CREATE OR REPLACE FUNCTION etl\_process()

RETURNS void AS $$

DECLARE

mean\_value numeric;

median\_value numeric;

mode\_value text;

BEGIN

COPY temp\_learner\_opportunity\_data(enrollment\_id, learner\_id, assigned\_cohort, apply\_date, status)

FROM 'C:/Users/Asus-PC/Downloads/learner\_opportunity\_data.csv' DELIMITER ',' CSV HEADER;

COPY temp\_marketing\_data("Ad Account Name", "Campaign name", "Delivery status", "Delivery level", "Result type", "Reporting starts", "Reach", "Outbound clicks", "Landing page views", "Results", "Cost per result", "Amount spent (AED)", "CPC", "rpc")

FROM 'C:/Users/Asus-PC/Downloads /marketing\_data.csv' DELIMITER ',' CSV HEADER;

COPY temp\_opportunity\_data(opportunity\_id, opportunity\_name, category, opportunity\_code, tracking\_questions)

FROM 'C:/Users/Asus-PC/Downloads /opportunity\_data.csv' DELIMITER ',' CSV HEADER;

COPY temp\_cohort\_data(cohort\_id, cohort\_code, start\_date, end\_date, size)

FROM 'C:/Users/Asus-PC/Downloads /cohort\_data.csv' DELIMITER ',' CSV HEADER;

COPY temp\_user\_data(learner\_id, country, degree, institution, major)

FROM 'C:/Users/Asus-PC/Downloads /user\_data.csv' DELIMITER ',' CSV HEADER;

COPY temp\_cognito\_data(user\_id, email, gender, UserCreateDate, UserLastModifiedDate, city, zip, state, birthdate)

FROM 'C:/Users/Asus-PC/Downloads /cognito\_data.csv' DELIMITER ',' CSV HEADER;

DELETE FROM temp\_learner\_opportunity\_data

WHERE enrollment\_id IS NULL OR learner\_id IS NULL OR assigned\_cohort IS NULL OR apply\_date IS NULL;

SELECT AVG(status) INTO mean\_value FROM temp\_learner\_opportunity\_data WHERE status IS NOT NULL;

UPDATE temp\_learner\_opportunity\_data SET status = mean\_value WHERE status IS NULL;

SELECT apply\_date INTO mode\_value

FROM temp\_learner\_opportunity\_data

GROUP BY apply\_date

ORDER BY COUNT(\*) DESC

LIMIT 1;

UPDATE temp\_learner\_opportunity\_data SET apply\_date = mode\_value WHERE apply\_date IS NULL;

DELETE FROM temp\_learner\_opportunity\_data a

USING temp\_learner\_opportunity\_data b

WHERE a.enrollment\_id = b.enrollment\_id

AND a.learner\_id = b.learner\_id

AND a.ctid < b.ctid;

SELECT PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY "Reach") INTO median\_value

FROM temp\_marketing\_data WHERE "Reach" IS NOT NULL;

UPDATE temp\_marketing\_data SET "Reach" = median\_value WHERE "Reach" IS NULL;

DELETE FROM temp\_marketing\_data a

USING temp\_marketing\_data b

WHERE a."Ad Account Name" = b."Ad Account Name"

AND a."Campaign name" = b."Campaign name"

AND a.ctid < b.ctid;

SELECT tracking\_questions INTO mode\_value

FROM temp\_opportunity\_data

GROUP BY tracking\_questions

ORDER BY COUNT(\*) DESC

LIMIT 1;

UPDATE temp\_opportunity\_data SET tracking\_questions = mode\_value WHERE tracking\_questions IS NULL;

DELETE FROM temp\_opportunity\_data a

USING temp\_opportunity\_data b

WHERE a.opportunity\_id = b.opportunity\_id

AND a.ctid < b.ctid;

SELECT PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY start\_date) INTO median\_value

FROM temp\_cohort\_data WHERE start\_date IS NOT NULL;

UPDATE temp\_cohort\_data SET start\_date = median\_value WHERE start\_date IS NULL;

SELECT PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY end\_date) INTO median\_value

FROM temp\_cohort\_data WHERE end\_date IS NOT NULL;

UPDATE temp\_cohort\_data SET end\_date = median\_value WHERE end\_date IS NULL;

DELETE FROM temp\_cohort\_data a

USING temp\_cohort\_data b

WHERE a.cohort\_id = b.cohort\_id

AND a.ctid < b.ctid;

UPDATE temp\_user\_data SET country = 'Unknown' WHERE country IS NULL;

DELETE FROM temp\_user\_data a

USING temp\_user\_data b

WHERE a.learner\_id = b.learner\_id

AND a.ctid < b.ctid;

UPDATE temp\_cognito\_data SET birthdate = '1900-01-01' WHERE birthdate IS NULL;

DELETE FROM temp\_cognito\_data a

USING temp\_cognito\_data b

WHERE a.user\_id = b.user\_id

AND a.ctid < b.ctid;

UPDATE temp\_learner\_opportunity\_data SET apply\_date = LOWER(apply\_date);

UPDATE temp\_marketing\_data SET "Ad Account Name" = LOWER("Ad Account Name");

UPDATE temp\_learner\_opportunity\_data SET status = NULL WHERE status < 0 OR status > 100;

INSERT INTO master\_table (

enrollment\_id, learner\_id, assigned\_cohort, apply\_date, status

)

SELECT

locd.enrollment\_id,

locd.learner\_id,

locd.assigned\_cohort,

locd.apply\_date,

locd.status

FROM temp\_learner\_opportunity\_data locd;

INSERT INTO master\_table (

"Ad Account Name", "Campaign name", "Delivery status", "Delivery level", "Result type", "Reporting starts", "Reach", "Outbound clicks", "Landing page views", "Results", "Cost per result", "Amount spent (AED)", "CPC", "rpc"

)

SELECT

mktd."Ad Account Name",

mktd."Campaign name",

mktd."Delivery status",

mktd."Delivery level",

mktd."Result type",

mktd."Reporting starts",

mktd."Reach",

mktd."Outbound clicks",

mktd."Landing page views",

mktd."Results",

mktd."Cost per result",

mktd."Amount spent (AED)",

mktd."CPC",

mktd."rpc"

FROM temp\_marketing\_data mktd;

INSERT INTO master\_table (

opportunity\_id, opportunity\_name, category, opportunity\_code, tracking\_questions

)

SELECT

opd.opportunity\_id,

opd.opportunity\_name,

opd.category,

opd.opportunity\_code,

opd.tracking\_questions

FROM temp\_opportunity\_data opd;

INSERT INTO master\_table (

cohort\_id, cohort\_code, start\_date, end\_date, size

)

SELECT

ctd.cohort\_id,

ctd.cohort\_code,

ctd.start\_date,

ctd.end\_date,

ctd.size

FROM temp\_cohort\_data ctd;

INSERT INTO master\_table (

learner\_id, country, degree, institution, major

)

SELECT

uld.learner\_id,

uld.country,

uld.degree,

uld.institution,

uld.major

FROM temp\_user\_data uld;

INSERT INTO master\_table (

user\_id, email, gender, UserCreateDate, UserLastModifiedDate, city, zip, state, birthdate

)

SELECT

ctd.user\_id,

ctd.email,

ctd.gender,

ctd.UserCreateDate,

ctd.UserLastModifiedDate,

ctd.city,

ctd.zip,

ctd.state,

ctd.birthdate

FROM temp\_cognito\_data ctd;

DROP TABLE IF EXISTS temp\_learner\_opportunity\_data;

DROP TABLE IF EXISTS temp\_marketing\_data;

DROP TABLE IF EXISTS temp\_opportunity\_data;

DROP TABLE IF EXISTS temp\_cohort\_data;

DROP TABLE IF EXISTS temp\_user\_data;

DROP TABLE IF EXISTS temp\_cognito\_data;

END;

$$ LANGUAGE plpgsql;

SELECT etl\_process();