**Jawaban**

1. https://lucid.app/lucidchart/73d370d4-82f8-4e37-868d-ac6fdbf86df3/edit?viewport\_loc=-69%2C-309%2C1735%2C872%2C0\_0&invitationId=inv\_ed0c2c65-edab-4630-90af-53de495453bc

2. Query DDL :

- Tabel Participants :

CREATE TABLE participants(nip SERIAL NOT NULL, name VARCHAR(100) NOT NULL, PRIMARY KEY (nip));

- Tabel Assignments :

CREATE TABLE assignments(id SERIAL NOT NULL, name VARCHAR(100) NOT NULL, PRIMARY KEY (id));

- Tabel Grades :

* CREATE TABLE grades(id SERIAL NOT NULL, grade INTEGER, assignment\_id BIGINT, participant\_nip BIGINT, PRIMARY KEY (id));
* ALTER TABLE grades ADD CONSTRAINT participant\_fk FOREIGN KEY (participant\_nip) REFERENCES participants(nip);
* ALTER TABLE grades ADD CONSTRAINT assignment\_fk FOREIGN KEY (assignment\_id) REFERENCES assignments(id);

3. Query DML :

a. SELECT a.name AS assignment\_name, g.grade

FROM assignments a

JOIN grades g ON a.id = g.assignment\_id

WHERE g.participant\_nip = 1 ORDER BY assignment\_name;

b. SELECT p.nip, p.name AS participant\_name,

case when AVG(g.grade) IS NULL then 0

else AVG(g.grade) end as average\_grade

FROM participants p

LEFT JOIN grades g

ON p.nip = g.participant\_nip

GROUP BY p.nip, p.name ORDER BY p.nip;

c. SELECT

CASE WHEN AVG\_grade BETWEEN 80 AND 100 THEN '80-100'

WHEN AVG\_grade BETWEEN 60 AND 79 THEN '60-79'

WHEN AVG\_grade BETWEEN 40 AND 59 THEN '40-59'

WHEN AVG\_grade BETWEEN 20 AND 39 THEN '20-39'

WHEN AVG\_grade BETWEEN 0 AND 19 THEN '0-19'

WHEN AVG\_grade is NULL THEN '0-19' END AS Rentang,

COUNT(nip) AS Jumlah\_Peserta

FROM ( SELECT p.nip, AVG(g.grade) AS AVG\_grade

FROM participants p LEFT JOIN grades g ON p.nip = g.participant\_nip GROUP BY p.nip)

AS AvgGrades GROUP BY Rentang ORDER BY Rentang;

d. SELECT

CASE WHEN AVG\_grade BETWEEN 80 AND 100 THEN '80-100'

WHEN AVG\_grade BETWEEN 60 AND 79 THEN '60-79'

WHEN AVG\_grade BETWEEN 40 AND 59 THEN '40-59'

WHEN AVG\_grade BETWEEN 20 AND 39 THEN '20-39'

WHEN AVG\_grade BETWEEN 0 AND 19 THEN '0-19'

WHEN AVG\_grade IS NULL THEN '0-19' END AS Rentang,

string\_agg(p.name, ', ') AS Nama\_Peserta,

COUNT(p.nip) AS Jumlah\_Peserta

FROM ( SELECT p.nip, AVG(g.grade) AS AVG\_grade

FROM participants p

LEFT JOIN grades g ON p.nip = g.participant\_nip

GROUP BY p.nip )

AS AvgGrades LEFT JOIN participants p

ON AvgGrades.nip = p.nip GROUP BY Rentang ORDER BY Rentang;

e.

* ALTER TABLE grades ADD COLUMN date DATE DEFAULT NOW();
* SELECT TO\_CHAR(distinct\_dates.date, 'YYYY-MM-DD') as tanggal, COALESCE(COUNT(grades.date), 0) AS Jumlah\_Penilaian

FROM (

SELECT generate\_series('2023-08-20'::date, '2023-09-01'::date, '1 day'::interval) AS date

) AS distinct\_dates

LEFT JOIN grades ON distinct\_dates.date = DATE(grades.date)

GROUP BY distinct\_dates.date ORDER BY distinct\_dates.date;

Hasil Screenshot













