

Level #3

Deskripsi

Nama : Zakaria

NIP : ARN222-17196

Format: Use Case dan Praktek

Penilaian: Share script via Git dan/atau dokumentasi

Objektif:

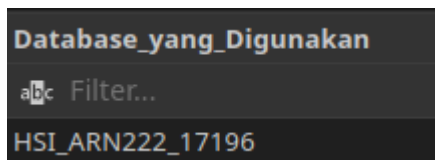
- Peserta mampu memahami dan melakukan Join Query.
- Peserta mampu menggunakan function built-in pada MariaDB/MySQL.
- Peserta mampu melakukan sub query.

Prolog

Dalam men-*design* sebuah database, tentunya terdapat beberapa data referensi yang menjadi penunjang pada data-data utama dalam database. Kali ini kita akan membuat beberapa referensi data pada database ini, serta keterkaitan antar tabel-tabel yang ada pada database.

Cek Database yang Digunakan

```
USE HSI_ARN222_17196;  
SELECT DATABASE() AS Database_yang_Digunakan;
```



Pre Requisites

Sebelum mengerjakan soal di bawah, silahkan melakukan task berikut ini.

1. Buatlah tabel *ref_evaluasi* dengan kolom-kolom di bawah ini.

Nama Kolom	Tipe Data	Panjang Data	Mandatory	Default Value
Kode	String	2	Ya	
Nama	String	30	Ya	
Bobot	Number		Ya	
Created_At	Date		Ya	Current time

Nama Kolom	Tipe Data	Panjang Data	Mandatory	Default Value
Updated_At	Date		Tidak	

- a. Kolom Kode adalah Primary Key.
- b. Isi data pada tabel ini dengan data sebagai berikut:
 - i. Kode: EH, Nama: Evaluasi Harian, Bobot: 40
 - ii. Kode: EP, Nama: Evaluasi Pekan, Bobot: 25
 - iii. Kode: EA, Nama: Evaluasi Akhir, Bobot: 35

Cek Describe tabel ref_evaluasi

DESCRIBE ref_evaluasi;

Field	Type	Null	Key	Default	Extra
kode	varchar(2)	NO	PRI	NULL	
nama	varchar(30)	NO		NULL	
bobot	int(11)	NO		NULL	
created_at	timestamp	NO		current_timestamp()	on update current_timestamp()
updated_at	timestamp	NO		0000-00-00 00:00:00	

Cek tabel ref_evaluasi

SELECT * FROM ref_evaluasi;

kode	nama	bobot	created_at	updated_at
EA	Evaluasi Akhir	35	2024-02-09 21:24:51	0000-00-00 00:00:00
EH	Evaluasi Harian	40	2024-02-09 21:24:51	0000-00-00 00:00:00
EP	Evaluasi Pekan	25	2024-02-09 21:24:51	0000-00-00 00:00:00

2. Buatlah tabel *nilai_peserta* dengan kolom-kolom di bawah ini.

Nama Kolom	Tipe Data	Panjang Data	Mandatory	Default Value
ID	Number		Ya	
NIP	String	12	Ya	
Jenis Evaluasi	String	2	Ya	
Mulai Evaluasi	Date		Ya	
Akhir Evaluasi	Date		Ya	
Nilai	Number		Ya	
Created_At	Date		Ya	Current time

Nama Kolom	Tipe Data	Panjang Data	Mandatory	Default Value
Updated_At	Date		Tidak	

Dengan kriteria sebagai berikut:

- ID adalah Primary Key, dengan auto increment.
- NIP adalah foreign key ke tabel *Peserta*.
- Jenis Evaluasi adalah foreign key ke tabel *ref_evaluasi*. Tambahkan klausul apabila value di tabel *ref_evaluasi* di-*update*, maka ter-*update* juga di tabel ini. Dan apabila value di tabel *ref_evaluasi* di-*delete*, maka tidak bisa (di-*restrict*).
- Kolom “Mulai Evaluasi” dan “Akhir Evaluasi” berisi tanggal dan waktu.
- Buatlah constraint agar nilai yang diisi pada kolom “Nilai” tidak boleh lebih dari 100.
- Insert sample data berikut (*lihat di Google Classroom*) ke tabel *nilai_peserta*.

query:

```
CREATE TABLE nilai_peserta (
  id INT NOT NULL AUTO_INCREMENT,
  nip VARCHAR(12) NOT NULL,
  jenis_evaluasi VARCHAR(2) NOT NULL,
  mulai_evaluasi TIMESTAMP NOT NULL,
  akhir_evaluasi TIMESTAMP NOT NULL,
  nilai INT NOT NULL,
  created_at TIMESTAMP NOT NULL,
  updated_at TIMESTAMP,
  PRIMARY KEY(id),
  CONSTRAINT `check_nilai` CHECK((nilai <= 100) & (nilai >=
0)),
  CONSTRAINT `fk_nilaipeserta_nip`
  FOREIGN KEY (nip) REFERENCES peserta(nip),
  CONSTRAINT `fk_nilaipeserta_jeniseval`
  FOREIGN KEY (jenis_evaluasi) REFERENCES ref_evaluasi(kode)
  ON UPDATE CASCADE
  ON DELETE RESTRICT
);
```

2F. Cek tabel *nilai_peserta* dengan limit 10

```
SELECT * FROM nilai_peserta LIMIT 10;
```

id	nip	jenis_evaluasi	mulai_evaluasi	akhir_evaluasi	nilai	created_at	updated_at
1	ARN999-01009	EH	2024-02-22 20:56:35	2024-02-22 21:22:13	61	0000-00-00 00:00:00	0000-00-00 00:00:00
2	ARN999-04009	EP	2024-01-25 22:45:56	2024-01-25 23:17:02	73	0000-00-00 00:00:00	0000-00-00 00:00:00
3	ARN999-06008	EH	2024-01-06 06:20:02	2024-01-06 06:25:07	75	0000-00-00 00:00:00	0000-00-00 00:00:00
4	ARN999-07015	EA	2024-01-15 23:31:37	2024-01-15 23:44:44	35	0000-00-00 00:00:00	0000-00-00 00:00:00
5	ARN999-06010	EH	2024-02-01 18:02:58	2024-02-01 18:21:53	38	0000-00-00 00:00:00	0000-00-00 00:00:00
6	ARN999-10020	EP	2024-01-31 04:43:07	2024-01-31 05:01:45	100	0000-00-00 00:00:00	0000-00-00 00:00:00
7	ARN999-10018	EP	2024-01-12 12:59:55	2024-01-12 13:06:29	35	0000-00-00 00:00:00	0000-00-00 00:00:00
8	ARN999-04017	EH	2024-02-02 05:26:28	2024-02-02 05:55:34	86	0000-00-00 00:00:00	0000-00-00 00:00:00
9	ARN999-08013	EH	2024-01-10 12:05:46	2024-01-10 12:29:27	62	0000-00-00 00:00:00	0000-00-00 00:00:00
10	ARN999-10011	EP	2024-01-16 02:07:48	2024-01-16 02:17:54	55	0000-00-00 00:00:00	0000-00-00 00:00:00

Soal

1. Lakukan task berikut ini.
 - a. Buat tabel *Kota* dan masukkan data referensi kota dan id-nya sesuai data pada poin 3.b di Soal Level #2. (Primary Key: kolom id)

Nama Kolom	Tipe Data	Panjang Data	Mandatory	Default Value
ID	String	3	Ya	
Nama Kota	String	50	Ya	
Created_At	Date		Ya	Current time
Updated_At	Date		Tidak	

Membuat tabel kota

```
CREATE TABLE kota (
  id VARCHAR(3) NOT NULL,
  nama_kota VARCHAR(50) NOT NULL,
  created_at TIMESTAMP NOT NULL,
  update_at TIMESTAMP,
  PRIMARY KEY (id)
);
```

Insert data pada tabel kota

```
INSERT INTO kota (id, nama_kota)
VALUES ('001', 'Jakarta'),
      ('002', 'Bekasi'),
      ('003', 'Depok'),
      ('004', 'Bogor'),
      ('005', 'Tangerang'),
      ('006', 'Aceh'),
      ('007', 'Medan'),
      ('008', 'Pekanbaru'),
      ('009', 'Padang'),
```

```
('010', 'Bandung');
```

Cek tabel kota

```
SELECT * FROM kota;
```

id	nama_kota	created_at	update_at
abc Filter...	abc Filter...	abc Filter...	abc Filter...
001	Jakarta	2024-02-10 14:41:25	0000-00-00 00:00:00
002	Bekasi	2024-02-10 14:41:25	0000-00-00 00:00:00
003	Depok	2024-02-10 14:41:25	0000-00-00 00:00:00
004	Bogor	2024-02-10 14:41:25	0000-00-00 00:00:00
005	Tangerang	2024-02-10 14:41:25	0000-00-00 00:00:00
006	Aceh	2024-02-10 14:41:25	0000-00-00 00:00:00
007	Medan	2024-02-10 14:41:25	0000-00-00 00:00:00
008	Pekanbaru	2024-02-10 14:41:25	0000-00-00 00:00:00
009	Padang	2024-02-10 14:41:25	0000-00-00 00:00:00
010	Bandung	2024-02-10 14:41:25	0000-00-00 00:00:00

- b. Buat relasi (foreign key) antara tabel *Peserta* dan *Kota*.

```
ALTER TABLE peserta
```

```
ADD CONSTRAINT `fk_kota`
```

```
FOREIGN KEY(domisili) REFERENCES kota(id);
```

- c. Tampilkan data nip, nama peserta, tempat lahir, tanggal lahir (format: "dd month yyyy", contoh: 31 Desember 2023) dan domisili (nama kota) dengan melakukan join antara tabel *Peserta* dan *Kota*. (saya limit 10)

```
SELECT p.nip,  
       CONCAT(p.nama_depan, ' ', p.nama_belakang) AS  
       'Nama',  
       p.tempat_lahir,  
       DATE_FORMAT(p.tanggal_lahir, "%d %M %Y") AS  
       'tanggal_lahir',  
       k.nama_kota AS domisili  
FROM peserta AS p INNER JOIN kota AS k  
ON p.domisili = k.id  
LIMIT 10;
```

2. Lakukan task berikut ini.

- a. Alter tabel *Peserta*, tambahkan kolom berikut.

Nama Kolom	Tipe Data	Panjang Data	Mandatory	Default Value
Angkatan	String	6	Tidak	
Grup	String	2	Tidak	

Add column angkatan

```
ALTER TABLE peserta
ADD COLUMN angkatan VARCHAR(6) NOT NULL AFTER alamat;
```

```
Add column grup
ALTER TABLE peserta
ADD COLUMN grup VARCHAR(2) NOT NULL AFTER angkatan;
```

```
describe tabel peserta
DESCRIBE peserta;
```

Field	Type	Null	Key	Default	Extra
nip	varchar(12)	NO	PRI	NULL	
nama_depan	varchar(50)	NO		NULL	
nama_belakang	varchar(50)	NO		NULL	
tempat_lahir	varchar(50)	NO		NULL	
tanggal_lahir	date	NO		NULL	
nomor_telepon	varchar(16)	NO		NULL	
email	varchar(50)	YES		NULL	
domisili	varchar(50)	NO	MUL	NULL	
alamat	varchar(200)	YES		NULL	
angkatan	varchar(6)	NO		NULL	
grup	varchar(2)	NO		NULL	
created_at	timestamp	NO		current_timestamp()	on update current_...
update_at	timestamp	NO		0000-00-00 00:00:00	

- b. Update kolom “Angkatan” dengan nilai 6 digit pertama dari kolom “NIP”.
- ```
UPDATE peserta
SET angkatan = LEFT(nip, 6),
 update_at = NOW();
```
- c. Update kolom “Grup” dengan nilai digit ke-8 dan digit ke-9 dari kolom “NIP”.
- ```
UPDATE peserta
SET grup = MID(nip, 8, 2),
    update_at = NOW();
```

```
cek setelah update
SELECT nip,
       CONCAT(nama_depan, ' ', nama_belakang) as nama,
       angkatan,
       grup
FROM peserta;
```

nip	nama	angkatan	grup
Filter...	Filter...	Filter...	Filter...
ARN999-01001	azhar susanto	ARN999	01
ARN999-01002	ikram prasetya	ARN999	01
ARN999-01003	wahyu hendarto	ARN999	01
ARN999-01004	roni darmawan	ARN999	01
ARN999-01006	fadil irawan	ARN999	01
ARN999-01007	arfan fadly	ARN999	01
ARN999-01008	alfarezi dollo	ARN999	01
ARN999-01009	isa darmawan	ARN999	01
ARN999-01010	ihsan susanto	ARN999	01
ARN999-01011	azam sanusi	ARN999	01
ARN999-01012	omar rhoma	ARN999	01
ARN999-01013	dafa rojali	ARN999	01
ARN999-01014	imran susanto	ARN999	01
ARN999-01015	efendi rohadi	ARN999	01
ARN999-01016	sandi irwansyah	ARN999	01
ARN999-01017	salim hendratmo	ARN999	01
ARN999-01018	razzan saputra	ARN999	01
ARN999-01020	rudi rojali	ARN999	01
ARN999-02001	randi saputro	ARN999	02
ARN999-02002	hakeem sanusi	ARN999	02
ARN999-02003	arfan dollo	ARN999	02

3. Dengan menggunakan *subquery*, tampilkan 5 besar peserta di grup yang nilai rata-ratanya (average Nilai Peserta dari seluruh Peserta pada grup tersebut) paling tinggi. Info yang ditampilkan adalah NIP, Nama Peserta, Umur Peserta, Nilai Peserta, dan Predikat.
 - a. Nama Peserta adalah gabungan antara kolom “Nama Depan” dan “Nama Belakang”.
 - b. Gunakan salah satu *date function* untuk menghitung Umur Peserta (menggunakan data pada kolom “Tanggal Lahir” di tabel *Peserta*).
 - c. Nilai Peserta dihitung dengan cara sebagai berikut:
 - i. Setiap evaluasi yang sama akan dihitung rata-ratanya.
 - ii. Hasil dari rata-rata tersebut akan dikalikan dengan bobot masing-masing evaluasi.
 - iii. Lalu hasil dari poin di atas dijumlahkan.
 - iv. Lakukan pembulatan dengan 2 angka di belakang koma.
 - v. “Predikat” dilihat dari “Nilai Peserta” dengan kriteria sebagai berikut (petunjuk: gunakan *flow control function*):
 1. 0 - 50: Rasib
 2. 51-60: Maqbul
 3. 61-75: Jayyid
 4. 76-90: Jayyid Jiddan
 5. 91-99: Mumtaz
 6. 100: Mumtaz Murtafi

```

SELECT tp.nip,
       tp.nama_peserta,
       tp.umur_peserta,
       tna.nilai_akhir,
       (CASE
        WHEN tna.nilai_akhir = 100 THEN "Mumtaz Murtafi"
        WHEN tna.nilai_akhir < 100 && tna.nilai_akhir >=91
THEN "Mumtaz"
        WHEN tna.nilai_akhir < 91 && tna.nilai_akhir >=76 THEN
"Jayyid Jiddan"
        WHEN tna.nilai_akhir < 76 && tna.nilai_akhir >=61 THEN
"Jayyid"
        WHEN tna.nilai_akhir < 61 && tna.nilai_akhir >=51 THEN
"Maqbul"
        WHEN tna.nilai_akhir < 51 && tna.nilai_akhir >=0 THEN
"Rasib"
        END) AS "Predikat"
FROM (
  SELECT tn.nip,
         ROUND(SUM(tn.n_je)/100, 2) AS nilai_akhir
  FROM (
    SELECT nip,
           jenis_evaluasi,
           AVG(np.nilai)*re.bobot AS n_je
    FROM nilai_peserta AS np
    INNER JOIN ref_evaluasi AS re
    ON np.jenis_evaluasi = re.kode
    GROUP BY np.nip, np.jenis_evaluasi
  ) as tn
  GROUP BY tn.nip
) AS tna
INNER JOIN (
  SELECT nip,
         CONCAT(nama_depan, ' ', nama_belakang) AS
nama_peserta,
         YEAR(NOW()) - YEAR(tanggal_lahir) AS umur_peserta
  FROM peserta
) AS tp
ON tna.nip = tp.nip
ORDER BY tna.nilai_akhir DESC
LIMIT 5;

```

Cara Membaca query:

1. line 17 – 23 untuk menentukan rata-rata tiap jenis evaluasi lalu dikalikan dengan bobotnya dan dialiaskan dengan tn (table nilai)
2. line 14 – 26 untuk mengelompokkan tiap jenis evaluasi berdasarkan nip-nya menggunakan sub-query dan dialiaskan dengan tna (table nilai akhir)

3. line 28 – 31 untuk membuat tabel peserta baru, yang akan di-join kan dengan tna (table nilai akhir)
4. line 1 – 35 inner join tna (table nilai akhir) dengan tp (table peserta) untuk menghasilkan ketentuan yang diinginkan soal

hasil:

nip	nama_peserta	umur_peserta	nilai_akhir	Predikat
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
ARN999-05006	latif irwansyah	78	78.48	Jayyid Jiddan
ARN999-12003	darwish wahyudi	59	78.05	Jayyid Jiddan
ARN999-05009	farhan hendratmo	17	77.89	Jayyid Jiddan
ARN999-08008	raafiq bustomi	39	77.78	Jayyid Jiddan
ARN999-07013	chamali setiawan	70	77.71	Jayyid Jiddan