



AT82.02

---

## DATA MODELING AND MANAGEMENT

---

### LAB9: DATA ENGINEERING

# Outline

---

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

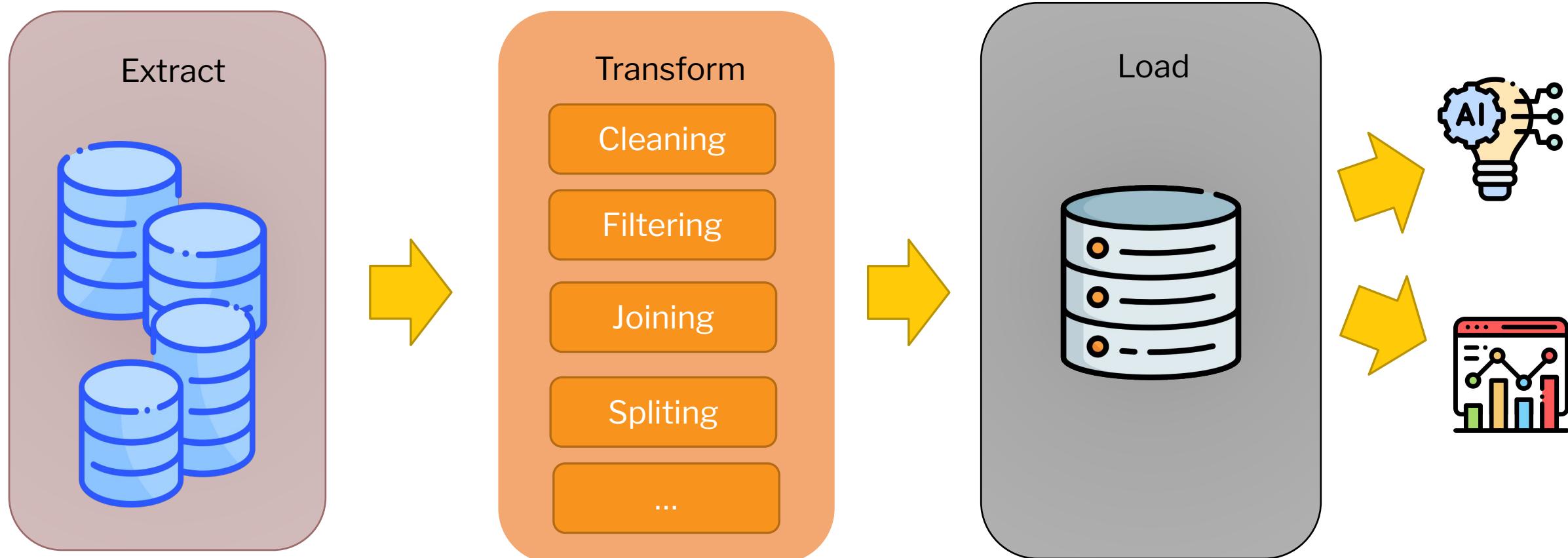
# Extract Transform Load

---

- **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

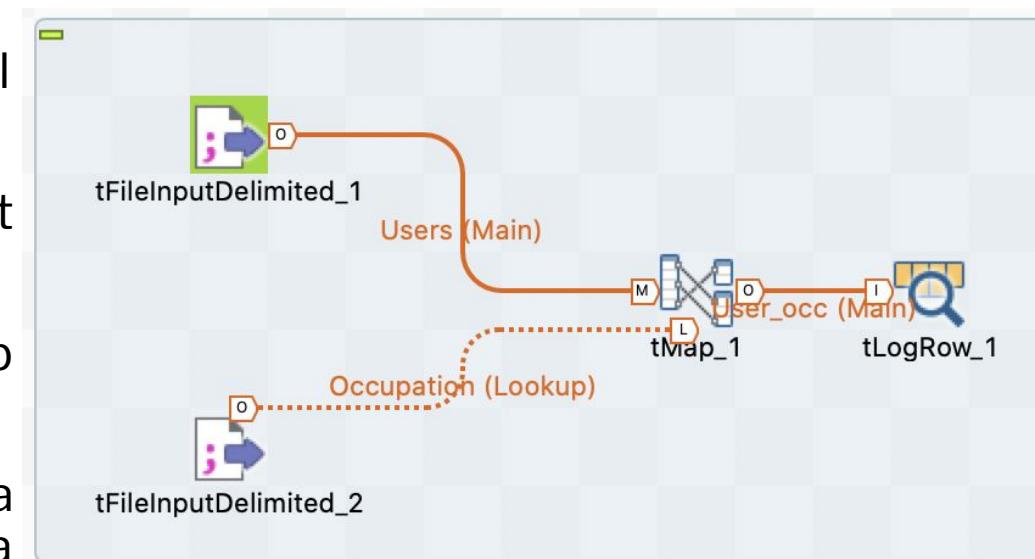
---

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



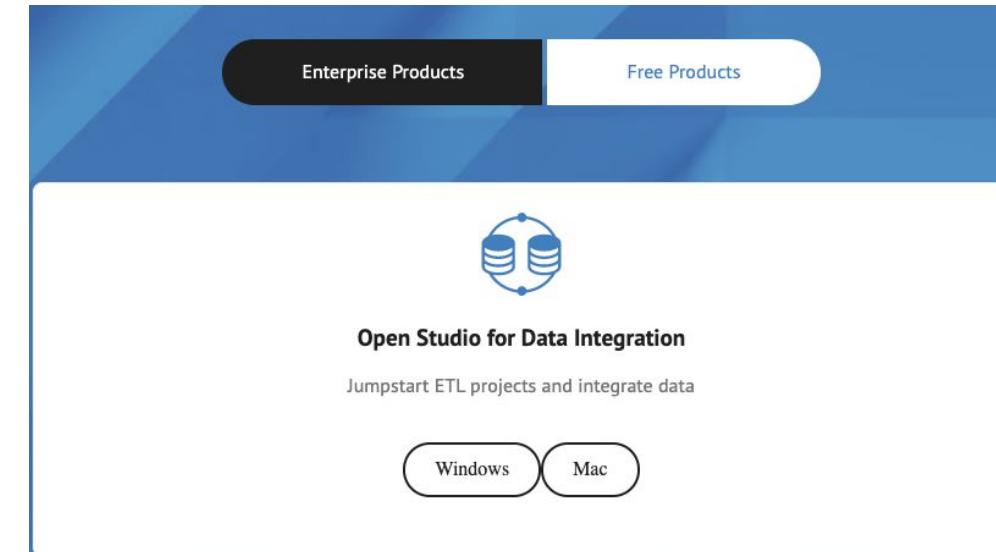
# Talend Installation

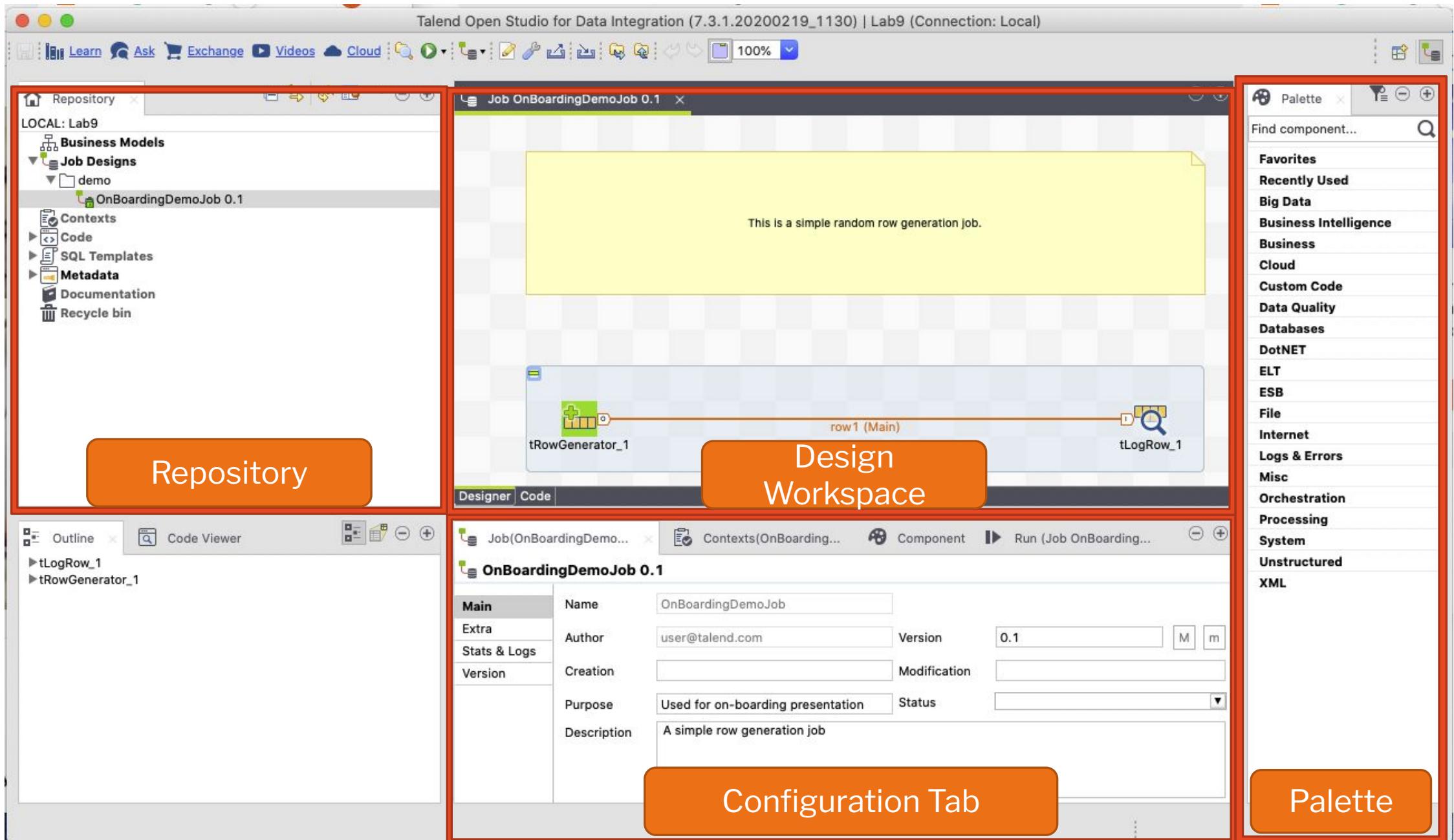
## **Open Studio for Data Integration**

<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	<b>tFileInputDelimited</b>	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 into 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	<b>tLogRow</b>	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	<b>tMap</b>	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	<b>tUniqRow</b>	Compares entries and sorts out duplicate entries from the input flow.

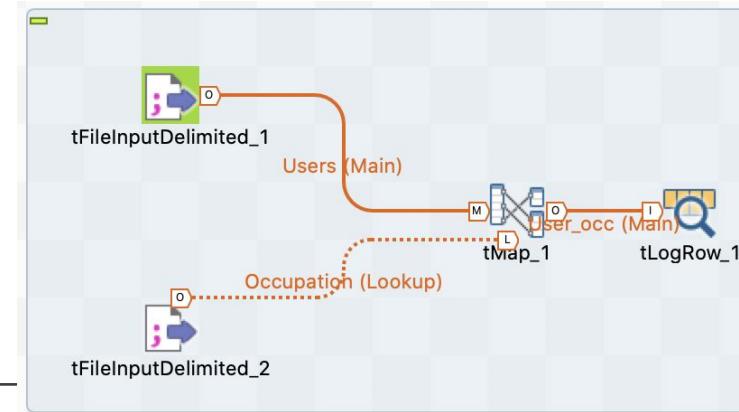
# Definition

# Job Design

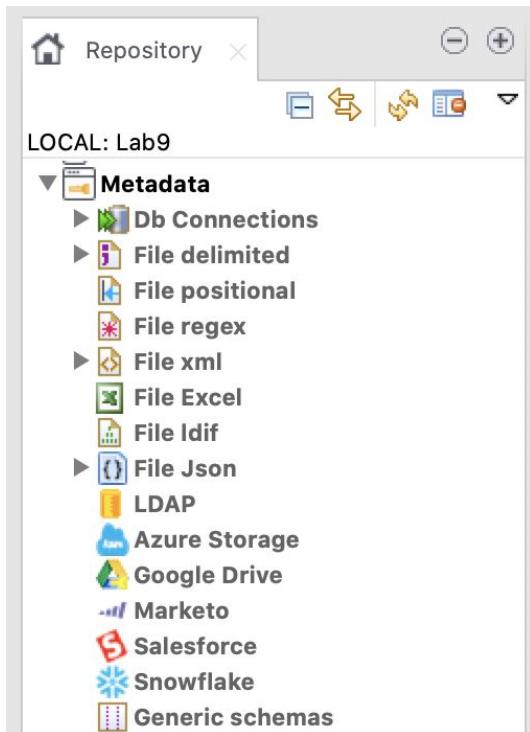
- This is the graphical representation of the business model.
  - In this design, one or more components are connected with each other to run a data integration process.

# 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.



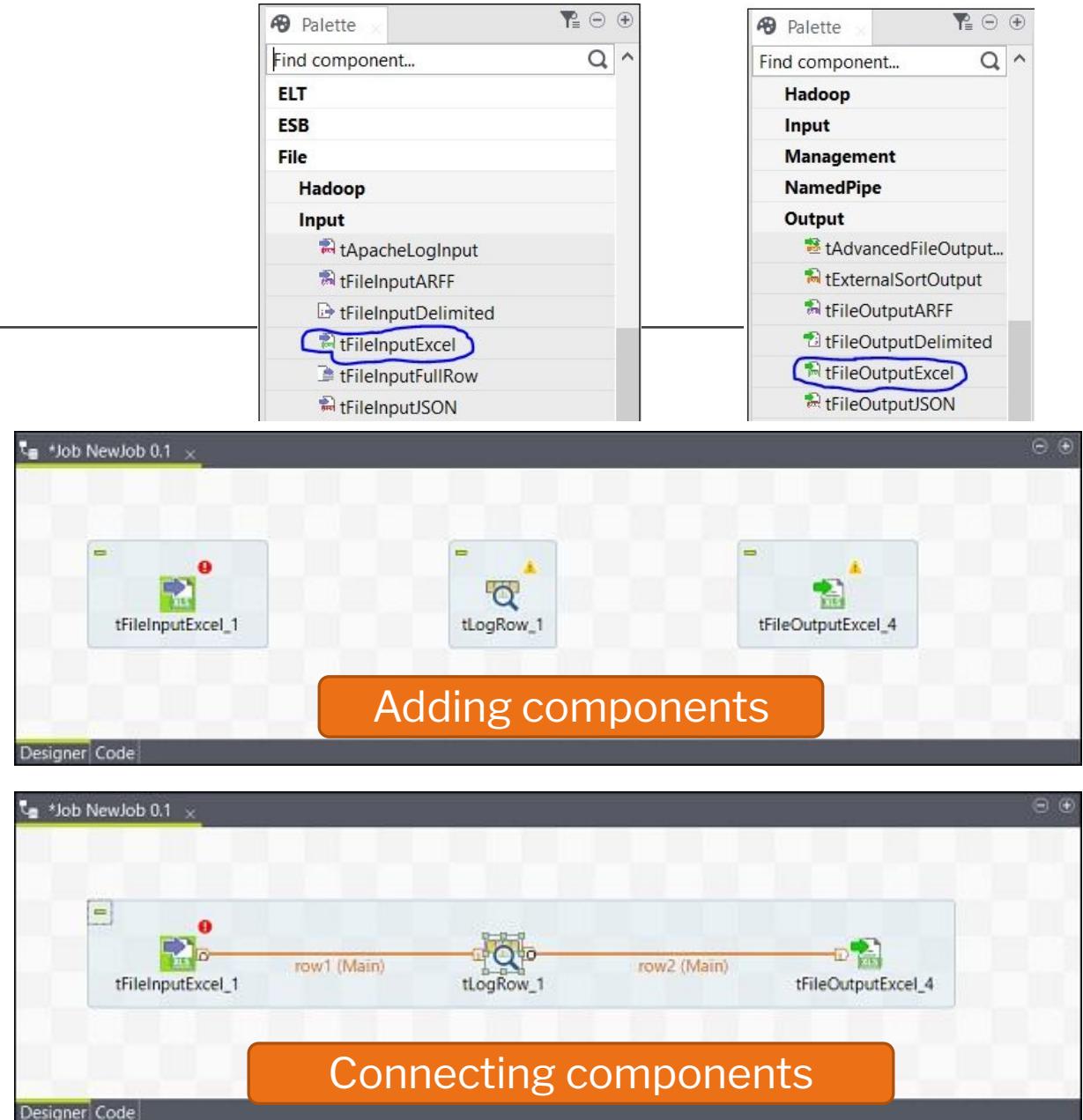
## Example Job



## 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 a three-step process for defining metadata:

- 1) **Select type of Metadata**: The "Metadata" section is selected in the left sidebar. A large orange arrow points from this step to the "File - Step 3 of 4" configuration screen.
- 2) **Set Configuration of the data**: The "File Settings" and "Rows To Skip" sections are visible on the right. A large orange arrow points from this step to the "File - Step 4 of 4" configuration screen.
- 3) **Set Schema**: The "Schema" section is shown on the right, displaying column definitions and data patterns. An orange box highlights the "Set Schema" button at the bottom of this screen.

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

**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  
Escape Char Settings: CSV  
Text Enclosure: ""

Rows To Skip

If any rows must be ignored, specify the following parameters  
Header: 1  
Footer:  
Skip empty row

Limit Of Rows

Preview | Output

UserID Gender BirthDate Occupation Zipcode

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

Set heading row as column names Refresh Preview

Set configuration

# 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

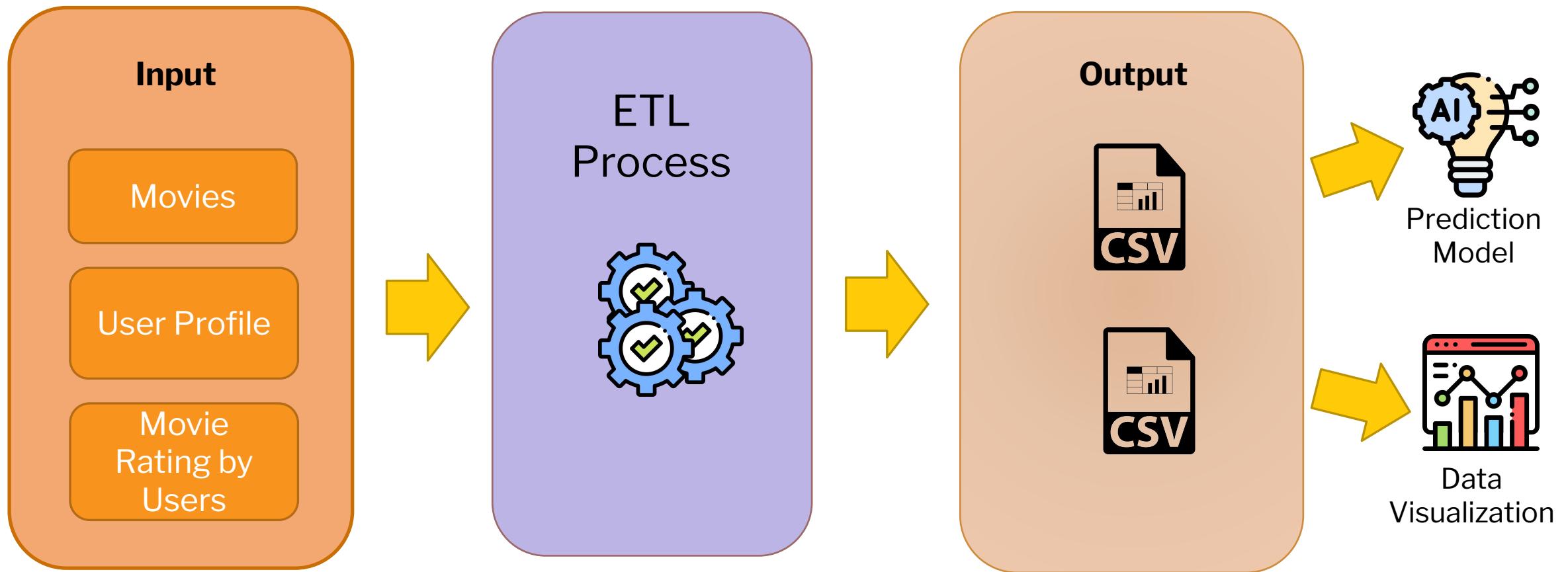
The variables used for **input**:

- Age , Gender , Occupation, Movie Category

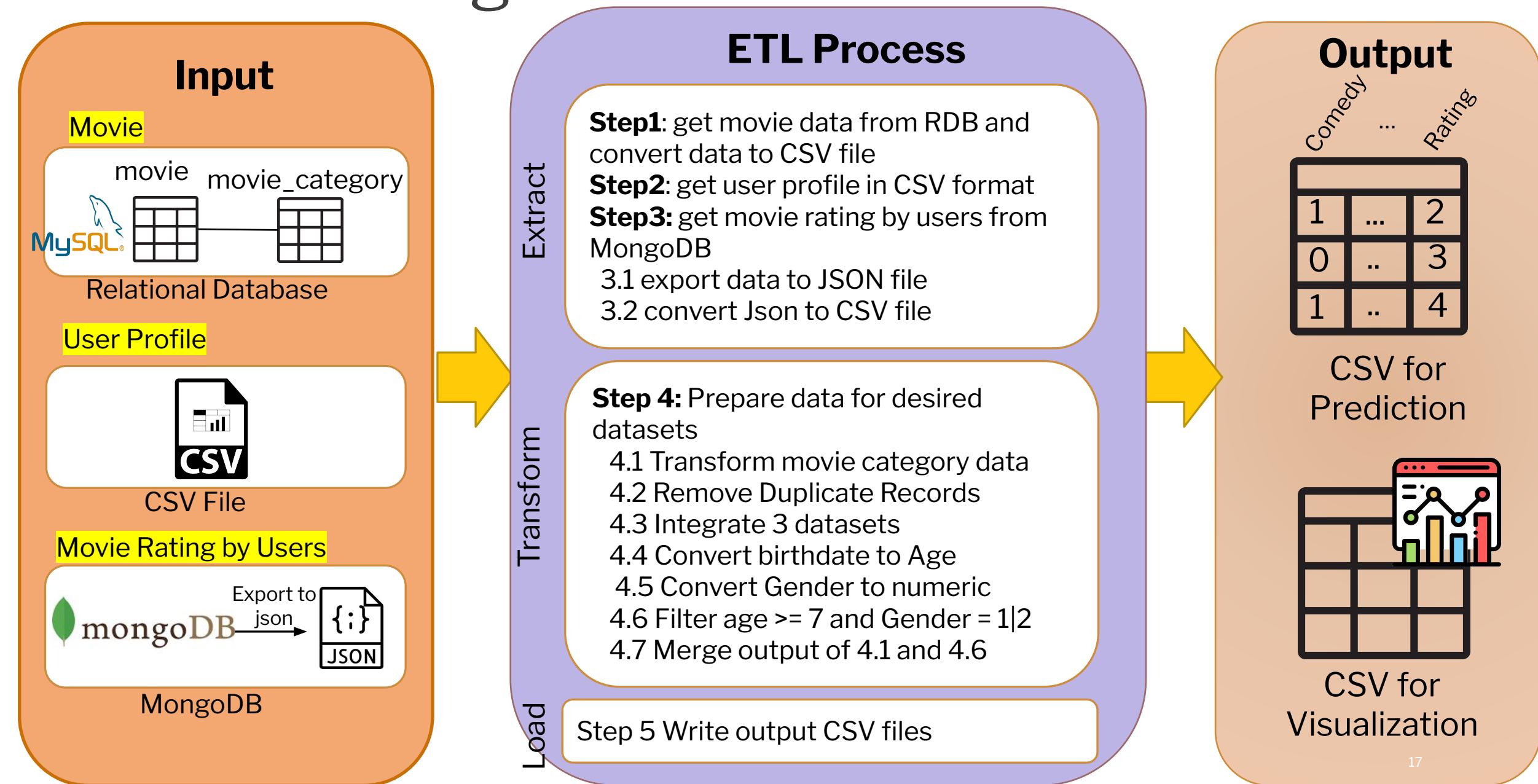
The variables used for **model prediction**:

- User Rating

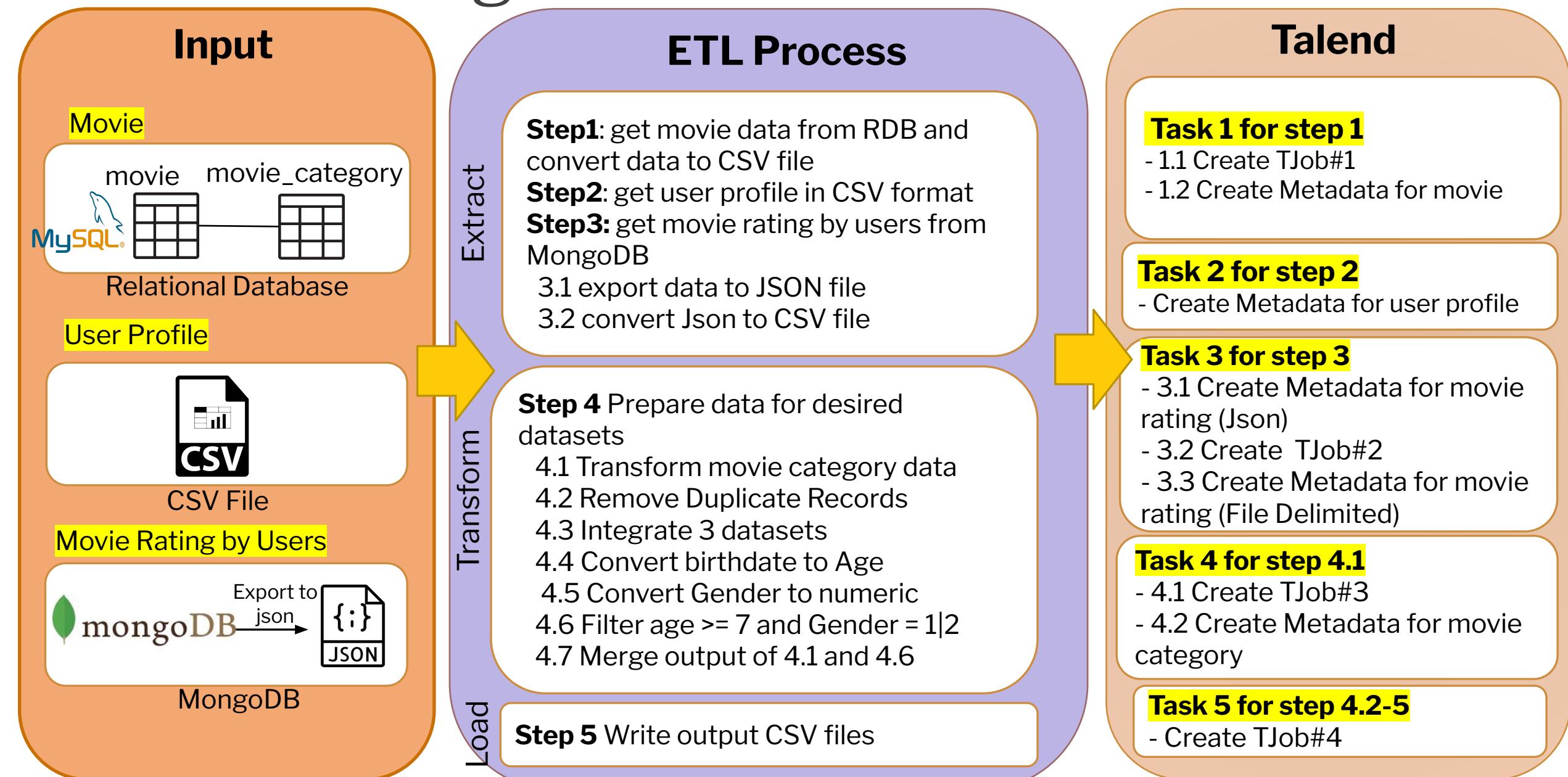
# Movie Rating Prediction



# Movie Rating Prediction



# Movie Rating Prediction



# Desired Output for Prediction

---

UserID	MovielID	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

Ratings

# Desired Output for Visualization

---

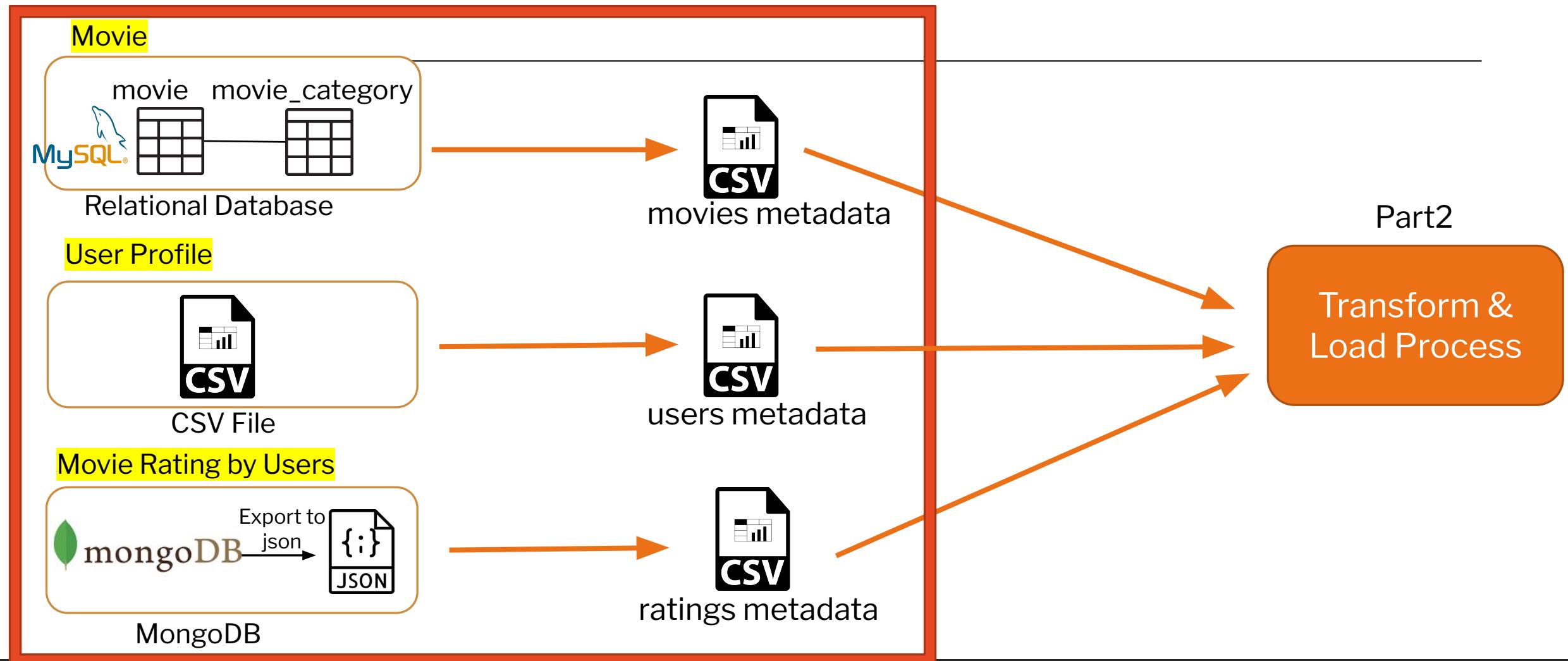
A	B	C	D	E	F	G	H	I
UserID	MovieID	Title	Year	Gender	Gender2	Age	Timestamp	Rating
2	1357	Shine (1996)	1996	Male	1	50	2015-10-18 15:24:38	5
2	3068	Verdict, The (1982)	1982	Male	1	50	2015-01-14 15:31:09	4
2	1537	Shall We Dance? (Shall We Dansu?) (1996)	1996	Male	1	50	2015-05-22 09:03:30	4
2	647	Courage Under 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	2127	Die Hard (1988)	1988	Male	1	50	2014-10-09 22:11:14	2

---

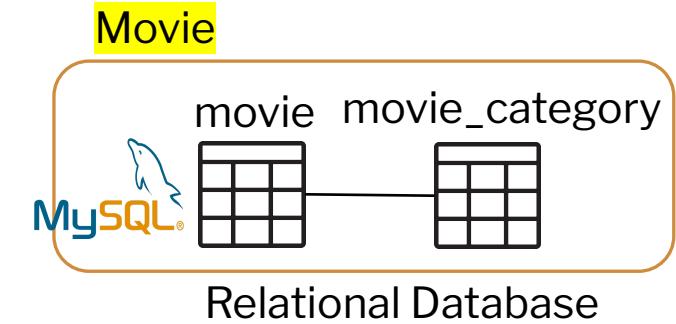
# Part 1

---

# Goal of Part 1



# Task 1 for step 1



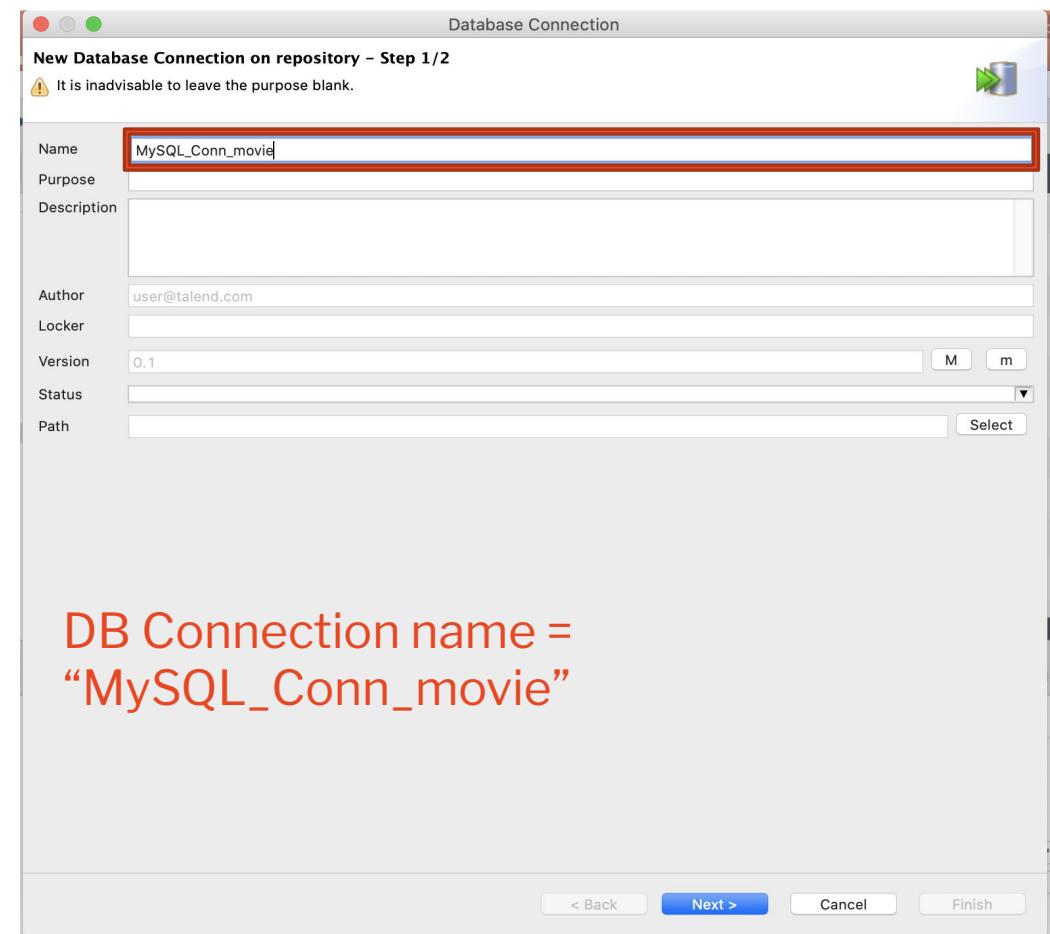
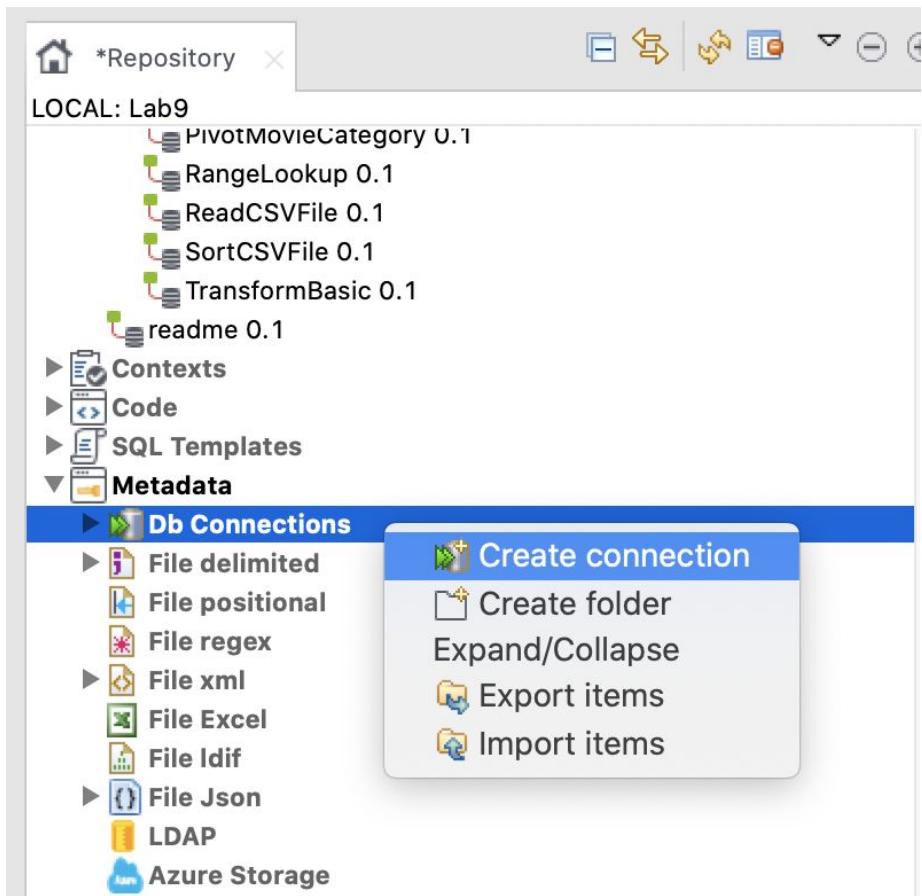
**Step 1:** Get movie data from RDB and convert data to CSV file

## Task 1:

- 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)

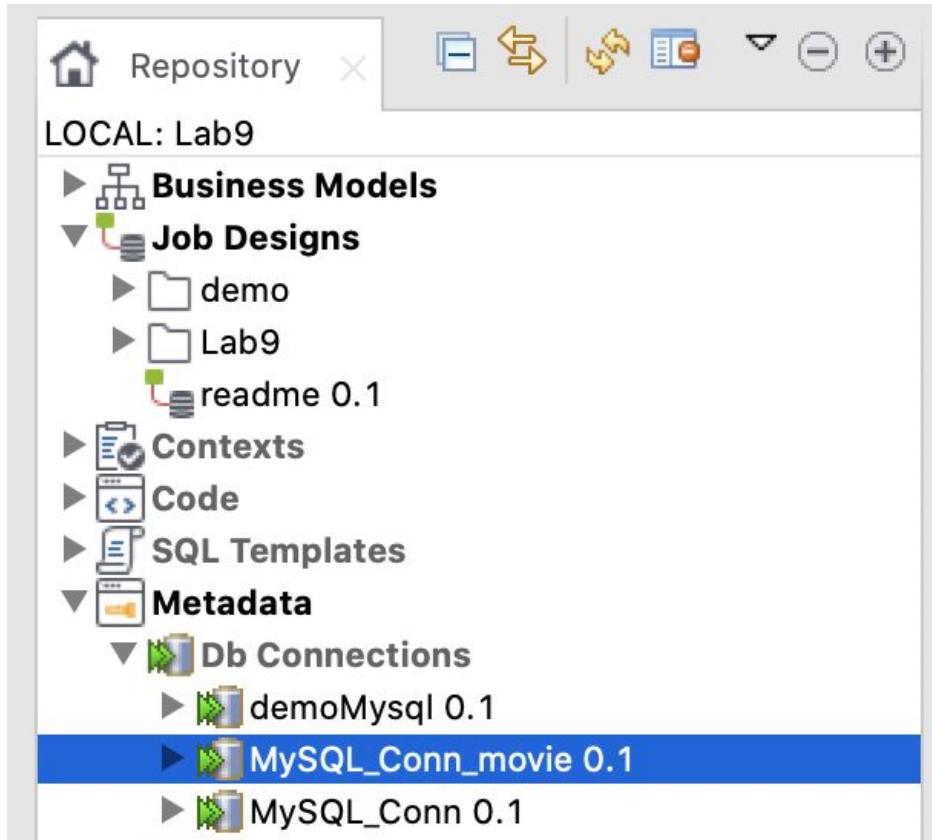
# Task 1

## 1.1 Create Metadata for DB Connection (1)



# Task 1

## 1.1 Create Metadata for DB Connection (2)



**Server:**

sql12.freemysqlhosting.net

**DB Name:** <dbname>

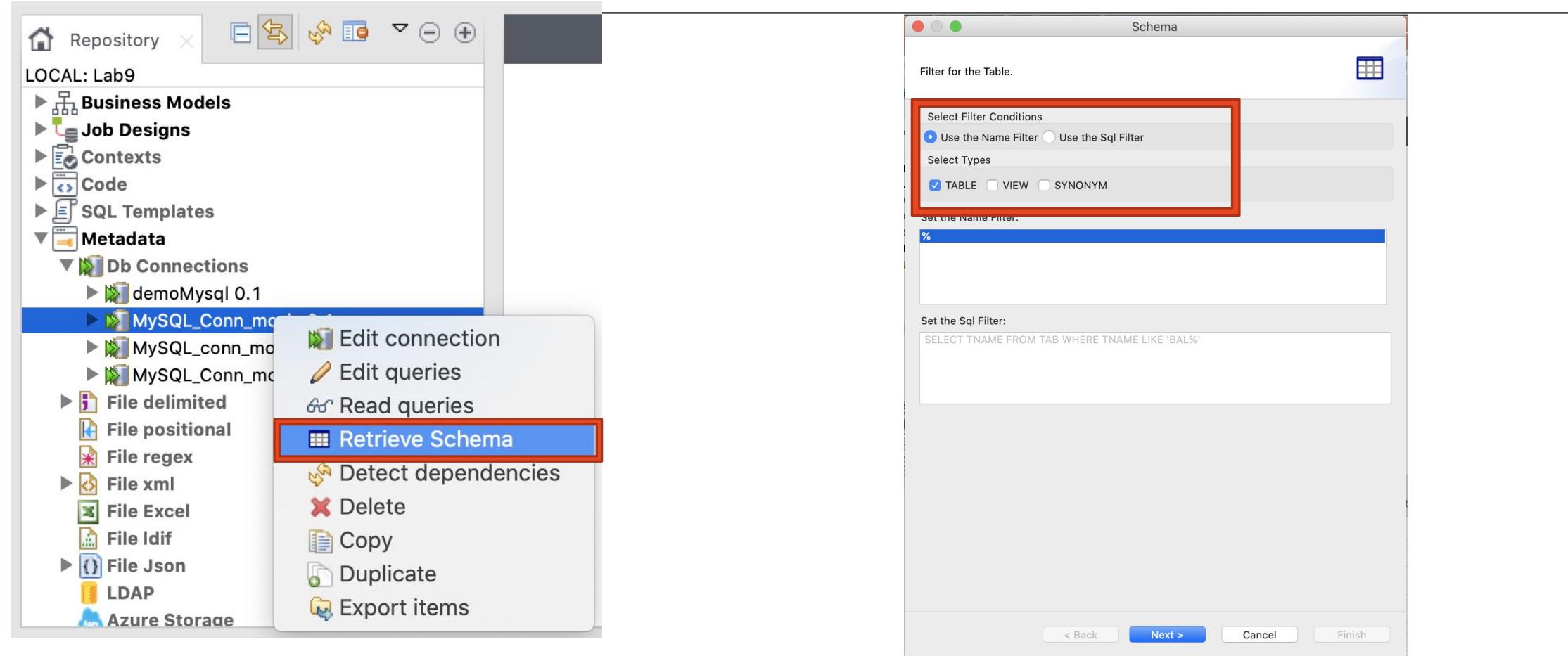
**Username:** <userID>

**Password:** <pw>

**Port number:** 3306

# Task 1

## 1.1 Create Metadata for DB Connection (3)



# Task 1

## 1.1 Create Metadata for DB Connection (4)

The screenshot shows two windows of MySQL Workbench's Schema editor.

**Left Window:** "Select Schema to create". It displays a table with columns "Name" and "Type". The table contains three rows: "sql12375315" (Type: CATALOG), "movie\_category" (Type: TABLE), and "movies" (Type: TABLE). The "movie\_category" and "movies" rows are selected and highlighted with a red border.

Name	Type
sql12375315	CATALOG
movie_category	TABLE
movies	TABLE

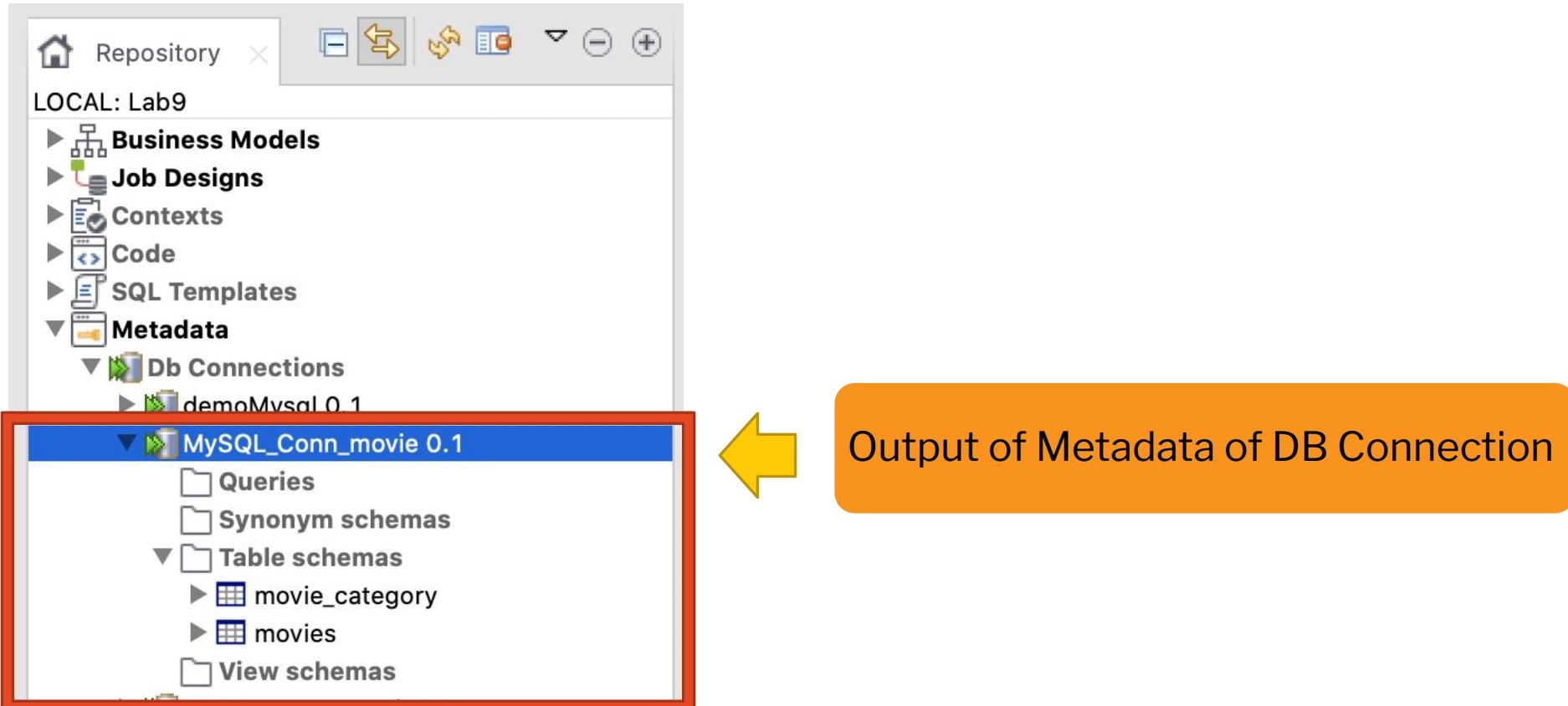
**Right Window:** "New Schema in connection "MySQL\_Conn\_movie3""

- Schema:** A list of existing schemas: "movie\_category" and "movies".
- Name:** "movies" (highlighted with a blue selection bar).
- Comment:** (empty text field)
- Type:** TABLE
- Based on table:** "movies" (dropdown menu, "Retrieve" button next to it).
- Note:** "Use the "Retrieve Schema" button to replace the current Schema by the table."
- Schema:** A table showing the structure of the "movies" table.

Column	Db Column	Key	DB Type	Type	N
MovieID	MovieID	✓	INT	Inte...	✓
Title	Title	✗	VARCHAR	String	✓
Year	Year	✗	INT	Inte...	✓

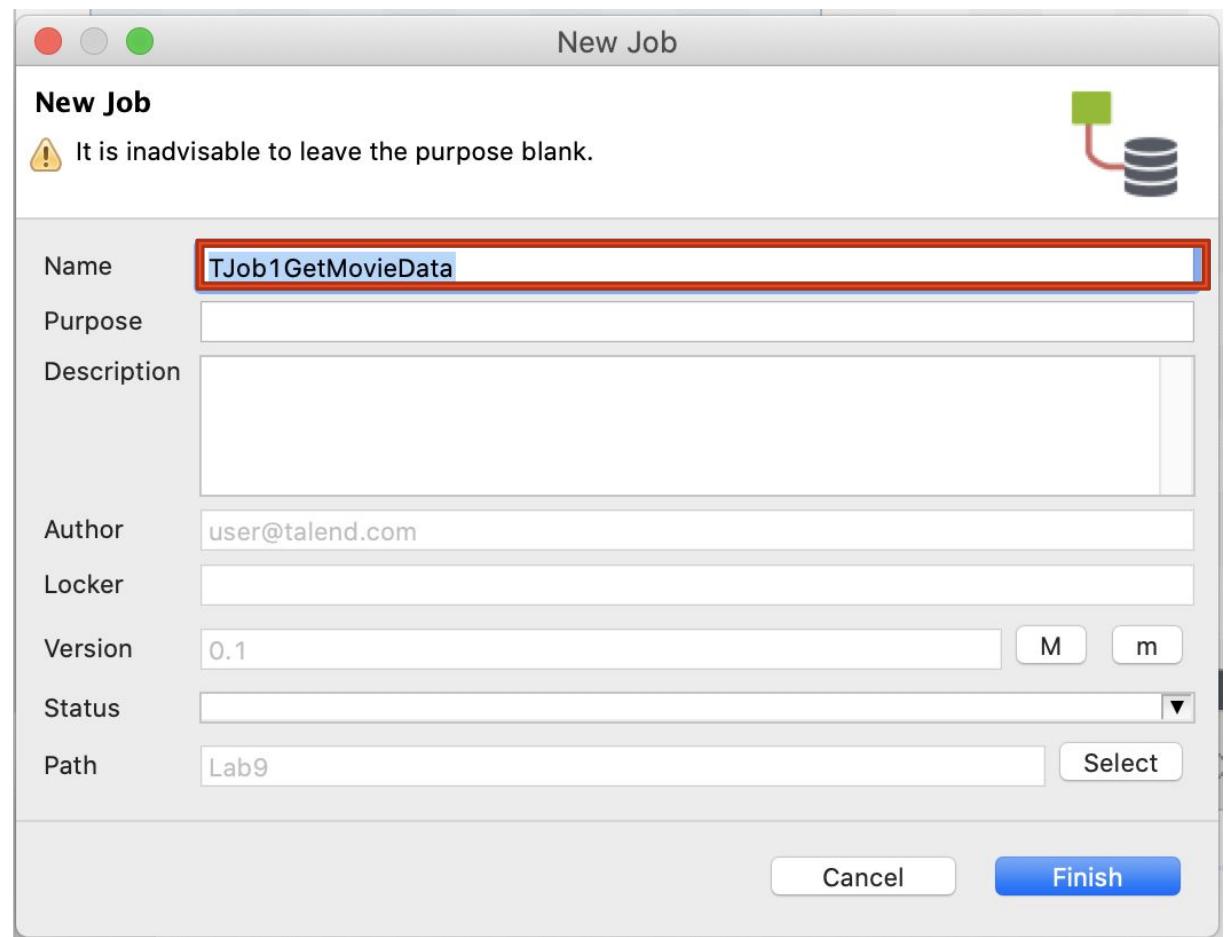
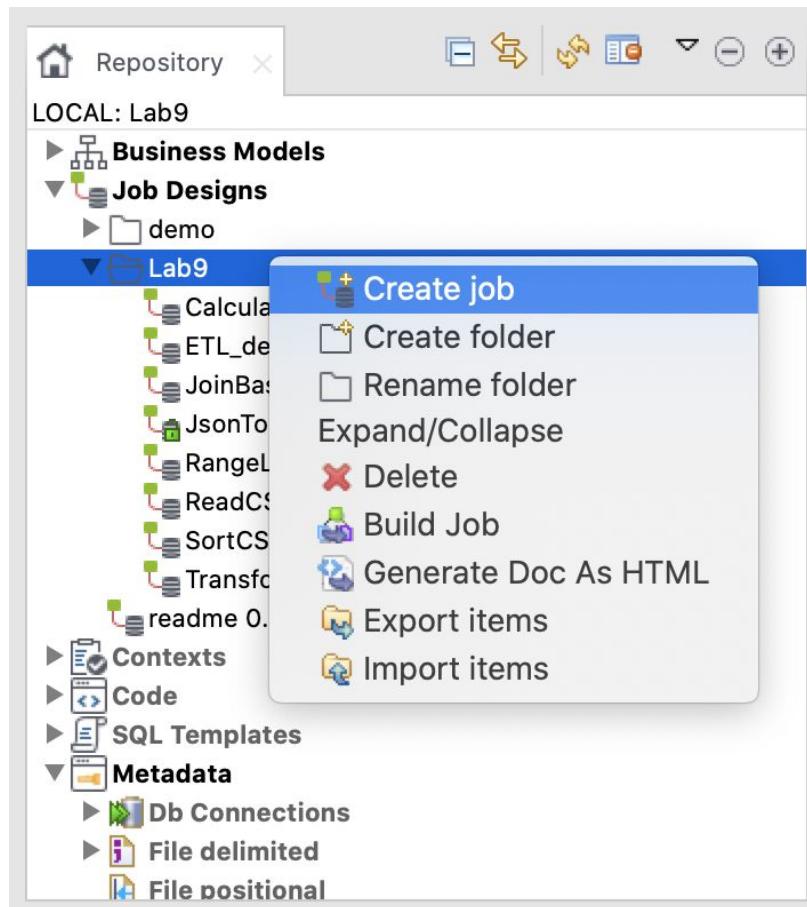
# Task 1

## 1.1 Create Metadata for DB Connection (5)



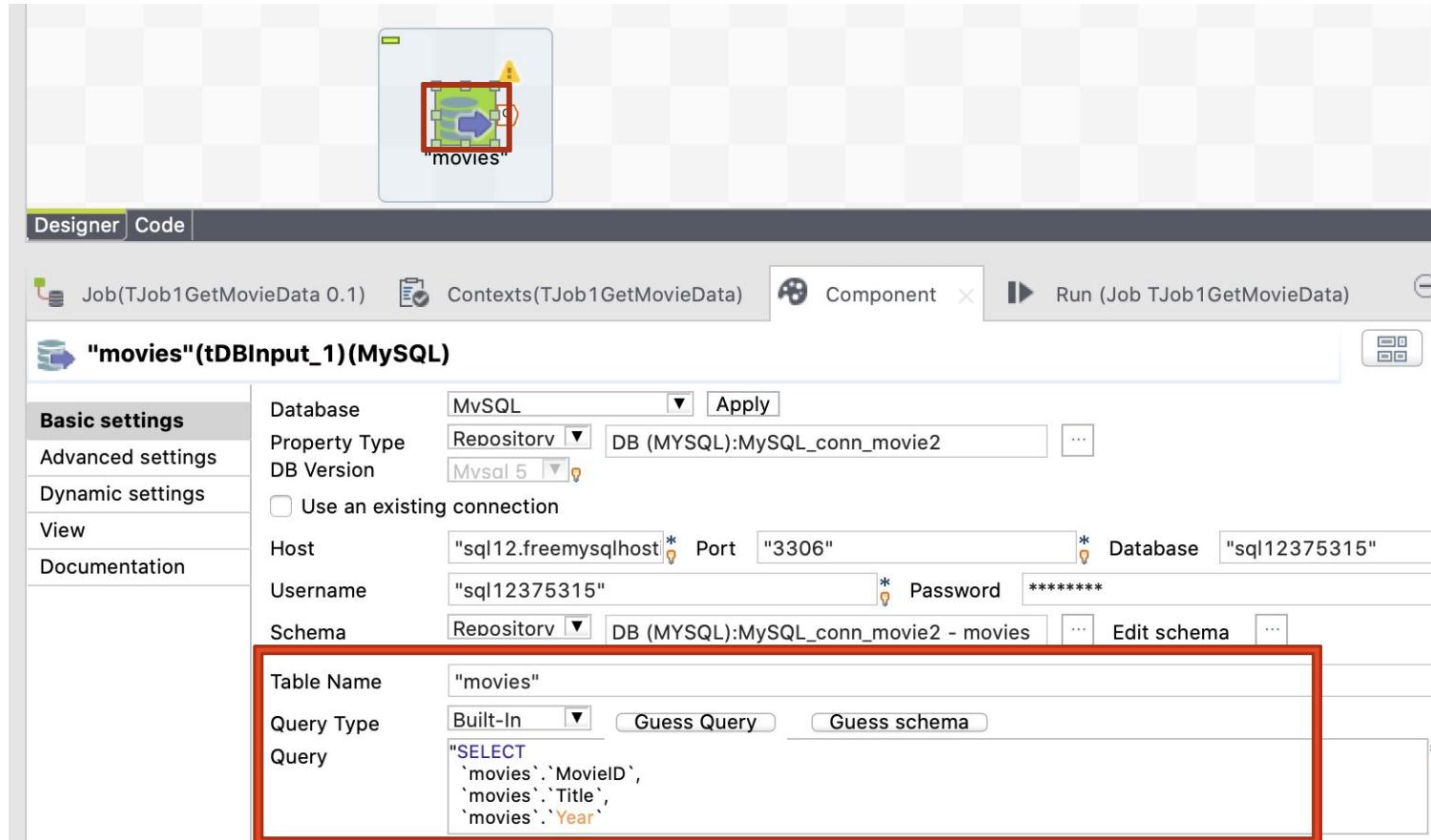
# Task 1

## 1.1 Create TJob#1: get movie data and explore



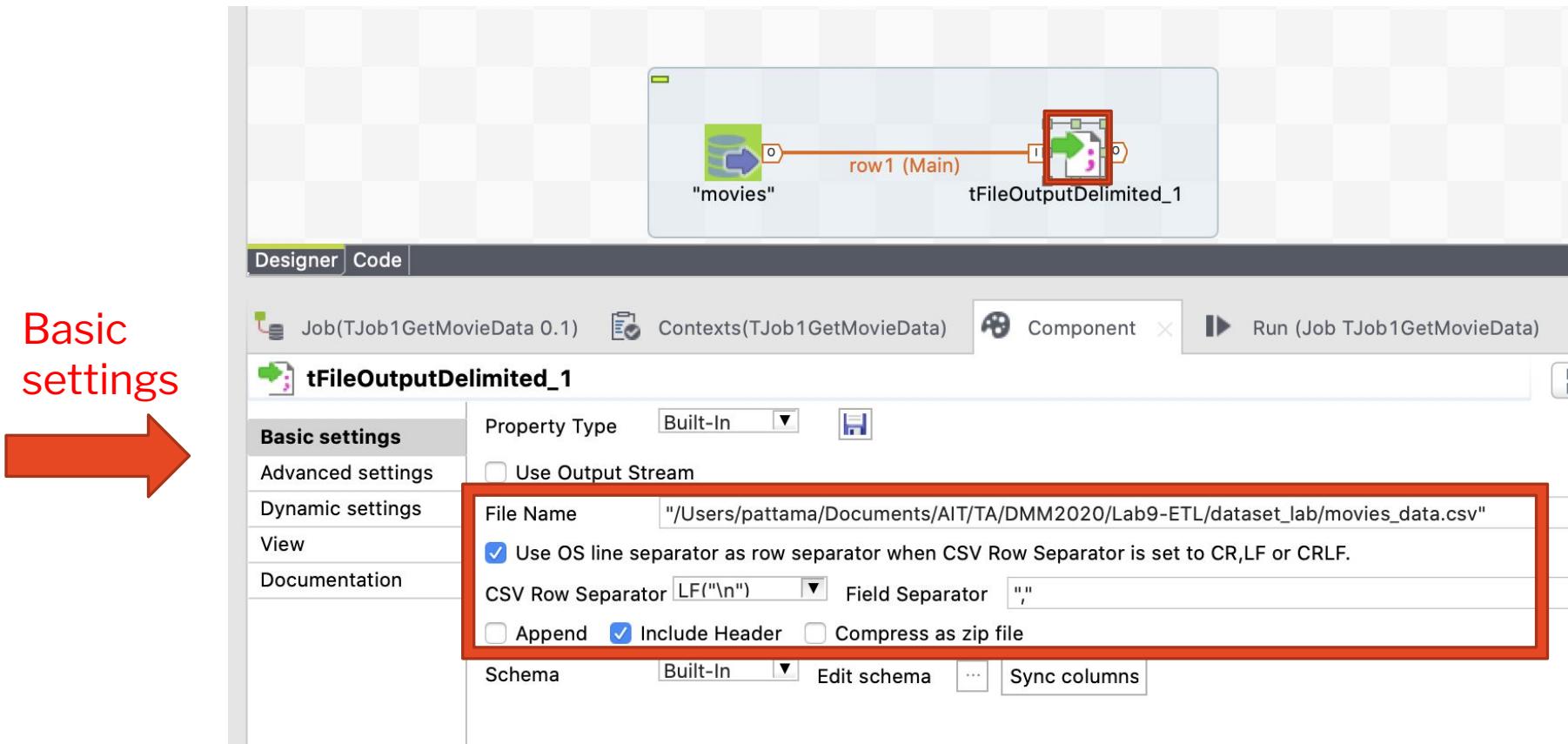
# Task 1

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



# Task1

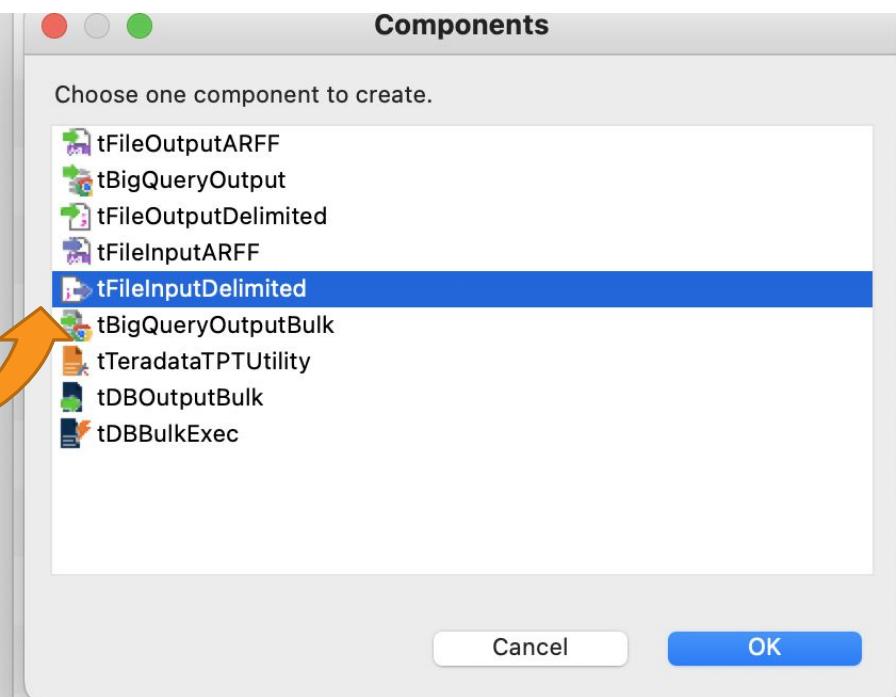
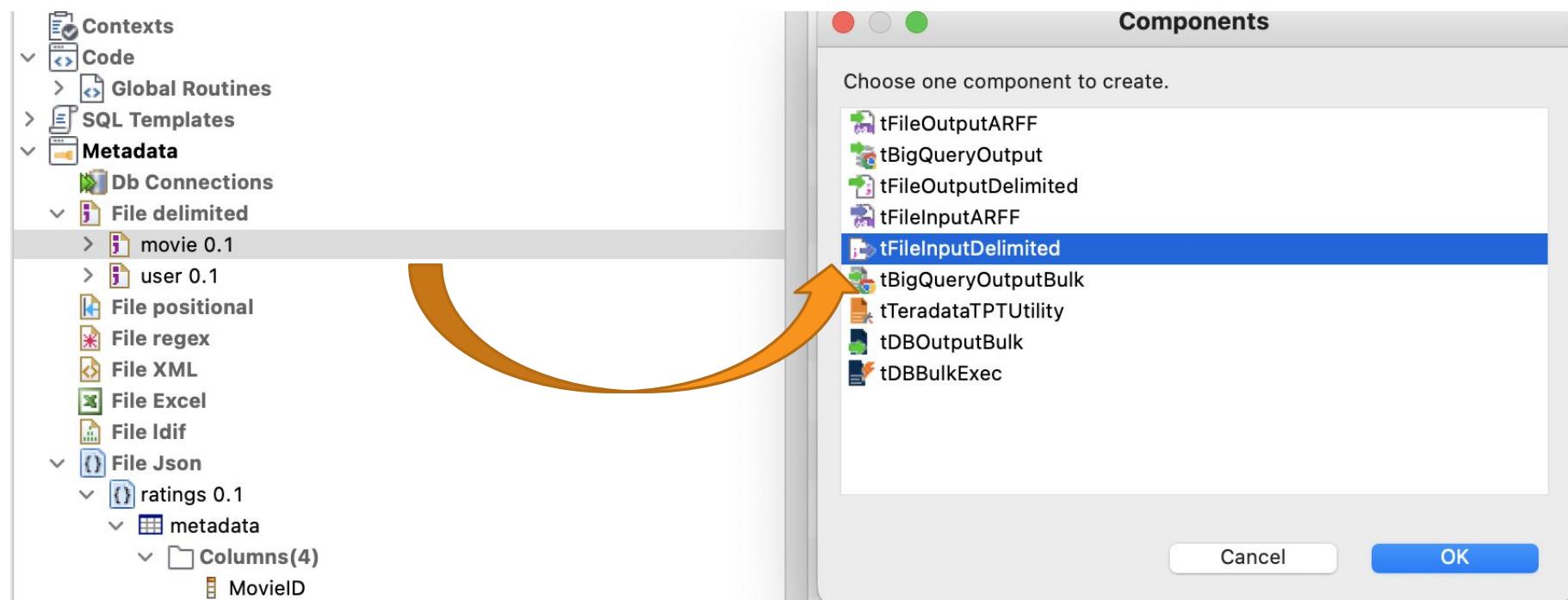
## 1.2 Create TJob#1: get movie data from RDB and convert data to CSV file (3)



# Task 1

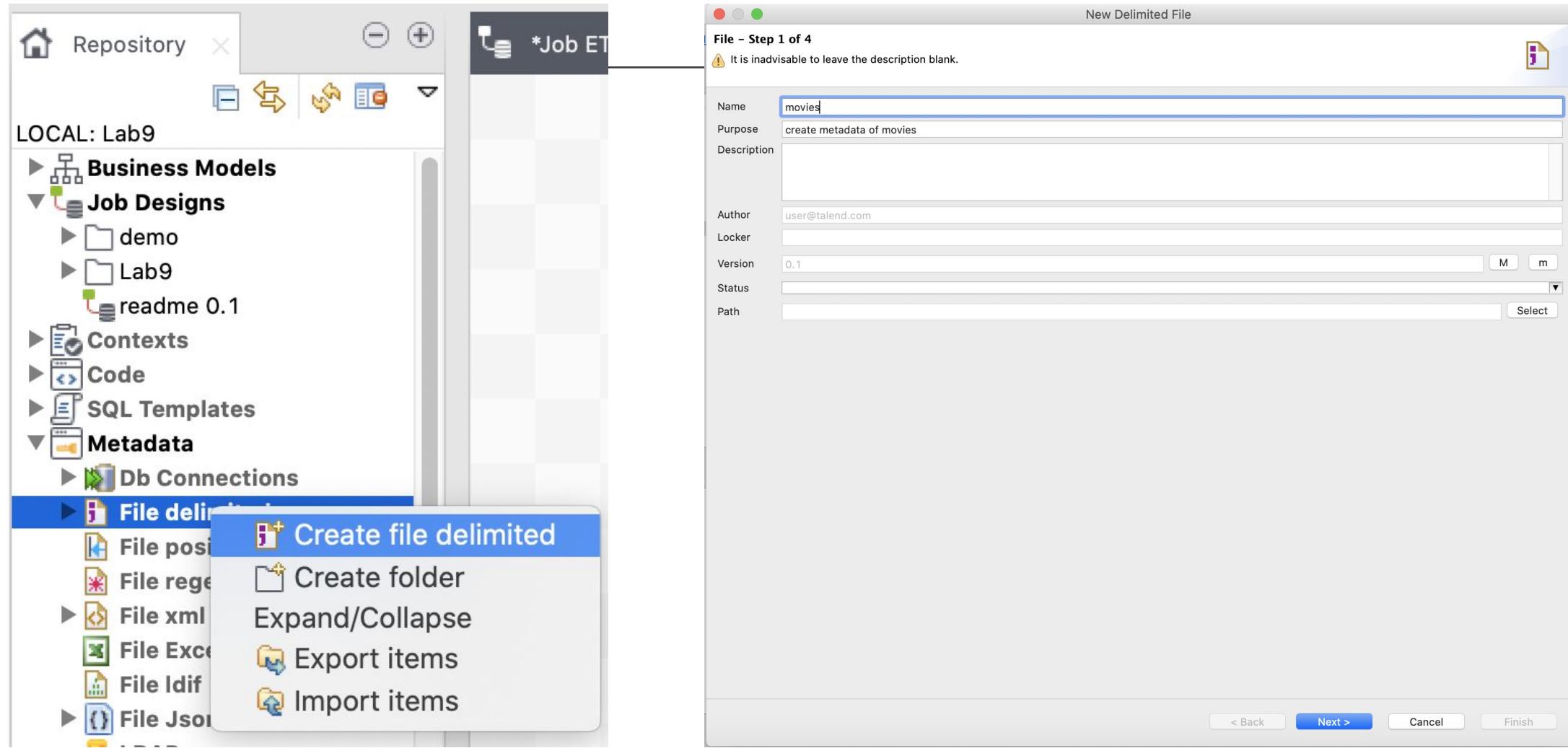
## 1.1 Create CSV to get movie data

Drag Metadata of table “movies” to Design Workspace



# Task 1

## 1.1 Create CSV Metadata for Movies (1)



# Task 1

## 1.1 Create CSV Metadata for Movies (2)

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

File – Step 3 of 3

Edit an existing Delimited File

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

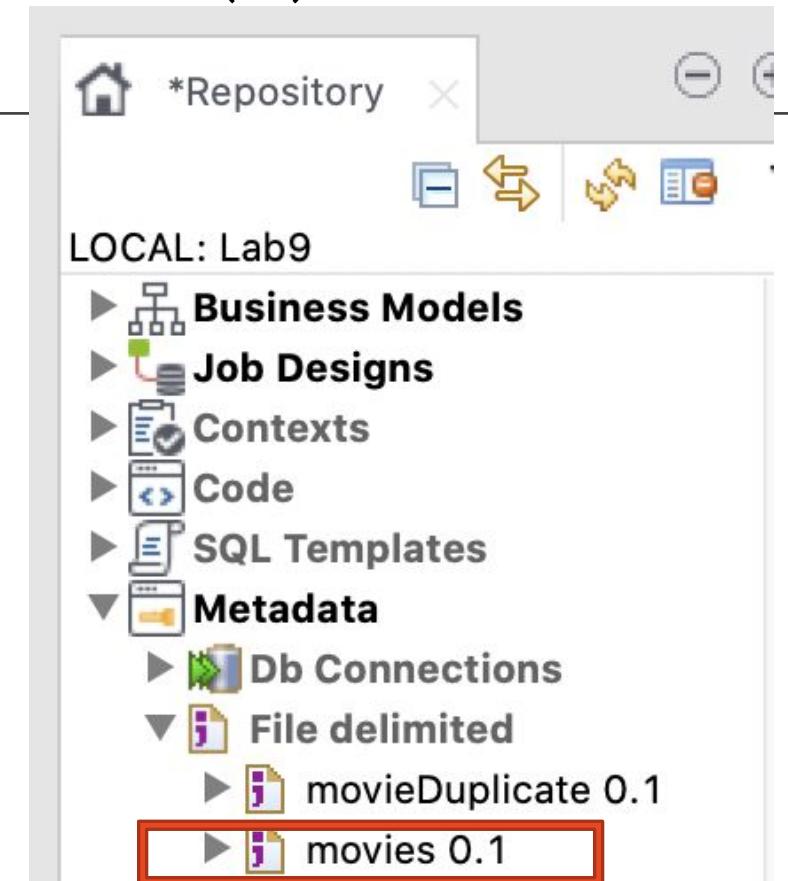
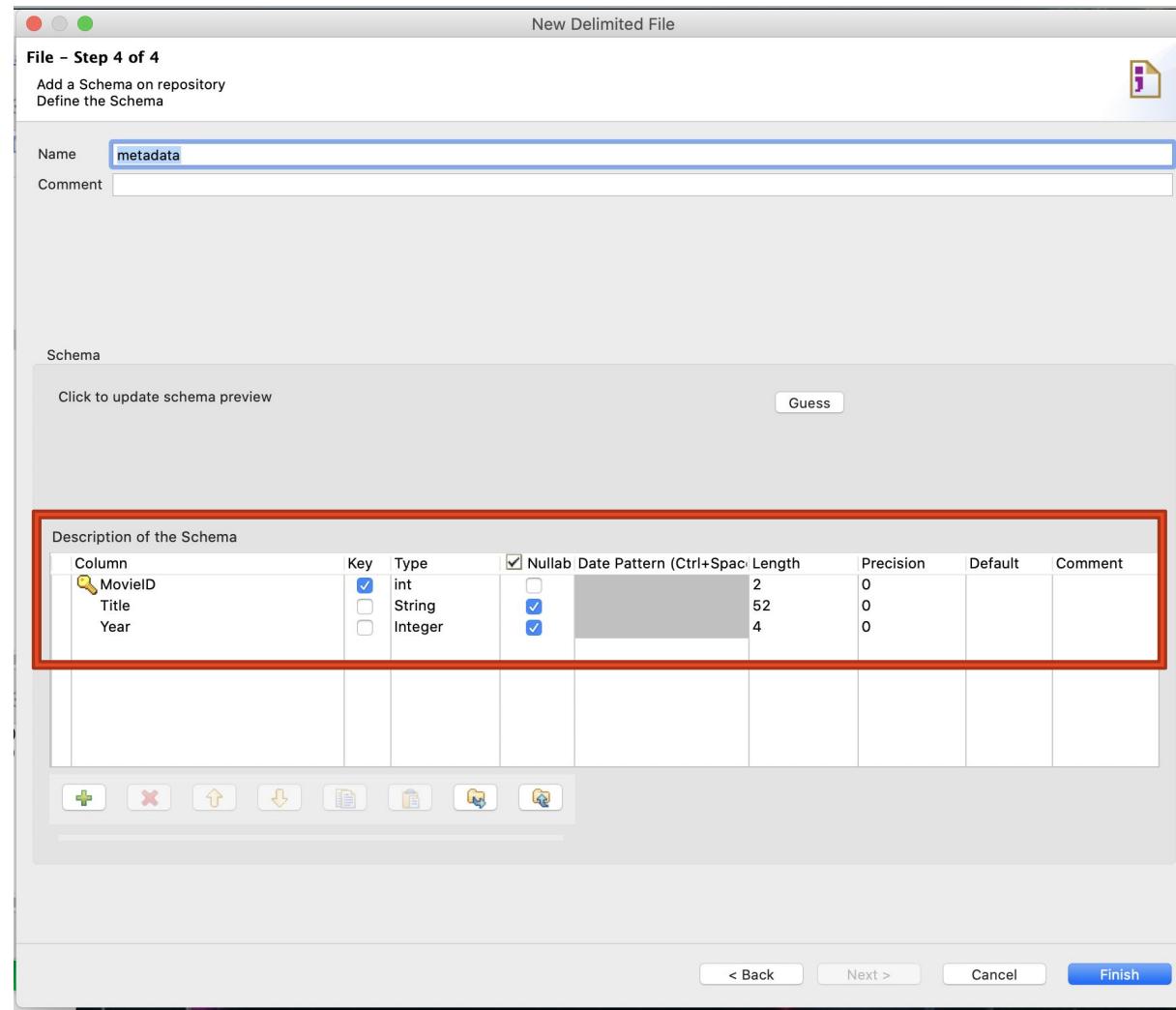
MovielID	Title	Year
1	Toy Story (1995)	1995
2	Jumanji (1995)	1995
3	Grumpier Old Men (1995)	1995
4	Waiting to Exhale (1995)	1995
5	Father of the Bride Part II (1995)	1995
6	Heat (1995)	1995
7	Sabrina (1995)	1995
8	Tom and Huck (1995)	1995

Export as context Revert Context

< Back Next > Cancel Finish

# Task 1

## 1.1 Create CSV Metadata for Movies (3)



# Task 1

## 1.2 Get movie data and explore

The screenshot shows the configuration interface for a **movie(tFileInputDelimited\_2)** component. On the left, a sidebar menu lists **Basic settings**, **Advanced settings**, **Dynamic settings**, **View**, and **Documentation**. A red arrow points to the **Basic settings** tab, which is currently selected. The main configuration area includes:

- Property Type:** Repository ▼ **DELIM:movie**
- Schema:** Repository ▼ **DELIM:movie - metadata** \* **Edit schema** ...
- A note: "When the input source is a stream or a zip file, footer and random shouldn't be bigger than 0."
- File name/Stream:** "/Users/swaruprajdhungana/Downloads/movies\_data.csv"
- CSV Row Separator:** LF("\n") **Field Separator:** ","
- CSV options:**  **Escape char:** ""
- Text enclosure:** ""
- Header:** 1 **Footer:** 0 **Limit:**
- Skip empty rows**
- Uncompress as zip file**
- Die on error**

# Task 1

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

The screenshot shows the Talend Designer interface. At the top, there is a job flow diagram with a green arrow pointing right labeled "movies", followed by a connector labeled "row1 (Main)", and a component labeled "tFileOutputDelimited\_1". Below the diagram, there are tabs for "Designer" and "Code", and a toolbar with icons for Job, Contexts, Component, and Run.

On the left, there is a sidebar with sections for "Advanced settings", "Basic settings", "Dynamic settings", "View", and "Documentation". A red arrow points from the text "Advanced settings" towards the sidebar.

The main panel displays the configuration for the "tFileOutputDelimited\_1" component. The "Advanced settings" section is expanded, showing several checkboxes:

- Advanced separator (for numbers)
- CSV options   Escape char  Text enclosure
- Create directory if does not exist
- Split output in several files
- Custom the flush buffer size
- Output in row mode

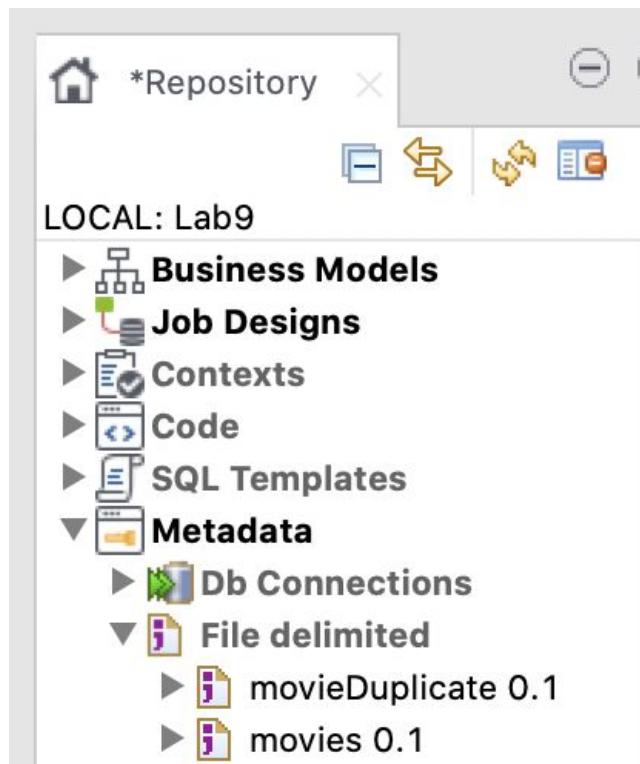
Below these, there is an "Encoding" dropdown set to "ISO-8859-15" and a checkbox for "Don't generate empty file". At the bottom of the advanced settings section, there is another checkbox:  Throw an error if the file already exist.

A red box highlights the "CSV options" section, and another red box highlights the "Throw an error if the file already exist" checkbox. To the right of the "Throw an error if the file already exist" box, the text "Uncheck to allow replace the existing file" is written in red.

# Output of Task 1

---

## 1. Metadata of File delimited “movies”





CSV File

## Task 2 for step 2

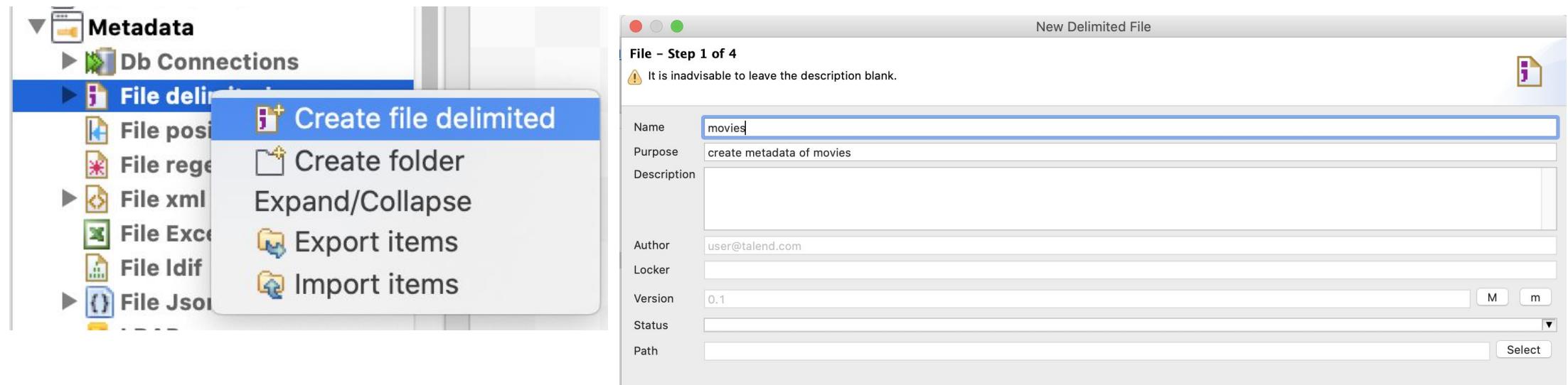
**Step 2:** get user profile in CSV format

### **Task 2:**

Create Metadata for user profile (File Delimited)

# Task 2

## Create CSV Metadata for User Profile (1)



# Task 2

## Create CSV Metadata for User Profile (2)

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 Corresponding Character: " , " (with escape 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

Escape Char Settings

CSV (selected) 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 Output

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

Export as context  Revert Context

# Task 2

## Create CSV Metadata for User Profile (3)

New Delimited File

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

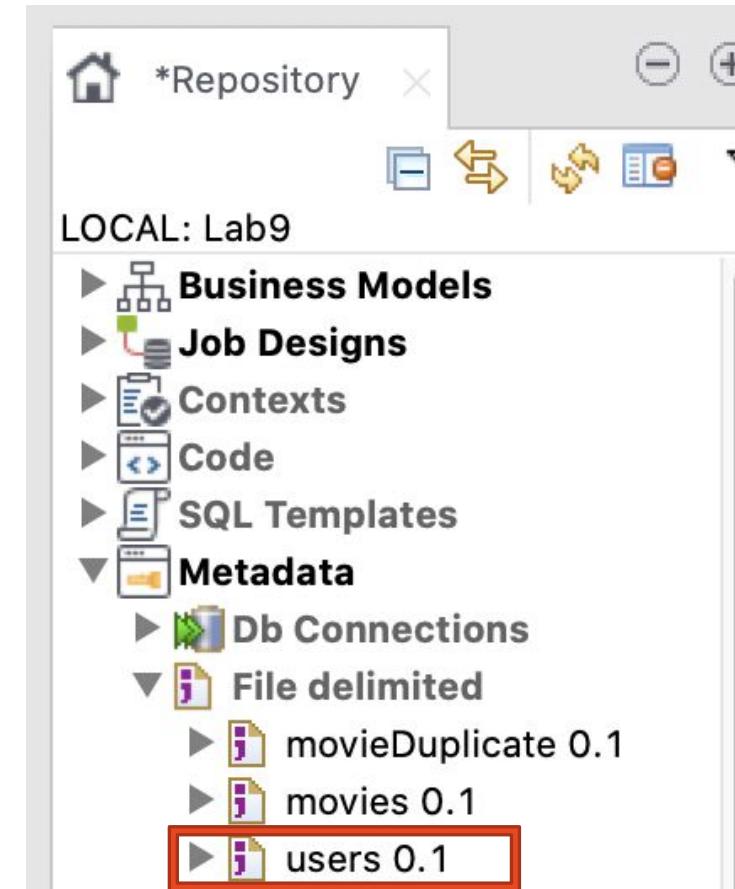
Name: metadata  
Comment:

Schema  
Click to update schema preview  
Guess

Description of the Schema

Column	Key	Type	Nullab	Date Pattern (Ctrl+Space Length)	Precision	Default	Comment
UserID	<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	
Zipcode	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		5	0	

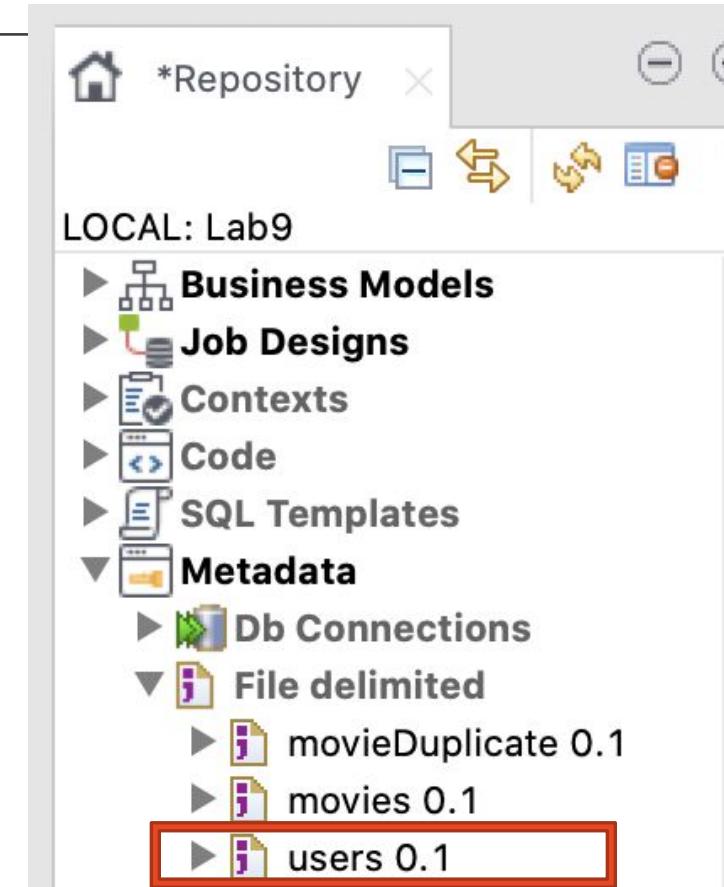
+ - ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂



# Output of Task 2

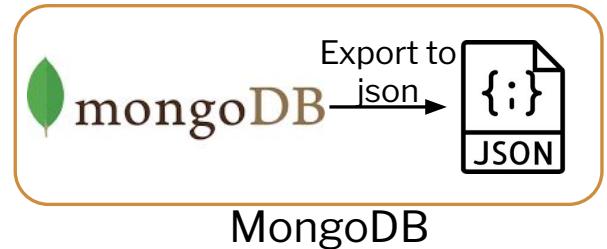
---

1. Metadata of File delimited “users”



## Task 3 for step 3

Movie Rating by Users



**Step 3:** get movie rating by users from MongoDB

### Task 3:

- Export data from MongoDB to JSON file

Atlas MongoDB

- 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)

Talend

# Task 3

## 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

# Task 3

## 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
```

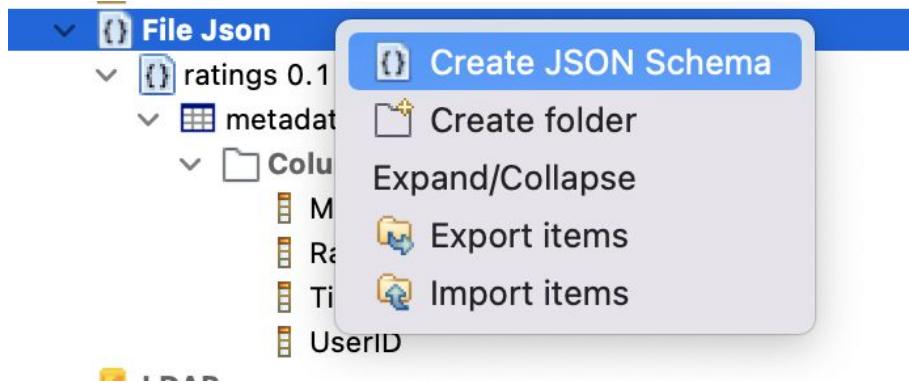
Export Data from MongoDB to JSONArray via MongoCompass

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

# Task 3

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

---



The screenshot shows the 'New Json File' dialog box, Step 1 of 5, titled 'File - Step 1 of 5'. It contains the following fields:

Field	Value
Name	rating_small
Purpose	(empty)
Description	(empty)
Author	user@talend.com
Locker	(empty)
Version	0.1
Status	(empty)
Path	(empty)

A warning message at the top states: "It is inadvisable to leave the purpose blank."

# Task 3

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

**File - Step 3 of 5**  
Add a Metadata File on repository  
Define the path of the file and the format settings

**File Settings**

Read By: **JsonPath**

Json: /Users/swaruprajdhungana/Downloads/rating\_small.json

Encoding: UTF-8

Limit: 0

**Schema Viewer**

```
$[*]
  _id
    $oid
    UserID
    MovieID
    Rating
    Timestamp
```

**File - Step 4 of 5**  
Add a Metadata File on repository  
Define the setting of the parse job

**Source Schema**

```
$[*]
  _id
    $oid
    UserID
    MovieID
    Rating
    Timestamp
```

**Target Schema**

Path loop expression: Absolute path expression: \$[\*]

Fields to extract:

Relative or absolute path expression	Column Name
UserID	UserID
MovieID	MovieID
Rating	Rating
<b>Timestamp</b>	<b>Timestamp</b>

Preview successful...

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
2	1357	5	2015-10-18 15:24:38
2	3068	4	2015-01-14 15:31:09
2	1527	4	2015-05-22 00:02:00

Export as context Revert Context

# Task 3

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

Date pattern = “yyyy-MM-dd HH:mm:ss”

The screenshot shows a software interface for defining a schema. On the left, a window titled "File - Step 5 of 5" displays the schema definition. It includes fields for "Name" (set to "metadata") and "Comment". Below this is a "Schema" section with a preview area labeled "Click to update schema preview" and a "Guess" button. A table titled "Description of the Schema" lists columns: UserID, MovieID, Rating, and Timestamp. The "Key" column has checkboxes for UserID, MovieID, and Rating, which are checked. The "Type" column shows Integer for UserID, MovieID, and Rating, and Date for Timestamp. The "Date Pattern" column for Timestamp is highlighted with a red box and contains the value "yyyy-MM-dd HH:mm:ss". The "Length" column for Timestamp shows values 3, 4, 1, and 19. The "Precision" and "Default" columns are empty. The "Comment" column is also empty. At the bottom of the schema window are standard file operations: +, -, up, down, save, and open.

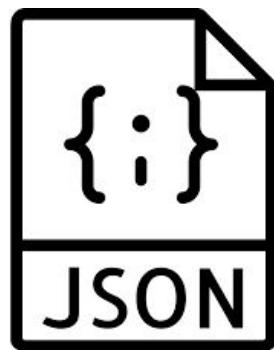
The right side of the interface shows a tree view of the schema structure:

- File regex
- File XML
- File Excel
- File Idif
- File Json (selected)
- ratings 0.1
  - metadata
    - Columns(4)
      - MovieID
      - Rating
      - Timestamp
      - UserID

# Task 3

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

---



Json Metadata  
(tInputFileJSON)



ratings.csv

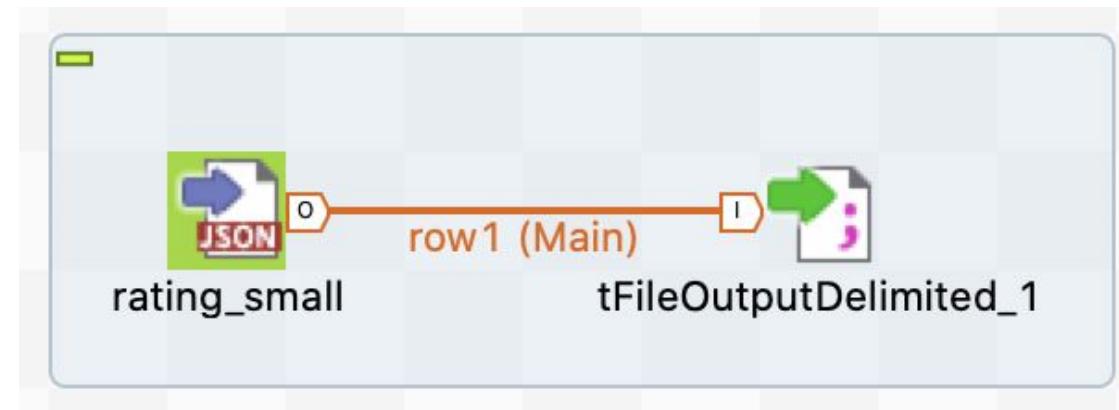
# Task 3

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

---

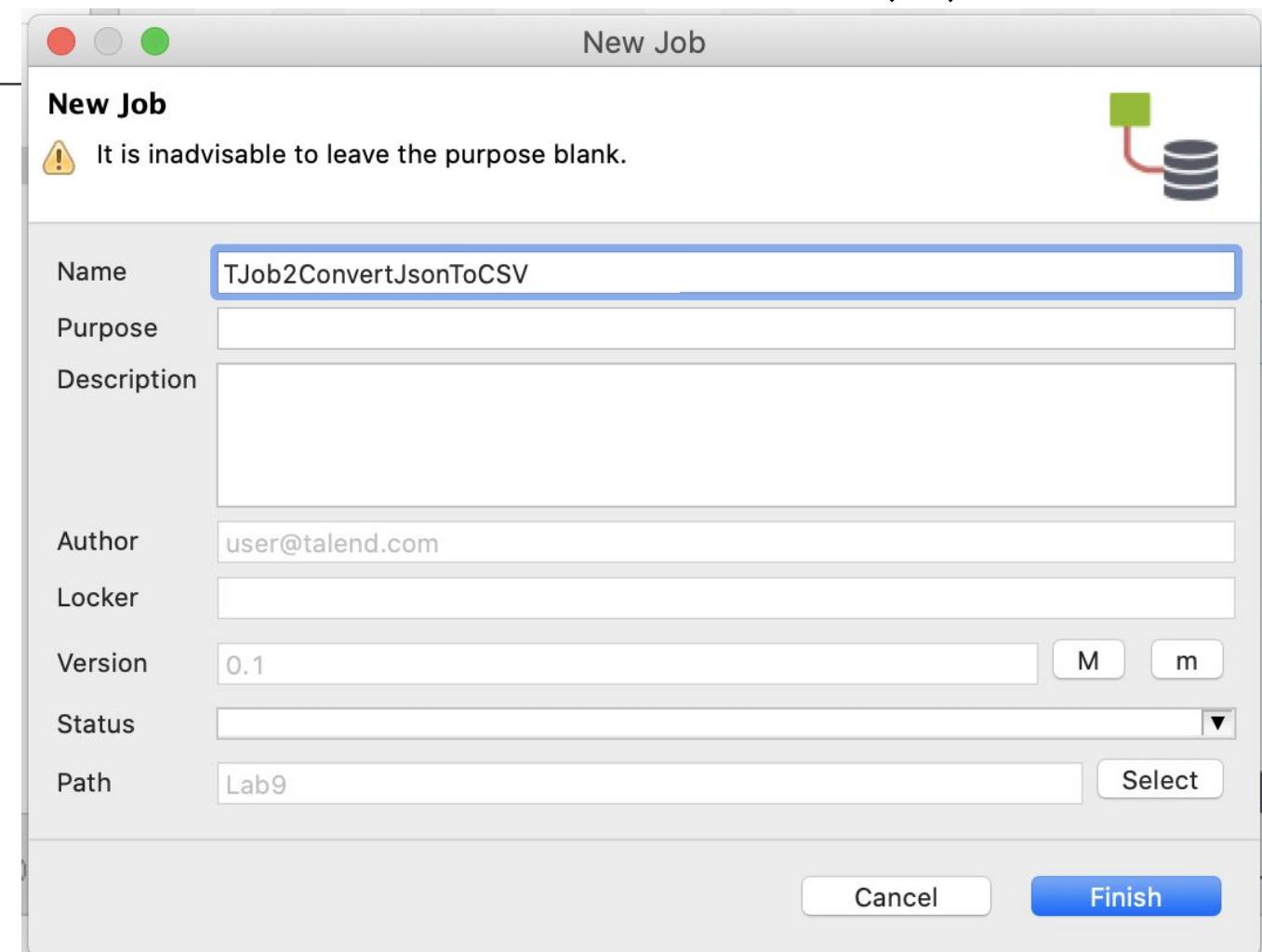
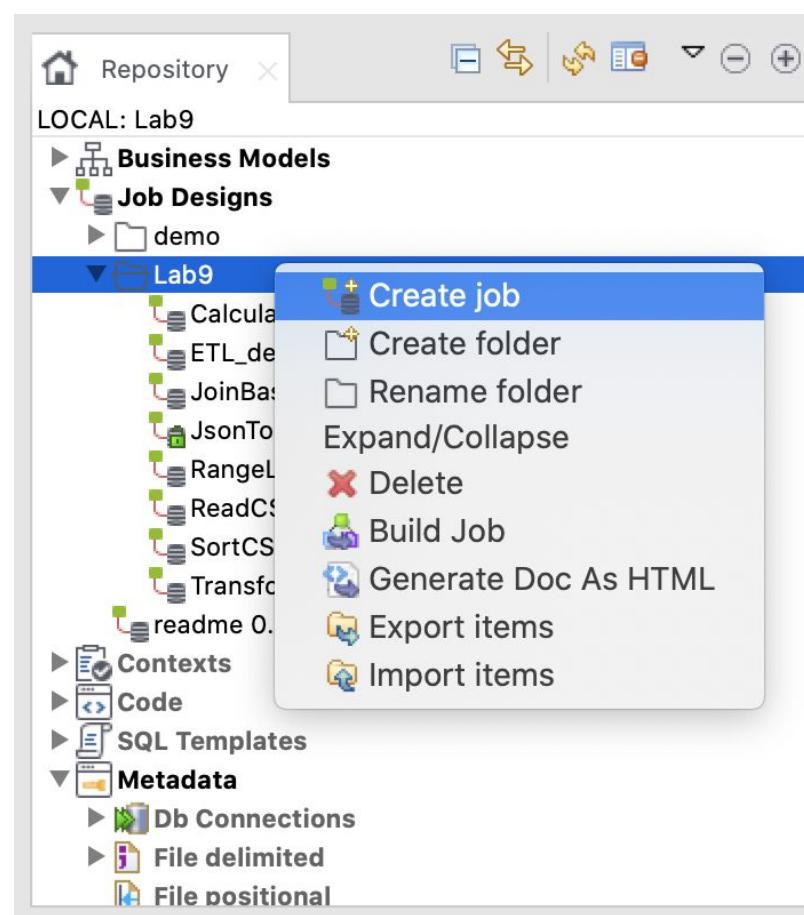
**Purpose :** Creating a Job to extract json data and convert to CSV file.

**Components:** tFileInputJson , tFileOutputDelimited



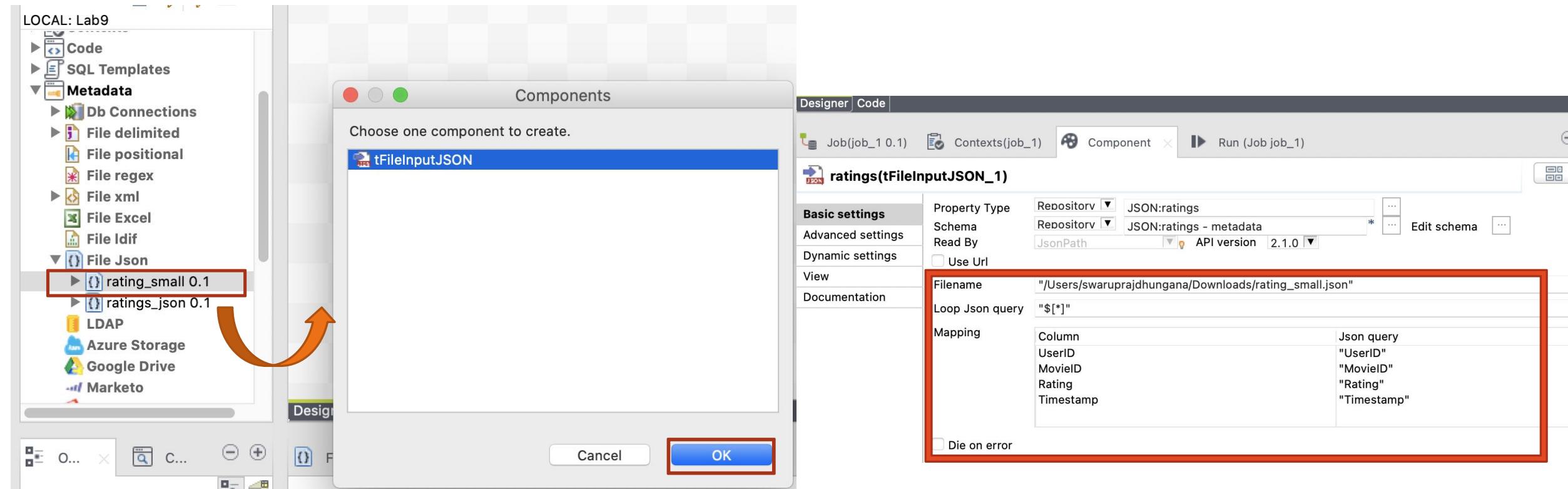
# Task 3

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



# Task 3

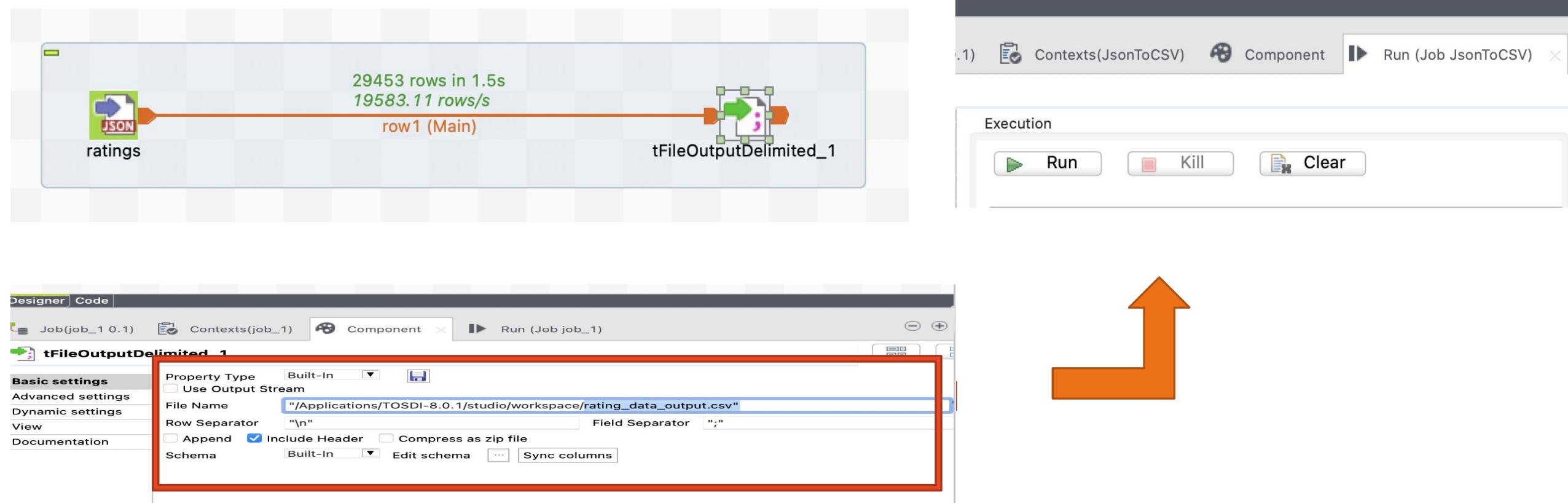
## 3.2 Create TJob#2: Convert Json to CSV File (2)



Drag ratings\_json to “design workspace” area and click “OK”

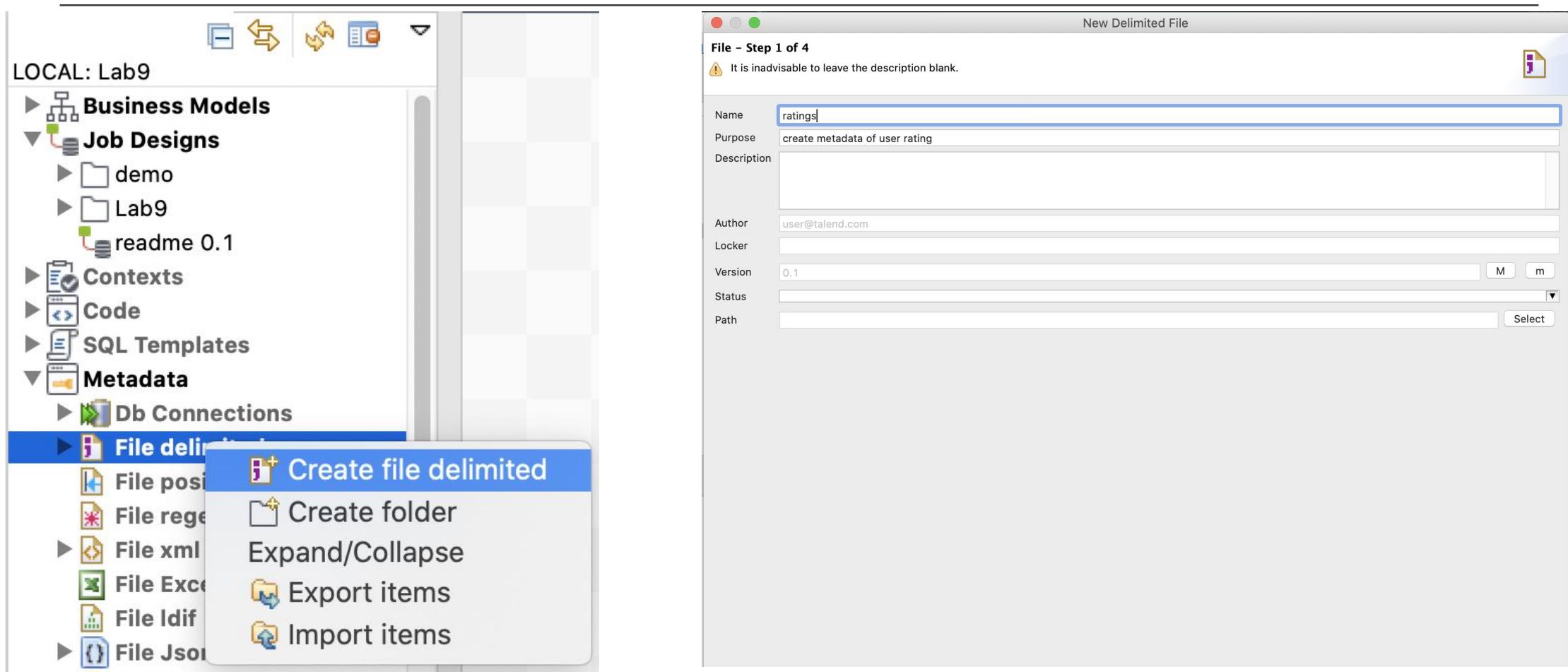
# Task 3

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



# Task 3

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



# Task 3

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

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

Escape Char Settings

CSV (selected)  
Delimited  
Escape Char: Empty  
Text Enclosure: Empty  
 Split row before field

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:

Preview | Output

Set heading row as column names

Column 0	Column 1	Column 2	Column 3
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

Same approach as Task 1

# Task 3

## 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:

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

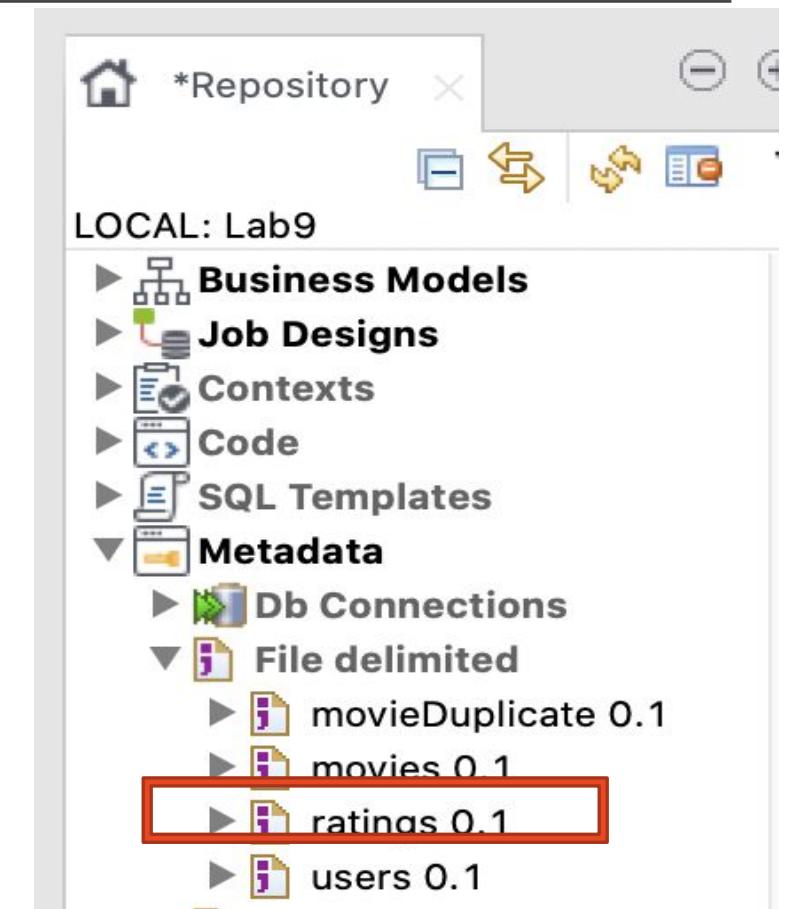
Schema

Click to update schema preview

Description of the Schema

Column	Key	Type	Nullable	Date Pattern (Ctrl+Space)	Length	Precision	Default	Comment
UserID	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		1	0		
MovieID	<input checked="" type="checkbox"/>	int	<input 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		

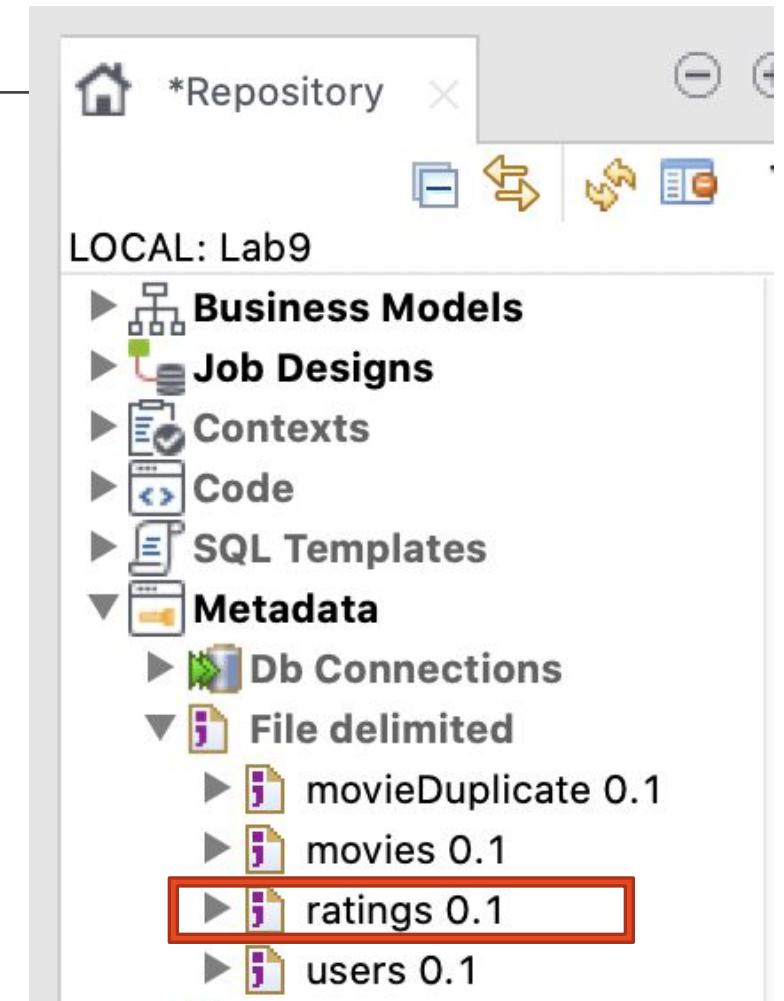
+ - < > << >> <<< >>>



# Output of Task 3

---

## 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

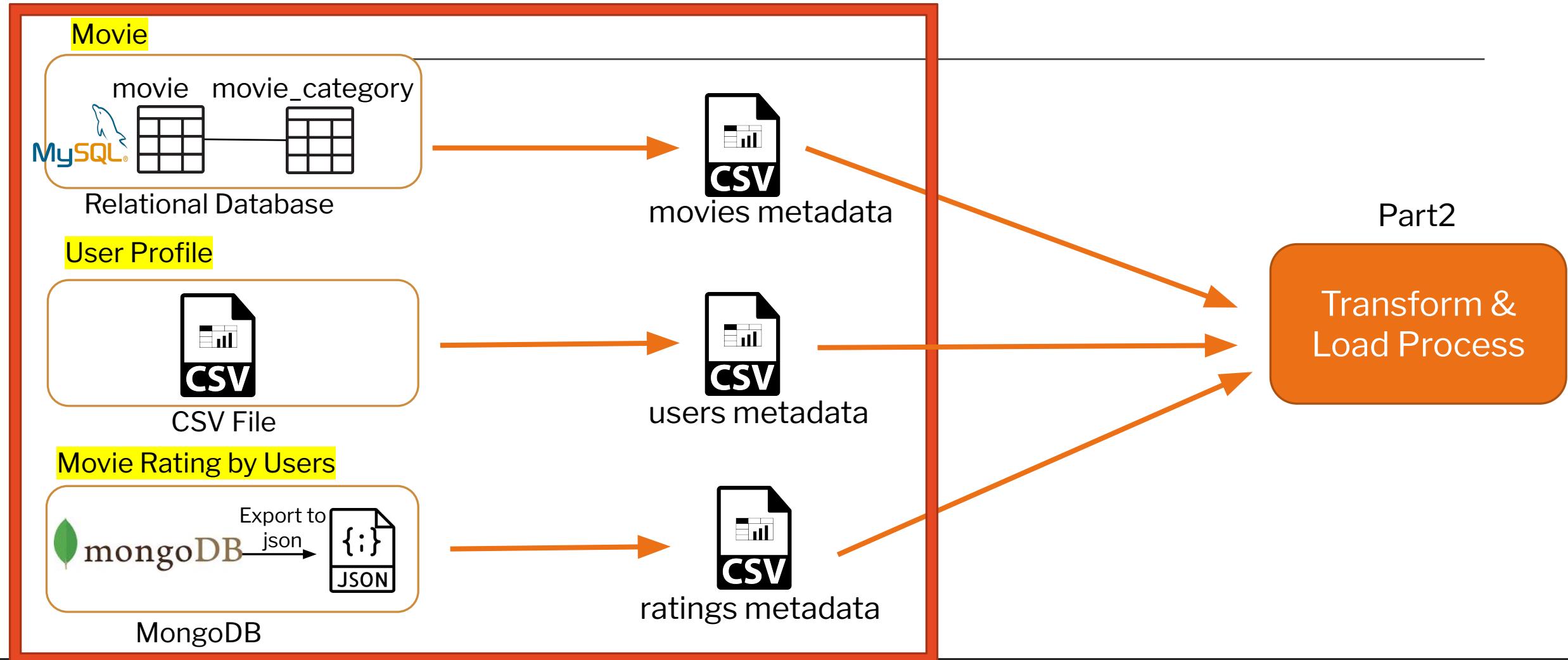
File Delimited Metadata

---

# Part 2

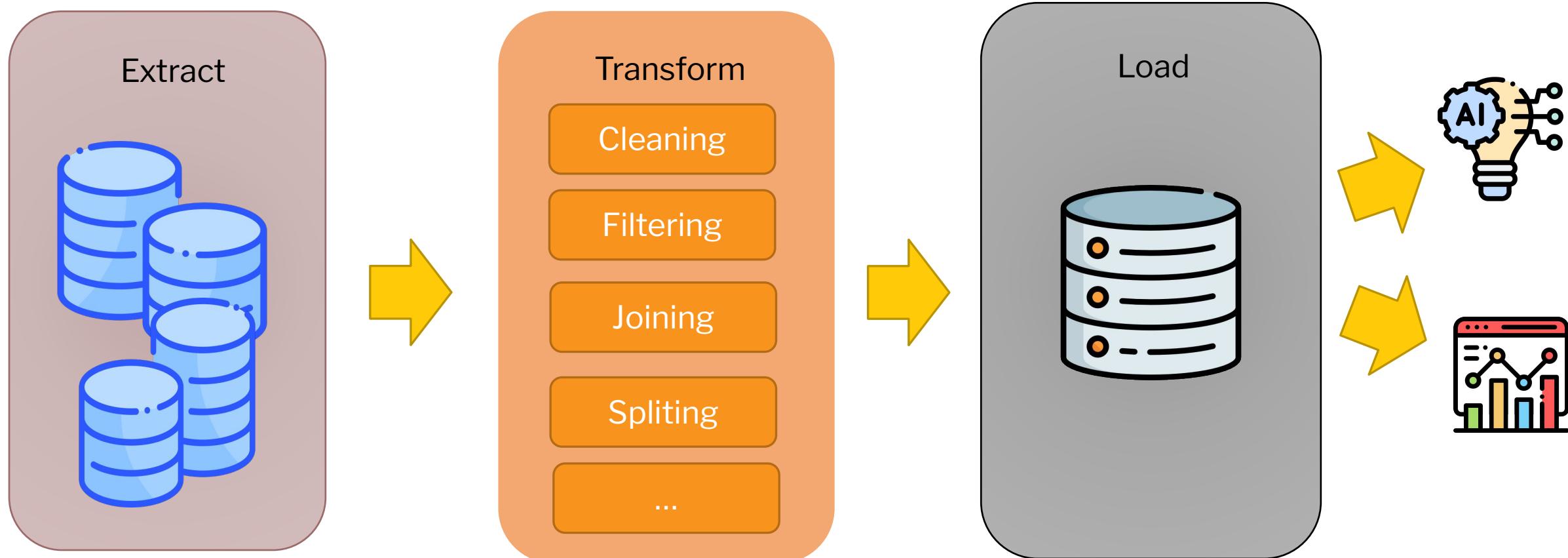
---

# Extraction Process in Part 1 (Recap)

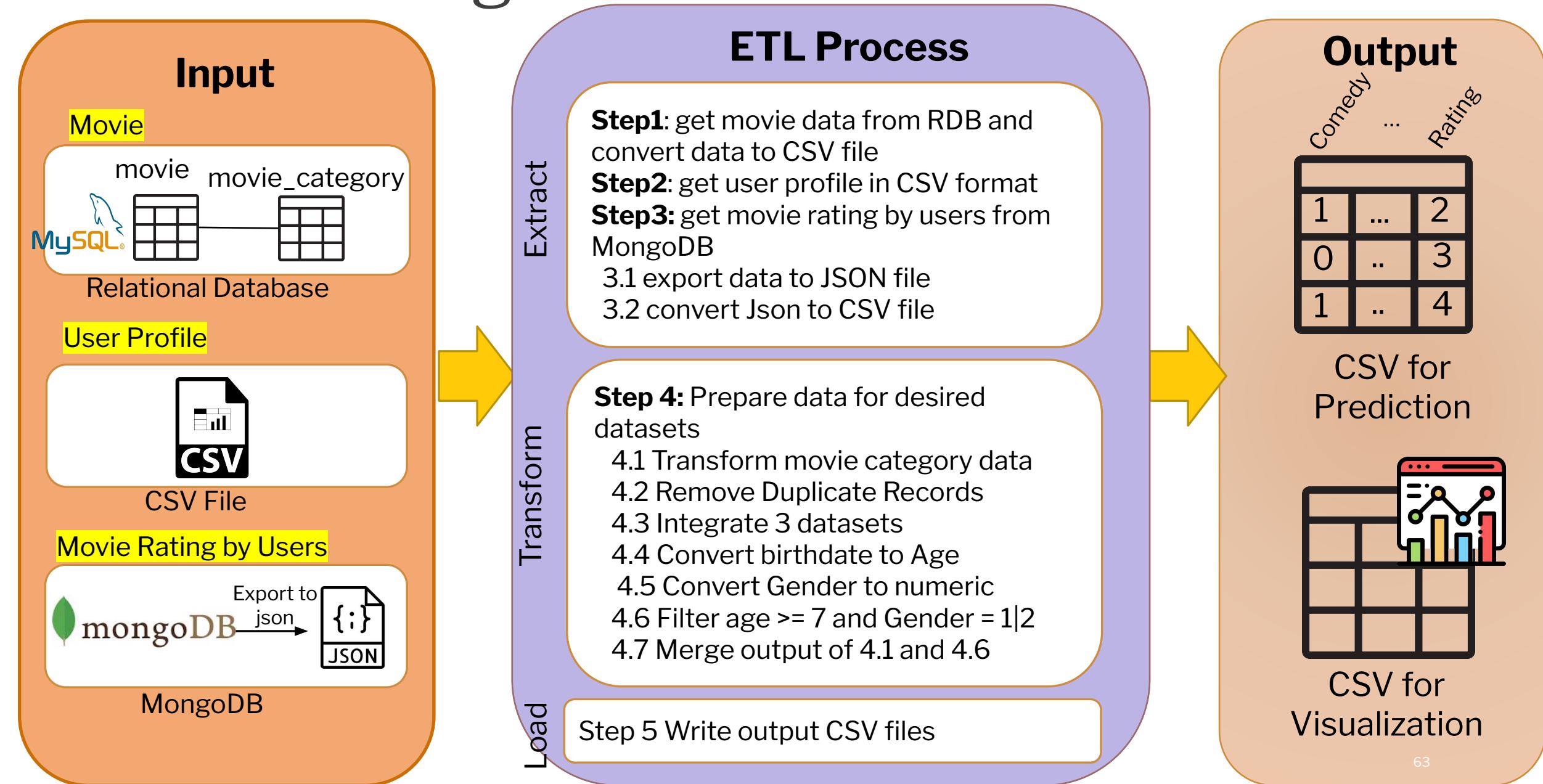


# ETL Process

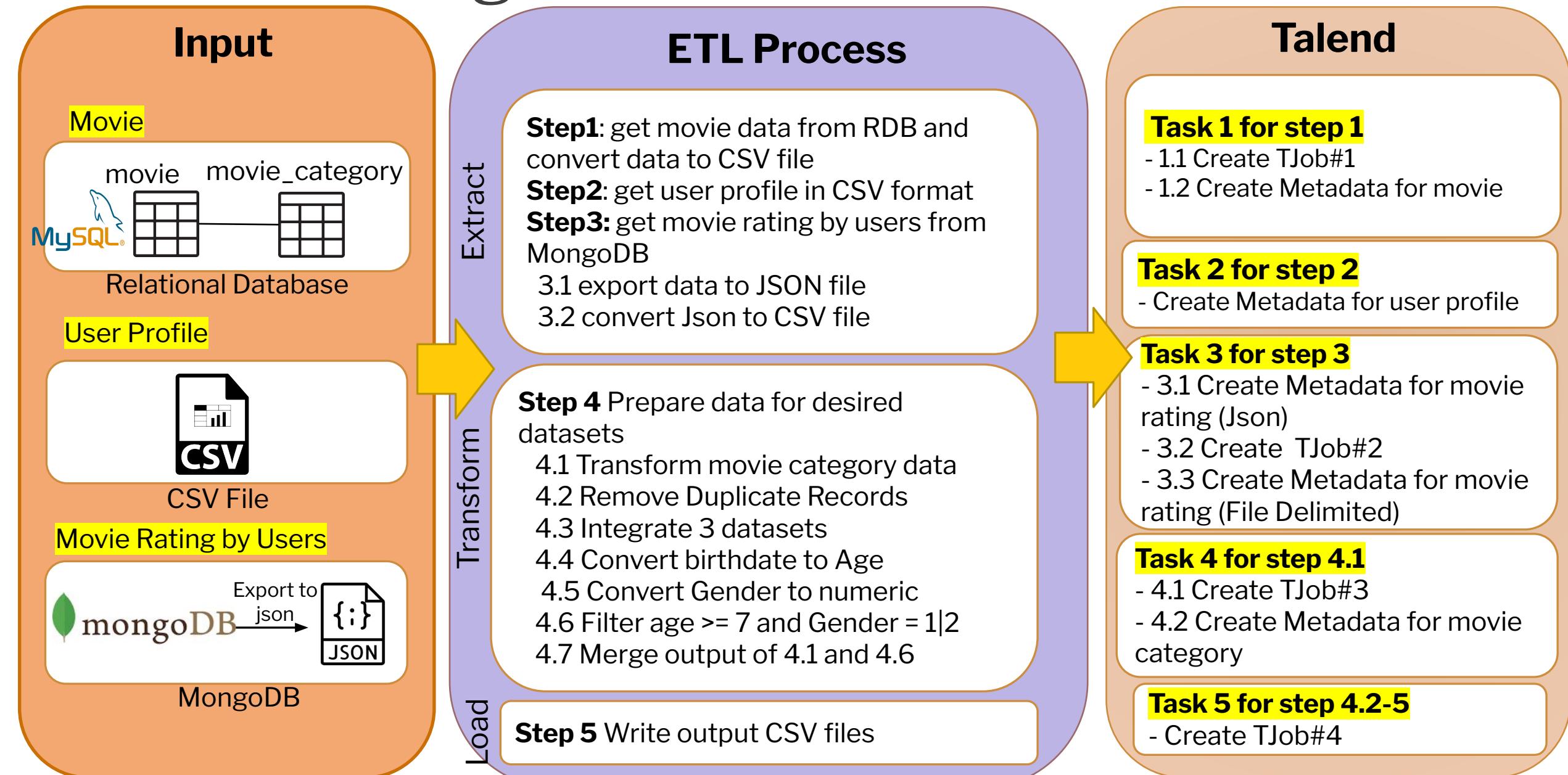
---



# Movie Rating Prediction



# Movie Rating Prediction



## **Task 4 for step 4.1**

---

**Step 4.1:** Transform movie category data

**Task 4:**

4.1 Create TJob#3: Transform movie category data

4.2 Create Metadata for movie category (File Delimited)

# 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	0	5
2	3068	1	50	16	0	0	0	0	0	0	1	0	0	0	4
2	1537	1	50	16	0	0	1	0	0	0	0	0	0	0	4
2	647	1	49	16	0	0	0	0	0	0	1	0	0	0	3
2	2194	1	50	16	0	0	0	0	0	0	1	1	1	1	4
2	648	1	50	16	0	0	0	1	0	0	0	1	0	0	4
2	2268	1	49	16	0	0	0	0	0	0	1	0	1	1	5
2	2628	1	50	16	0	0	0	1	1	0	0	1	0	0	3
2	1103	1	50	16	0	0	0	0	0	0	1	0	0	0	3
2	2916	1	50	16	0	0	0	1	0	0	0	1	0	0	3
2	3468	1	50	16	0	0	0	0	0	0	1	0	0	0	3
2	1210	1	49	16	0	0	0	1	0	1	0	1	0	0	5
2	1792	1	50	16	0	0	0	0	0	0	0	1	0	0	4
2	1687	1	49	16	0	0	0	0	0	0	0	1	0	0	3
2	1213	1	50	16	0	0	0	0	0	0	1	0	1	1	3
2	3578	1	50	16	0	0	0	0	0	0	1	1	0	0	2
2	2881	1	50	16	0	0	0	0	0	0	0	0	1	0	5
2	3030	1	49	16	0	0	1	0	0	0	1	0	0	0	3
2	1217	1	49	16	0	0	0	0	0	0	1	0	0	0	4
2	3105	1	50	16	0	0	0	0	0	0	0	1	0	0	3
2	434	1	50	16	0	0	0	1	0	0	0	1	1	1	4

Gender

Age Occupation

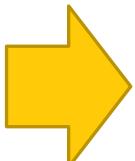
Movie category

Rating 66

# Task 4

## 4.1 Create TJob#3: Transform movie category data

MovieID	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



Change movie category to numeric values for prediction model

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

Desired output of movie category

movie category table

# Task 4

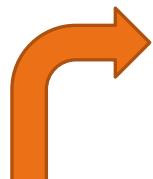
## 4.1 Create TJob#3: Transform movie category data

MovieID	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:

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



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

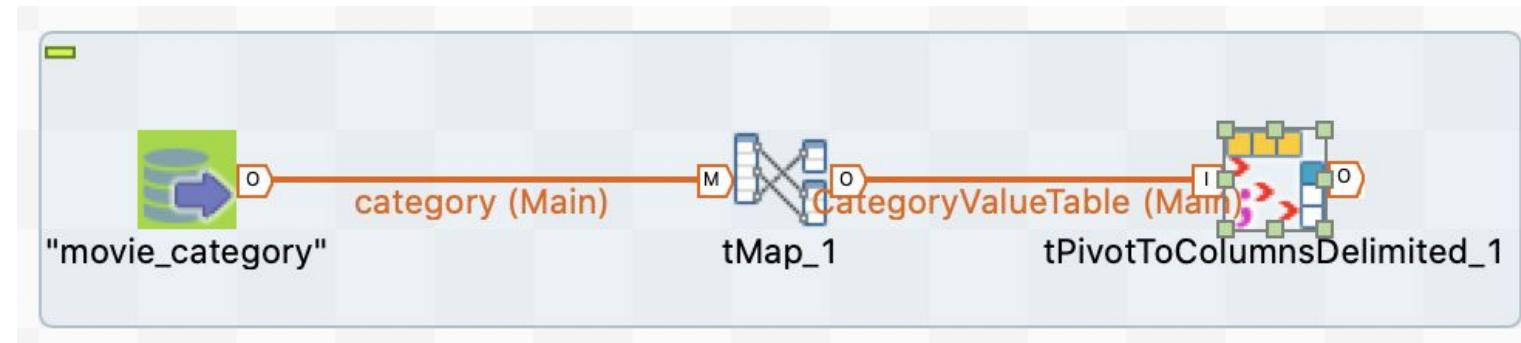
# Task 4

## 4.1 Create TJob#3: Transform movie category data

---

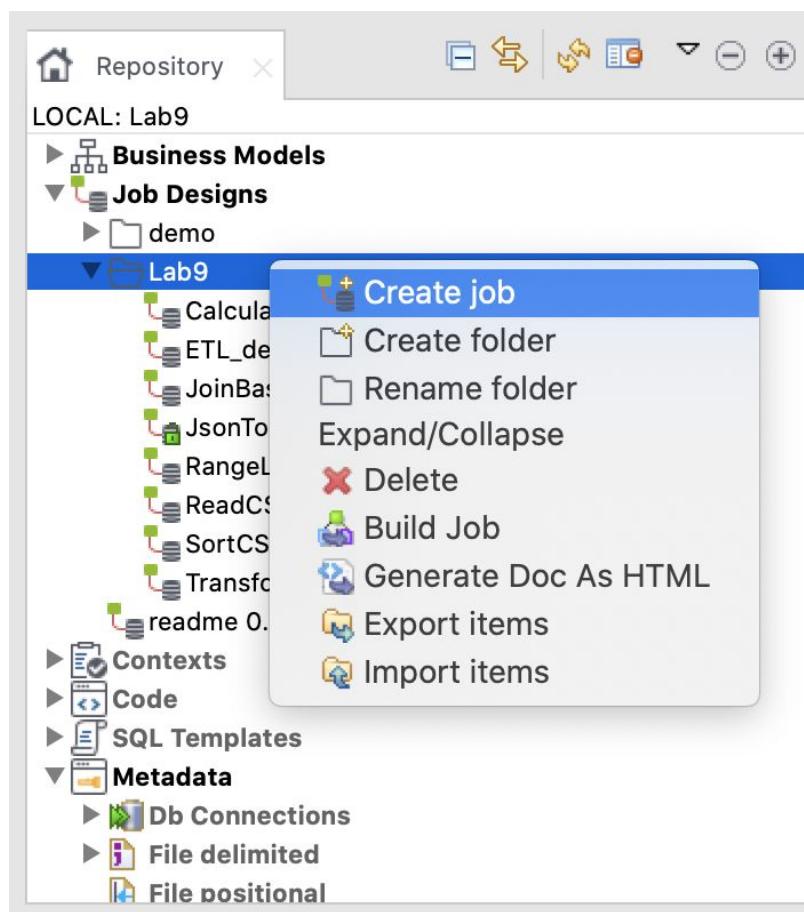
**Purpose :** Transform movie category data for prediction model

**Components:** tDBInput(MySQL) , tmap, tPivotToColumnsDelimited



# Task 4

## 4.1 Create TJob#3: Transform movie category data



The screenshot shows the 'New Job' dialog box. At the top right is a small icon of two databases with a red and green line connecting them. The dialog has several fields:

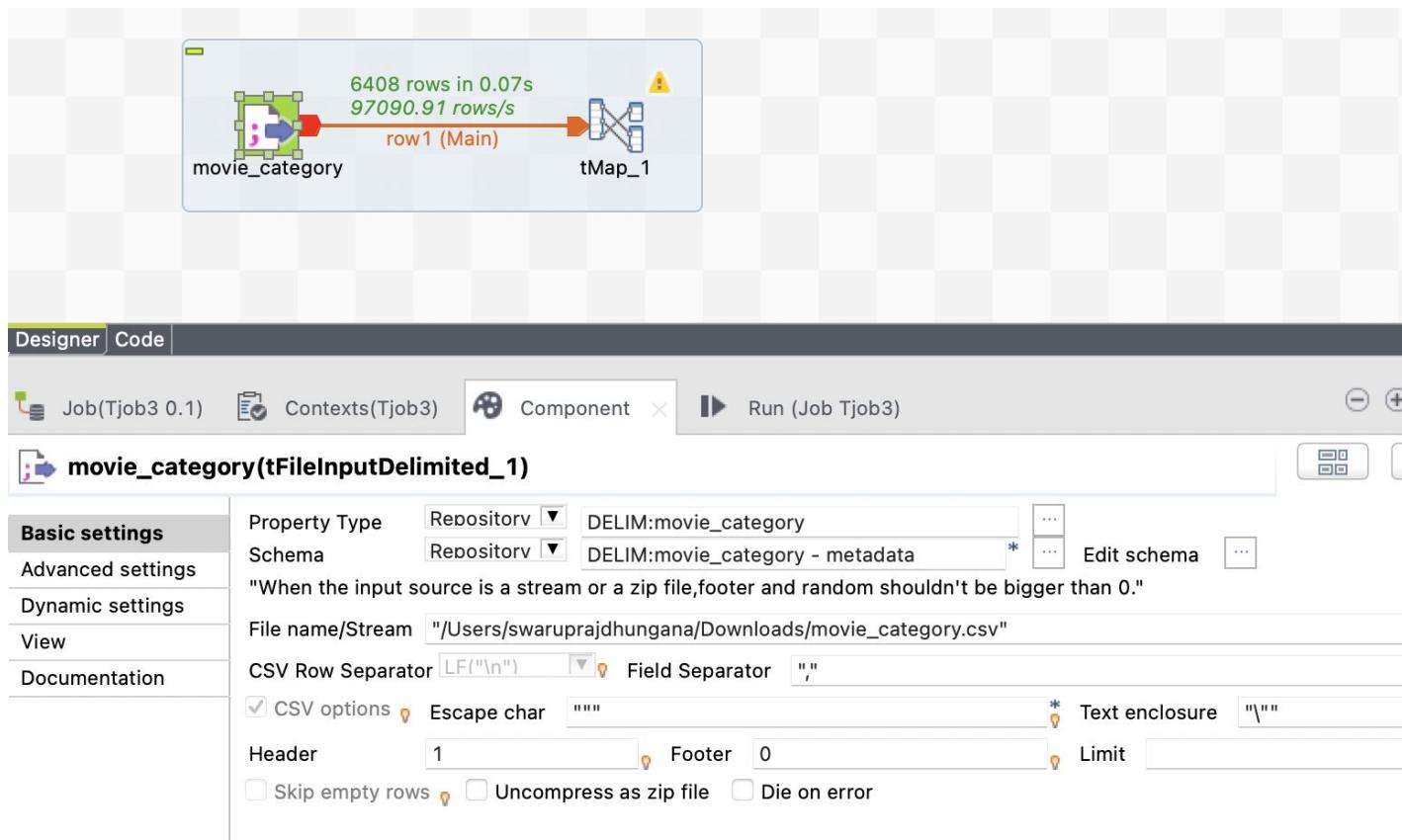
- Name:** TJob3TransformMovieCategory
- Purpose:** (empty)
- Description:** (empty)
- Author:** user@talend.com
- Locker:** (empty)
- Version:** 0.1
- Status:** (dropdown menu)
- Path:** Lab9

At the bottom right are 'Cancel' and 'Finish' buttons. Above the 'Name' field, there is a warning message: "It is inadvisable to leave the purpose blank."

# Task4

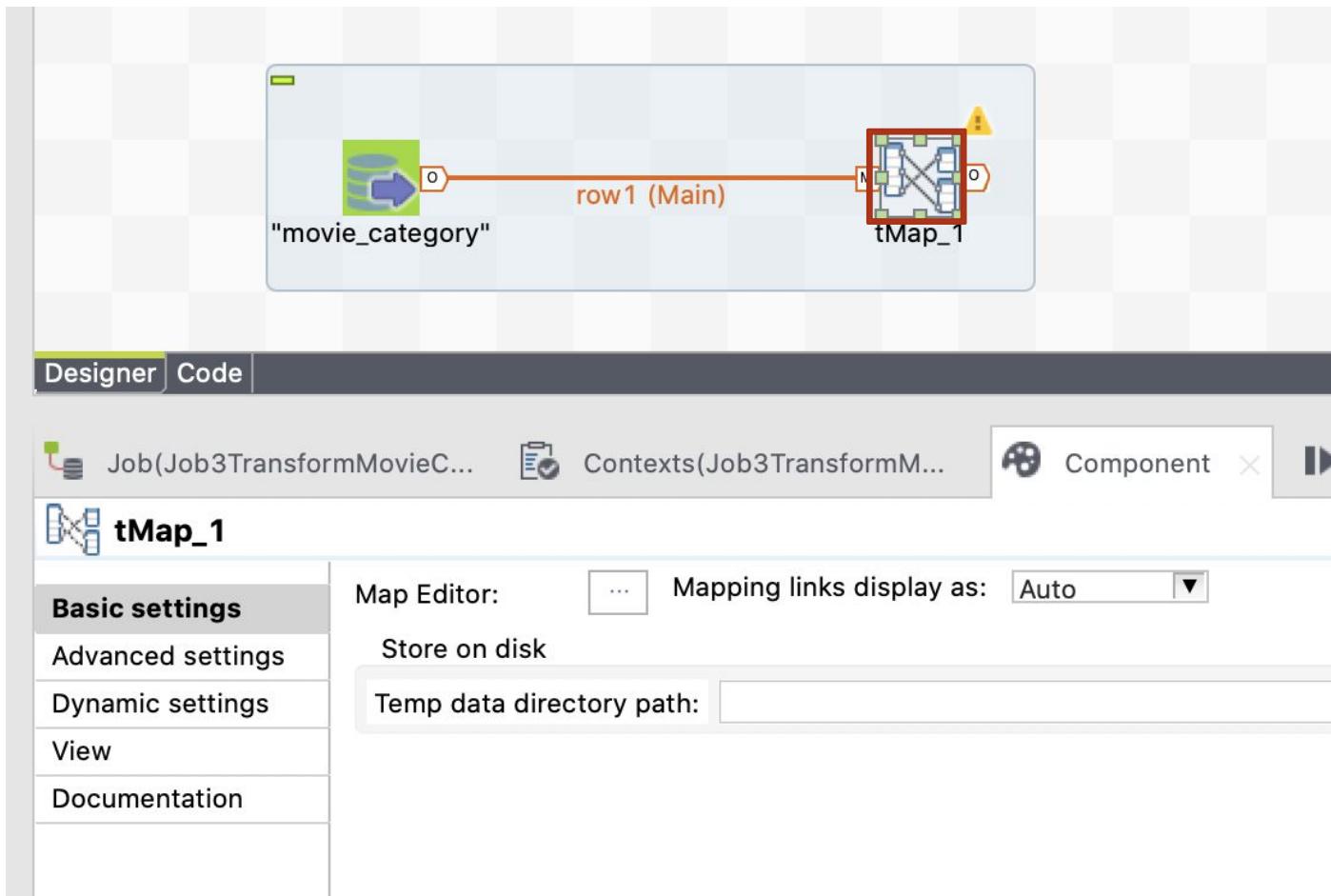
## 4.1 Create TJob#3: Transform movie category data

Drag Metadata of table “**movie\_category**” to Design Workspace



# 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

# Task 4

## 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

The screenshot shows the Talend Studio interface with the following components:

- Row Editor:** A central workspace where a row named "row1" is being edited. It contains columns "MovielD" and "Category".
- Value Tables:** A panel on the right labeled "valueTables" containing a single entry: "row1.MovieID" with a value of "1".
- Schema Editor:** A bottom panel showing the schema for both the input and output rows. The input row has columns "MovielD" (Key, Integer) and "Category" (String). The output row has columns "MovieID" (Key, int), "Category" (String), and "Values" (int).

Annotations with orange circles:

- Annotation 1: "Add column “Values”" pointing to the "Values" column in the Schema editor.
- Annotation 2: "Set value = 1 for all records in column “Values”" pointing to the value "1" in the valueTables panel.

# Task 4

## 4.1 Create TJob#3: Transform movie category data

---

Add tLogRow to check output  
Example Output:

MovielD  
Category  
Values

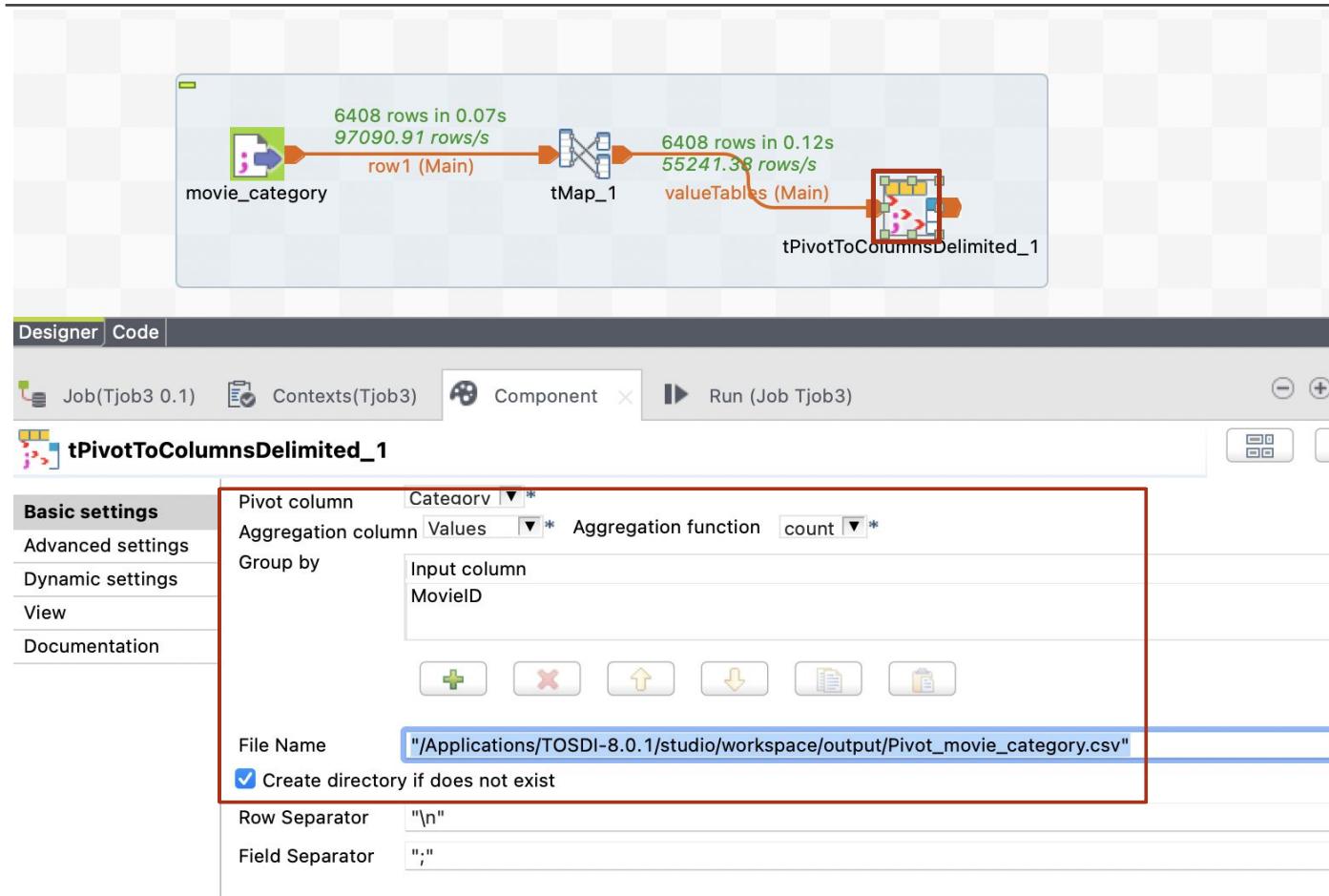
The screenshot shows the Talend Designer interface with the 'Designer' tab selected. The top bar includes tabs for 'Designer' (selected), 'Code', and other components like 'Job', 'Contexts', 'Component', and 'Run'. Below the bar, the title 'Job Job3TransformMovieCategory' is displayed. On the left, a sidebar lists 'Basic Run' options: 'Debug Run', 'Advanced settings', 'Target Exec', and 'Memory Run'. The main area is titled 'Execution' and contains three buttons: 'Run' (highlighted with a red border), 'Kill', and 'Clear'. To the right of the buttons is a text box showing the output of the job. The output is a list of movie IDs, categories, and values, with the first few lines highlighted by a red box.

MovielD	Category	Values
3945	Children's	1
3946	Action	1
3946	Drama	1
3946	Thriller	1
3947	Thriller	1
3948	Comedy	1
3949	Drama	1
3950	Drama	1
3951	Drama	1

Example Output

# Task 4

## 4.1 Create TJob#3: Transform movie category data



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

# Task 4

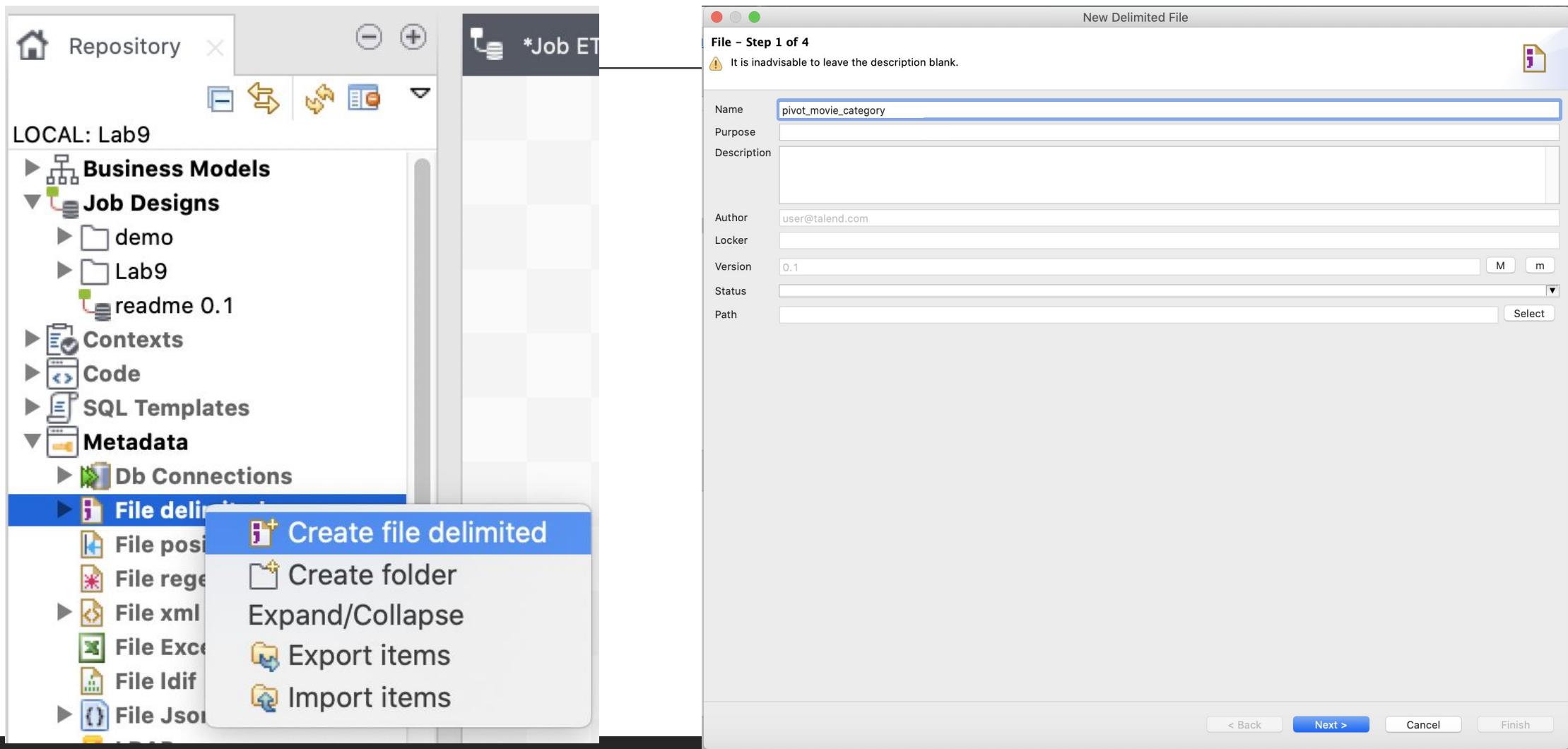
## 4.1 Create TJob#3: Transform movie category data

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

Output after pivot column

# Task 4

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



# Task4

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

# Task 4

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

The screenshot shows the Talend Data Integration interface. On the left, the "New Delimited File" dialog is open at Step 4 of 4, titled "File - Step 4 of 4". It shows a schema named "metadata" with a comment field. The "Schema" section displays a table structure for a "MovieID" column, where the "Type" is set to "int". A red box highlights the "Type" column. On the right, the "Repository" view shows a tree structure under "LOCAL: Lab9". The "Metadata" node is expanded, showing "Db Connections" and "File delimited". The "File delimited" node is also expanded, showing files like "movieDuplicate 0.1", "movies 0.1", and "pivot\_movie\_category 0.1", which is also highlighted with a red box.

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
MovielD	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		2	0	0	
Animation	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Children_s	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Comedy	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Adventure	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Fantasy	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Romance	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Drama	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Action	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Crime	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	
Thriller	<input type="checkbox"/>	Integer	<input type="checkbox"/>			0	0	

< Back    Next >    Cancel    Finish

Repository

LOCAL: Lab9

Metadata

Db Connections

- demoMySQL 0.1
- MySQL\_Conn\_movie 0.1
- MySQL\_Conn 0.1

File delimited

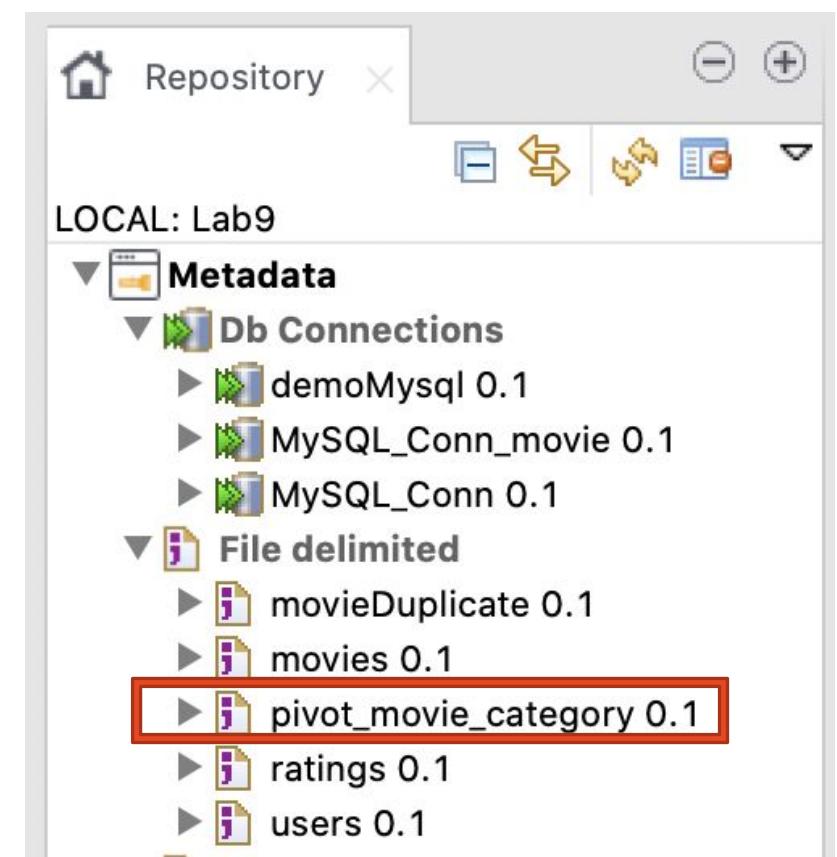
- movieDuplicate 0.1
- movies 0.1
- pivot\_movie\_category 0.1
- ratings 0.1
- users 0.1

# Output of Task 4

---

1. Metadata of File Delimited “pivot\_movie\_category”

Using this metadata in Tjob#4 !!!



## **Task 5 for step 4.2-5**

---

**Step 4.2-5:** Prepare dataset for desired datasets

**Task5:**

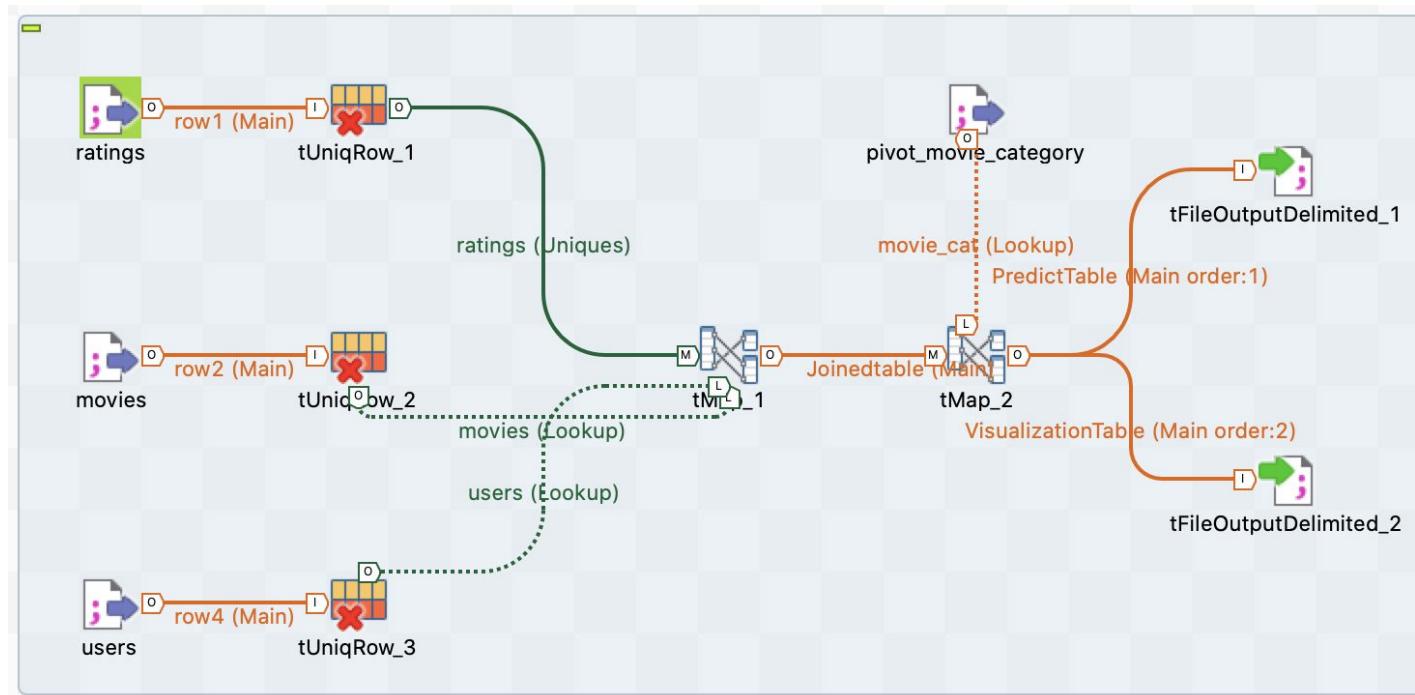
Create TJob#4: Prepare dataset for desired datasets (Main Job)

# Task5

Create TJob#4: Prepare dataset for desired datasets (Main Job)

**Purpose :** Prepare dataset for desired datasets

**Components:** tFileInputDelimited, tUniqRow, tMap, tFileOutputDelimited\_1



## tUniqRow

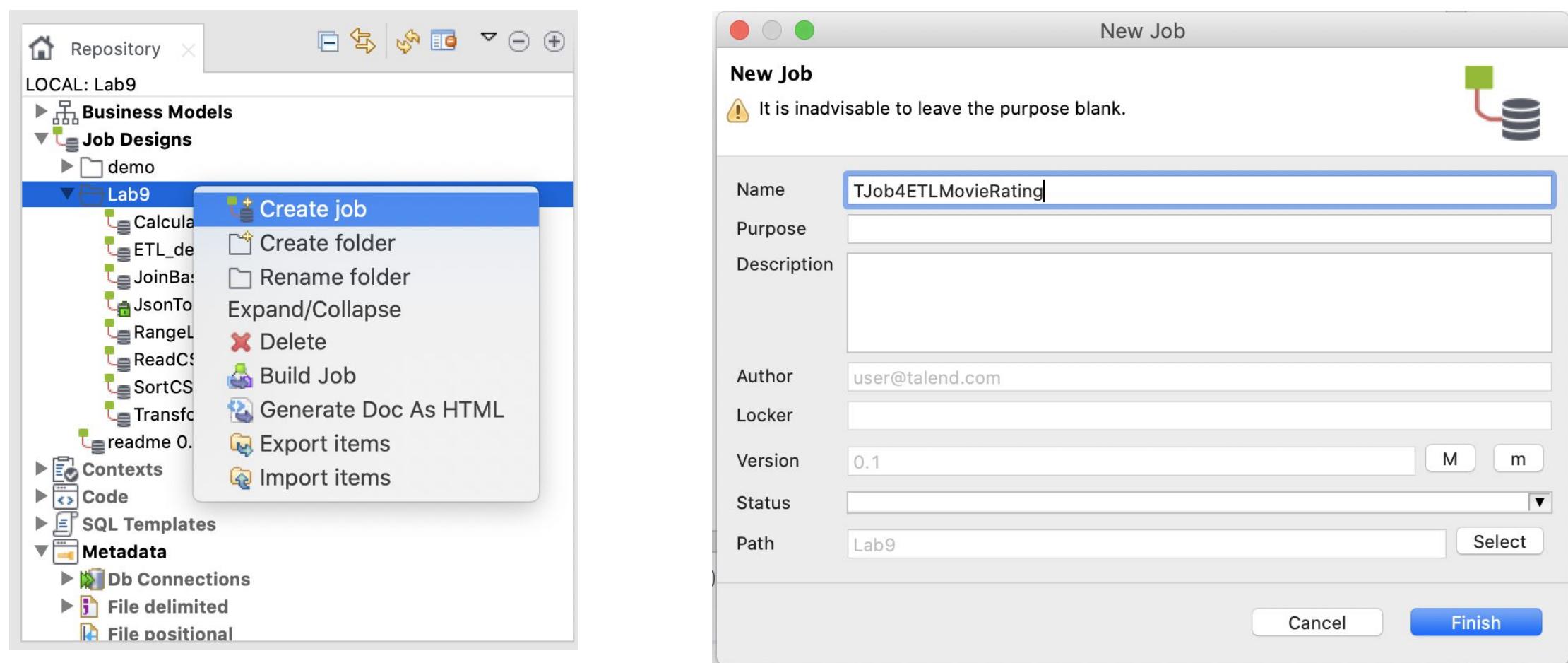
Compares entries and sorts out duplicate entries from the input flow.

## tMap

get data from one or more sources transforms data sends the transformed data to one or more destinations.

# Task5

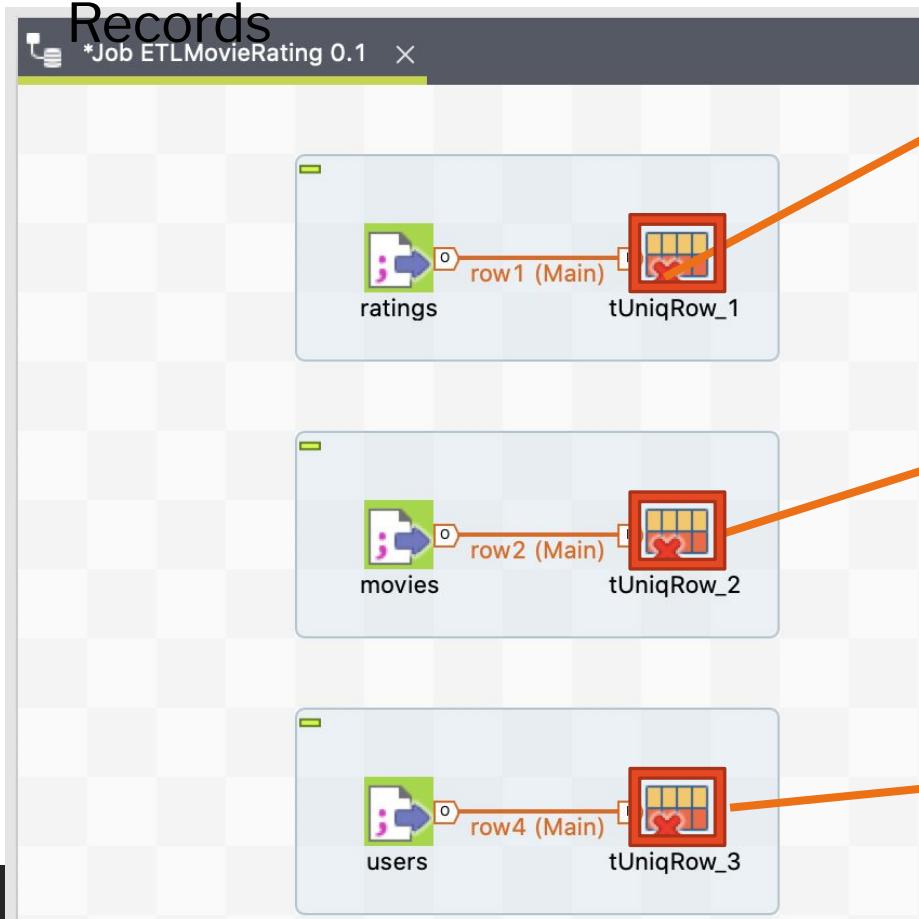
## Create TJob#4: Prepare dataset for desired datasets



# 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"/>

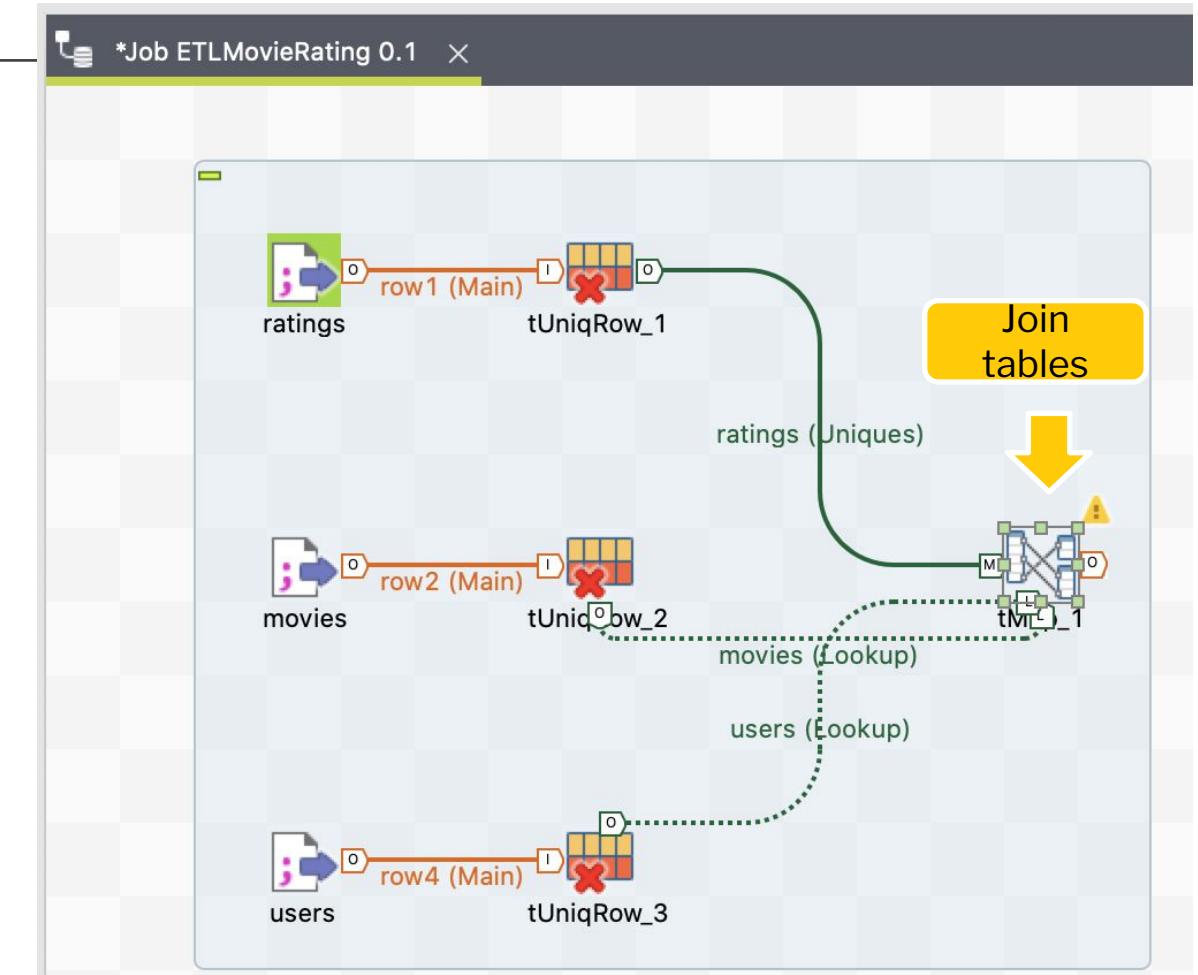
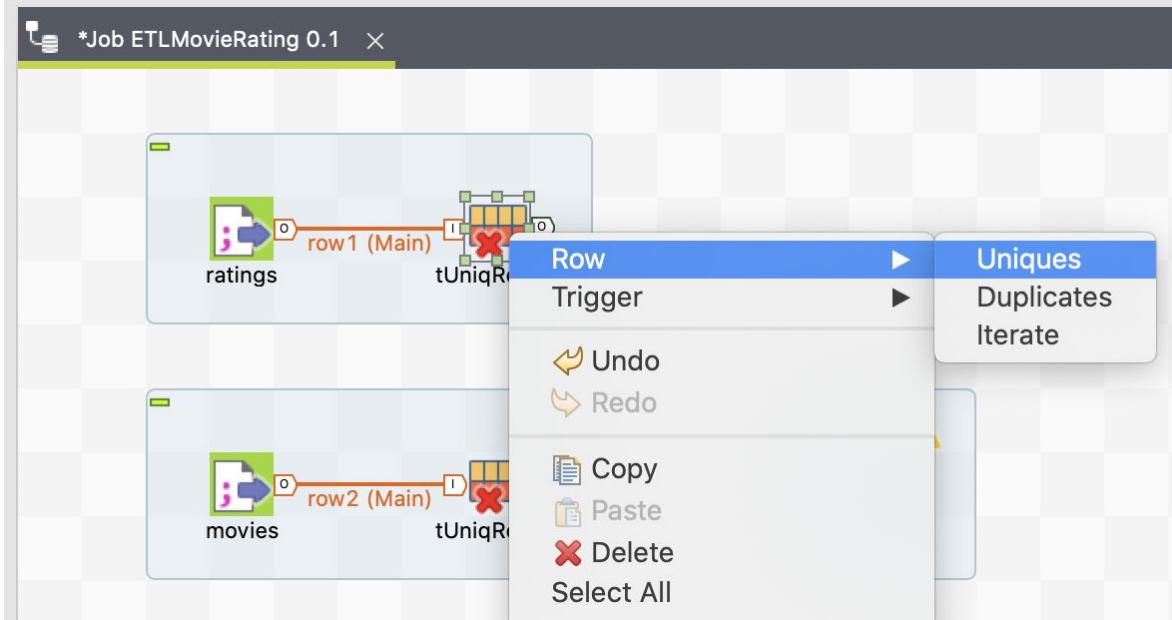
**tUniqRow\_3**

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

Input Tables

Set join model = Inner join

Output Table "Joinedtable"

Talend Open Studio for Data Integration - tMap - tMap\_1

Var

Joinedtable

Expression

- ratings.UserID
- ratings.MovieID
- movies.Title
- movies.Year
- ratings.Rating
- ratings.Timestamp
- users.Gender
- users.BirthDate
- users.Occupation
- users.Zipcode

Column

- UserID
- MovieID
- Title
- Year
- Rating
- Timestamp
- Gender
- BirthDate
- Occupation
- Zipcode

Joinedtable

Column	Type	Key	Nullab	Date Pattern (Ctrl+F)	Length	Precision	Default	Comment
UserID	Integer	✓	✓	✓	2	0		
Gender	String		✓	✓	6	0		
BirthDate	Date		✓	✓	10	0		"yyyy-MM-dd"
Occupation	Integer		✓	✓	2	0		
Zipcode	String		✓	✓	5	0		

Column	Type	Key	Nullab	Date Pattern (Ctrl+F)	Length	Precision	Default	Comment
UserID	int	✓	✓		1	0		
MovieID	Integer	✓	✓		4	0		
Title	String		✓		52	0		
Year	Integer		✓		4	0		
Rating	Integer		✓		1	0		
Timestamp	Date		✓		6	0		"yyyy-MM-dd H...
Gender	String		✓					

# 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

User

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

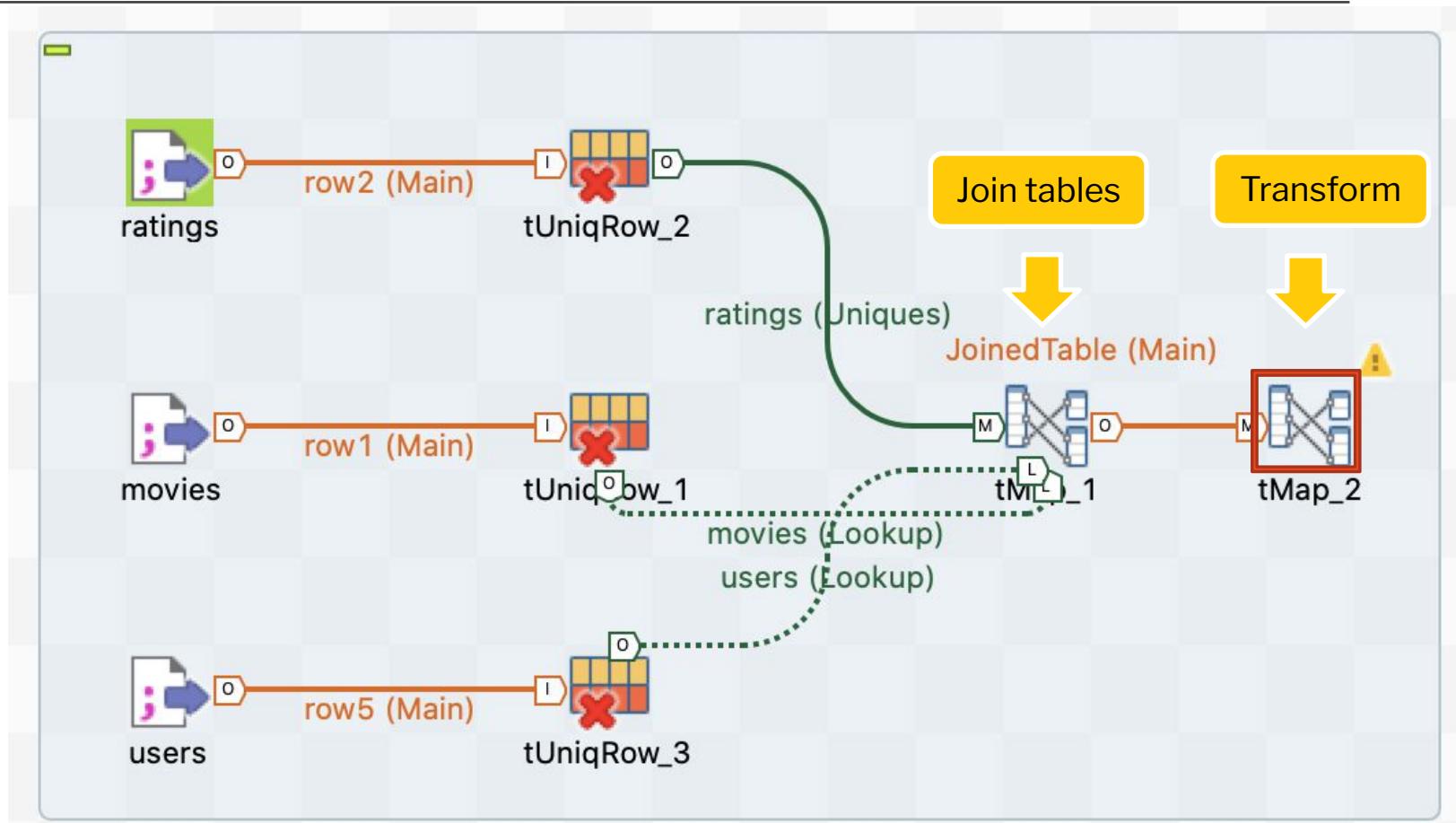
## Create TJob#4: Prepare dataset for desired datasets

### Transform Data:

Step 4.4: Convert birthdate to Age

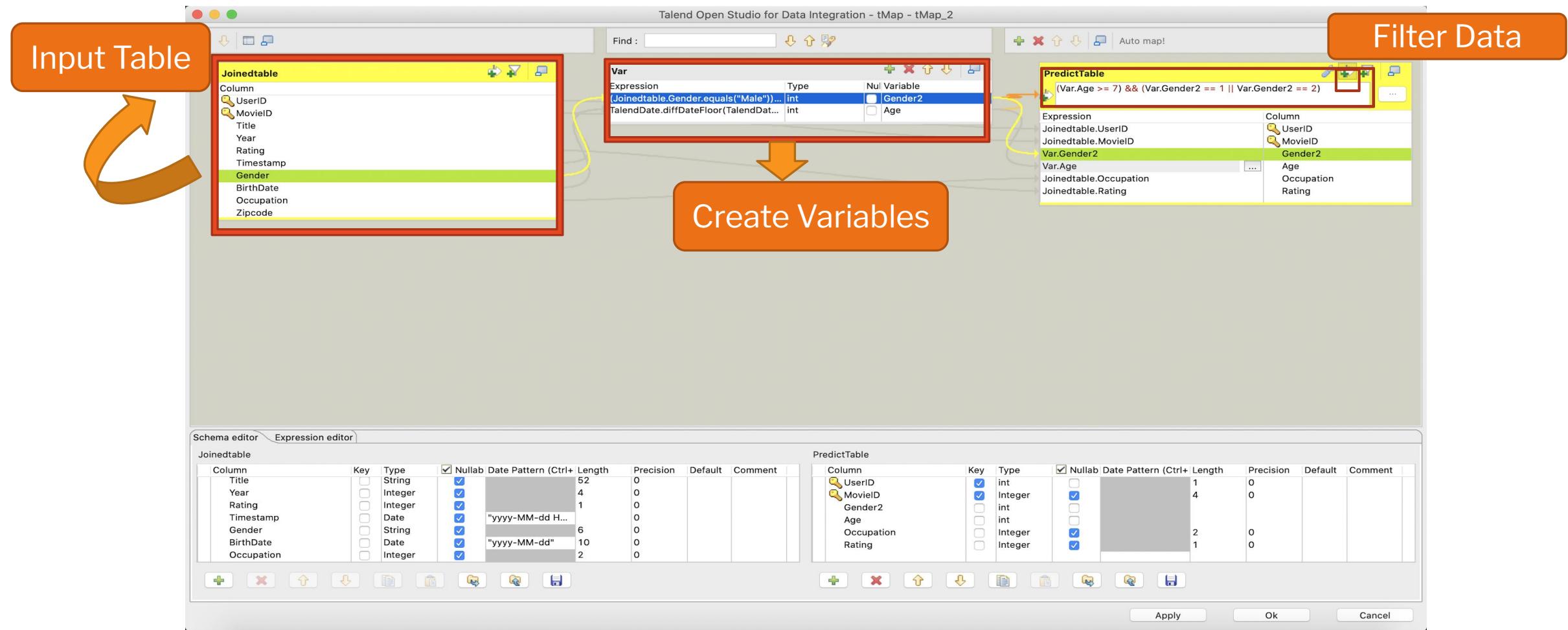
Step 4.5: Convert Gender to numeric

Step 4.6: Filter age  $\geq 7$  and  
Gender = 1|2



# Task5

## Create TJob#4: Prepare dataset for desired datasets



# Task5

## Create TJob#4: Prepare dataset for desired datasets

The screenshot shows the 'Expression Builder' window in Talend Open Studio. In the 'Var' section, there is a variable named 'Gender2' with the expression: `(Joinedtable.Gender.equals("Male"))?1:(Joinedtable.Gender.equals("Female"))?2:0`. An orange arrow points from this expression to the 'Expression' field in the main panel. The main panel shows a 'PredictTable' component with an 'Expression' condition: `(Var.Age >= 7) && (Var.Gender2 == 1 || Var.Gender2 == 2)`.

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

The screenshot shows the 'Expression Builder' window in Talend Open Studio. In the 'Var' section, there is a variable named 'Age' with the expression: `TalendDate.diffDateFloor(TalendDate.parseDate("yyyy-MM-dd", TalendDate.formatDate("yyyy-MM-dd", Joinedtable.Timestamp)), Joinedtable.BirthDate, "yyyy")`. An orange arrow points from this expression to the 'Expression' field in the main panel. The main panel shows a 'PredictTable' component with an 'Expression' condition: `(Var.Age >= 7) && (Var.Gender2 == 1 || Var.Gender2 == 2)`.

**Create variable named “Age”  
-> Calculate age**

# Task5

## Create TJob#4: Prepare dataset for desired datasets

---

### Create

#### Variables

Age

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

Call java method

Gender2

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

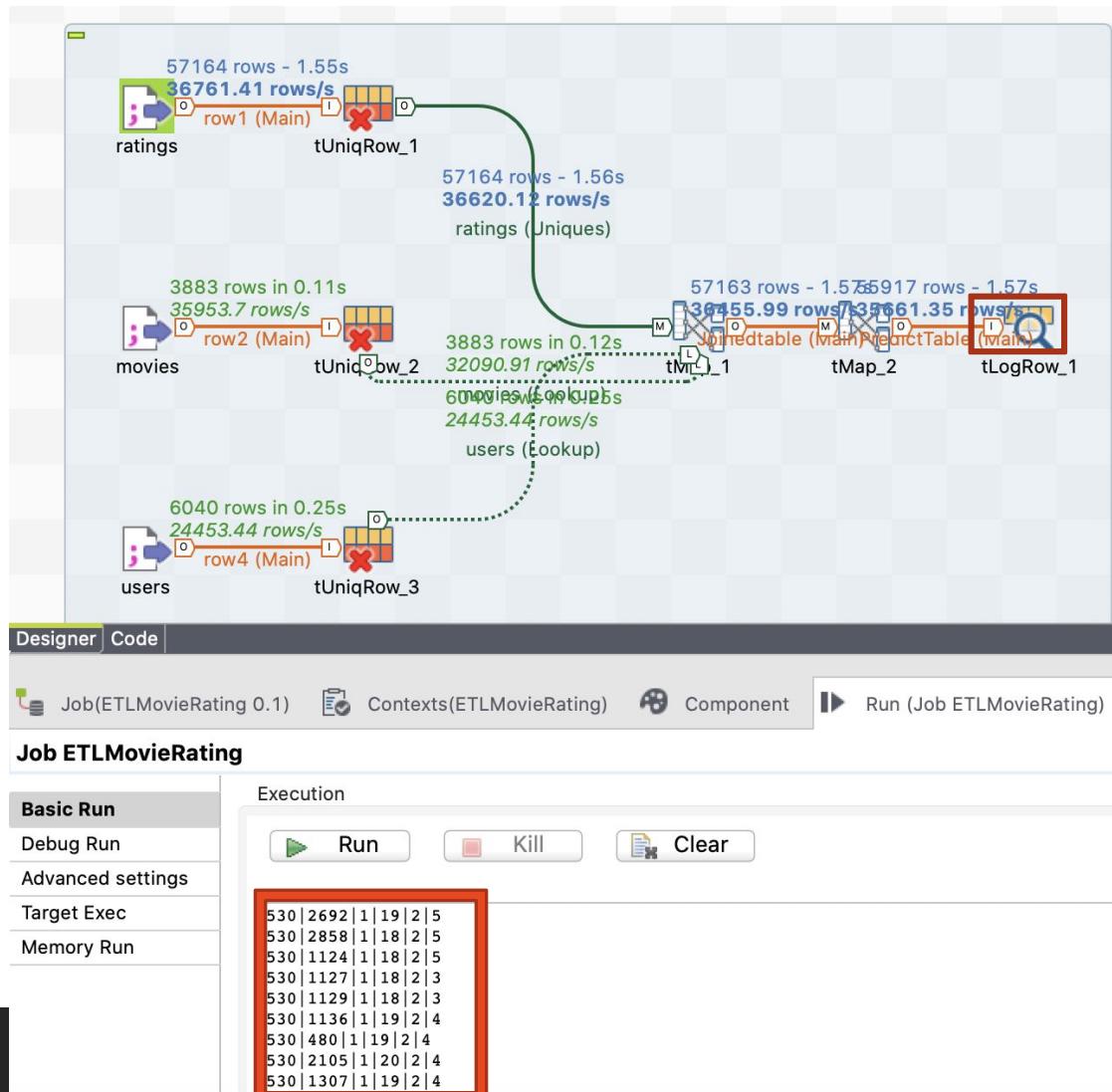
IF/ELSE Statement

### Filter Data

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

# Task5

## Create TJob#4: Prepare dataset for desired datasets



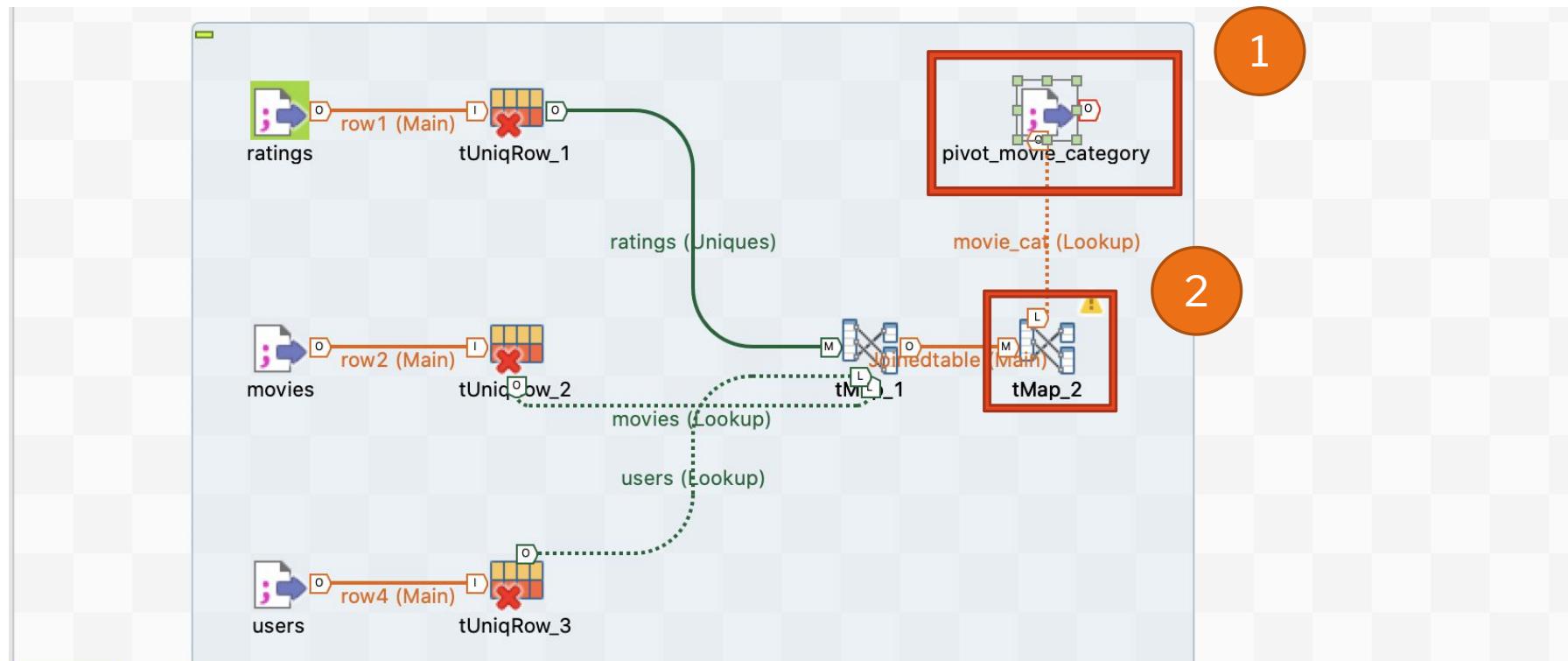
- Add tLogRow to check output
- Example Output:
  - MovieID
  - UserID
  - Gender2
  - Age
  - Occupation
  - Rating

# Task5

## Create TJob#4: Prepare dataset for desired datasets

### Step 4.7: Merge output of 4.1 and 4.6

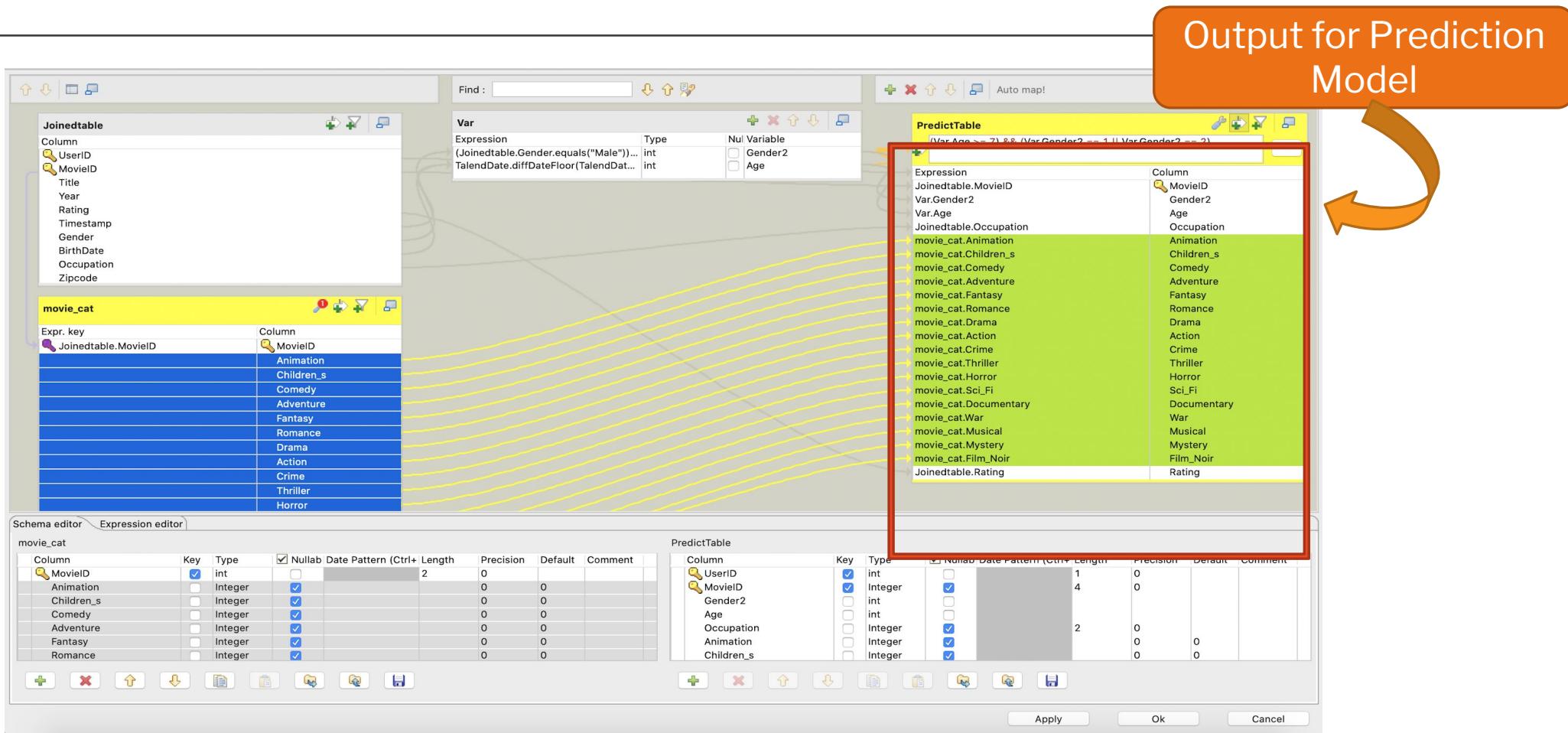
- Add File Delimited Metadata “pivot\_movie\_category” to merge the output at tMap\_2
- Double click at tMap\_2



# Task 5

## Create TJob#4: Prepare dataset for desired datasets

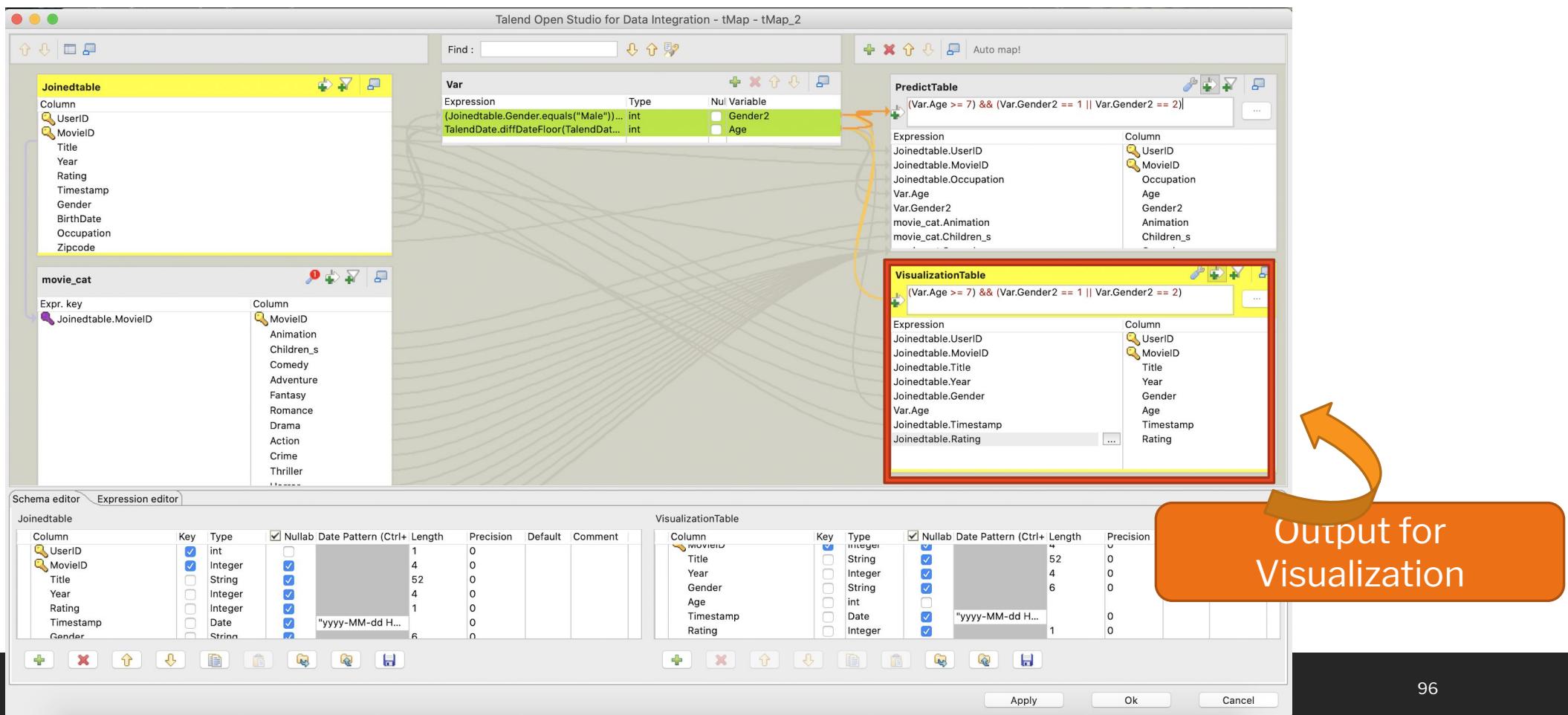
Step 4.7:  
Merge output of  
4.1 and 4.6 for  
Prediction Model



# Task5

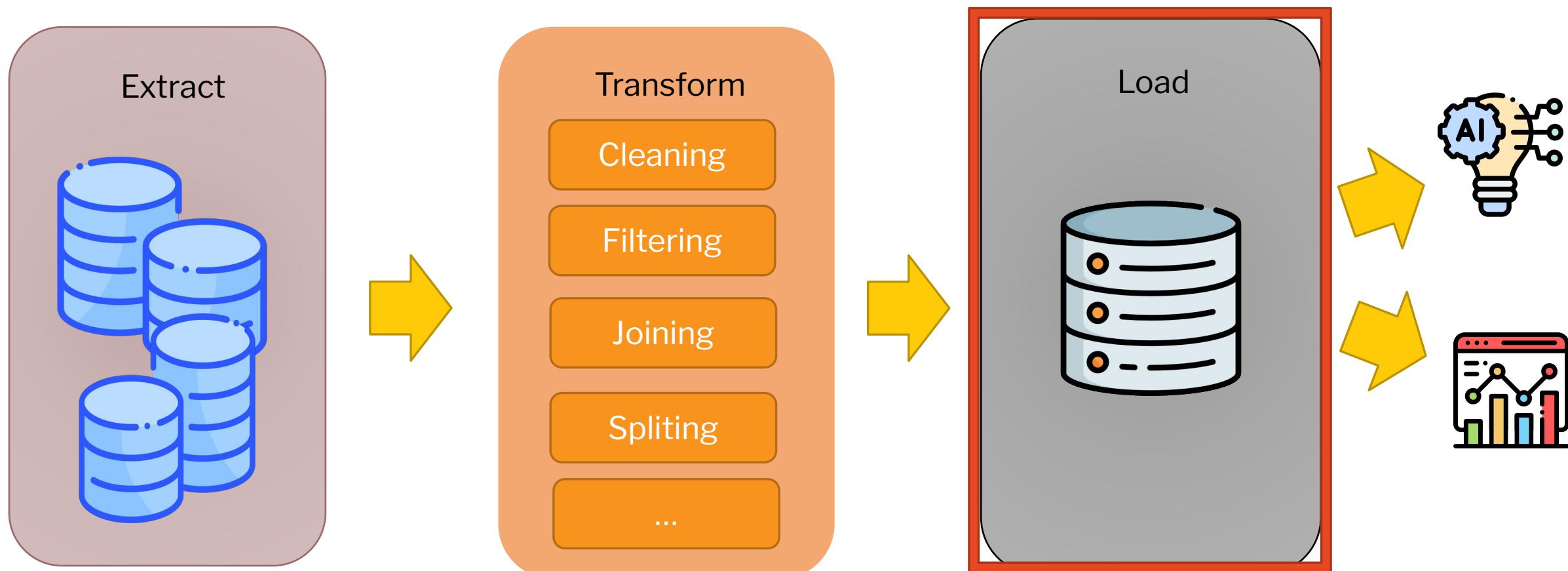
## Create TJob#4: Prepare dataset for desired datasets

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



# ETL Process

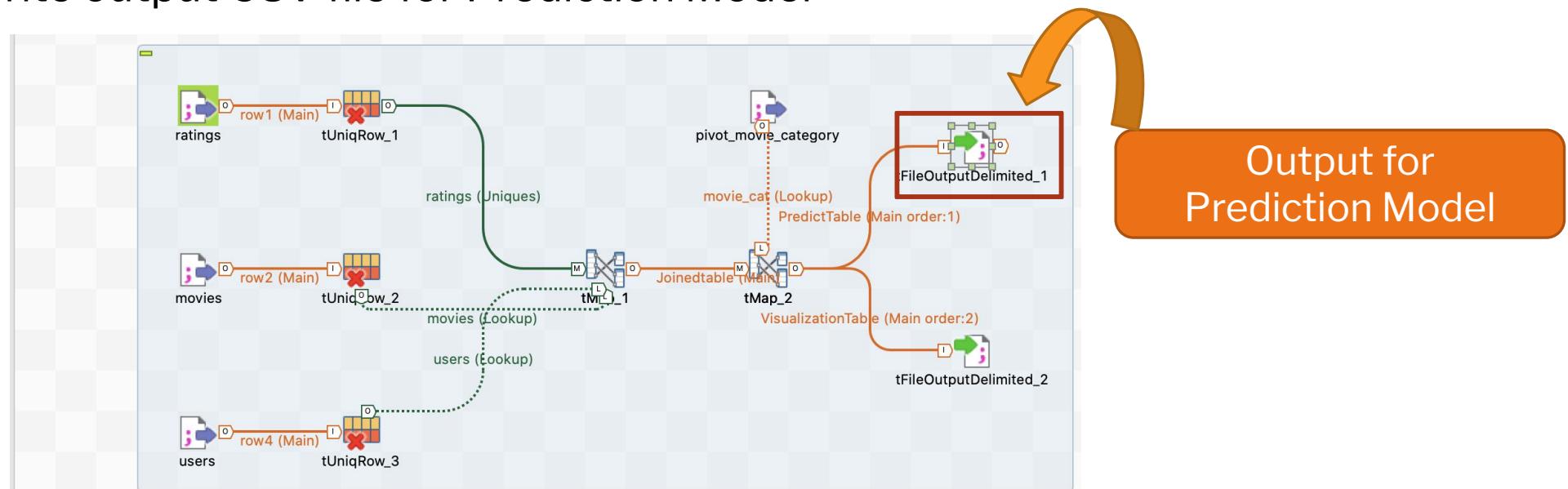
---



# Task5

## Create TJob#4: Prepare dataset for desired datasets

- Step 5: Write output CSV file for Prediction Model

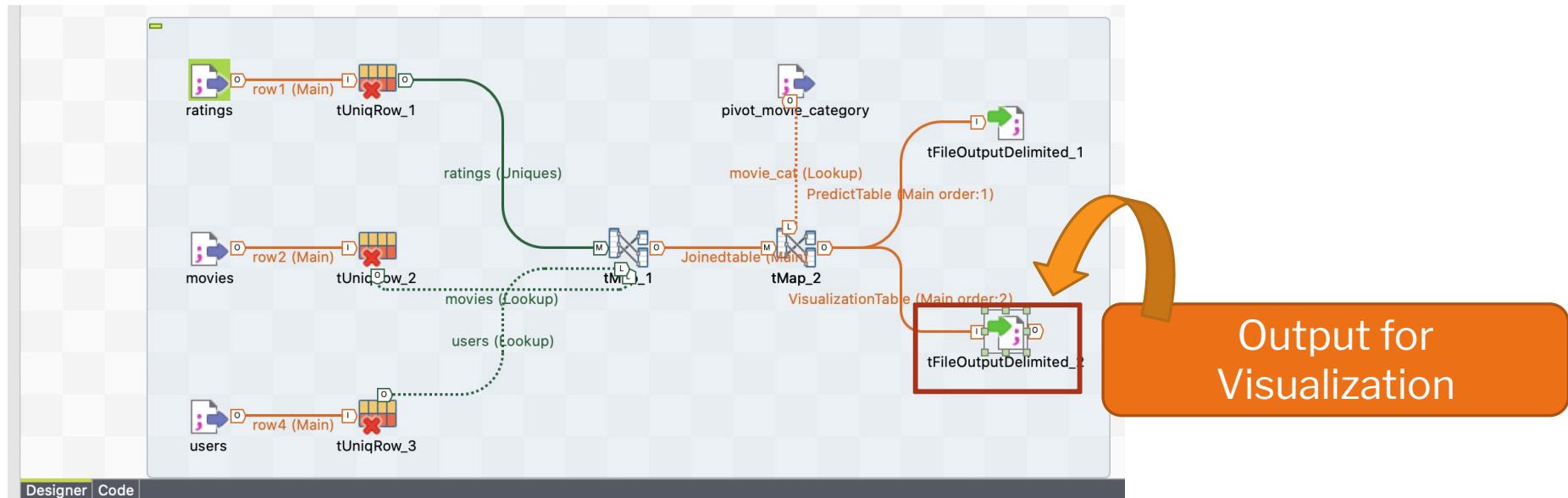


Output for  
Prediction Model

# 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 Task 5

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	0	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	0	1	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	0	1	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	1	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 Task 5

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	Death Wish (1974)	1	50	2014-10-22 00:11:11	2		

Output for  
Visualization

# References

---

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



Thank you.

---