**Python Flask Commands**

# **Hands-on 4 – MySQL database commands**

**Installing MySQL CLI –**

**https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install.html**

**export MYSQLSH\_PROMPT\_THEME='MySQL > '**

**Connect to RDS MySQL –**

**\connect admin@<rds-endpoint>:3306**

**Create Customer table –**

**use financedb;**

**create table customer (customer\_id bigint, first\_name varchar(40),last\_name varchar(40),email varchar(40), phone\_number varchar(20), dob date, gender varchar(10), national\_id varchar(20),address\_line\_1 varchar(100), address\_line\_2 varchar(100),city varchar(100) , country varchar(100), postal\_code varchar(20),created\_at timestamp, updated\_at timestamp);**

**Explore Customer table –**

**show tables;**

**select count(\*) from customer;**

**select \* from customer limit 2;**

**describe customer;**

# **Hands-on 6 – Caching commands**

**Load Customer table in Aurora using LOAD command from S3 file**

**load data s3 file 's3://<bucket>/finance\_customer\_data\_5M.csv' into table customer fields terminated by ',' ignore 1 lines;**

**Install Flask Caching library**

**pip3 install Flask-Caching**

**Update customer table**

**update customer set first\_name='3aayaam' where customer\_id=106;**

# **Hands-on 7 – Deploy Flask**

**Run Flask app using FLASK\_APP variable**

**export FLASK\_APP='FlaskApp1.py'**

**flask run --port=8000**

**nohup flask run --port=8000 &**

**Run multiple apps in the same machine**

**export FLASK\_APP='FlaskApp1.py'**

**flask run --port=8001**

**export FLASK\_APP='FlaskApp2.py'**

**flask run --port=8002**

**export FLASK\_APP='FlaskApp1.py'**

**nohup flask run --port=8001 &**

**export FLASK\_APP='FlaskApp2.py'**

**nohup flask run --port=8002 &**

**Install Gunicorn**

**pip3 install gunicorn**

**Run Flask application using Gunicorn app server**

**gunicorn -w 4 -b 0.0.0.0:9001 'FlaskApp1:app1'**

**gunicorn -w 4 -b 0.0.0.0:9002 'FlaskApp2:app2'**

**nohup gunicorn -w 4 -b 0.0.0.0:9001 'FlaskApp1:app1' &**

**nohup gunicorn -w 4 -b 0.0.0.0:9002 'FlaskApp2:app2' &**

**Run Flask application using Gunicorn on EC2 instances**

**chmod 400 3aayaam-python-flask-key-pair.pem**

**ssh -i 3aayaam-python-flask-key-pair.pem <user>@<ip-addr>**

**python3 --version**

**pip3 install flask**

**pip3 install gunicorn**

**chmod 777 FlaskApp1.py**

**gunicorn -w 4 -b 0.0.0.0:9001 'FlaskApp1:app1'**

**gunicorn -w 4 -b 0.0.0.0:9002 'FlaskApp2:app2'**

# **Hands-on 8 – Web Application using HTML**

**Create accounts table**

**create table accounts (**

**account\_id varchar(15),**

**account\_type varchar(4),**

**customer\_id int,**

**balance decimal(10,2),**

**currency\_code varchar(4),**

**acc\_opening\_date timestamp,**

**last\_update\_date timestamp**

**);**

**create index accounts\_account\_id\_ix on accounts (account\_id);**

**create index accounts\_customer\_id\_ix on accounts (customer\_id);**

**Load Account table in Aurora using LOAD command from S3 file**

**load data s3 file 's3://<bucket>/bank\_account\_data.csv' into table accounts fields terminated by ',' ignore 1 lines;**

**Check no. of accounts for a customer, whether it matches the API**

**select distinct customer\_id from accounts;**

**select customer\_id, count(\*) as no\_of\_accounts from accounts group by customer\_id;**

**select customer\_id, count(\*) as no\_of\_accounts from accounts where customer\_id = <value> group by customer\_id;**

**select customer\_id, account\_id from accounts where customer\_id = <value>;**