

Nama: Alvaro Austin

NPM: 2106752180

Kelas: A

Asumsi:

A. Default value semua attribute adalah Null. Sehingga total food dan note akan null apabila belum pernah dibuat.

B. Provinsi Yogyakarta adalah DI. Yogyakarta

B. PT. Gurmanas adalah Nama Restoran (RName) -> Konfirmasi dengan Kak Gitan

1. Tampilkan nama restoran dan cabangnya yang berlokasi di Provinsi Yogyakarta dengan rating > 4.5.

```
SELECT RESTAURANT.RName, RESTAURANT.RBranch
FROM RESTAURANT
WHERE RESTAURANT.Province = 'DI. Yogyakarta' AND
RESTAURANT.Rating > 4.5;
```

2. Tampilkan nama makanan apa saja yang dijual oleh restoran-restoran yang bekerja sama dengan PT. Gurmanas yang memiliki harga dengan rentang Rp 10.000,00 - Rp 50.000,00.

```
SELECT FOOD.FoodName
FROM FOOD
WHERE RName = 'PT. Gurmanas' AND (PRICE BETWEEN 10000 AND
50000);
```

3. Tampilkan semua nama makanan yang dijual oleh Restoran ABX yang mengandung bahan makanan kacang. **Petunjuk: Anda harus membuat nested query menggunakan keyword IN dengan benar.**

```
SELECT FOOD_INGREDIENT.FoodName
FROM FOOD_INGREDIENT
WHERE FOOD_INGREDIENT.RName IN
      (SELECT FOOD.RName FROM FOOD WHERE FOOD.RName = 'ABX')
AND
FOOD_INGREDIENT.Ingredient IN
      (SELECT Ingredient.Id FROM INGREDIENT WHERE
INGREDIENT.Name = 'Kacang');
```

Nama: Alvaro Austin

NPM: 2106752180

Kelas: A

4. Pelanggan yang membeli makanan memiliki opsi untuk menambahkan note/keterangan terkait pesanan yang dilakukan. Tampilkan nama makanan yang dipesan oleh pelanggan bernama Carissa Magnolia yang memiliki note atau keterangan tambahan.

Petunjuk: Anda harus membuat nested query menggunakan keyword EXISTS dengan benar

Asumsi: Apabila Carissa Magnolia mempunyai lebih dari 1 makanan yang sama dengan transaksi masing-masing maka makanan tersebut akan masuk pada query ini

```
SELECT TRANSACTION_FOOD.FoodName
FROM TRANSACTION_FOOD
WHERE EXISTS (SELECT * FROM USER WHERE USER.Email =
TRANSACTION_FOOD.Email AND USER.FName = 'Carissa' AND
USER.LName = 'Magnolia' AND Note IS NOT NULL);
```

5. Tampilkan nama dari kurir yang pernah mengantarkan makanan yang dipesan dari restoran "KFC" cabang "Margonda", tetapi belum pernah mengantarkan makanan dari restoran "KFC" cabang "Lenteng Agung".

```
(SELECT USER.FName, USER.LName
FROM USER, TRANSACTION_ACTOR, COURIER, TRANSACTION,
TRANSACTION_FOOD
WHERE USER.Email = TRANSACTION_ACTOR.Email AND
TRANSACTION_ACTOR.Email = COURIER.Email AND COURIER.Email =
TRANSACTION.CourierId AND TRANSACTION.Email =
TRANSACTION_FOOD.Email AND TRANSACTION.Datetime =
TRANSACTION_FOOD.Datetime AND TRANSACTION_FOOD.RName =
'KFC' AND TRANSACTION_FOOD.RBranch = 'Margonda')
```

EXCEPT

```
(SELECT USER.FName, USER.LName
FROM USER, TRANSACTION_ACTOR, COURIER, TRANSACTION,
TRANSACTION_FOOD
```

Nama: Alvaro Austin

NPM: 2106752180

Kelas: A

```
WHERE USER.Email = TRANSACTION_ACTOR.Email AND  
TRANSACTION_ACTOR.Email = COURIER.Email AND COURIER.Email =  
TRANSACTION.CourierId AND TRANSACTION.Email =  
TRANSACTION_FOOD.Email AND TRANSACTION.Datetime =  
TRANSACTION_FOOD.Datetime AND TRANSACTION_FOOD.RName =  
'KFC' AND TRANSACTION_FOOD.RBranch = 'Lenteng Agung');
```

6. Tampilkan nama dan email pelanggan yang belum pernah memesan makanan pada aplikasi SIREST.

```
(SELECT USER.FName, USER.LName, USER.Email  
FROM USER, TRANSACTION_ACTOR, CUSTOMER  
WHERE USER.Email = TRANSACTION_ACTOR.Email AND  
TRANSACTION_ACTOR.Email = CUSTOMER.Email) EXCEPT  
(SELECT USER.FName, USER.LName, USER.Email  
FROM USER, TRANSACTION_ACTOR, CUSTOMER, TRANSACTION,  
TRANSACTION_FOOD  
WHERE USER.Email = TRANSACTION_ACTOR.Email AND  
TRANSACTION_ACTOR.Email = CUSTOMER.Email AND CUSTOMER.Email  
= TRANSACTION.Email  
AND TRANSACTION.Email = TRANSACTION_FOOD.EMAIL AND  
TRANSACTION.Datetime = TRANSACTION_FOOD.Datetime);
```

Nama: Alvaro Austin

NPM: 2106752180

Kelas: A

7. Tampilkan nama dan email pelanggan yang sudah pernah menggunakan semua metode pembayaran yang disediakan oleh aplikasi SIREST.

```
SELECT USER.FName, USER.LName, USER.Email
FROM USER, TRANSACTION
WHERE TRANSACTION.Email = USER.Email
GROUP BY USER.FName, USER.LName, USER.Email
HAVING COUNT(PMId) IN (
SELECT COUNT(*)
FROM PAYMENT_METHOD);
```

8. Tampilkan nama dan harga makanan untuk makanan yang paling laku (memiliki jumlah pembelian item terbanyak) di restoran "KFC" cabang "Margonda".

```
SELECT FOOD.FoodName, FOOD.Price
FROM FOOD, TRANSACTION_FOOD
WHERE FOOD.RName = TRANSACTION_FOOD.Rname AND FOOD.RBranch
= TRANSACTION_FOOD.RBranch AND FOOD.RName = 'KFC' AND
FOOD.RBranch = 'Margonda' AND FOOD.FoodName =
TRANSACTION_FOOD.FoodName
GROUP BY FOOD.FoodName, FOOD.Price
ORDER BY SUM(TRANSACTION_FOOD.Amount) DESC LIMIT 1;
```

9. Untuk setiap restoran, tampilkan jumlah menu, harga minimum menu dan harga maksimum menu yang ditawarkan di restoran tersebut.

```
SELECT RESTAURANT.RName, RESTAURANT.RBranch, COUNT(*),
MIN(FOOD.Price), MAX(FOOD.Price)
FROM FOOD, RESTAURANT
WHERE FOOD.RName = RESTAURANT.RName AND FOOD.RBranch =
RESTAURANT.RBranch
GROUP BY RESTAURANT.RName, RESTAURANT.RBranch;
```

Nama: Alvaro Austin

NPM: 2106752180

Kelas: A

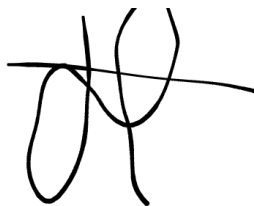
10. Tampilkan nama promo yang saat ini masih berlaku / digunakan oleh sekurang-kurangnya 10 restoran (perhatikan bahwa promo yang sudah lewat dianggap tidak valid / tidak diperhitungkan).

Petunjuk: Anda dapat menggunakan keyword `current_timestamp` untuk mendapatkan nilai timestamp saat ini.

```
SELECT PromoName
FROM PROMO, RESTAURANT_PROMO
WHERE (CURRENT_TIMESTAMP BETWEEN RESTAURANT_PROMO.starttime
AND RESTAURANT_PROMO.endtime) AND (PROMO.Id IN
(SELECT Pid
FROM RESTAURANT_PROMO
GROUP BY Pid
HAVING COUNT(Pid) >= 10
))
GROUP BY PromoName;
```

Code of conduct:

Saya menyatakan bahwa tugas ini saya kerjakan dengan usaha sendiri. Saya tidak menyalin jawaban dari sumber manapun. Saya bertanggung jawab menjaga agar jawaban tugas saya tidak disalin oleh peserta lainnya.



Alvaro Austin