1

1. Metadata of DB Connection “MySQL_Conn_movie”
2. Metadata of File delimited “movies”



Movie Rating Prediction





CSV File

Task2 for step 2

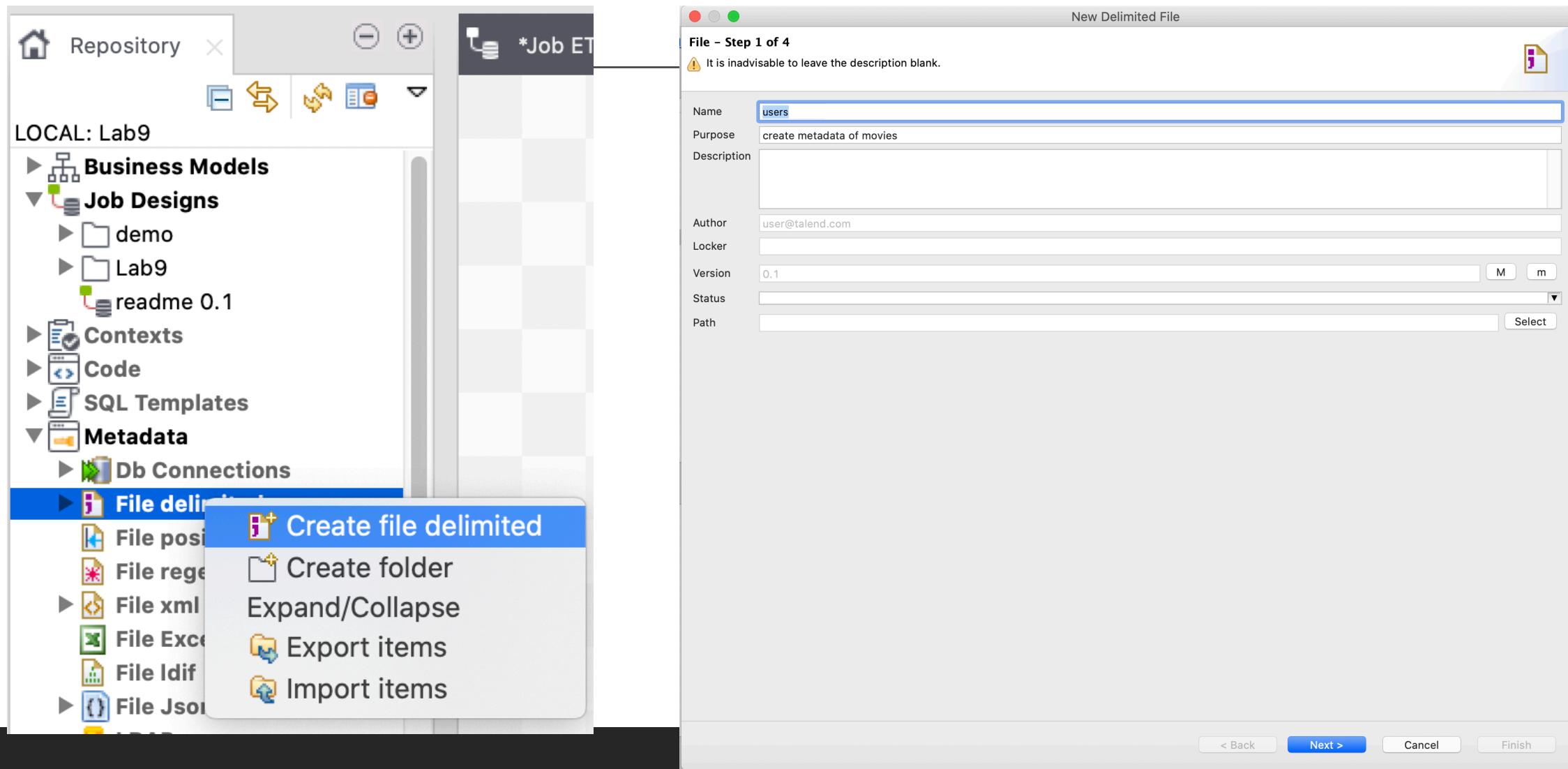
Step2: get user profile in CSV format

Task2:

Create Metadata for user profile (File Delimited)

Task2

Create CSV Metadata for User Profile (1)



Task2

Create CSV Metadata for User Profile (2)

Edit an existing Delimited File

File - Step 2 of 3
Edit an existing Metadata File on repository
Update the path of the file and the format settings

File Settings

Server: Localhost 127.0.0.1

File: /Users/pattama/Documents/AIT/TA/DMM2020/Lab9-ETL/dataset/users_data.csv

Format: UNIX

File Viewer

UserID	Gender	BirthDate	Occupation	Zipcode
1	Female	2019-05-10	10	48067
2	Male	1964-11-10	16	70072
3	Male	1995-06-27	15	55117
4	Male	1975-11-07	7	02460
5	Male	1995-05-06	20	55455
6	Female	1970-02-27	9	55117
7	Male	1985-06-05	1	06810
8	Male	1995-05-27	12	11413
9	Male	1995-03-13	17	61614
10	Female	1985-07-29	1	95370
11	Female	1995-10-01	1	04093
12	Male	1995-01-21	12	32793
13	Male	1975-06-18	1	93304
14	Male	1985-06-09	0	60126
15	Male	1995-01-02	7	22903
16	Female	1985-07-17	0	20670
17	Male	1970-11-19	1	95350
18	Female	2002-05-27	3	05825

< Back Next > Cancel **Finish**

New Delimited File

File - Step 3 of 4
Add a Metadata File on repository
Define the setting of the parse job

File Settings

Encoding: UTF-8

Field Separator: Comma

Row Separator: Standard EOL

Rows To Skip

If any rows must be ignored, specify the following parameters

Header: 1

Footer:

Skip empty row

Escape Char Settings

CSV Delimited

Escape Char: Empty

Text Enclosure: ""

Split row before field

Limit Of Rows

If the number of lines must be limited, specify this number

Limit:

Preview

Set heading row as column names Refresh Preview

UserID	Gender	BirthDate	Occupation	Zipcode
1	Female	2019-05-10	10	48067
2	Male	1964-11-10	16	70072
3	Male	1995-06-27	15	55117
4	Male	1975-11-07	7	02460
5	Male	1995-05-06	20	55455
6	Female	1970-02-27	9	55117
7	Male	1985-06-05	1	06810
8	Male	1995-05-27	12	11413

Output

Set heading row as column names Refresh Preview

Export as context Revert Context

< Back Next > Cancel **Finish**

Task2

Create CSV Metadata for User Profile (3)

The screenshot shows the Talend Data Integration environment. On the left, a 'New Delimited File' dialog is open, Step 4 of 4, titled 'File - Step 4 of 4'. It shows a schema for a 'metadata' file. A red box highlights the 'Description of the Schema' table, which includes columns for Column, Key, Type, Nullability, Date Pattern, Precision, Default, and Comment. The 'BirthDate' row is selected, showing a date pattern of 'yyyy-MM-dd' with a length of 10. On the right, the Talend Repository browser displays a tree structure under 'LOCAL: Lab9'. A red box highlights the 'users 0.1' item under the 'File delimited' node.

Description of the Schema

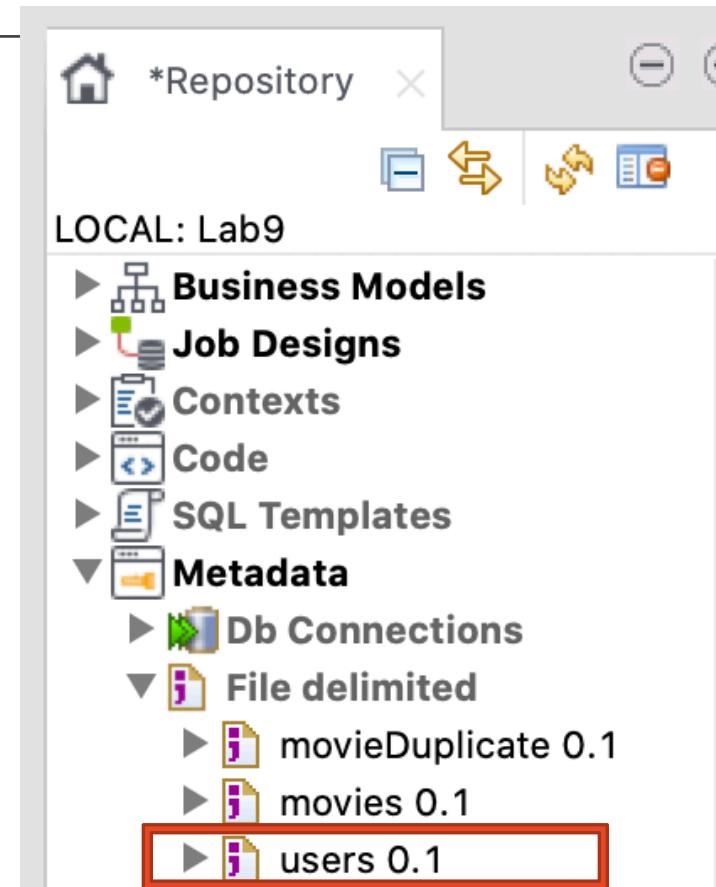
Column	Key	Type	Nullability	Date Pattern (Ctrl+Space Length)	Precision	Default	Comment
UserID	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		2	0	
Gender	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		6	0	
BirthDate	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd"	10	0	
Occupation	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		2	0	

***Repository**

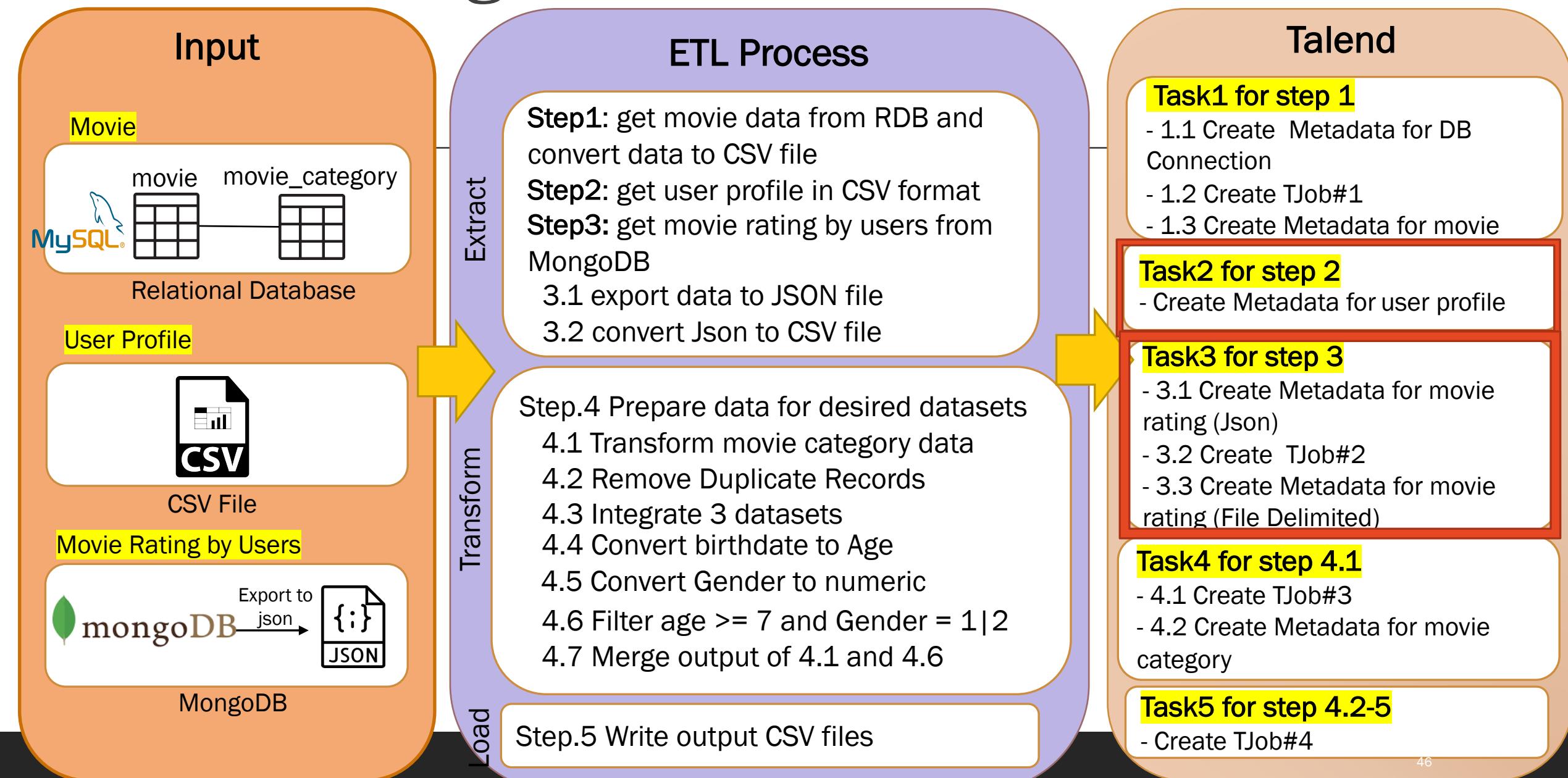
- Business Models
- Job Designs
- Contexts
- Code
- SQL Templates
- Metadata
 - Db Connections
 - File delimited
 - movieDuplicate 0.1
 - movies 0.1
 - users 0.1**

Output of Task2

-
1. Metadata of File delimited “users”



Movie Rating Prediction



Task3 for step 3

Movie Rating by Users



Step3: get movie rating by users from MongoDB

Task3:

- Export data from MongoDB to JSON file
- Convert Json to CSV file
 - 3.1 Create Metadata for movie rating (Json)
 - 3.2 Create TJob#2
 - 3.3 Create Metadata for movie rating (File Delimited)

Atlas MongoDB

Talend

Task3

Export data from MongoDB to JSON file

```
{  
    "_id" : ObjectId("5faab9f2b3fcdbbf3a340cf"),  
    "UserID" : 1,  
    "MovieID" : 1193,  
    "Rating" : 5,  
    "Timestamp" : "2015-05-14 08:48:14"  
}  
{  
    "_id" : ObjectId("5faab9f2b3fcdbbf3a340d0"),  
    "UserID" : 1,  
    "MovieID" : 661,  
    "Rating" : 3,  
    "Timestamp" : "2015-11-08 12:41:07"  
}  
{  
    "_id" : ObjectId("5faab9f2b3fcdbbf3a340d1"),  
    "UserID" : 1,  
    "MovieID" : 914,  
    "Rating" : 3,  
    "Timestamp" : "2015-02-27 19:56:12"  
}  
{  
    "_id" : ObjectId("5faab9f2b3fcdbbf3a340d2"),  
    "UserID" : 1,  
    "MovieID" : 3408,  
    "Rating" : 4,  
    "Timestamp" : "2015-08-05 06:46:28"  
}  
{  
    "_id" : ObjectId("5faab9f2b3fcdbbf3a340d3"),  
    "UserID" : 2,  
    "MovieID" : 1357,  
    "Rating" : 5,  
    "Timestamp" : "2015-10-18 15:24:38"  
}
```

Atlas MongoDB

movie rating collection in MongoDB

Task3

Export data from MongoDB to JSON file

- Export Data from MongoDB to JSONArray via Terminal

Example:

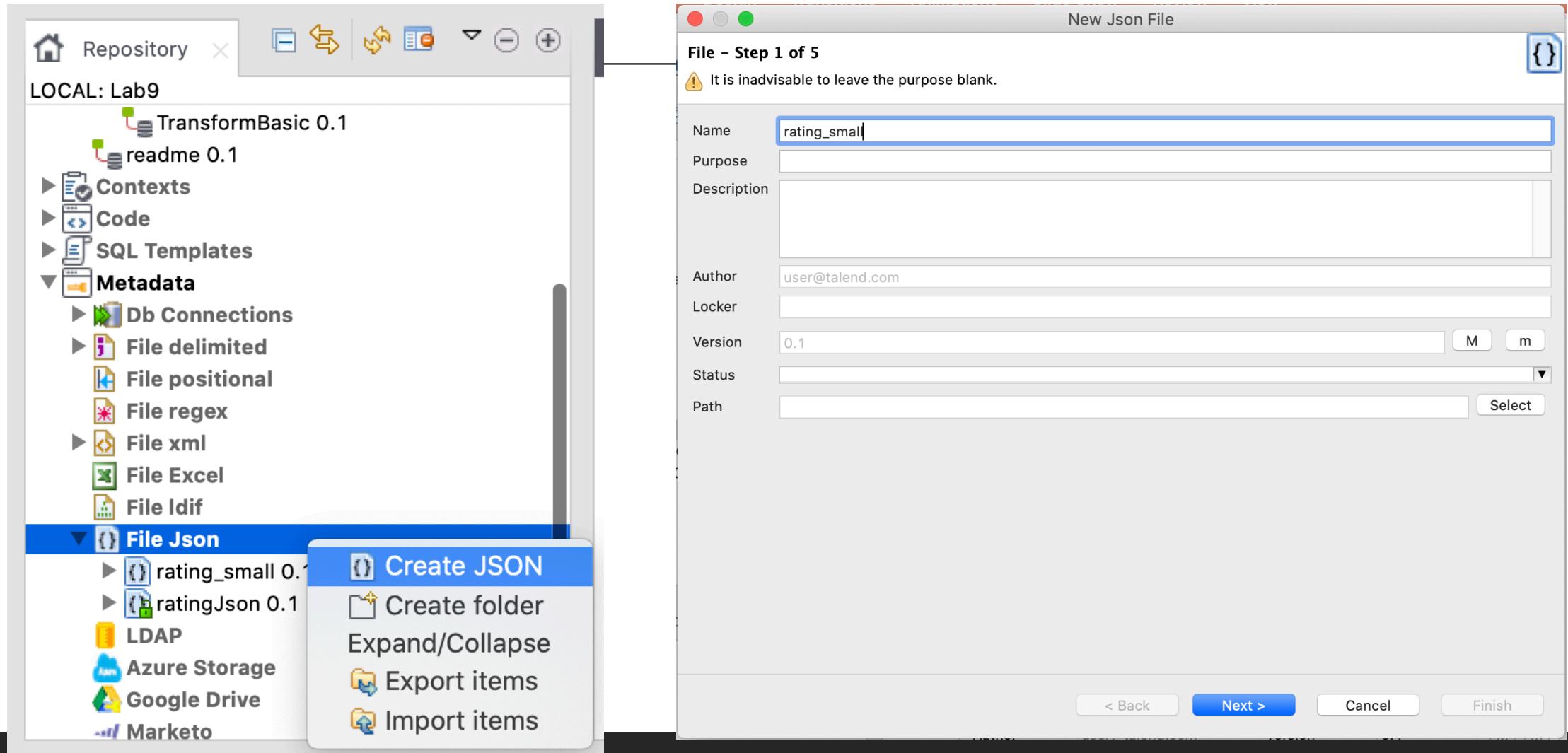
```
mongoexport --uri="mongodb://localhost:27017/dmm2020" --collection=ratings --jsonArray  
--out=/Users/pattama/Documents/AIT/TA/DMM2020/Lab9-ETL/datasets/ratings.json
```

- Export Data from MongoDB to JSONArray via MongoCompass

URL: <https://youtu.be/L6usQ1KbSRs>

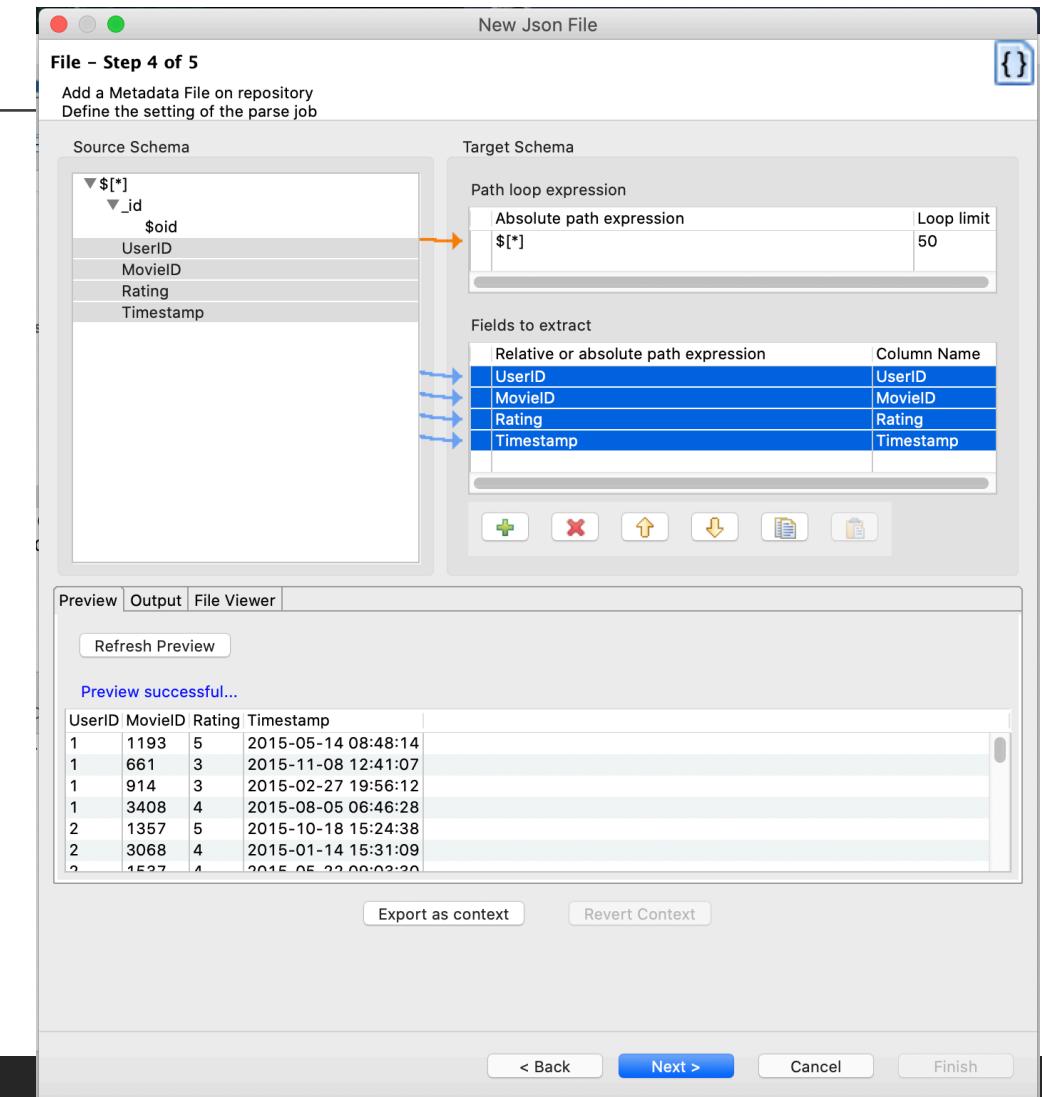
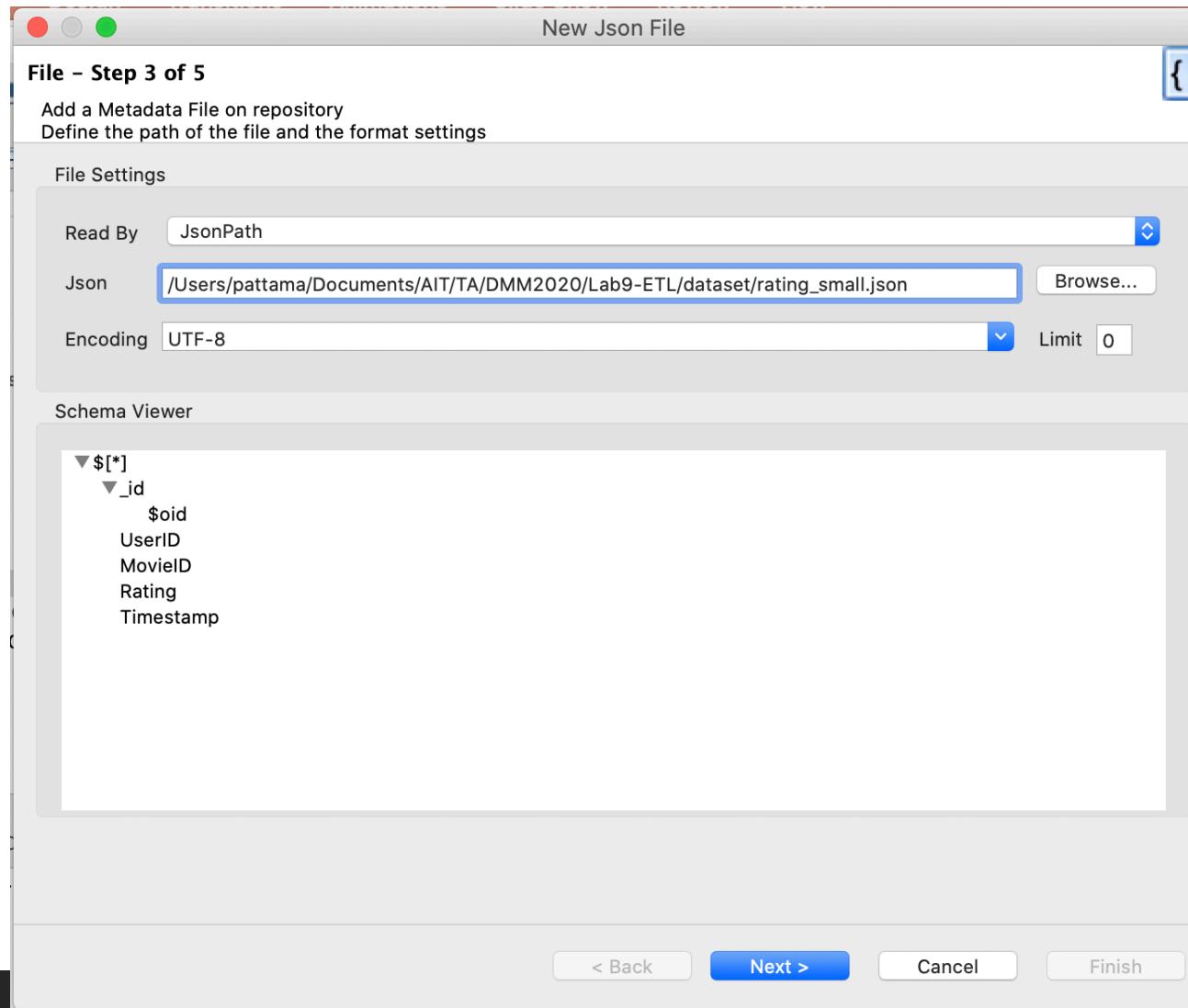
Task3

3.1 Create Metadata for movie rating (Json) (1)



Task3

3.1 Create Metadata for movie rating (Json)(2)



Task3

3.1 Create Metadata for movie rating (Json)(3)

The screenshot shows the Talend Data Integration environment. On the left, a 'New Json File' dialog is open, Step 5 of 5, titled 'File - Step 5 of 5'. It shows a schema for a 'metadata' file with four columns: UserID (Integer), MovieID (Integer), Rating (Integer), and Timestamp (Date). The 'Timestamp' column has a 'Nullab Date Pattern (Ctrl+Shift+P)' set to 'yyyy-MM-dd HH:mm:ss'. A red box highlights this pattern. Below the table, the text 'Date pattern = "yyyy-MM-dd HH:mm:ss"' is displayed. At the bottom are buttons for '< Back', 'Next >', 'Cancel', and 'Finish'. On the right, a 'Repository' browser window is open, showing a local repository named 'Lab9'. Under 'File Json', two items are listed: 'rating_small 0.1' and 'ratings_json 0.1', with 'rating_small 0.1' also highlighted by a red box.

New Json File

File - Step 5 of 5

Add a Schema on repository
Define the Schema

Name: metadata

Comment:

Schema

Click to update schema preview

Guess

Description of the Schema

Column	Key	Type	Nullab Date Pattern (Ctrl+Shift+P)	Length	Precision	Default	Comment
UserID	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>	3	0		
MovieID	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>	4	0		
Rating	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>	1	0		
Timestamp	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd HH:mm:ss"	19	0	

Date pattern = "yyyy-MM-dd HH:mm:ss"

< Back Next > Cancel Finish

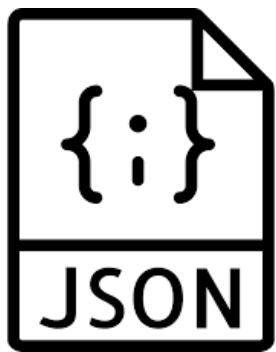
*Repository

LOCAL: Lab9

- File delimited
- File positional
- File regex
- File xml
- File Excel
- File Idif
- ▼ { } File Json
 - { } rating_small 0.1
 - { } ratings_json 0.1
- LDAP
- Azure Storage

Task3

3.2 Create TJob#2: Convert Json to CSV File



Json Metadata (tInputFileJSON)



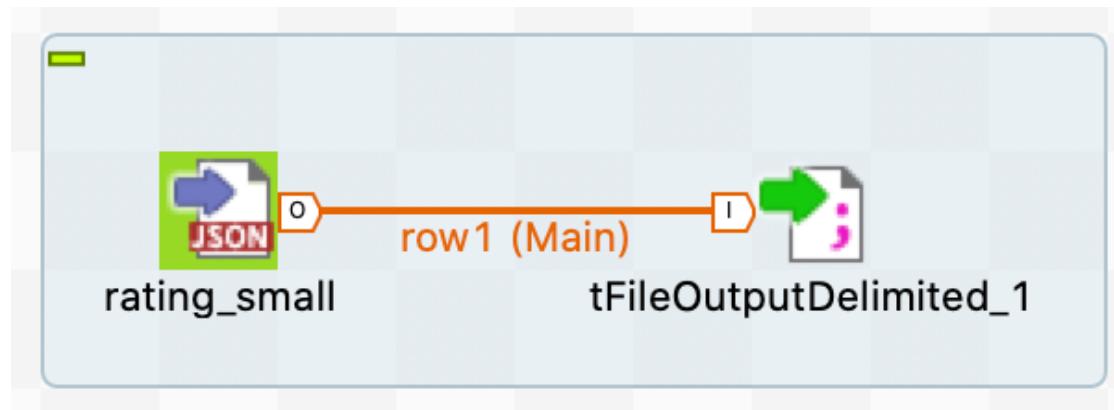
ratings.csv

Task3

3.2 Create TJob#2: Convert Json to CSV File

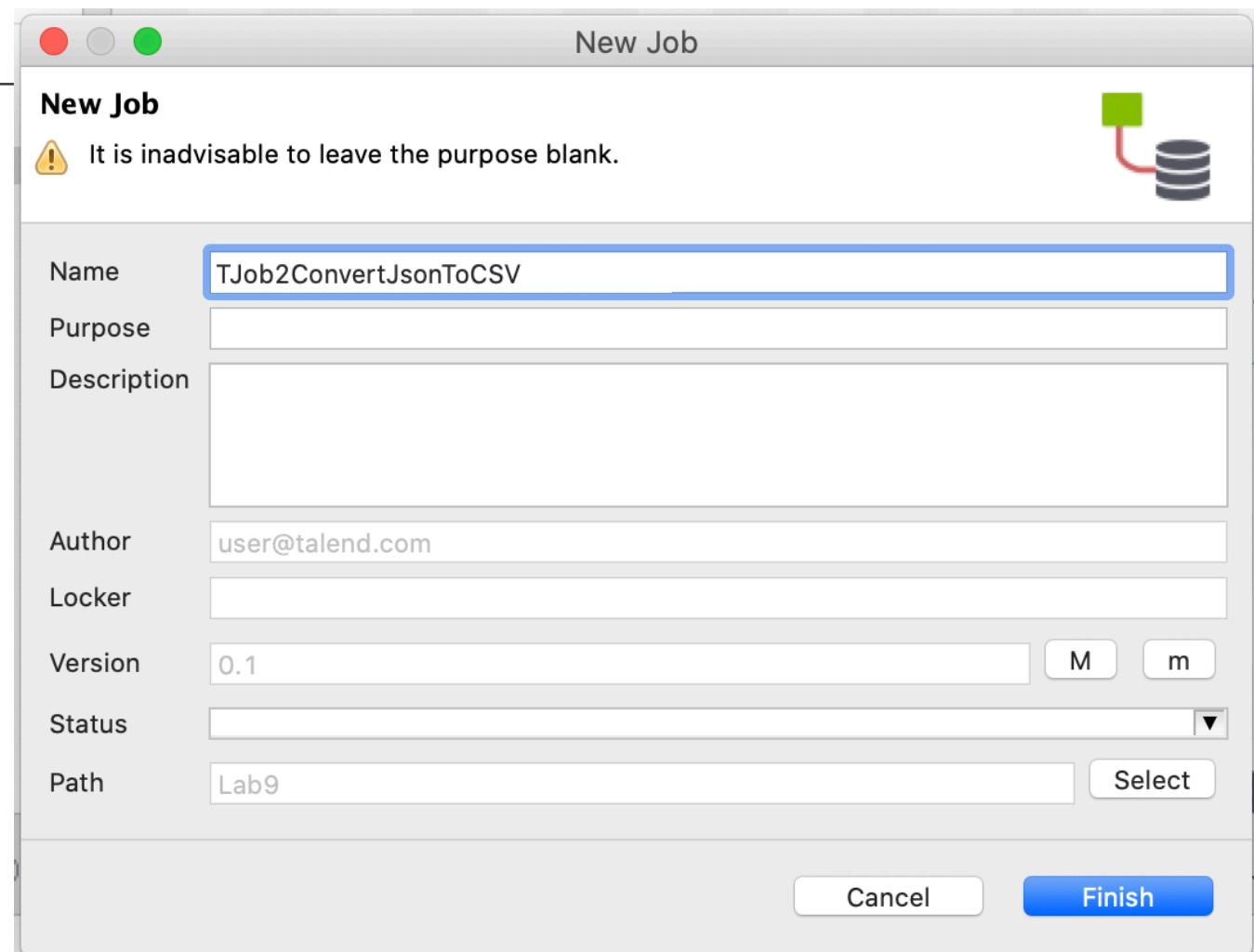
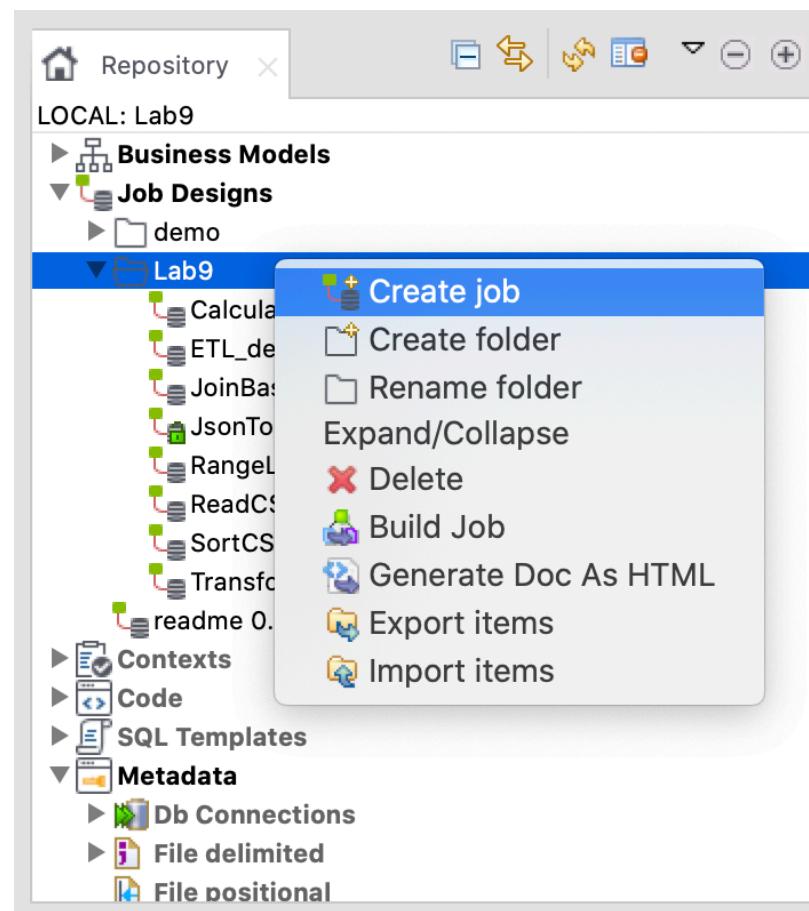
Purpose : Creating a Job to extract json data and convert to CSV file.

Components: tFileInputJson , tFileOutputDelimited



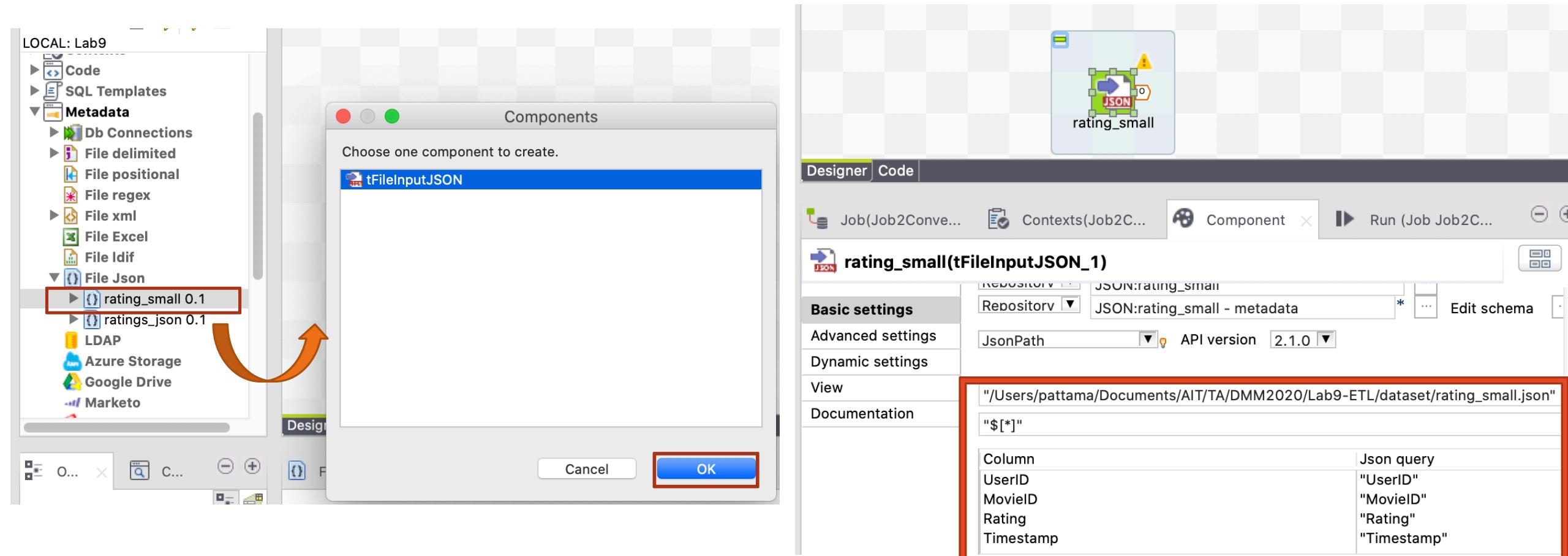
Task3

3.2 Create TJob#2: Convert Json to CSV File (1)



Task3

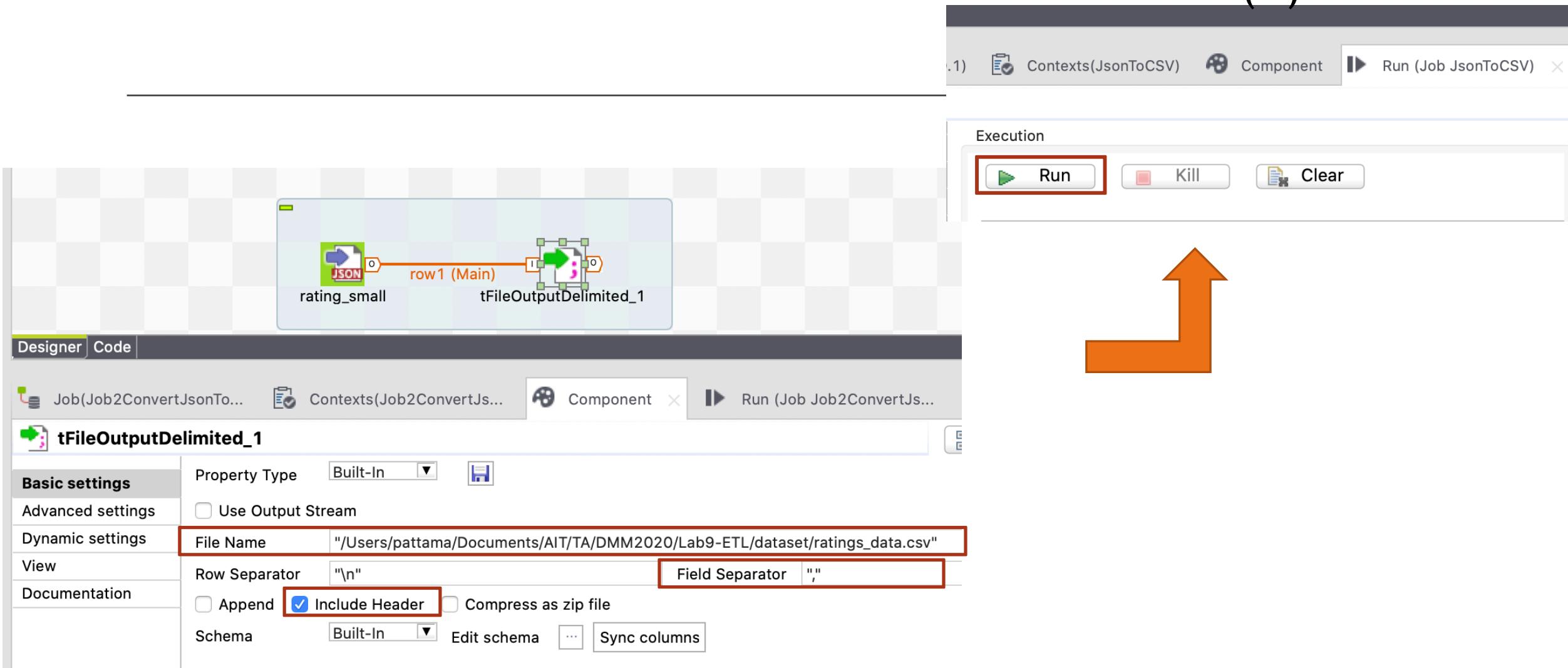
3.2 Create TJob#2: Convert Json to CSV File (2)



Drag ratings_json to “design workspace” area and click “OK”

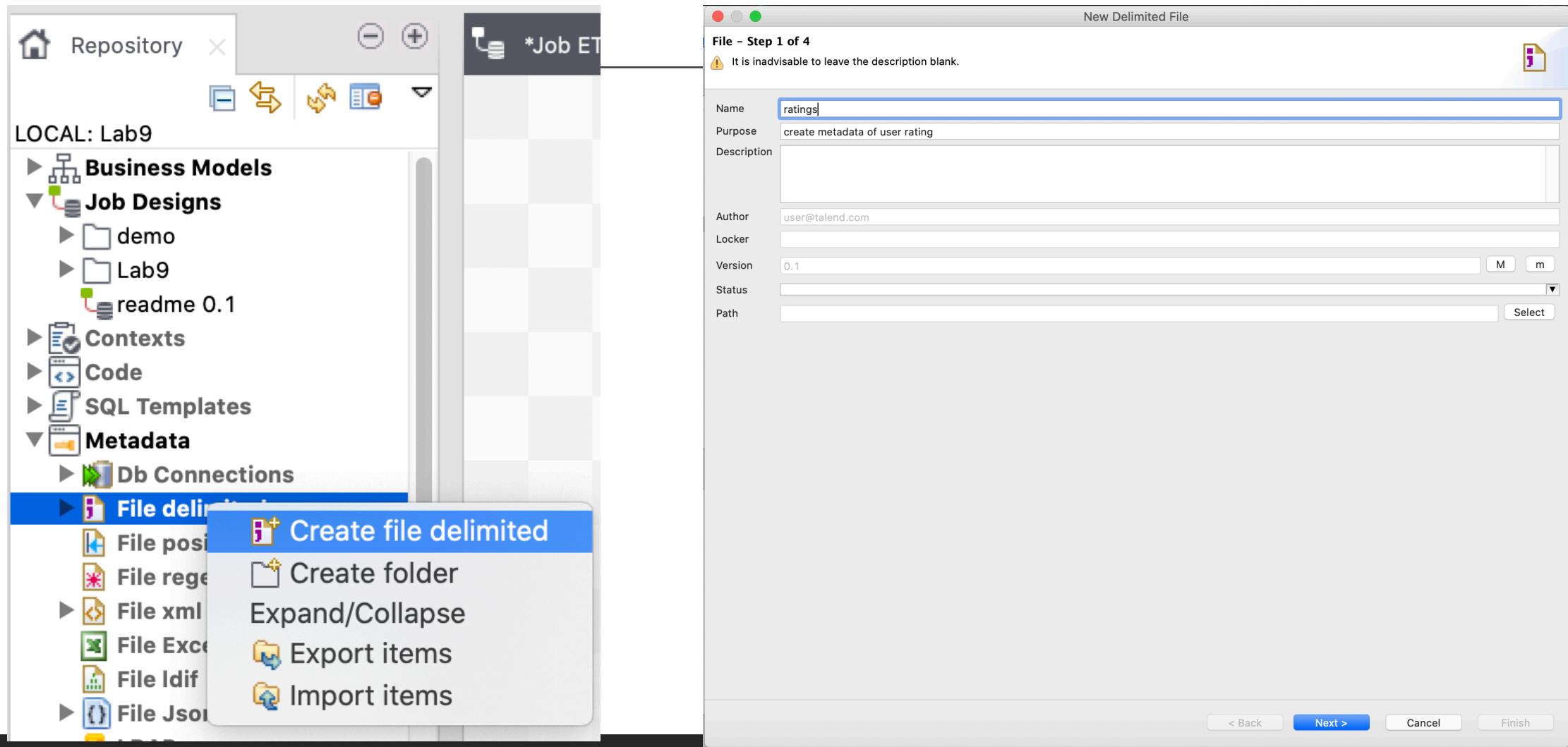
Task3

3.2 Create TJob#2: Convert Json to CSV File (3)



Task3

3.3 Create Metadata for movie rating (File Delimited)(1)



Task3

3.3 Create Metadata for movie rating (File Delimited)(2)

The image shows two side-by-side windows for creating metadata for a CSV file. Both windows have a title bar 'New Delimited File' and a header 'File - Step 2 of 4'.

Left Window (Step 2 of 4):

- File Settings:**
 - Server: Localhost 127.0.0.1
 - File: /Users/pattama/Documents/AIT/TA/DMM2020/Lab9-ETL/dataset/ratings_data.csv
 - Format: UNIX
- File Viewer:** Displays the first few lines of the CSV file:

```
UserID,MovieID,Rating,Timestamp
1,1193,5,2015-05-14 08:48:14
1,661,3,2015-11-08 12:41:07
1,914,3,2015-02-27 19:56:12
1,3408,4,2015-08-05 06:46:28
1,2355,5,2015-11-30 01:21:58
1,1197,3,2014-08-01 06:04:41
1,1287,5,2015-11-28 15:32:20
1,2804,5,2015-10-22 02:47:32
1,594,4,2014-10-15 14:36:13
1,919,4,2014-12-03 15:06:31
1,595,5,2015-04-24 09:47:01
1,938,4,2014-08-12 02:28:33
1,2398,4,2015-11-11 14:30:28
1,2918,4,2014-04-14 21:57:57
1,1035,5,2014-02-28 10:54:14
1,2791,4,2015-11-21 05:39:23
1,2687,3,2014-06-05 11:32:11
1,2018,4,2014-11-11 19:22:11
```
- Buttons:** < Back, Next >, Cancel, Finish.

Right Window (Step 3 of 4):

- File Settings:**
 - Encoding: UTF-8
 - Field Separator: Comma
 - Row Separator: Standard EOL
- Rows To Skip:** Header checked (1), Footer unchecked, Skip empty row unchecked.
- Escape Char Settings:** CSV selected, Delimited unchecked. Escape Char: Empty, Text Enclosure: Empty, Split row before field unchecked.
- Limit Of Rows:** Limit unchecked.
- Preview:** Shows the first few rows of the CSV file with column headers: UserID, MovieID, Rating, Timestamp.
- Buttons:** Set heading row as column names (checked), Refresh Preview, Export as context, Revert Context, < Back, Next >, Cancel, Finish.

Task3

3.3 Create Metadata for movie rating (File Delimited)(3)

New Delimited File

File - Step 4 of 4
Add a Schema on repository
Define the Schema

Name: metadata
Comment:

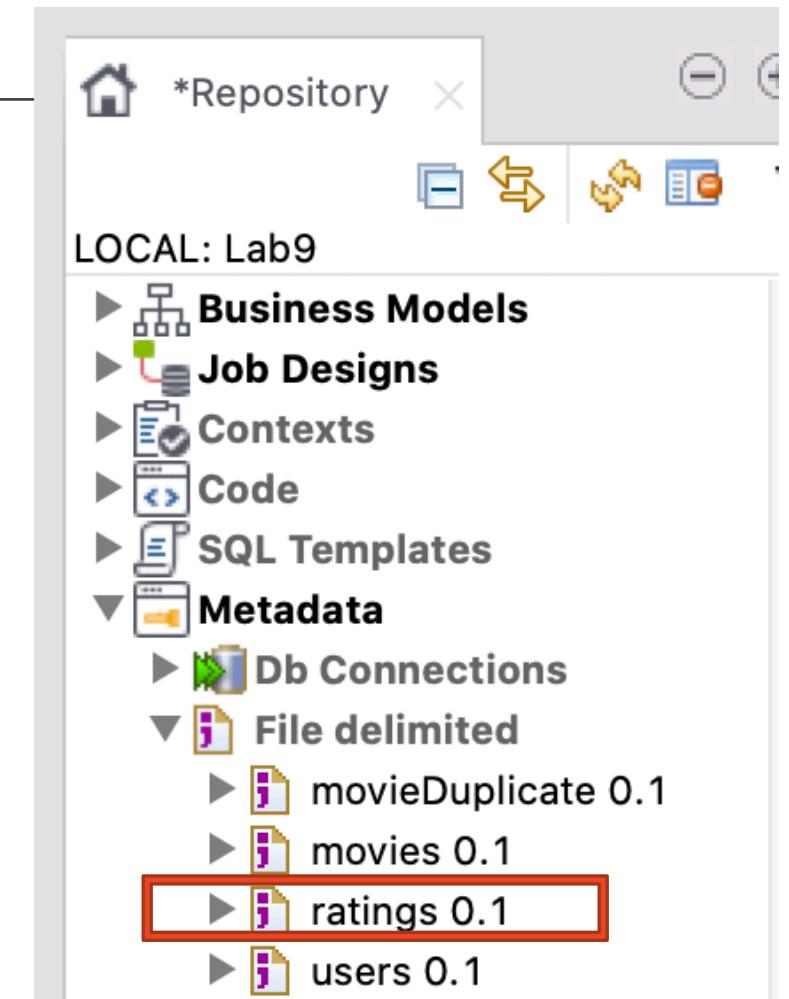
Schema
Click to update schema preview Guess

Description of the Schema

Column	Key	Type	Nullab	Date Pattern (Ctrl+Space)	Length	Precision	Default	Comment
UserID		int	<input checked="" type="checkbox"/>		1	0		
MovielD		int	<input checked="" type="checkbox"/>		4	0		
Rating		Integer	<input checked="" type="checkbox"/>		1	0		
Timestamp		Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd HH:mm:ss"	19	0		

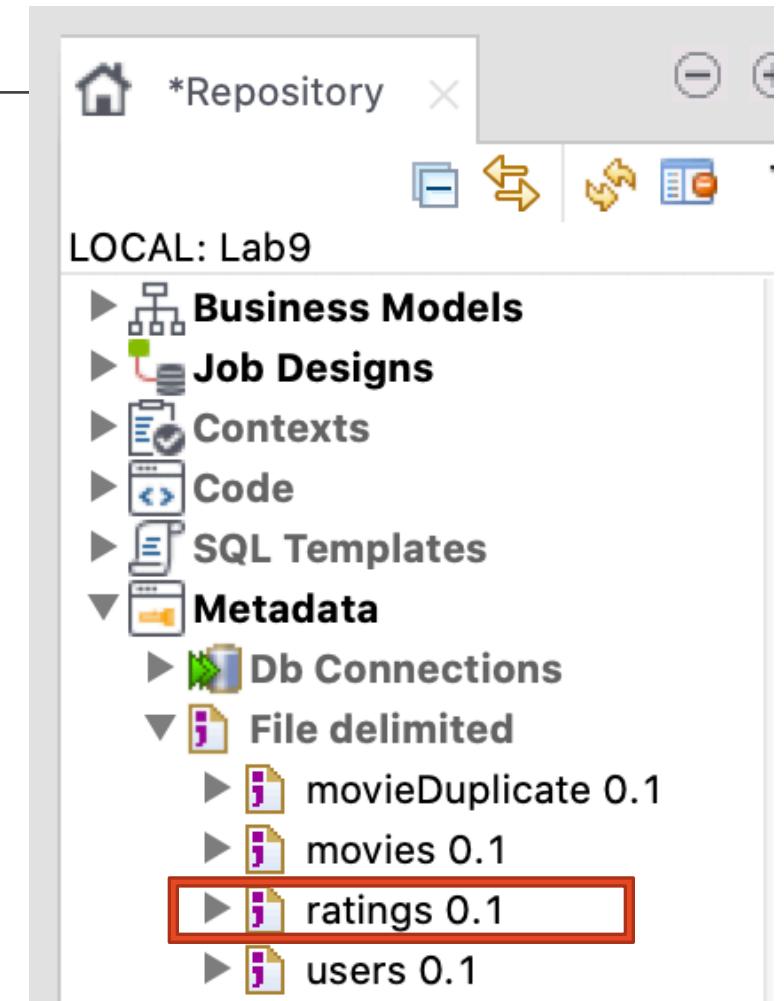
Date pattern = "yyyy-MM-dd HH:mm:ss"

< Back Next > Cancel Finish



Output of Task3

1. Metadata of File Delimited “ratings”



The Results from Extraction Process

MovieID	Title	Year
1	Toy Story (1995)	1995
2	Jumanji (1995)	1995
3	Grumpier Old Men (1995)	1995
4	Waiting to Exhale (1995)	1995
5	Father of the Bride Part II (1995)	1995
6	Heat (1995)	1995
7	Sabrina (1995)	1995
8	Tom and Huck (1995)	1995
9	Sudden Death (1995)	1995
10	GoldenEye (1995)	1995
11	American President, The (1995)	1995
12	Dracula: Dead and Loving It (1995)	1995
13	Balto (1995)	1995
14	Nixon (1995)	1995
15	Cutthroat Island (1995)	1995
16	Casino (1995)	1995

Movies

UserID	Gender	BirthDate	Occupation	Zipcode
1	Female	2019-05-10	10	48067
2	Male	1964-11-10	16	70072
3	Male	1995-06-27	15	55117
4	Male	1975-11-07	7	02460
5	Male	1995-05-06	20	55455
6	Female	1970-02-27	9	55117
7	Male	1985-06-05	1	06810
8	Male	1995-05-27	12	11413
9	Male	1995-03-13	17	61614
10	Female	1985-07-29	1	95370
11	Female	1995-10-01	1	04093
12	Male	1995-01-21	12	32793
13	Male	1975-06-18	1	93304
14	Male	1985-06-09	0	60126
15	Male	1995-01-02	7	22903

Users

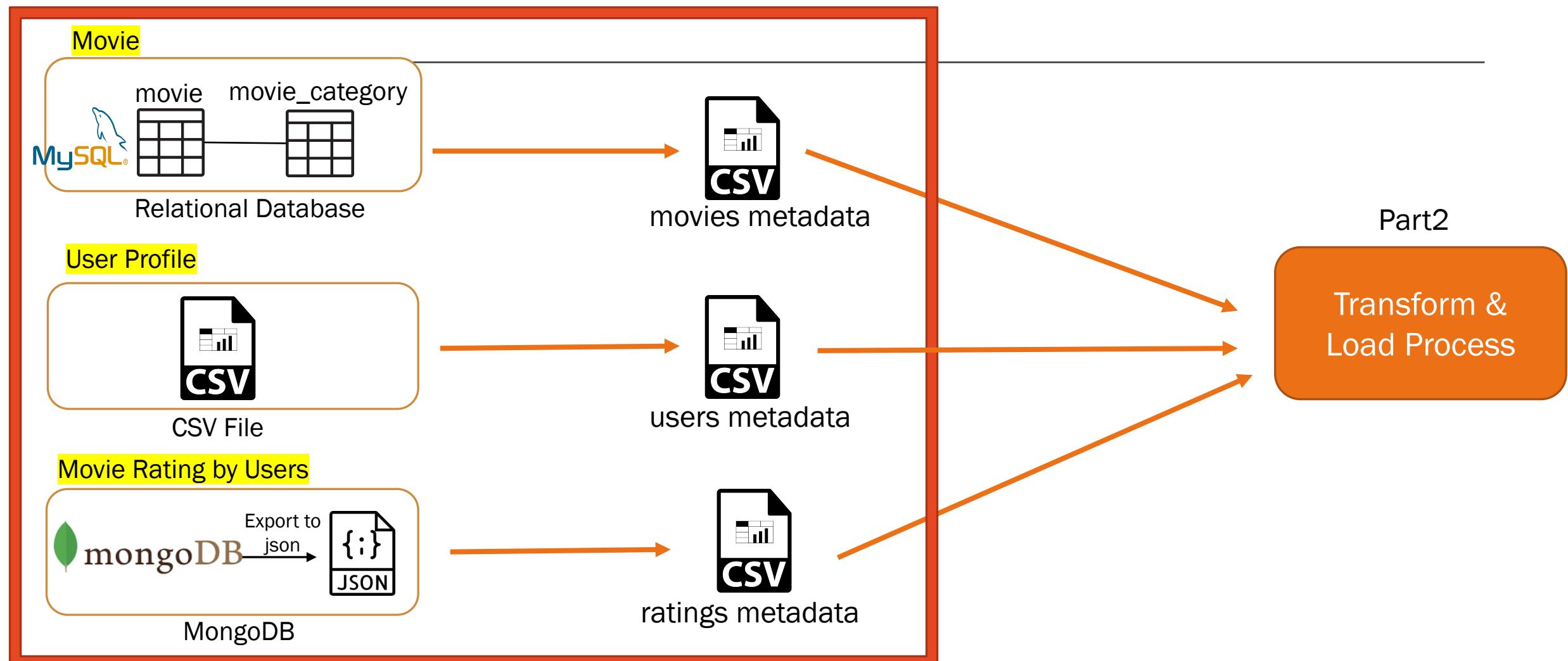
UserID	MovieID	Rating	Timestamp
1	1193	5	2015-05-14 08:48:14
1	661	3	2015-11-08 12:41:07
1	914	3	2015-02-27 19:56:12
1	3408	4	2015-08-05 06:46:28
1	2355	5	2015-11-30 01:21:58
1	1197	3	2014-08-01 06:04:41
1	1287	5	2015-11-28 15:32:20
1	2804	5	2015-10-22 02:47:32
1	594	4	2014-10-15 14:36:13
1	919	4	2014-12-03 15:06:31
1	595	5	2015-04-24 09:47:01
1	938	4	2014-08-12 02:28:33
1	2398	4	2015-11-11 14:30:28
1	2918	4	2014-04-14 21:57:57
1	1035	5	2014-02-28 10:54:14
1	2791	4	2015-11-21 05:39:23

Ratings

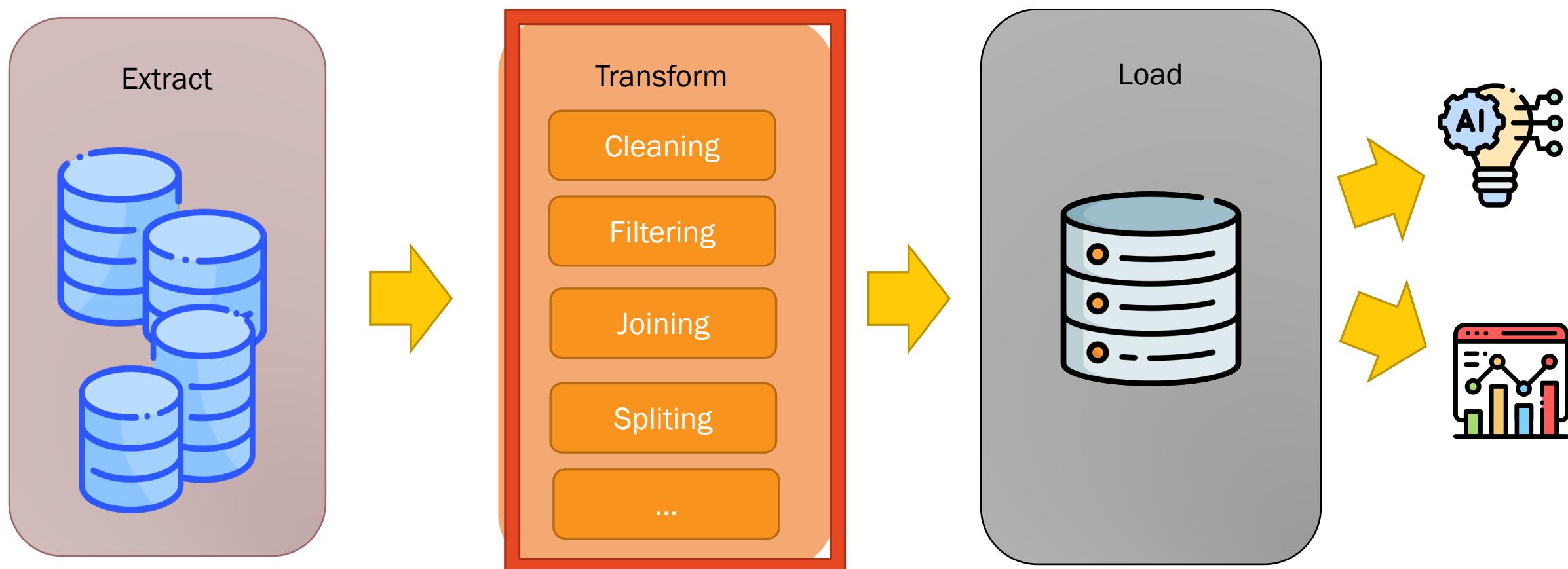
File Delimited Metadata

Part 2

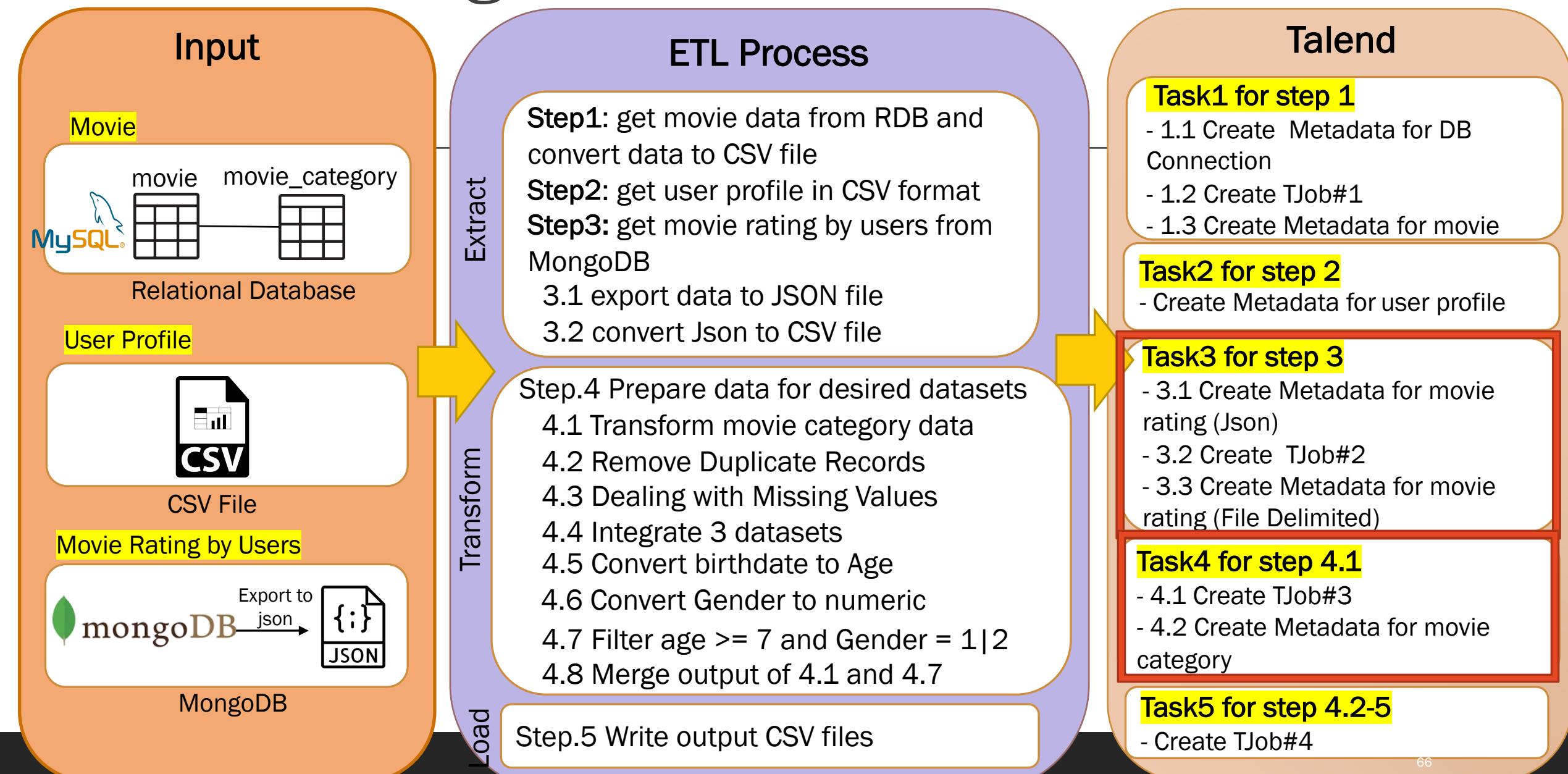
Extraction Process in Part1 (Recap)



ETL Process



Movie Rating Prediction



Task4 for step 4.1

Step4.1: Transform movie category data

Task4:

4.1 Create TJob#3: Transform movie category data

4.2 Create Metadata for movie category (File Delimited)

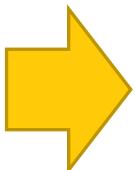
Task4

4.1 Create TJob#3: Transform movie category data

MovielD	Category
1	Animation
1	Children's
1	Comedy
2	Adventure
2	Children's
2	Fantasy
3	Comedy
3	Romance
4	Comedy
4	Drama
5	Comedy



movie category table



➤ Change movie category to numeric values for prediction model

MovielD	Animation	Children's	...	Romance
1	1	1	...	0
2	0	1	...	0
3	0	0	...	1

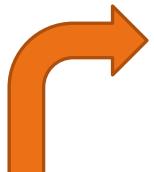
Desired output of movie category

Task4

4.1 Create TJob#3: Transform movie category data

MovielID	Category	Values
1	Animation	1
1	Children's	1
1	Comedy	1
2	Adventure	1
2	Children's	1
2	Fantasy	1
3	Comedy	1
3	Romance	1
4	Comedy	1
4	Drama	1
5	Comedy	1
6	Action	1
6	Crime	1
6	Thriller	1

movie category table



Pivot Table:

- Set value = 1 for all records in column “Values”
- Pivot “Category” column to header

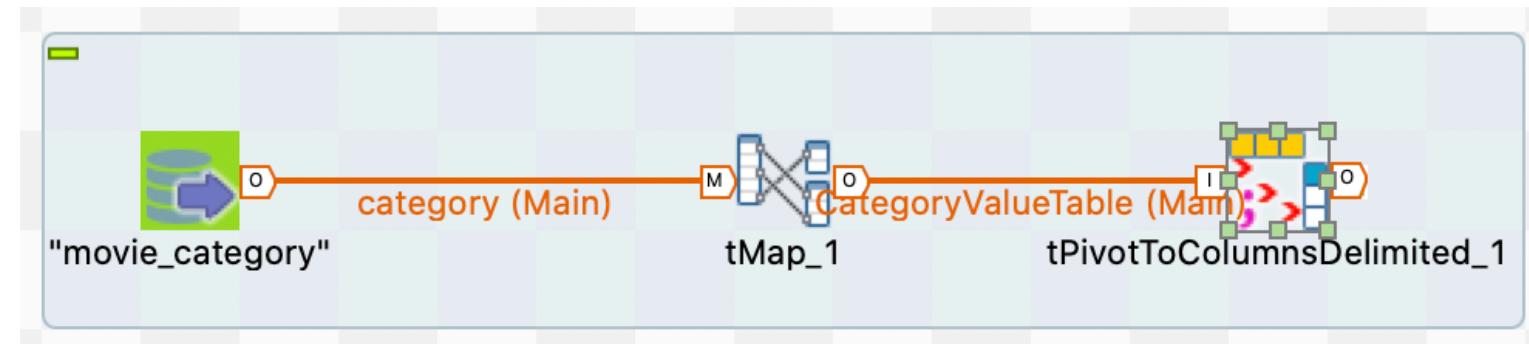
MovielID	Animation	Children's	Romance
1	1	1	...	0
2	0	1	...	0
3	0	0	...	1

Task4

4.1 Create TJob#3: Transform movie category data

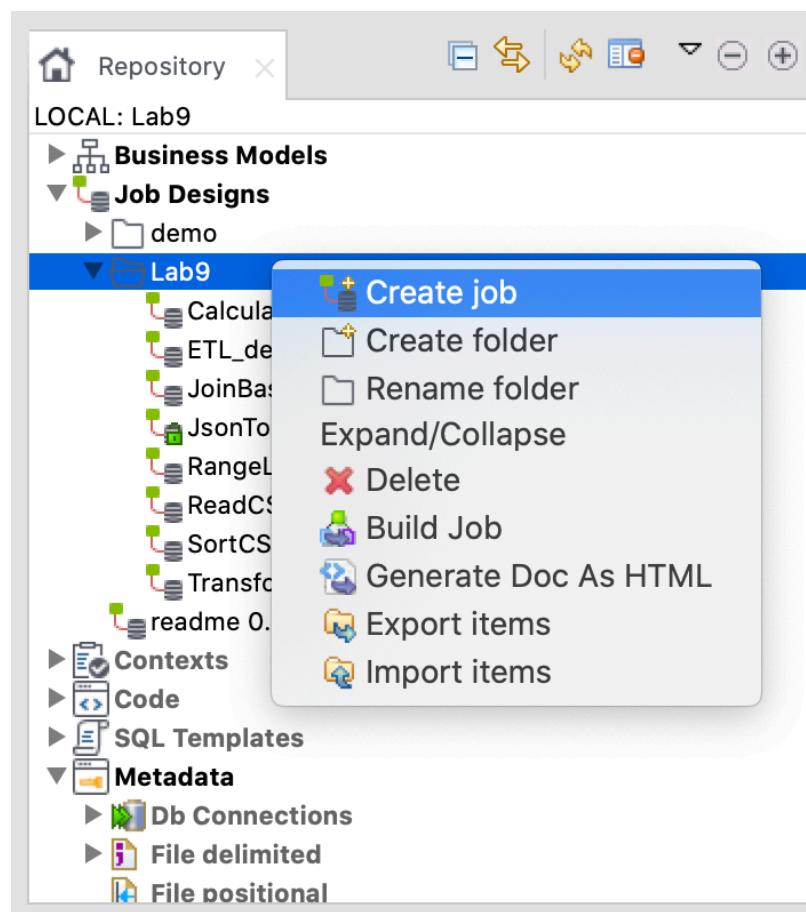
Purpose : Transform movie category data for prediction model

Components: tDBInput(MySQL) , tmap, tPivotToColumnsDelimited



Task4

4.1 Create TJob#3: Transform movie category data



The screenshot shows the 'New Job' dialog box. The 'Name' field is filled with 'TJob3TransformMovieCategory'. The 'Purpose' field is empty. The 'Description' field is empty. The 'Author' field contains 'user@talend.com'. The 'Locker' field is empty. The 'Version' field is set to '0.1'. The 'Status' field is empty. The 'Path' field is set to 'Lab9'. At the bottom right, there are 'Cancel' and 'Finish' buttons. A warning message above the form states: 'It is inadvisable to leave the purpose blank.' There is also a small icon of a database with a red line pointing to it.

Task4

4.1 Create TJob#3: Transform movie category data

- Drag Metadata of table “movie_category” under “MySQL_Conn_movie” to Design Workspace

The screenshot shows the Talend Data Integration workspace. On the left, the Project Explorer displays a connection named "MySQL_Conn_movie 0.1" which contains a "Table schemas" folder with an entry for "movie_category". A large orange arrow points from this entry towards the center of the screen. In the center, a "Components" dialog box is open, listing various MySQL components. The "tDBInput(MySQL)" component is selected and highlighted with a blue background. At the bottom right of the dialog are "Cancel" and "OK" buttons. To the right of the dialog is the main workspace. It shows a job structure with a "movie_category" component connected to a "tMap_1" component. Below this, a detailed configuration window for the "movie_category" component is open. The "Basic settings" tab is selected, showing the following configuration:

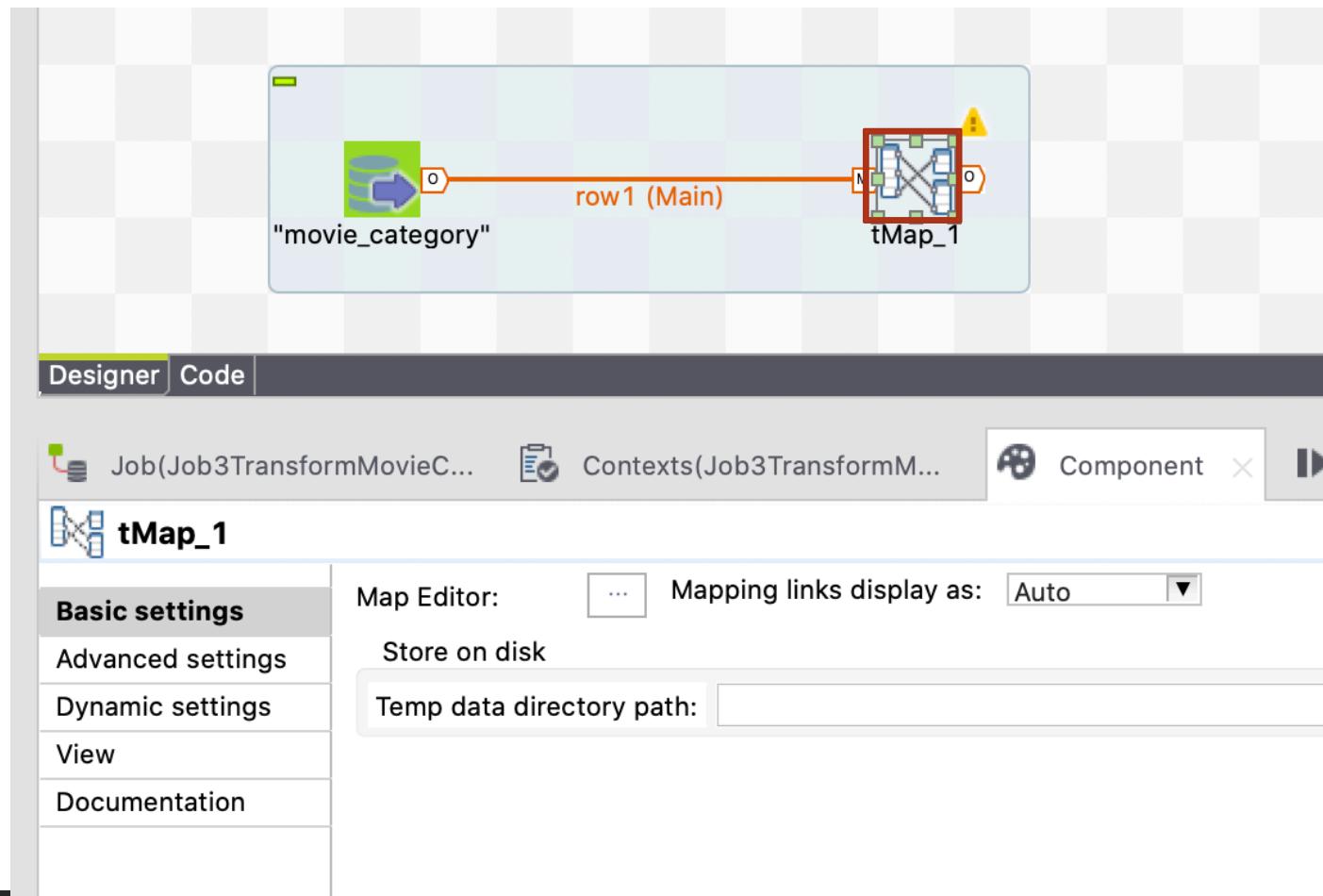
Database	Mysql	Apply
Property Type	Repository	DB (MYSQ...):MySQL_Conn_movie
DB Version	Mvsal 5	
<input type="checkbox"/> Use an existing connection		
Host	sql12.freemysqlhost*	Port 3306*
Username	sql12375315*	Password *****
Schema	Repostor	DB (MYSQ...):MySQL_Conn_movie - movie_cat

The "Table Name" field is set to "movie_category". The "Query Type" is set to "Built-In". The "Query" field contains the following SQL code, which is highlighted with a red box:

```
SELECT
`movie_category`.`MovieID`,
`movie_category`.`Category`
FROM `movie_category`
```

Task4

4.1 Create TJob#3: Transform movie category data



- **tMap Component**
 - get data from one or more sources
 - transforms data
 - sends the transformed data to one or more destinations.
- Double click at tMap component

Task4

4.1 Create TJob#3: Transform movie category data

Set value = 1 for all records in column "Values"

MovielD	Category	Values
1	Animation	1
1	Children's	1
1	Comedy	1
2	Adventure	1
2	Children's	1
2	Fantasy	1
3	Comedy	1
3	Romance	1
4	Comedy	1
4	Drama	1
5	Comedy	1
6	Action	1
6	Crime	1
6	Thriller	1

Talend Open Studio for Data Integration - tMap - tMap_1

category

Column
MovielD
Category

Var

CategoryValueTable

Expression
category.MovielD
category.Category

Column
MovielD
Category
Values

Set value = 1 for all records in column "Values"

Add column "Values"

1

2

Schema editor Expression editor

category

Column Key Type Nullab Date Pattern Length Preciso Default Comm

MovielD int ✓ 10 100 0

Category String

CategoryValueTable

Column Key Type Nullab Date Pattern Length Preciso Default Comm

MovielD int ✓ 10 100 0

Category String

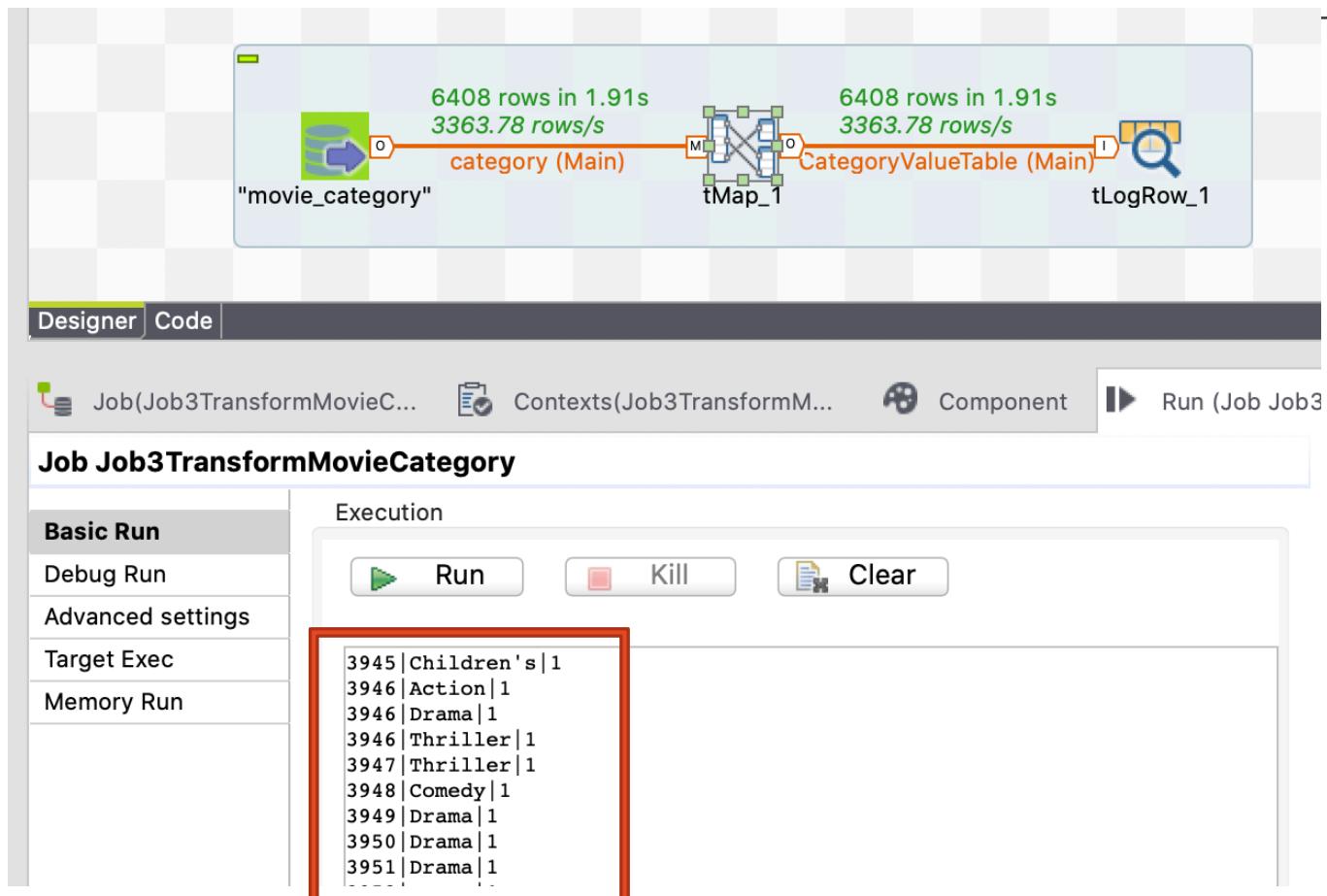
Values int

1

Apply Ok Cancel

Task4

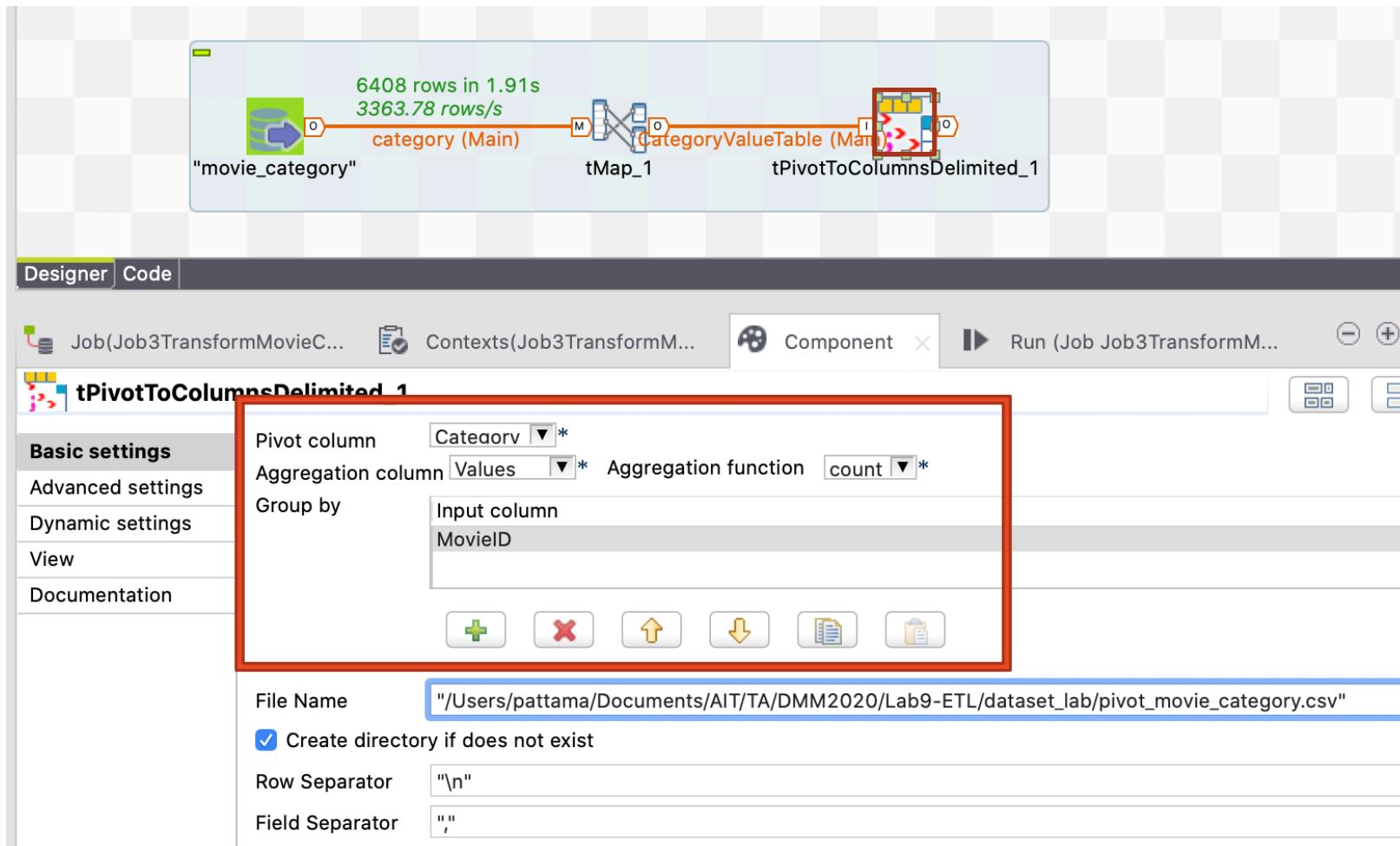
4.1 Create TJob#3: Transform movie category data



- Add tLogRow to check output
- Example Output:
 - MovieID
 - Category
 - Values

Task4

4.1 Create TJob#3: Transform movie category data



- Pivot column: Category
- Aggregation column: Values
- Aggregation function: count
- Group by: MovieID

Task4

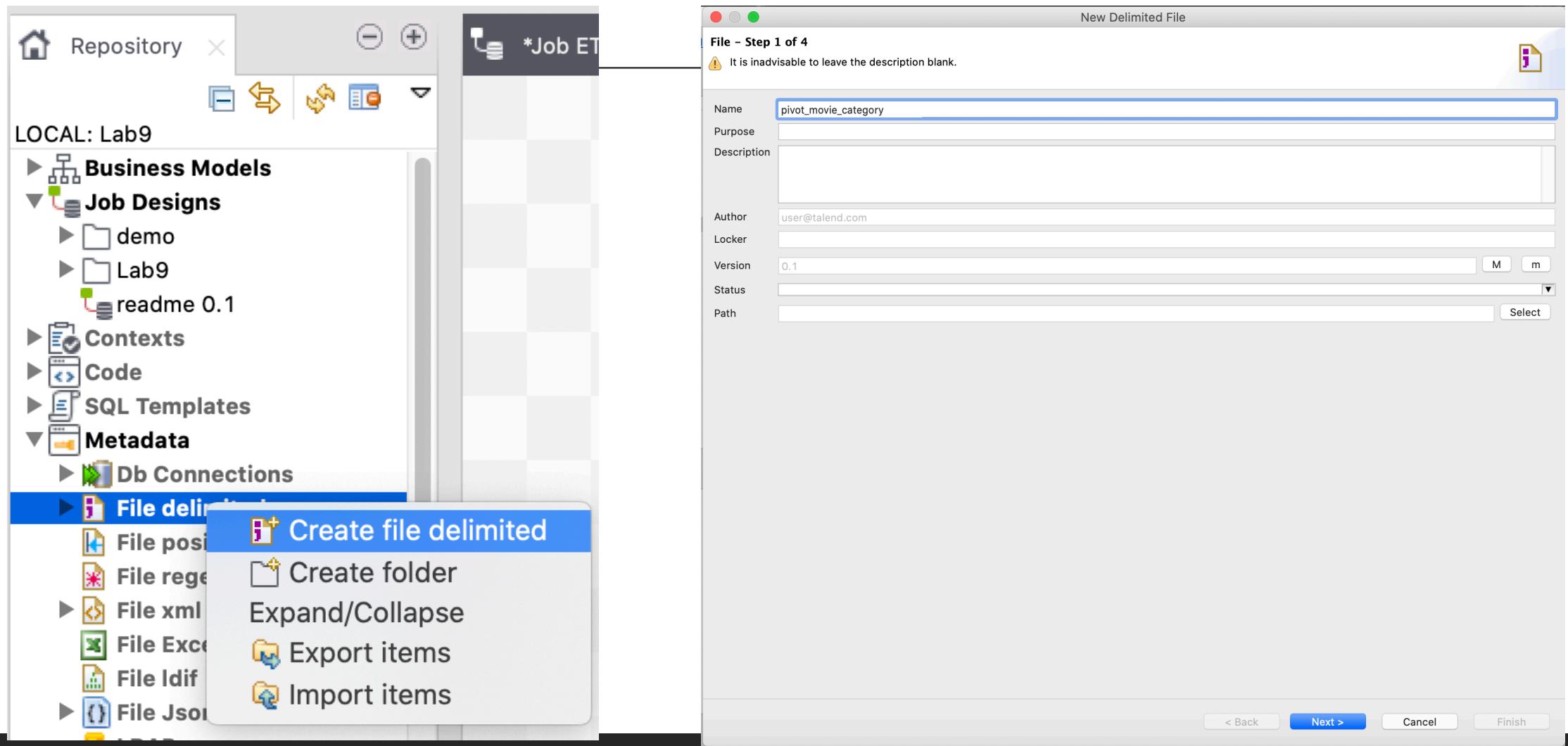
4.1 Create TJob#3: Transform movie category data

MovieID	Animation	Children's	Comedy	Adventure	Fantasy	Romance	Drama	Action	Crime	Thriller	Horror	Sci-Fi	Thriller	Horror	Sci-Fi
1	1	1	1												
2		1		1	1										
3			1			1									
4			1				1								
5			1					1							
6									1	1	1				
7			1			1									
8		1		1											
9									1						
10				1					1	1					
11			1				1	1							
12			1								1				
13	1	1													
14								1							
15				1		1			1						
16									1		1				
17								1	1						

Output after pivot column

Task4

4.2 Create Metadata for movie category (File Delimited) (1)



Task4

4.2 Create Metadata for movie category (File Delimited) (2)

The image shows two side-by-side windows for creating a delimited file. Both windows have a title bar 'New Delimited File' and are labeled 'File - Step 2 of 4' and 'File - Step 3 of 4' respectively.

File - Step 2 of 4: This window is for defining the path of the file and format settings. It includes fields for Server (localhost 127.0.0.1), File (pivot_movie_category.csv located at /Users/pattama/Documents/AIT/TA/DMM2020/Lab9-ETL/dataset), and Format (UNIX). A 'File Viewer' section displays the first few rows of the CSV file, which lists MovieID and various genres separated by commas.

MovieID	Animation	Children's	Comedy	Adventure	Fantasy	Romance	Drama	Action	Crime	Thriller	Horror	Sci-Fi	Documentary	War	Musical	Mystery	Film-Noir	Western
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

File - Step 3 of 4: This window is for defining the setting of the parse job. It includes sections for File Settings (Encoding: UTF-8, Field Separator: Comma, Row Separator: Standard EOL), Escape Char Settings (CSV selected), and Rows To Skip (Header checked, value 1). A 'Preview' tab shows the parsed data with column names and values corresponding to the genres listed in the file viewer.

MovieID	Animation	Children's	Comedy	Adventure	Fantasy	Romance	Drama	Action	Crime	Thriller	Horror	Sci-Fi	Documentary	War	Musical	Mystery	Film-Noir
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Task4

4.2 Create Metadata for movie category (File Delimited) (3)

The screenshot shows two windows from the Talend Data Integration environment.

Left Window: New Delimited File - Step 4 of 4

This window is titled "New Delimited File" and is on "Step 4 of 4: Define the Schema". It shows a schema for a file named "metadata".

Schema Definition:

Column	Type	Length	Precision	Default	Comment
MovielD	int	2	0	0	
Animation	Integer			0	
Children_s	Integer			0	
Comedy	Integer			0	
Adventure	Integer			0	
Fantasy	Integer			0	
Romance	Integer			0	
Drama	Integer			0	
Action	Integer			0	
Crime	Integer			0	
Thriller	Integer			0	

Right Window: Repository

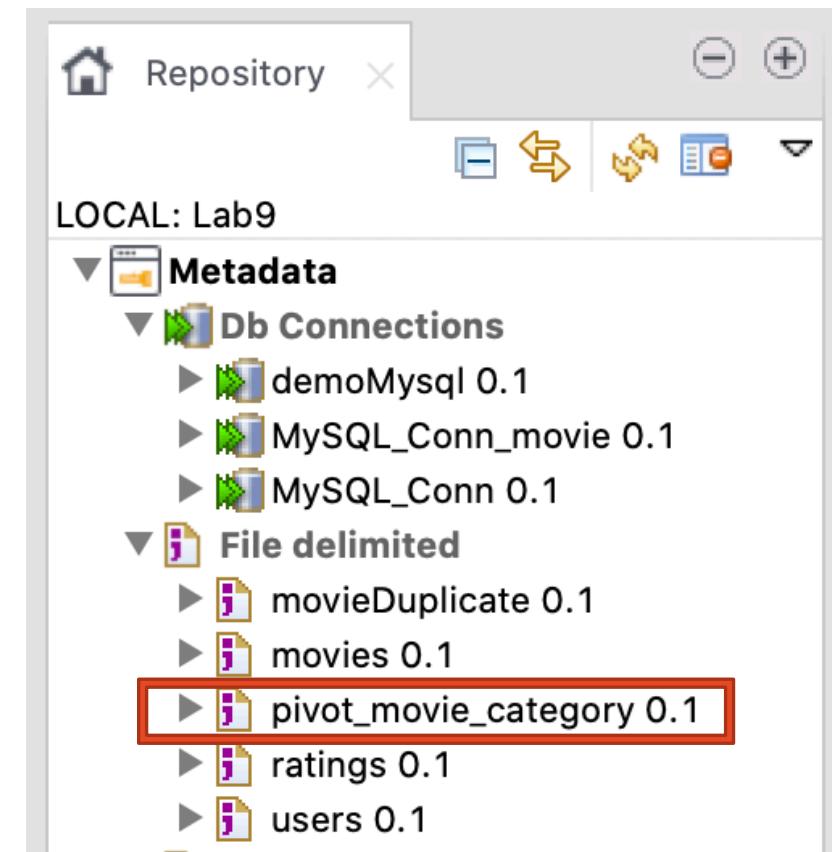
This window shows the "Repository" view for "LOCAL: Lab9". It lists various metadata items:

- Db Connections
 - demoMySQL 0.1
 - MySQL_Conn_movie 0.1
 - MySQL_Conn 0.1
- File delimited
 - movieDuplicate 0.1
 - movies 0.1
 - pivot_movie_category 0.1 (highlighted with a red box)
 - ratings 0.1
 - users 0.1

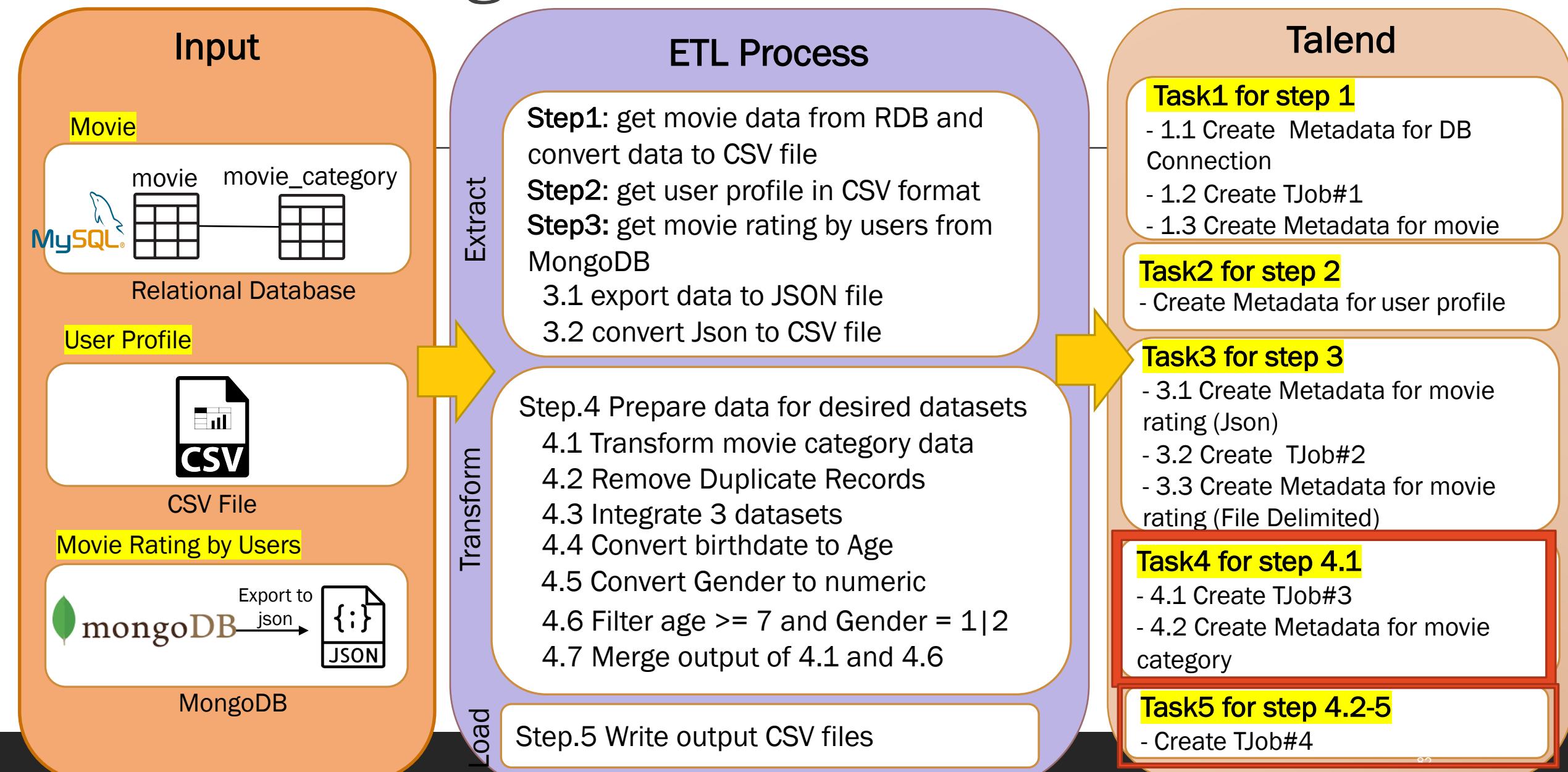
Output of Task4

1. Metadata of File Delimited “pivot_movie_category”

Using this metadata in Tjob#4 !!!



Movie Rating Prediction



Task5 for step 4.2-5

Step4.2-5: Prepare dataset for desired datasets

Task5:

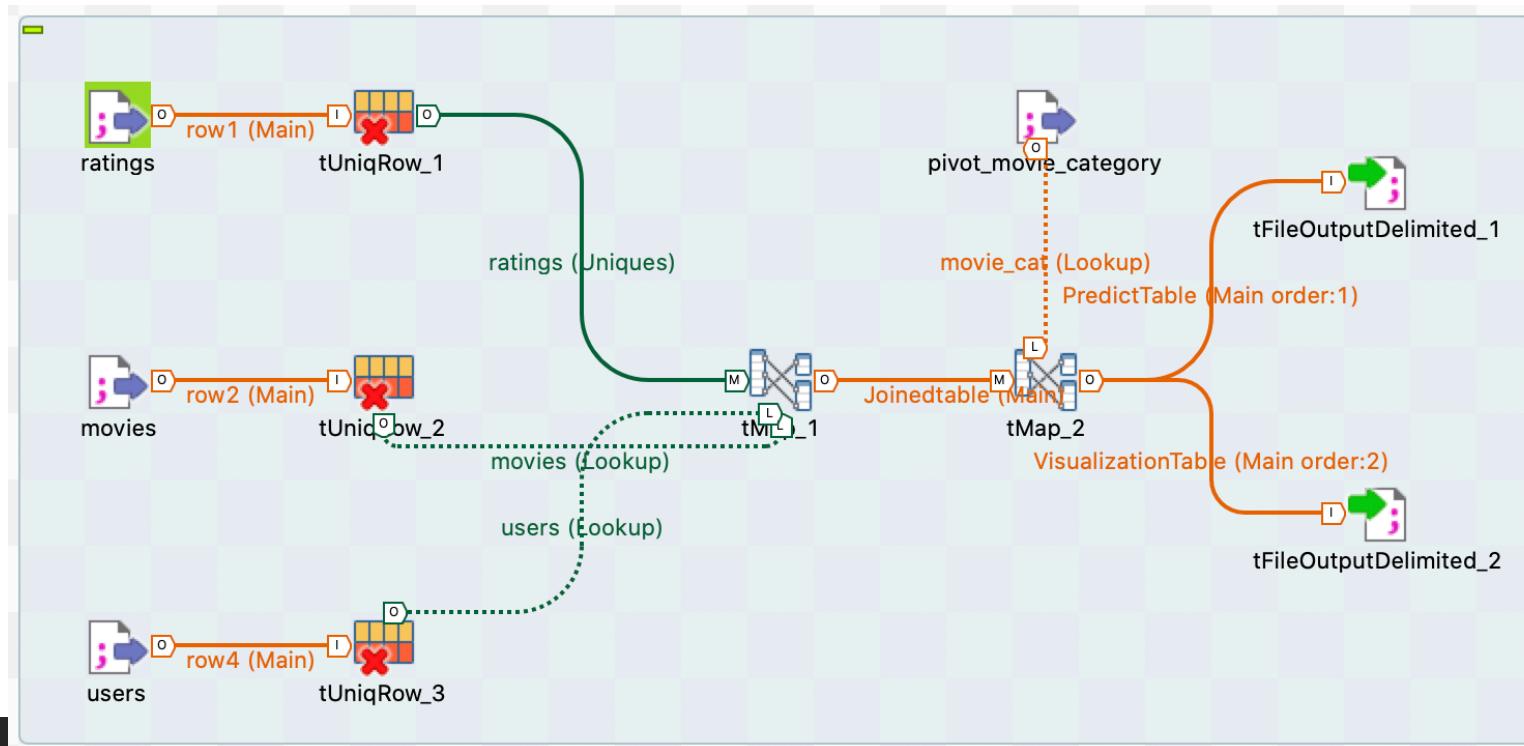
Create TJob#4

Task5

Prepare dataset for desired datasets

Purpose : Prepare dataset for desired datasets

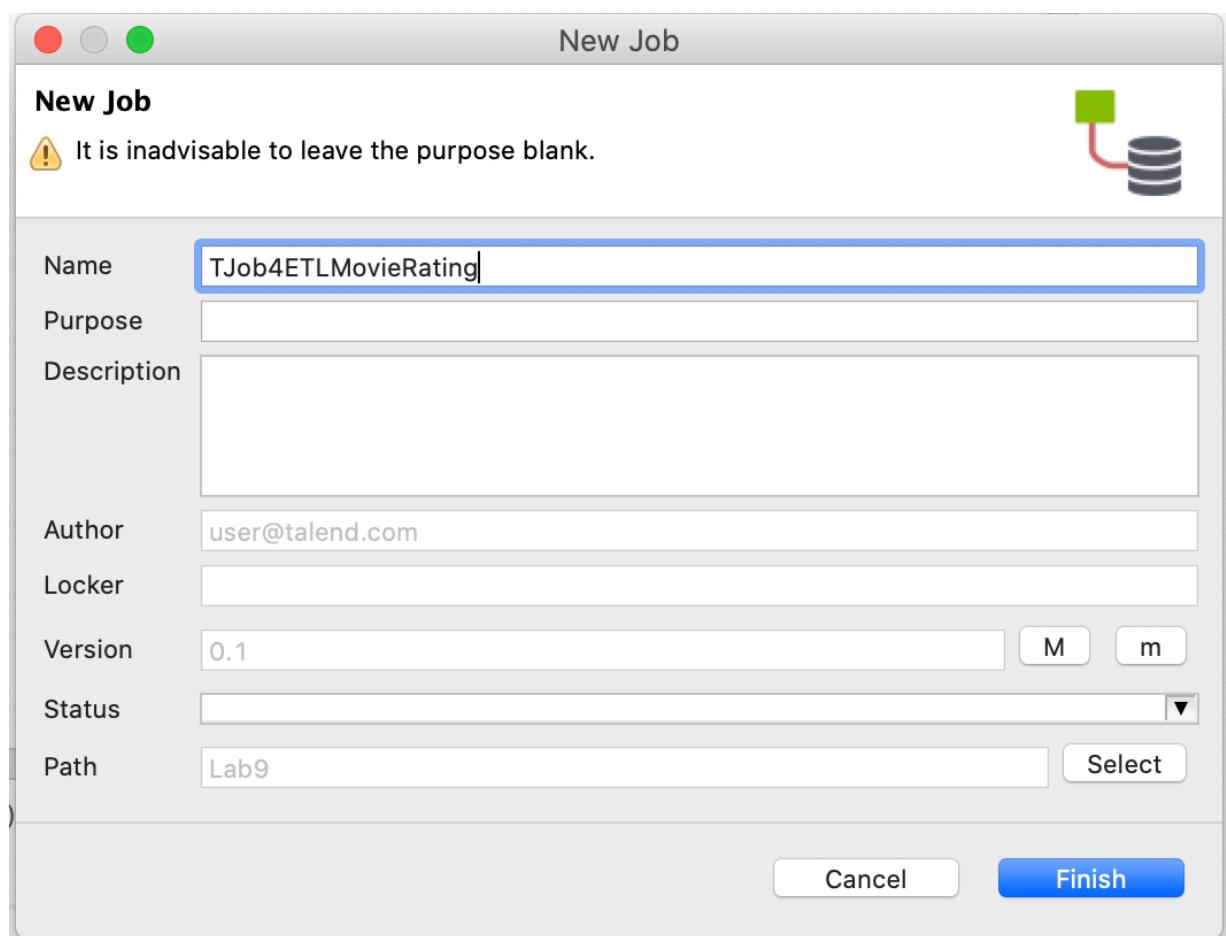
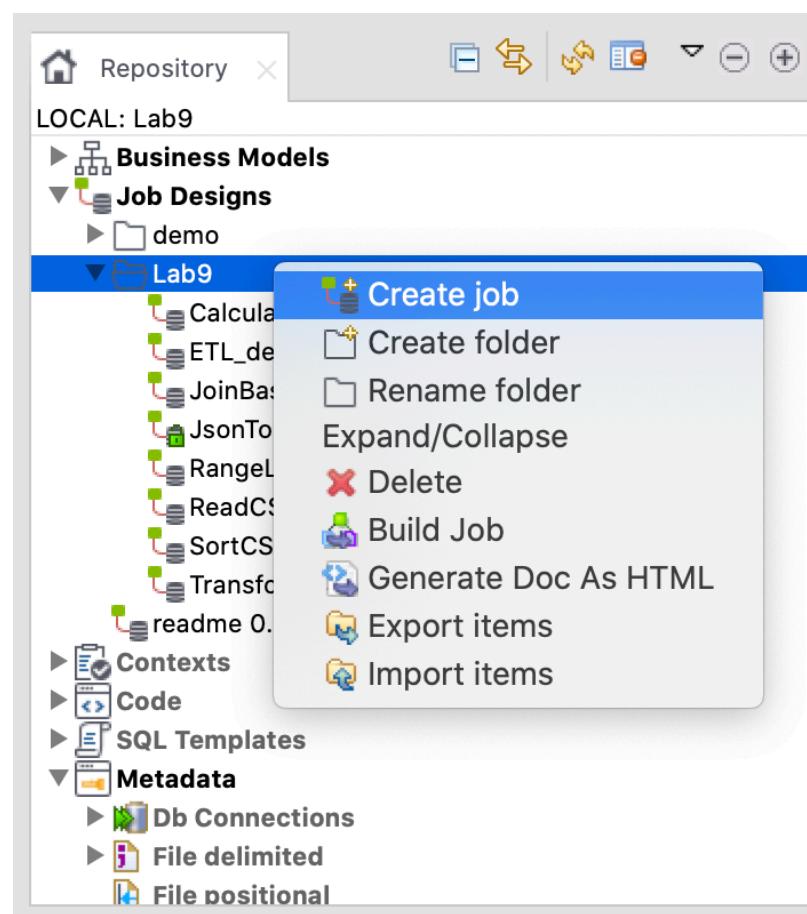
Components: tFileInputDelimited, tUniqRow, tMap, tFileOutputDelimited_1



- **tUniqRow** - Compares entries and sorts out duplicate entries from the input flow.
- **tMap**
 - get data from one or more sources
 - transforms data
 - sends the transformed data to one or more destinations.

Task5

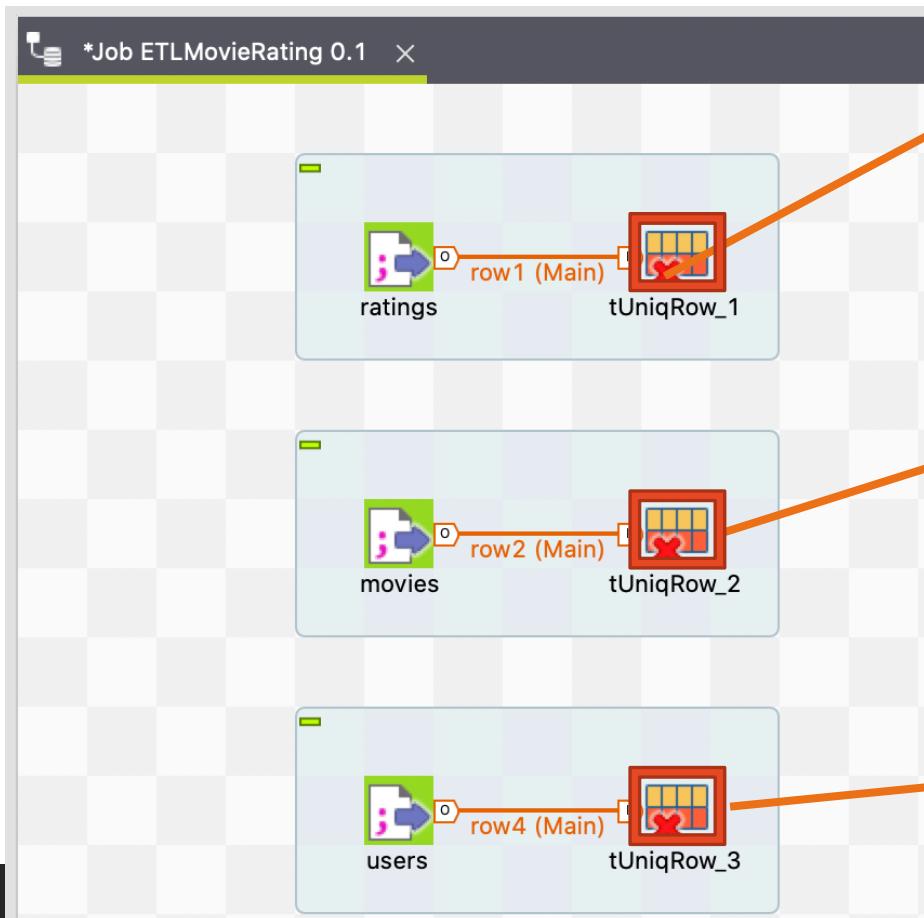
Create TJob#4: Prepare dataset for desired datasets



Task5

Create TJob#4: Prepare dataset for desired datasets

Step 4.2 Remove Duplicate Records



tUniqRow_1

Column	Key attribute
UserID	<input checked="" type="checkbox"/>
MovielD	<input checked="" type="checkbox"/>
Rating	<input checked="" type="checkbox"/>
Timestamp	<input checked="" type="checkbox"/>

tUniqRow_2

Column	Key attribute
MovielD	<input checked="" type="checkbox"/>
Title	<input checked="" type="checkbox"/>
Genres	<input checked="" type="checkbox"/>

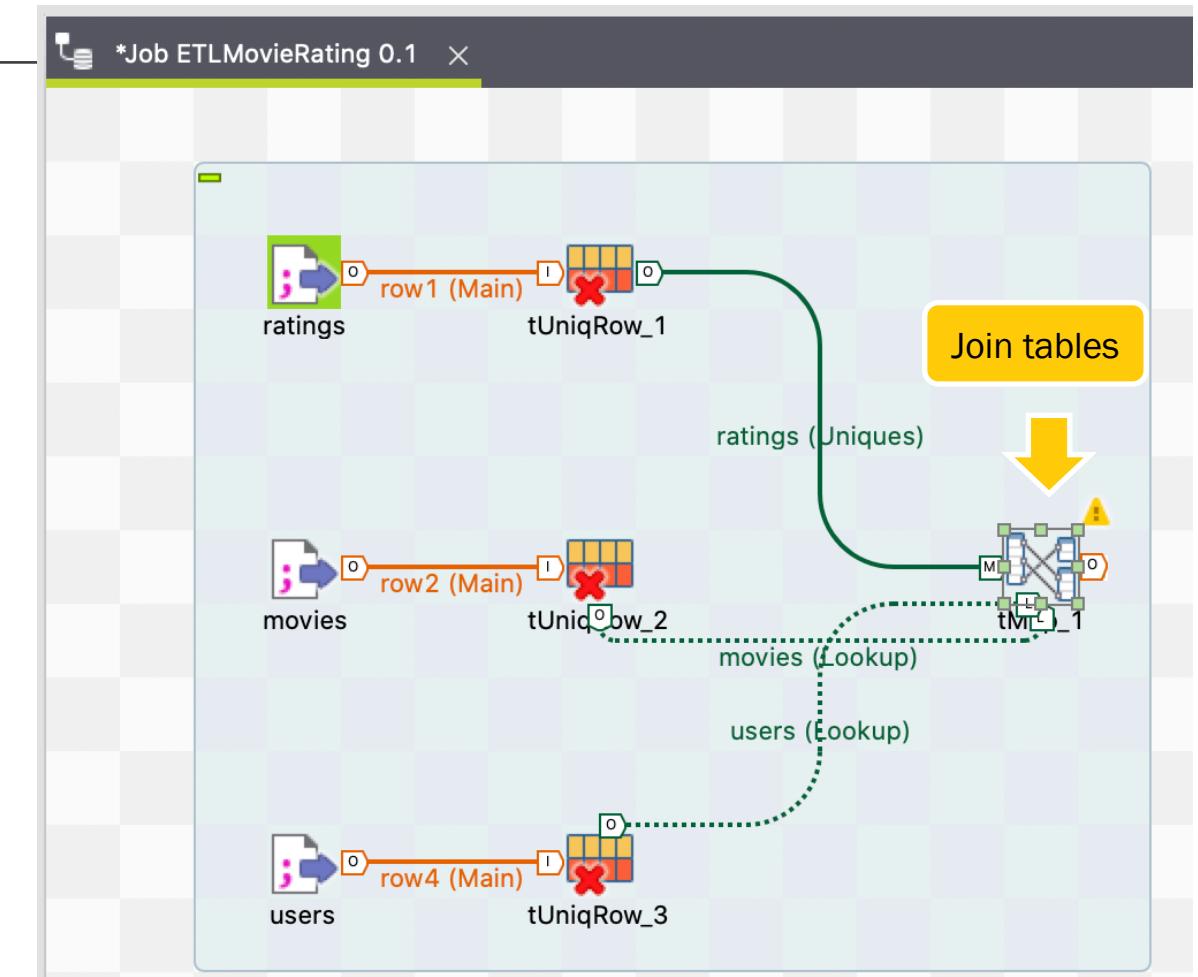
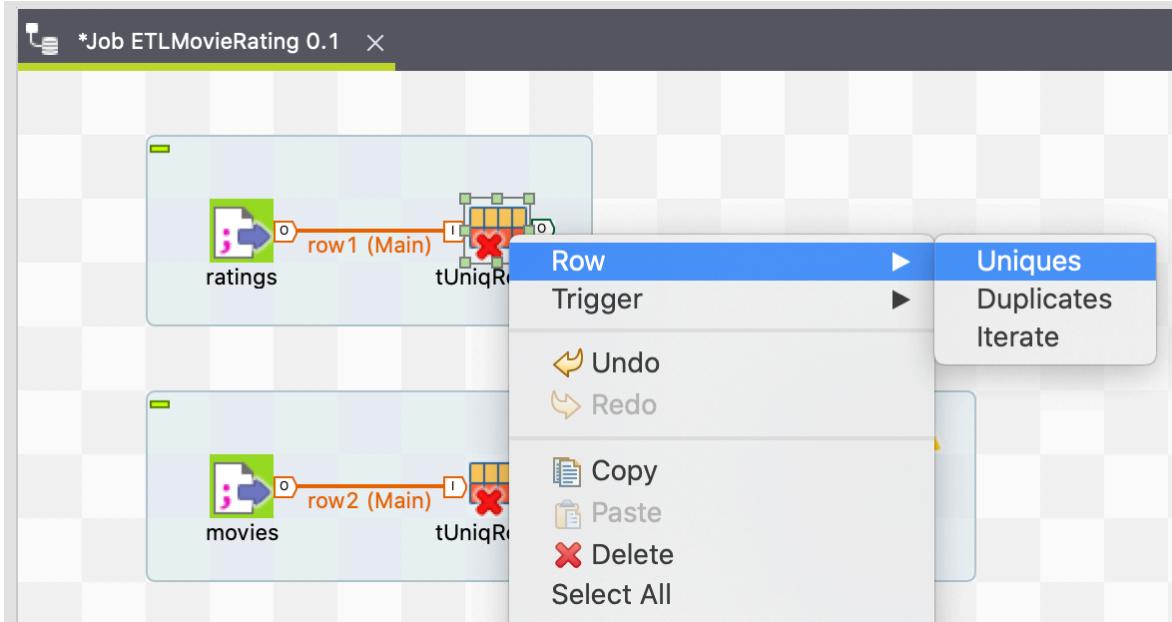
tUniqRow_3

Column	Key attribute
UserID	<input checked="" type="checkbox"/>
Gender	<input checked="" type="checkbox"/>
BirthDate	<input checked="" type="checkbox"/>
Occupation	<input checked="" type="checkbox"/>

Task5

Create TJob#4: Prepare dataset for desired datasets

Step 4.3 Integrate 3 datasets



Task5

Create TJob#4: Prepare dataset for desired datasets

Step 4.3 Integrate 3 datasets

The screenshot shows the Talend Open Studio interface for creating a data integration job. The main window displays the tMap component, which is used to integrate three datasets:

- Input Tables:** The 'movies' dataset is highlighted with a red box. Annotations indicate that the 'Match Model' is set to 'Unique match' and the 'Join Model' is set to 'Inner Join'. A callout box states: "Set join model = Inner join".
- Output Table:** The resulting dataset is named "Joinedtable", shown in the 'Var' tab.

The 'Var' tab lists the columns of the Joinedtable:

Column
ratings.UserID
ratings.MovieID
movies.Title
movies.Genres
ratings.Rating
ratings.Timestamp
users.Gender
users.BirthDate
users.Occupation
users.Zipcode

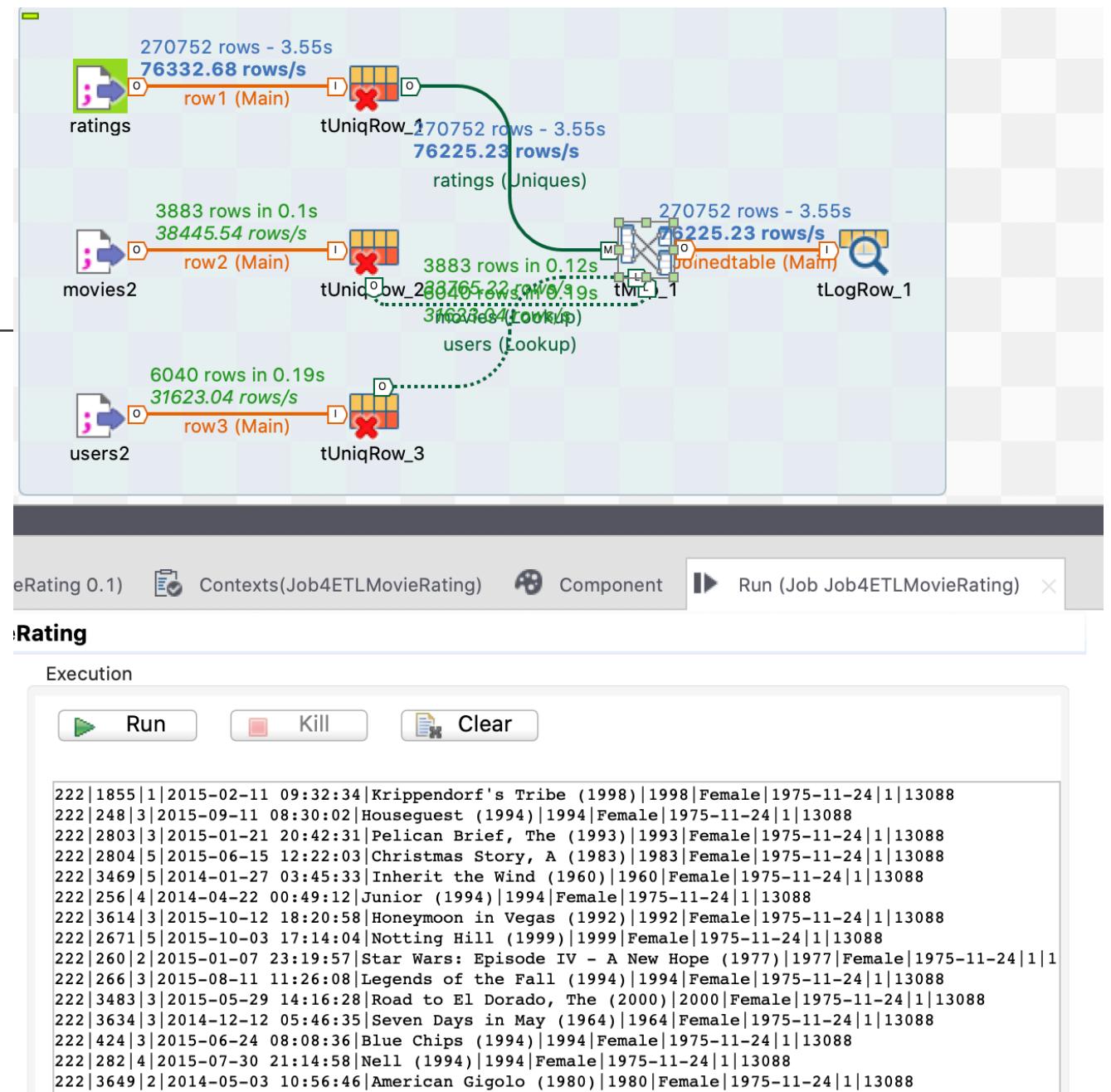
The 'Schema editor' and 'Expression editor' tabs at the bottom show the detailed schema for the 'movies' and 'Joinedtable' datasets, including column names, data types, and length specifications.

Task5

Create TJob#4: Prepare dataset for desired datasets

Step 4.3 Integrate 3 datasets

Output of step 4.3 is “Joinedtable”



Task5

Create TJob#4: Prepare dataset for desired datasets

UserID	Gender	BirthDate	Occupation	Zipcode
1	Female	2019-05-10	10	48067
2	Male	1964-11-10	16	70072
3	Male	1995-06-27	15	55117
4	Male	1975-11-07	7	02460
5	Male	1995-05-06	20	55455
6	Female	1970-02-27	9	55117
7	Male	1985-06-05	1	06810
8	Male	1995-05-27	12	11413
9	Male	1995-03-13	17	61614
10	Female	1985-07-29	1	95370
11	Female	1995-10-01	1	04093
12	Male	1995-01-21	12	32793
13	Male	1975-06-18	1	93304
14	Male	1985-06-09	0	60126
15	Male	1995-01-02	7	22903

Users

Step 4.4: Convert birthdate to Age

Step 4.5: Convert Gender to numeric

Transform Data

- Convert birth date to age (numeric values)
- Convert Gender to numeric values (Male=1,Female=2)

Filtering Data

- Filter age more than 7 years

Task5

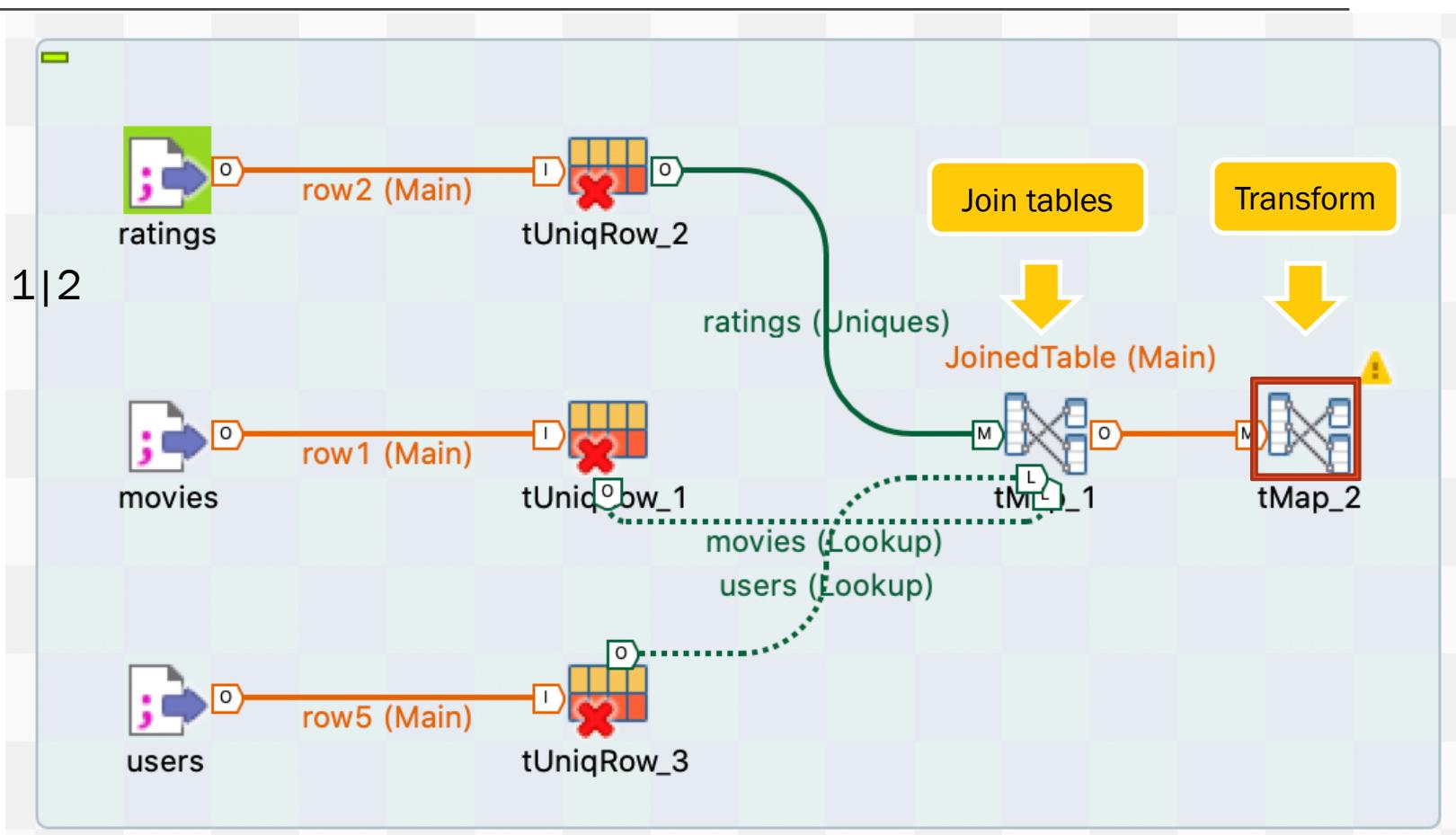
Create TJob#4: Prepare dataset for desired datasets

Transform Data:

Step 4.4: Convert birthdate to Age

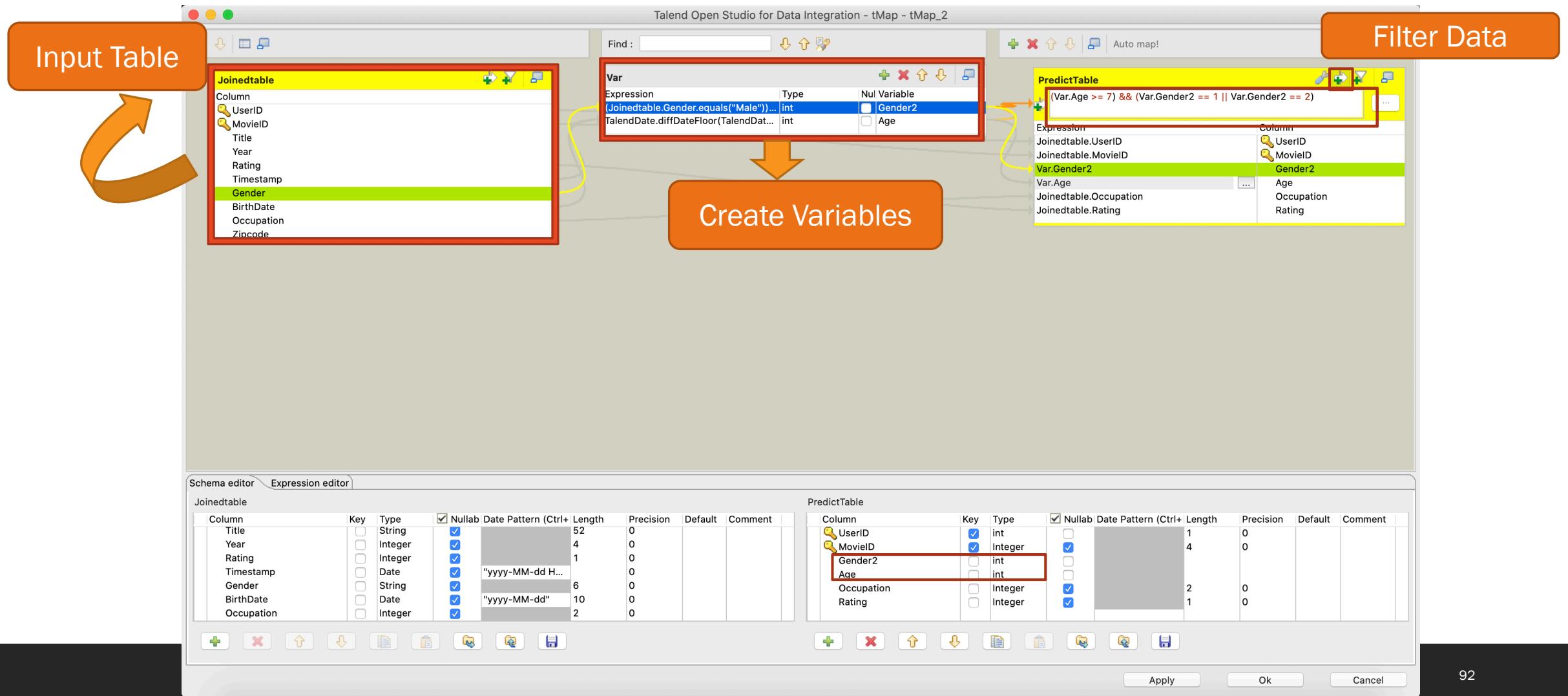
Step 4.5: Convert Gender to numeric

Step 4.6: Filter age ≥ 7 and Gender = 1|2



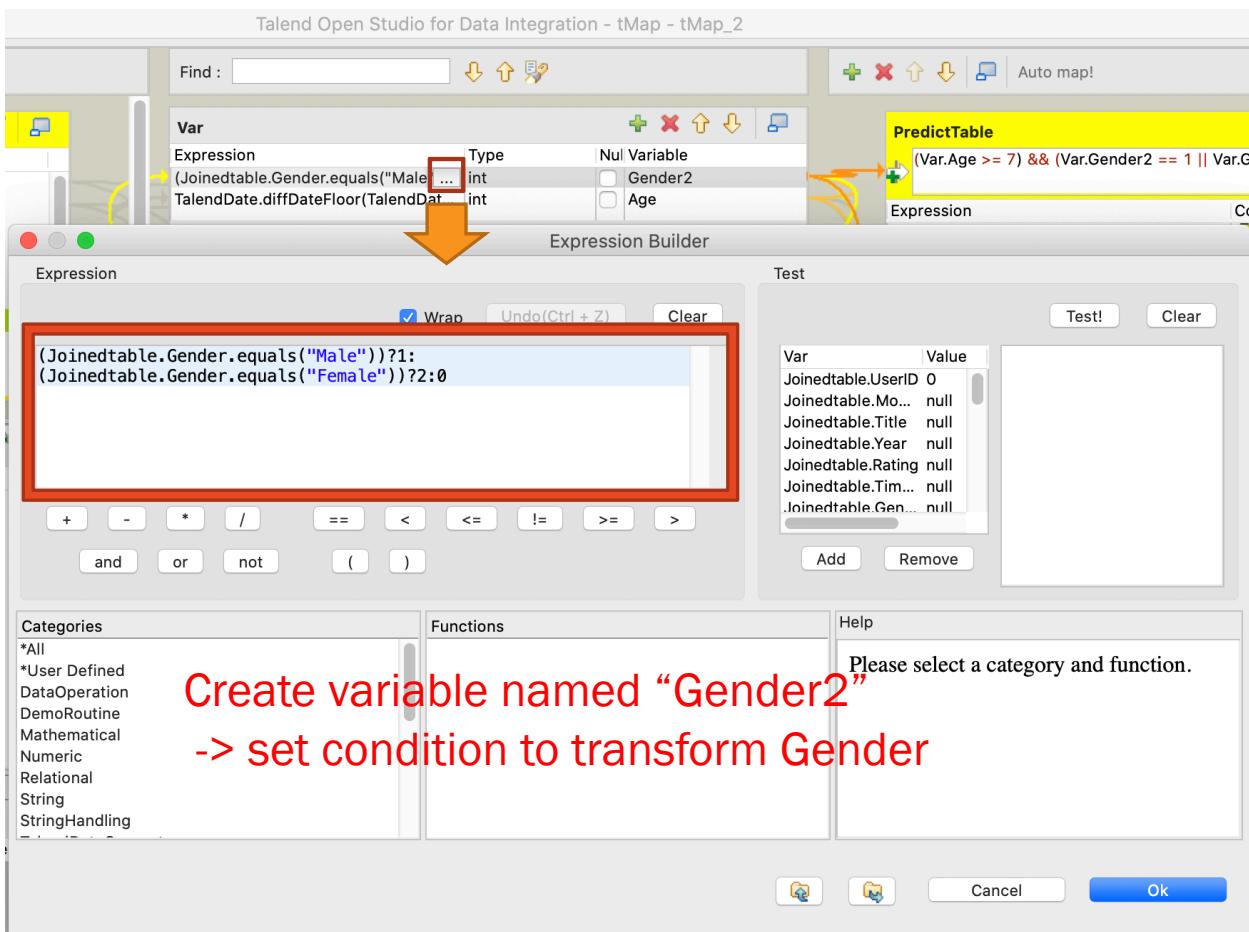
Task5

Create TJob#4: Prepare dataset for desired datasets

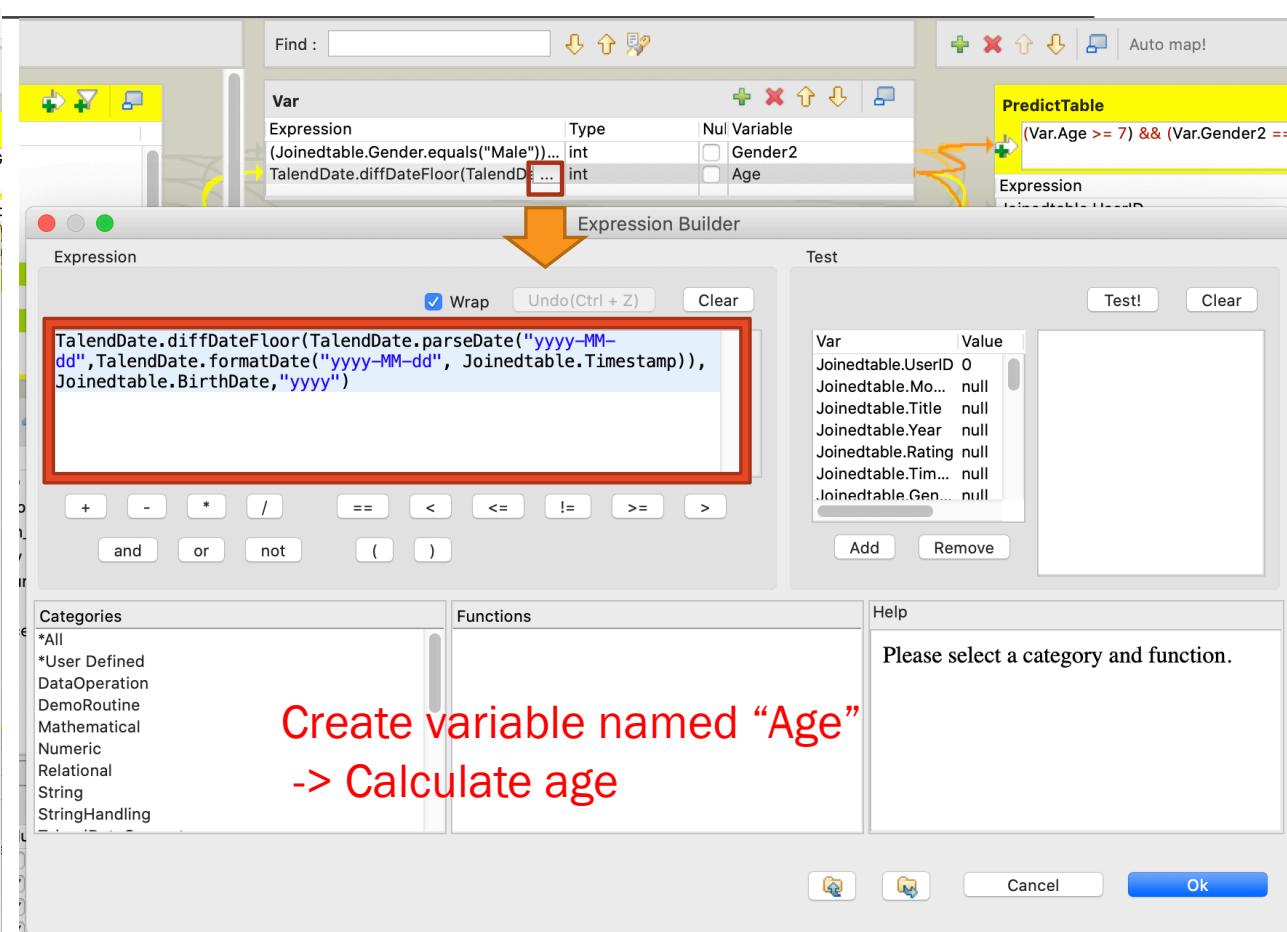


Task5

Create TJob#4: Prepare dataset for desired datasets



Create variable named “Gender2”
-> set condition to transform Gender



Create variable named “Age”
-> Calculate age

Task5

Create TJob#4: Prepare dataset for desired datasets

Create Variables

Age(int)

TalendDate.diffDateFloor(TalendDate.parseDate("yyyy-MM-dd", TalendDate.formatDate("yyyy-MM-dd", Joinedtable.Timestamp)), Joinedtable.BirthDate, "yyyy")

Call java method

Gender2
(String)

(Joinedtable.Gender.equals("Male"))?1:
(Joinedtable.Gender.equals("Female"))?2:0

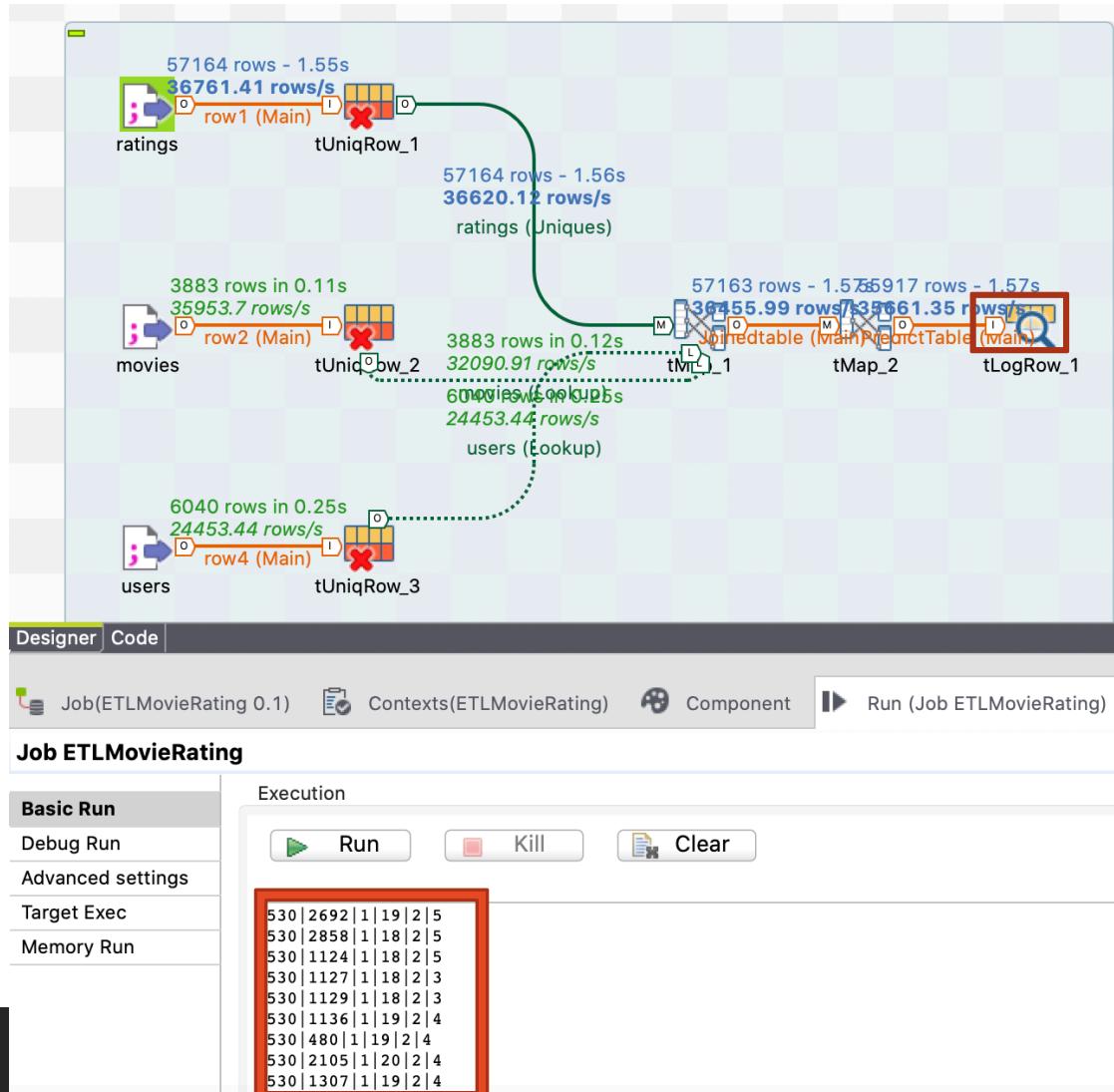
IF/ELSE Statement

Filter Data

(Var.Age >= 7) && (Var.Gender2 == 1 || Var.Gender2 == 2)

Task5

Create TJob#4: Prepare dataset for desired datasets



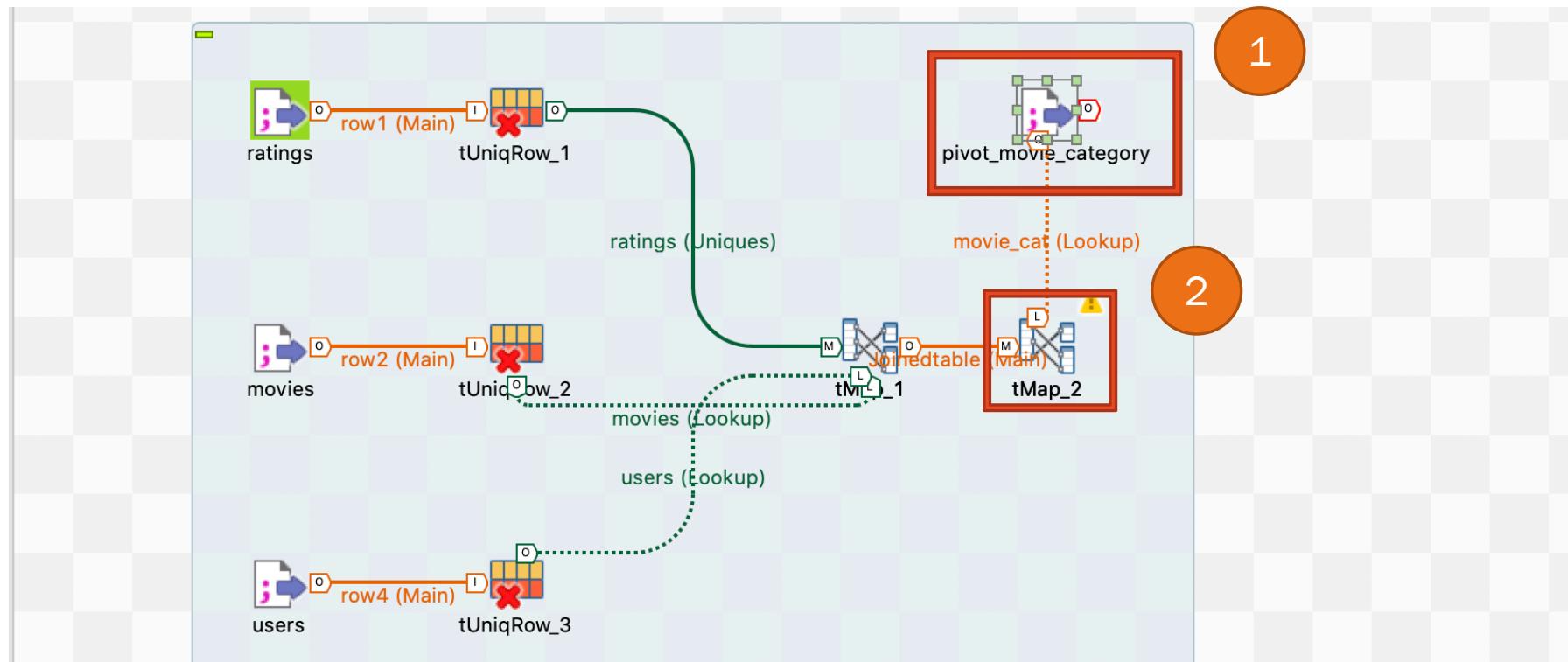
- Add tLogRow to check output
- Example Output:
 - MovieID
 - UserID
 - Gender2
 - Age
 - Occupation
 - Rating

Task5

Create TJob#4: Prepare dataset for desired datasets

➤ Step 4.7: Merge output of 4.1 and 4.6

- Add File Delimited Metadata “pivot_movie_category” to merge the output and merge output at tMap_2
- Double click at tMap_2



Task5

Create TJob#4: Prepare dataset for desired datasets

- Step 4.7: Merge output of 4.1 and 4.6 for Prediction Model

Output for Prediction Model

The screenshot shows the Talend Data Integration environment with three main components:

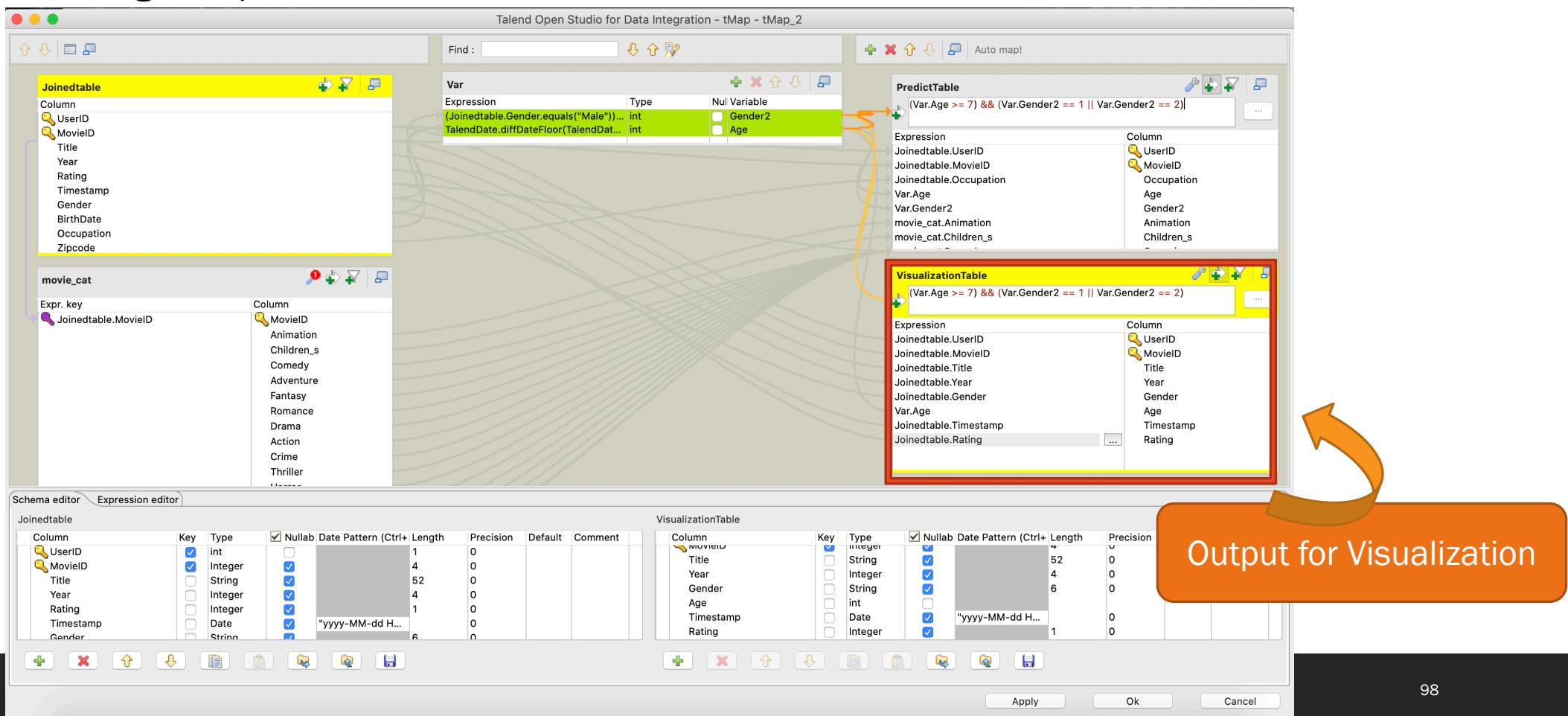
- Schema editor:** Displays the schema for the **movie_cat** table, which includes columns like MovieID, Animation, Children_s, Comedy, Adventure, Fantasy, Romance, Drama, Action, Crime, Thriller, Horror, Sci_Fi, Documentary, War, Musical, Mystery, and Film_Noir.
- Expression editor:** Shows a variable named **Var** with two expressions:
 - (Joinedtable.Gender.equals("Male")... int
 - TalendDate.diffDateFloor(TalendDate... intand a null variable **Var.Age**.
- PredictTable:** A table where rows from the **movie_cat** table are mapped to columns in the **PredictTable**. The mapping is as follows:

Joinedtable.MovieID	Var.Gender2	Var.Age	Joinedtable.Occupation	Column
movie_cat.Animation			movie_cat.Animation	Animation
movie_cat.Children_s			movie_cat.Children_s	Children_s
movie_cat.Comedy			movie_cat.Comedy	Comedy
movie_cat.Adventure			movie_cat.Adventure	Adventure
movie_cat.Fantasy			movie_cat.Fantasy	Fantasy
movie_cat.Romance			movie_cat.Romance	Romance
movie_cat.Drama			movie_cat.Drama	Drama
movie_cat.Action			movie_cat.Action	Action
movie_cat.Crime			movie_cat.Crime	Crime
movie_cat.Thriller			movie_cat.Thriller	Thriller
movie_cat.Horror			movie_cat.Horror	Horror
movie_cat.Sci_Fi			movie_cat.Sci_Fi	Sci_Fi
movie_cat.Documentary			movie_cat.Documentary	Documentary
movie_cat.War			movie_cat.War	War
movie_cat.Musical			movie_cat.Musical	Musical
movie_cat.Mystery			movie_cat.Mystery	Mystery
movie_cat.Film_Noir			movie_cat.Film_Noir	Film_Noir
Joinedtable.Rating				Rating

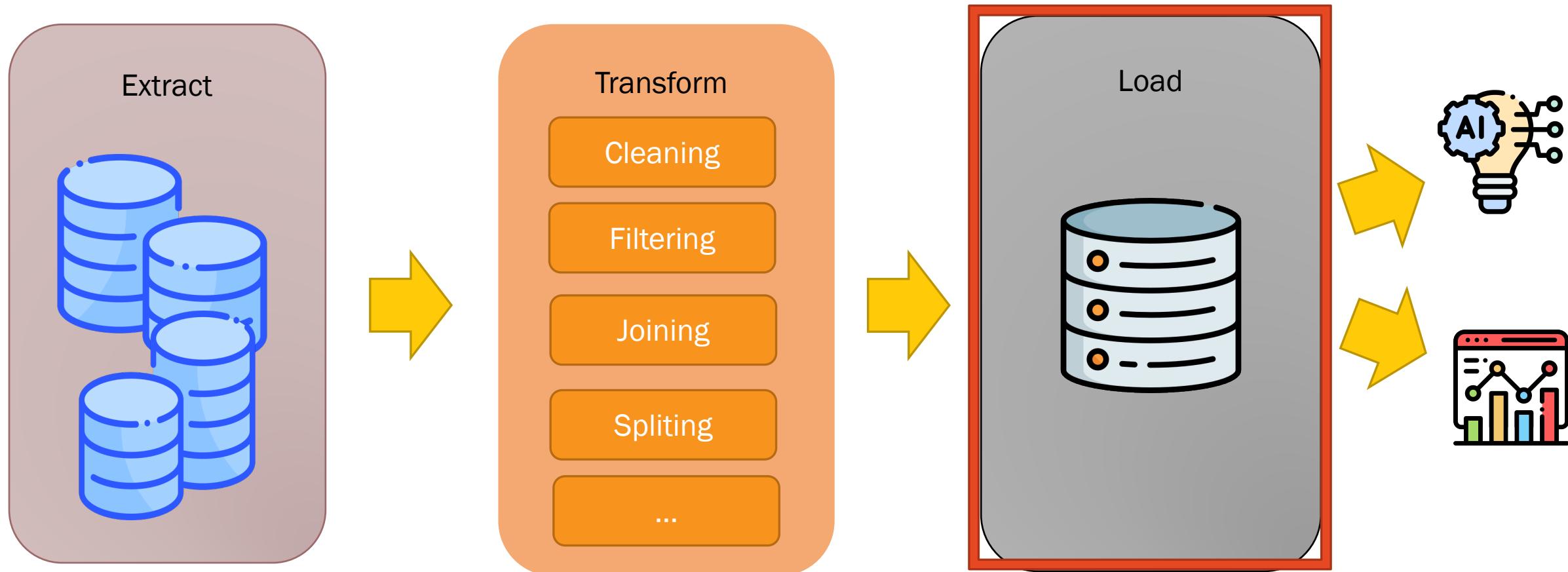
Task5

Create TJob#4: Prepare dataset for desired datasets

- Step 4.7: Merge output of 4.1 and 4.6 for Visualization



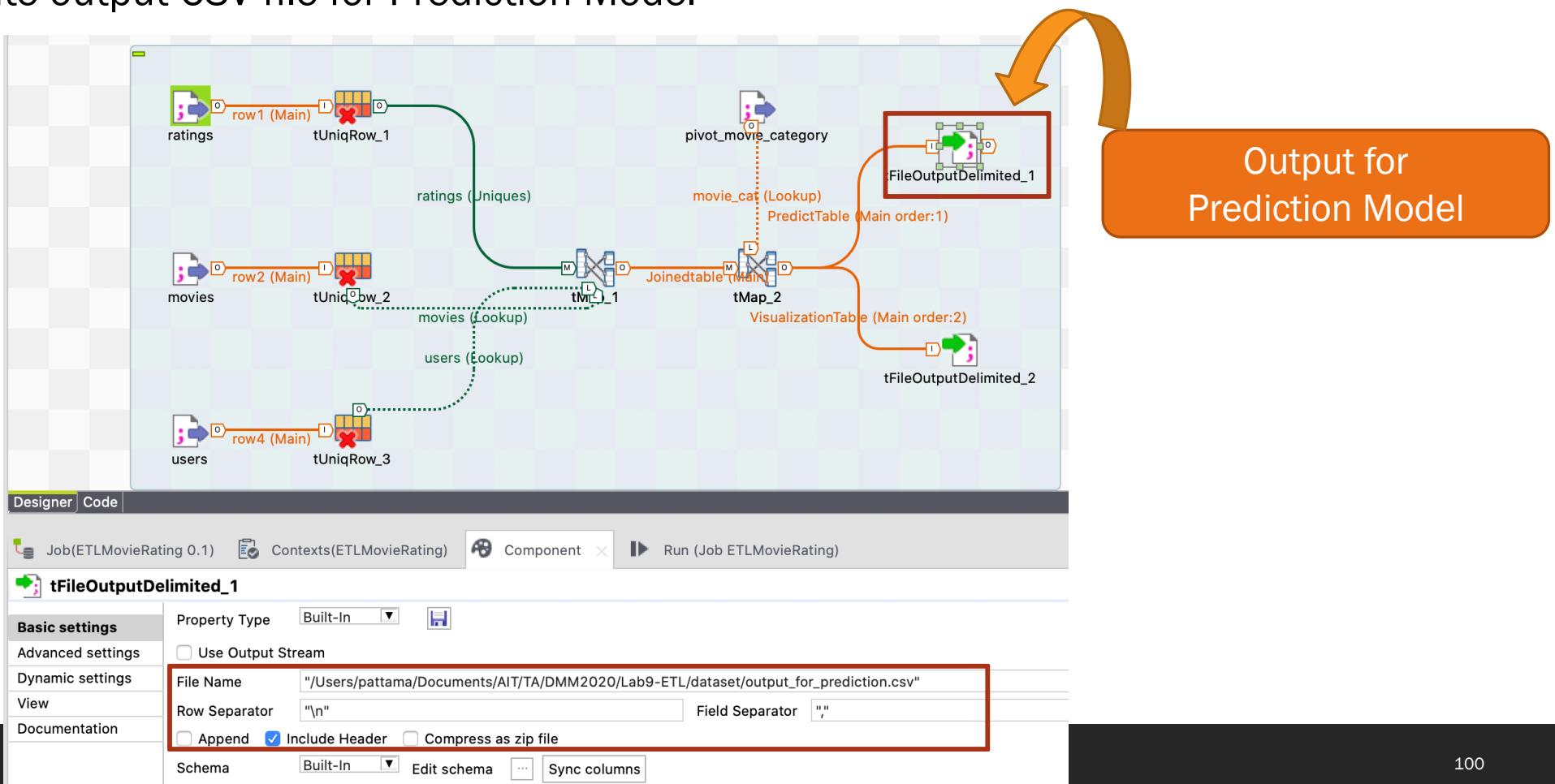
ETL Process



Task5

Create TJob#4: Prepare dataset for desired datasets

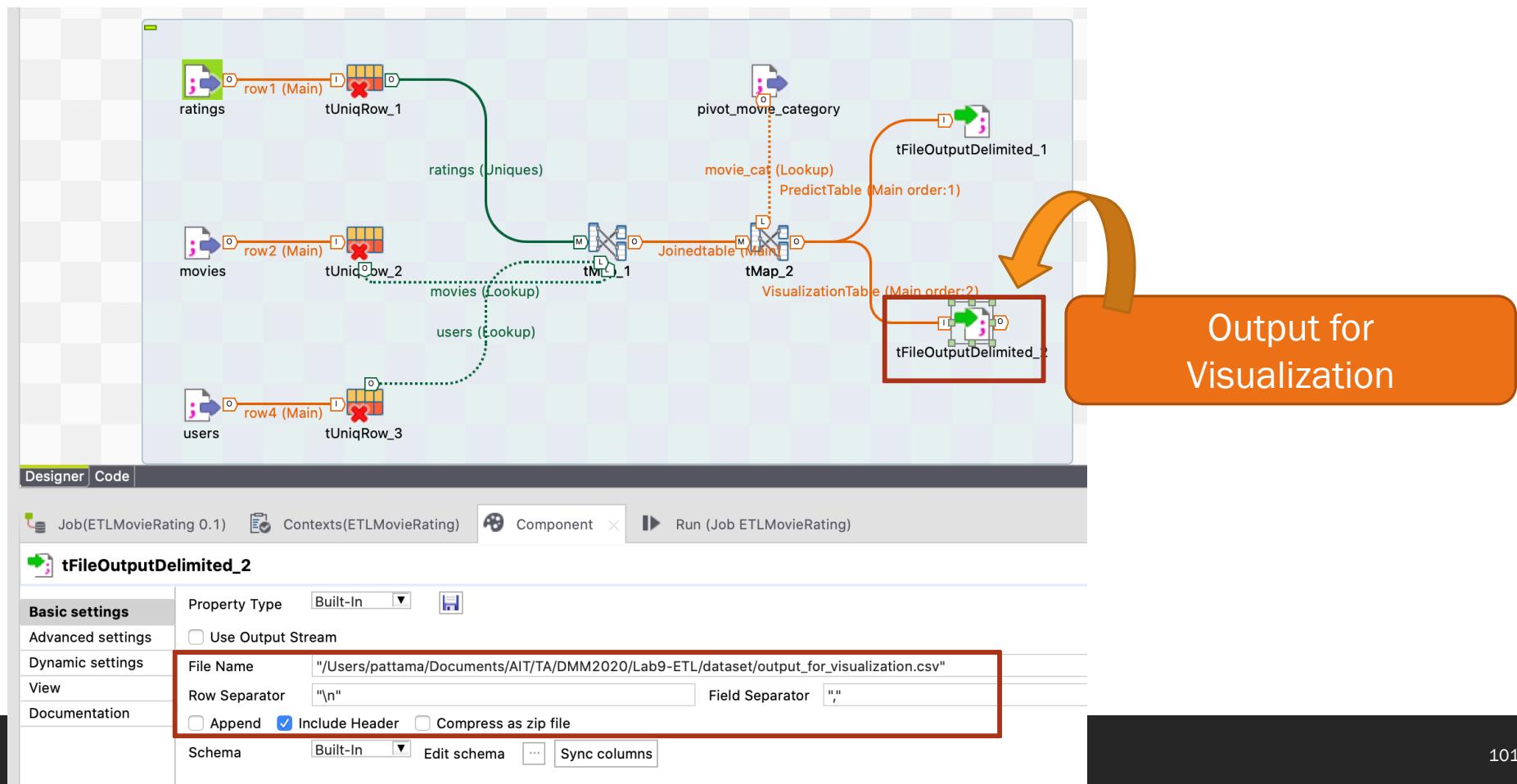
- Step 5: Write output CSV file for Prediction Model



Task5

Create TJob#4: Prepare dataset for desired datasets

- Step 5: Write output CSV file for Visualization



Output of Task5

- 1.** Dataset for Prediction Model “Movie Rating Prediction ”
- 2.** Dataset for Visualization

Output of Task5

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
UserID	MovieID	Gender2	Age	action	comedy	romantic	fantacy	Fantasy	Romance	Drama	Action_1	Crime	Thriller	Horror	Sci_Fi												
2	1357	1	50	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	3068	1	50	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	1537	1	50	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	647	1	49	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	2194	1	50	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	
2	648	1	50	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	2268	1	49	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
2	2628	1	50	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	
2	1103	1	50	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	2916	1	50	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	
2	3468	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	1210	1	49	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	
2	1792	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	
2	1687	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	
2	1213	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	
2	3578	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	
2	2881	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	
2	3030	1	49	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	1217	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	3105	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	
2	434	1	50	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
2	2126	1	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	
2	3107	1	50	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	
2	3108	1	50	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Output for
Prediction Model

Output of Task5

A	B	C	D	E	F	G	H	I
UserID	MovielID	Title	Year	Gender	Gender2	Age	Timestamp	Rating
2	1357	Shine (1996)	1996	Male	1	50	2015-10-18 15:24:38	5
2	3068	Verdict, The (1982)	1982	Male	1	50	2015-01-14 15:31:09	4
2	1537	Shall We Dance? (Shall We Dansu?) (1996)	1996	Male	1	50	2015-05-22 09:03:30	4
2	647	Courage Under Fire (1996)	1996	Male	1	49	2014-02-21 07:02:49	3
2	2194	Untouchables, The (1987)	1987	Male	1	50	2015-10-25 03:23:12	4
2	648	Mission: Impossible (1996)	1996	Male	1	50	2015-01-07 10:58:19	4
2	2268	Few Good Men, A (1992)	1992	Male	1	49	2014-05-27 15:12:42	5
2	2628	Star Wars: Episode I - The Phantom Menace (1999)	1999	Male	1	50	2015-11-03 14:11:03	3
2	1103	Rebel Without a Cause (1955)	1955	Male	1	50	2015-06-07 05:36:56	3
2	2916	Total Recall (1990)	1990	Male	1	50	2015-09-19 18:07:05	3
2	3468	Hustler, The (1961)	1961	Male	1	50	2015-08-17 09:42:45	5
2	1210	Star Wars: Episode VI - Return of the Jedi (1983)	1983	Male	1	49	2014-02-09 14:01:03	4
2	1792	U.S. Marshalls (1998)	1998	Male	1	50	2015-07-02 00:47:44	3
2	1687	Jackal, The (1997)	1997	Male	1	49	2014-05-11 10:57:57	3
2	1213	GoodFellas (1990)	1990	Male	1	50	2014-11-28 21:19:25	2
2	3578	Gladiator (2000)	2000	Male	1	50	2015-06-24 00:51:45	5
2	2881	Double Jeopardy (1999)	1999	Male	1	50	2014-12-29 21:27:23	3
2	3030	Yojimbo (1961)	1961	Male	1	49	2014-10-09 16:27:37	4
2	1217	Ran (1985)	1985	Male	1	49	2014-03-12 16:41:31	3
2	3105	Awakenings (1990)	1990	Male	1	50	2015-10-17 05:59:39	4
2	434	Cliffhanger (1993)	1993	Male	1	50	2015-10-03 23:12:06	2
2	2126	Snake Eyes (1998)			1	49	2014-02-04 15:38:02	3
2	2127	Reindeer Games (2000)			1	50	2014-10-22 20:11:11	2

Output for
Visualization

References

1. <https://www.talend.com/resources/discovering-talend-studio/>
2. <https://www.tutorialspoint.com/talend/index.htm>



Thank you.
