

Azis Muslim
Role: Data Engineer

1. Create script to create table for each object
 - a. Employee
 - b. Position History

Query	Query History
1	CREATE TABLE employee(2 eid int NOT NULL PRIMARY KEY , 3 employeeid VARCHAR(25) , 4 fullname VARCHAR(25) NOT NULL , 5 birthdate DATE NOT NULL , 6 address VARCHAR(25) NOT NULL); 7 8 INSERT INTO employee(eid, employeeid, fullname, birthdate, address) 9 VALUES (1, 10105001, 'Ali Anton', '19-Aug-82', 'Jakarta Utara'); 10 11 INSERT INTO employee(eid, employeeid, fullname, birthdate, address) 12 VALUES (2, 10105002, 'Rara Siva', '1-Jan-82', 'Mandalika'); 13 14 INSERT INTO employee(eid, employeeid, fullname, birthdate, address) 15 VALUES (3, 10105003, 'Rini Aini', '20-Feb-82', 'Sumbawa Besar'); 16 17 INSERT INTO employee(eid, employeeid, fullname, birthdate, address) 18 VALUES (4, 10105004, 'Budi', '22-Feb-82', 'Mataram Kota'); 19 20

Data Output	Messages	Notifications
INSERT 0 1		
Query returned successfully in 69 msec.		

Figure.1

Gambar di atas adalah query sql untuk membuat tabel employee dengan kolom eid, employeeid, fullname, birthdate, address. Setelah itu pada query selanjutnya adalah query sql untuk melakukan input value yang ada pada tabel di setiap kolomnya.

Query Query History

1 CREATE TABLE position_history(
2 pid int NOT NULL PRIMARY KEY,
3 posid VARCHAR(25) NOT NULL,
4 postitle VARCHAR(25) NOT NULL,
5 employeeid VARCHAR(25) NOT NULL,
6 startdate DATE NOT NULL,
7 enddate DATE NOT NULL);
8
9 INSERT INTO position_history(pid, posid, postitle, employeeid, startdate, enddate)
10 VALUES(1, 50000, 'IT Manager', 10105001,'1-Jan-2022', '28-Feb-2022');
11
12 INSERT INTO position_history(pid, posid, postitle, employeeid, startdate, enddate)
13 VALUES(2, 50001, 'IT Sr. Manager', 10105001,'1-Mar-2022', '31-Dec-2022');
14
15 INSERT INTO position_history(pid, posid, postitle, employeeid, startdate, enddate)
16 VALUES(3, 50002, 'Programmer Analyst', 10105002,'1-Jan-2022', '28-Feb-2022');
17
18 INSERT INTO position_history(pid, posid, postitle, employeeid, startdate, enddate)
19 VALUES(4, 50003, 'Sr. Programmer Analyst', 10105002,'1-Mar-2022', '31-Dec-2022');
20
21 INSERT INTO position_history(pid, posid, postitle, employeeid, startdate, enddate)
22 VALUES(5, 50004, 'IT Admin', 10105003, '1-Jan-2022', '28-Feb-2022');
23
24 INSERT INTO position_history(pid, posid, postitle, employeeid, startdate, enddate)
25 VALUES(6, 50005, 'IT Secretary', 10105003,'1-Mar-2022', '31-Dec-2022');
26

Data Output Messages Notifications

INSERT 0 1

Query returned successfully in 214 msec.

Figure. 2

Pada figure.2 di atas adalah query sql untuk create tabel position history dengan kolom pid, posid, employeeid, startdate, dan enddate. Selanjutnya pada query sql di bawahnya adalah query sql untuk melakukan input value dari kolom pada tabal yang telah di create

Query

Query History

Scratch Pad

x

1

SELECT * FROM employee

2

INNER JOIN position_history

3

ON employee.employeeid = position_history.employeeid

Data Output

Messages

Notifications

	eid integer	employeeid character varying (25)	fullname character varying (25)	birthdate date	address character varying (25)	pid integer	posid character varying (25)	postitle character varying (25)	employ charact
1	1	10105001	Ali Anton	1982-08-19	Jakarta Utara	2	50001	IT Sr. Manager	10105001
2	1	10105001	Ali Anton	1982-08-19	Jakarta Utara	1	50000	IT Manager	10105001
3	2	10105002	Rara Siva	1982-01-01	Mandalika	4	50003	Sr. Programmer Analy...	10105002
4	2	10105002	Rara Siva	1982-01-01	Mandalika	3	50002	Programmer Analyst	10105002
5	3	10105003	Rini Aini	1982-02-20	Sumbawa Besar	6	50005	IT Secretary	10105003
6	3	10105003	Rini Aini	1982-02-20	Sumbawa Besar	5	50004	IT Admin	10105003

Figure. 3

Pada figure.3 di atas menjelaskan query untuk melihat kolom yang terdapat pada tabel employee dan tabel position history. Query yang digunakan adalah inner join untuk menggabungkan keseluruhan tabel yang kesesuaian.

1. Data Warehouse Architecture Design

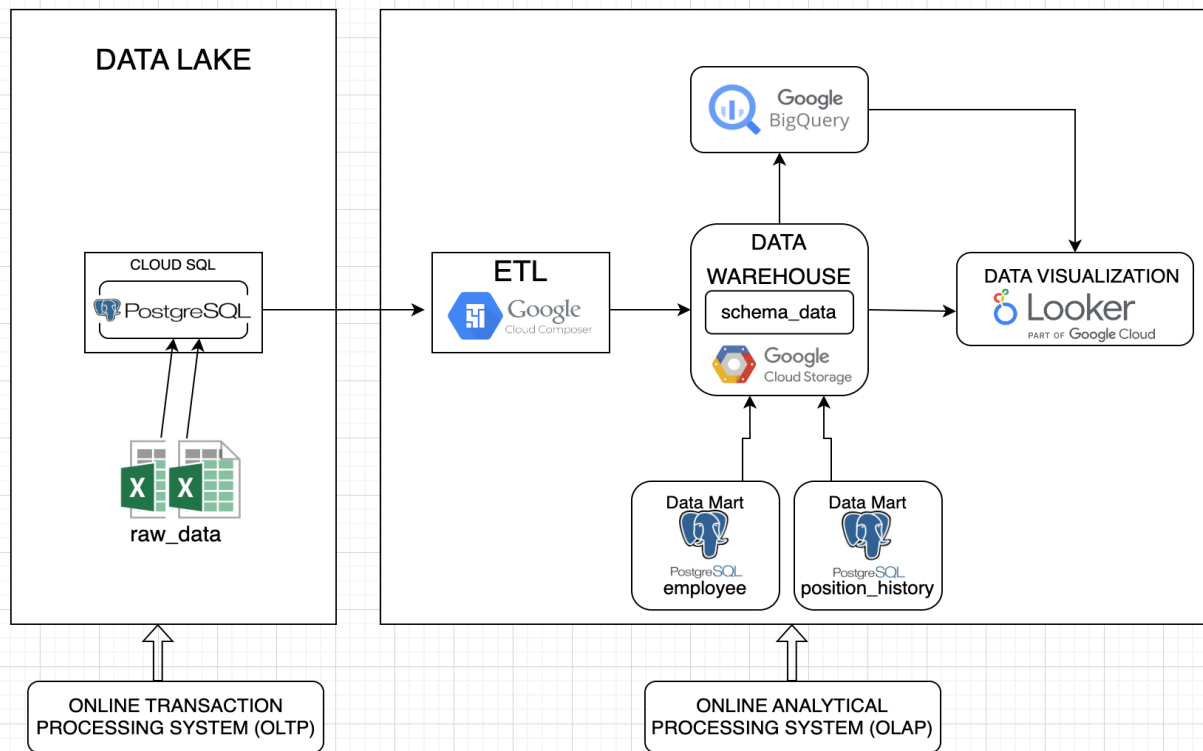


Figure.4

Pada figure.4 di atas menjelaskan arsitektur sederhana mengenai data pipeline untuk data warehouse yang berisikan tabel employee dan tabel position history. Pada data lake tempat dimana semua data dari berbagai sumber dan berbagai jenis di simpan. Pada data lake di atas jenis storage yang digunakan adalah cloud sql. Cloud sql adalah jenis storage yang ada pada google cloud yang dapat digunakan untuk menyimpan structured data. Platform yang digunakan untuk menyimpan structured data tersebut adalah postgresql. Alasannya adalah karena postgresql memungkinkan untuk di scale up jika data yang disimpan nantinya akan terus bertambah banyak.

Proses selanjutnya adalah proses ETL, pada diagram di atas saya menempatkan Google composer sebagai fitur yang dapat digunakan untuk ETL. Sebagai penjelasan singkat, ETL sendiri adalah proses ekstraksi, transformasi, dan load dari berbagai jenis data yang di ingest dari data lake. Sebelum masuk ke dalam data warehouse raw data harus dilakukan clean up terlebih dahulu.

Data yang sudah di proses dari dirty data to clean data akan di simpan pada data warehouse atau fitur pada google cloud platform yang bisa disebut google cloud storage. Pada google cloud storage inilah data warehouse dan jenis schema databasenya dibuat.

Proses terakhir dari data pipeline adalah visualisasi data. Berdasarkan figure.4 visualisasi data dapat dilakukan dengan menggunakan fitur google cloud platform yaitu looker. Looker memungkinkan data analyst untuk melakukan visualisasi data berdasarkan query sql, namun query harus dilakukan melalui big query yang secara otomatis terhubung ke looker. Dari hasil query yang sudah di running hasil visualisasinya akan langsung muncul karena big query dan looker secara default sudah terintegrasi.