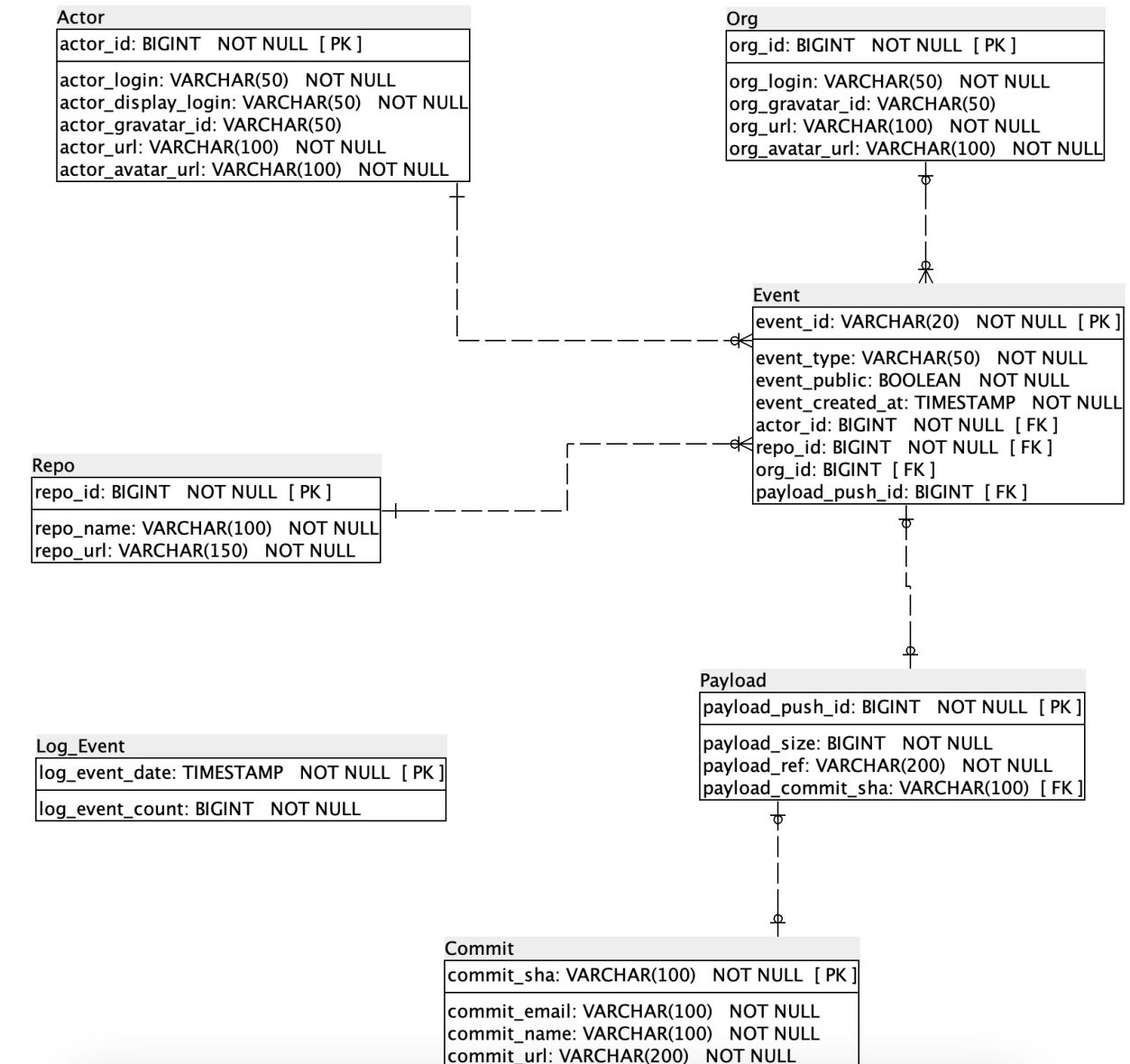


Lab5 – Project Airflow

Data model



** Table **Log_Event** is used to collect the log for loading of table **Event**.

Project implementation instruction

1. Reach the project repository '/swu-ds525/05-creating-and-scheduling-data-pipelines'

: \$ `cd 05-creating-and-scheduling-data-pipelines`

```
(base) JC@Napchins-MacBook-Air swu-ds525 % cd 05-creating-and-scheduling-data-pipelines
(base) JC@Napchins-MacBook-Air 05-creating-and-scheduling-data-pipelines %
```

2. Setup Environment (** ONLY FOR LINUX on 1ST TIME SETUP **)

: \$ `mkdir -p ./dags ./logs ./plugins`

: \$ `echo -e "AIRFLOW_UID=$(id -u)" > .env`

```
mkdir -p ./dags ./logs ./plugins
echo -e "AIRFLOW_UID=$(id -u)" > .env
```

3. Prepare the environment workspace thru 'docker-compose.yml'

: \$ `docker-compose up`

```
docker-compose up
```

These services will be up

- Apache Airflow : for task scheduling
- Postgres : for database
- Adminer : for Postgres access thru web service

4. Open Airflow and Postgres thru web service

Airflow :

The screenshot shows the Apache Airflow web interface in a browser. The address bar shows 'localhost:8080/home'. The top navigation bar includes the Airflow logo and links for DAGs, Datasets, Security, Browse, Admin, and Docs. The main heading is 'DAGs'. Below this, there are filters for 'All' (44), 'Active' (1), and 'Paused' (43). A search box 'Filter DAGs by tag' is also present. The table below lists DAGs with columns for DAG, Owner, Runs, Schedule, and Last Run. The first DAG listed is 'dataset_consumes_1', owned by 'airflow', with a status of 'consumes' and 'dataset-scheduled'.

DAG	Owner	Runs	Schedule	Last Run
dataset_consumes_1	airflow	0	Dataset	

Postgres : password = 'postgres'

Adminer 4.8.1

Language: English

Login

System	PostgreSQL
Server	warehouse
Username	postgres
Password
Database	postgres

☐ Permanent login

** below are Postgres connection information, setup in docker-compose.yaml

```
warehouse:
  image: postgres:13
  environment:
    POSTGRES_USER: postgres
    POSTGRES_PASSWORD: postgres
    POSTGRES_DB: postgres
  volumes:
    - warehouse-db-volume:/var/lib/postgresql/data
```

5. Check Airflow schedule

Frequency: Hourly

01/11/2022 13:49:20 25 All Run Types All Run States Clear Filters

Auto-refresh ☐

DAG: lab5_airflow Schedule: @hourly Next Run: 2022-11-01, 13:00:00

get_files
create_table
etl
log_event

Duration: 00:00:08

lab5_airflow / 2022-11-01, 13:00:00 UTC

Mark Failed Mark Success

Re-run: Clear existing tasks Queue up new tasks

Status	success
Run ID	scheduled__2022-11-01T12:00:00+00:00
Run type	scheduled
Run duration	00:00:08
Last scheduling decision	2022-11-01, 13:20:07 UTC
Started	2022-11-01, 13:19:59 UTC
Ended	2022-11-01, 13:20:07 UTC

** Schedule was run successfully

6. Check table creation and data loading in Postgres

← → ↻ localhost:8088/?pgsql=warehouse&username=postgres&db=postgres&ns=public

Language: English PostgreSQL » warehouse » postgres » Schema: public

Adminer 4.8.1

DB: postgres Schema: public

SQL command Import Export Create table

select actor
select committed
select event
select log_event
select org
select payload
select repo

Schema: public

Alter schema Database schema

Tables and views

Search data in tables (7)

<input type="checkbox"/>	Table	Engine	Collation	Data Length?	Index Length?	Data Free	Auto Increment	Rows?	Comment?
<input type="checkbox"/>	actor	table		24,576	40,960	?	?	127	
<input type="checkbox"/>	committed	table		40,960	65,536	?	?	152	
<input type="checkbox"/>	event	table		16,384	40,960	?	?	150	
<input type="checkbox"/>	log_event	table		8,192	16,384	?	?	0	
<input type="checkbox"/>	org	table		8,192	16,384	?	?	0	
<input type="checkbox"/>	payload	table		16,384	40,960	?	?	90	
<input type="checkbox"/>	repo	table		16,384	40,960	?	?	135	
	7 in total		en_US.utf8	131,072	262,144	0			

** Tables were created and data were loaded properly

7. Check data in tables

Table **log_event** : the timestamp and record count were inserted

← → ↻ localhost:8088/?pgsql=warehouse&username=postgres&db=postgres&ns=public&select=log_event

Language: English PostgreSQL » warehouse » postgres » public » Select: log_event

Adminer 4.8.1

DB: postgres Schema: public

SQL command Import Export Create table

select actor
select committed
select event
select log_event
select org
select payload
select repo

Select: log_event

Select data Show structure Alter table New item

Select Search Sort Limit 50 Action Select

SELECT * FROM "log_event" LIMIT 50 (0.001 s) Edit

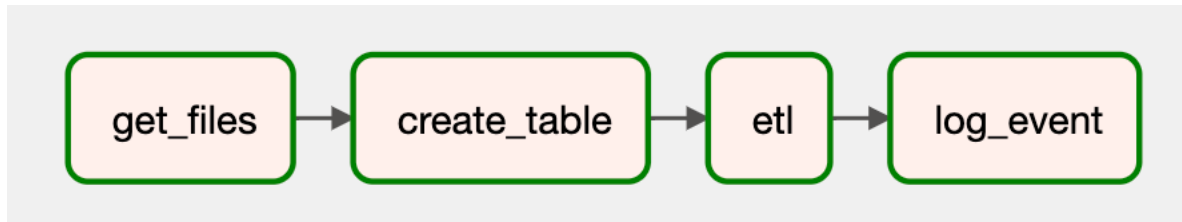
<input type="checkbox"/> Modify	log_event_date	log_event_count
<input type="checkbox"/> edit	2022-11-01 14:11:32.548421	150

Whole result 1 row Modify Save Selected (0) Edit Clone Delete Export (1)

Import

Appendix

1. Workflow



- Get data files
- Create tables if not exist
- Loading the data if not exist
- Log the loading information (timestamp of loading & record count) for table **Event**

2. Using Xcoms to store returned output of task

Xcoms variable:

	Key	Value	Timestamp	Dag Id	Task Id	Run Id	Map Index	Execution Date
<input type="checkbox"/>	return_value	[30, 30, 30, 30, 30]	2022-11-01, 15:11:02	lab5_airflow	etl	scheduled_2022-11-01T14:00:00+00:00		2022-11-01, 14:00:00

Access Xcoms variable:

```
def _log_event(**context):  
    ti = context['ti']  
    event_cnt = ti.xcom_pull(task_ids = 'etl', key = 'return_value')
```

3. DAG schedule

```
with DAG(  
    'lab5_airflow',  
    start_date = timezone.datetime(2022, 11, 1),  
    schedule = '@hourly',  
    tags = ['lab5'],  
    catchup = False,  
) as dag:
```

Start: 2022-11-01

Frequency: Hourly

DS525 - Chin Lertvipada - 64199130039