PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

Praktikum 4

Database Security

Jumat, 25 Oktober 2024

Oleh:
13522053 - Erdianti Wiga Putri Andini
13522092 - Sa’ad Abdul Hakim

IF3140 - Sistem Basis Data

2024

Petunjuk

1. Kerjakan setiap soal praktikum ini dengan baik
2. Gunakan asumsi pada praktikum ini bahwa jumlah pegawai dapat ditambah sewaktu-waktu, sedangkan jumlah divisi sesuai pada soal saja
3. Nama database: **airport**
4. **Dilarang Menyontek**

AirportDB


passengerdetails
passenger_id
birthdate
sex
street
city
zip
country
emailaddress
telephoneno
passenger
passenger_id
passportno
firstname
lastname
airport_geo
airport_id
name
city
country
latitude
longitude
geolocation
booking
booking_id
flight_id
seat
passenger_id
price
flight
flight_id
flightno
from
to
departure
arrival
airline_id
airplane_id
airplane
airplane_id
capacity
type_id
airline_id
airplane_type
type_id
identifier
description
airport
airport_id
iata
icao
name
employee
employee_id
firstname
lastname
birthdate
sex
street
city
zip
country
emailaddress
telephoneno
salary
department
username
password
airline
airline_id
iata
airlinename
base_airport
weatherdata
log_date
time
station
temp
humidity
airpressure
wind
weather
winddirection
flightschedule
flightno
from
to
departure
arrival
airline_id
monday
tuesday
wednesday
thursday
friday
saturday
sunday

Kena Audit

Pak Ardneham (memiliki akun labdas), sebagai orang yang sangat sibuk, mengelola basis data AirportDB tanpa membuat restriksi-restriksi. Setiap kebutuhan akses ke AirportDB oleh rekan kerja atau atasannya akan menggunakan kredensial miliknya sebagai superadmin. Akan tetapi, karena proses audit, maka Pak Ardneham harus memastikan bahwa akses ke AirportDB harus sesuai dengan kebutuhan dari setiap pengguna serta semua orang yang akan menggunakan harus login dengan kredensial sendiri.

Setelah melalui proses membuat dokumen kebutuhan, maka didapatkan kebutuhan akses sebagai berikut:

a. Pak Tius, seorang pegawai divisi Airport Manager, memiliki pekerjaan untuk memasukkan informasi airport dan airline baru yang terbentuk di seluruh dunia, serta mengubah informasinya jika ada yang harus diperbarui
b. Bu Didot, manajer dari departemen Marketing, membutuhkan akses ke basis data AirportDB. Diketahui bahwa setiap manajer hanya boleh mendapatkan informasi employee dari departemen yang sama. Informasi yang diperbolehkan pun hanya nama lengkap pegawai, alamat email, dan gaji.
c. Admin InSystems, Pak Cello, diberikan hak untuk membuat booking dan memperbarui informasi booking
d. Divisi Ticketing yang dipimpin oleh Pak Jay memerlukan informasi passenger serta bookingnya. Divisi ini memiliki tugas untuk melakukan booking dan mengubah informasi passenger hanya pada informasi email dan nomor telepon saja berdasarkan informasi ID dari setiap passenger. Pak Jay mendelegasikan hak akses ini kepada Bu Rifi yang sedang dalam masa probation sehingga aksesnya hanya 6 bulan saja
e. Pak Julala, ketua dari divisi air traffic control, membutuhkan informasi flight dan untuk setiap flight, jenis airplane-nya. Selain mendapatkan informasi, divisi yang dipimpin Pak Julala juga memiliki kewenangan untuk mengubah data flight apabila terjadi kendala mendadak. Maka dari itu, divisi ini memiliki akses untuk seluruh data flight. Di sisi lain, pihak para maskapai meminta pihak bandara untuk membatasi pemberian data airplane. Dari data airplane dan typenya, data yang boleh diberikan hanyalah id dari pesawat, kapasitas dari pesawat, dan identifier dari jenis pesawat itu sendiri.
f. Madame Marie Antoinette merupakan seorang ahli cuaca. Seorang ahli cuaca hanya memiliki tugas mengisi informasi weather serta menghapusnya jika ada yang salah. Ia tidak diperbolehkan untuk mengubah data yang telah dimasukkan. Madame Antoinette hanya berwenang pada wilayah Eropa, yaitu station 1. Oleh karena itu, pastikan juga Madame Antoinette hanya dapat membaca data cuaca di wilayah terkait saja.

Berdasarkan spesifikasi kebutuhan ini, sebagai anak magang Pak Ardneham, bantulah untuk mengimplementasikan seluruh kebutuhan ini pada AirportDB. Selain itu, isilah matriks akses yang terlampir pada dokumen ini. Anda diperbolehkan untuk membuat *view* jika membantu pekerjaan Anda!

Matriks Akses Tabel AirportDB

Role/Tabel	passengerdetails	passenger	booking	flight	airplane	flightschedule	airplane_type	airline	airport	airport_geo	employee	weatherdata
Airport Manager								W, U	W, U			
Manajer Marketing												
Admin InSystems			W, U									
Divisi Ticketing	R, U	R	R, W									
AirTraffic Control				R, U								
WeatherExpert												W, D

Note: Isi dengan hak akses yang sesuai untuk setiap divisi/orang dengan tabel yang ada. Tetap kosongkan *cell* jika suatu entitas tidak memiliki hak akses terhadap suatu objek

Daftar Hak Akses: Akses Read (R), Akses Write (W), Akses Update (U), Akses Delete (D)

Matriks Akses View AirportDB (jika ada)

Role	View employee_department	View view_airplane	View view_station1
Airport Manager			
Manajer Marketing	R		
Admin ERP InSystems			
Divisi Ticketing			
AirTrafficControl		R	
WeatherExpert			R

Note: Isi dengan hak akses yang sesuai untuk setiap divisi/orang dengan tabel yang ada. Tetap kosongkan *cell* jika suatu entitas tidak memiliki hak akses terhadap suatu objek. Tambahkan kolom sesuai kebutuhan.

Daftar Hak Akses: Akses Read (R), Akses Write (W), Akses Update (U), Akses Delete (D)

Implementasi View (jika ada)

1. employee_department

Query Definisi	`CREATE VIEW employee_department`
	`AS SELECT CONCAT(firstname, ' ',lastname) as fullname, emailaddress, salary`
	`FROM employee`
	`WHERE department = 'Marketing';`

Query Eksekusi (Limit 2)	`SELECT * FROM employee_department LIMIT 2;`

SS Eksekusi Query Definisi

```
airport=# CREATE VIEW employee_department
AS SELECT CONCAT(firstname, ' ',lastname) as fullname, emailaddress, salary
FROM employee
WHERE department = 'Marketing';
CREATE VIEW
```

SS Eksekusi Query Eksekusi

```
airport=# SELECT * FROM employee_department LIMIT 2;
    fullname    |         emailaddress          | salary
----------------+-------------------------------+---------
 Greg Harris    | Ben.Huff@mysmart.mymobile.ph | 1851.54
 Rodney Dillard | Alice.Harris@vmobl.com       | 1795.10
(2 rows)
```

2. view_airplane

Query Definisi	`CREATE VIEW view_airplane AS`
	`SELECT a.airplane_id, a.capacity, t.identifier`
	`FROM airplane a JOIN airplane_type t ON a.type_id = t.type_id`
	`JOIN flight f ON a.airplane_id = f.airplane_id;`

Query Eksekusi (Limit 2)	`SELECT * FROM view_airplane LIMIT 2;`

SS Eksekusi Query Definisi

```
airport=# CREATE VIEW view_airplane AS
SELECT a.airplane_id, a.capacity, t.identifier
FROM airplane a JOIN airplane_type t ON a.type_id = t.type_id
JOIN flight f ON a.airplane_id = f.airplane_id;
CREATE VIEW
```

SS Eksekusi Query Eksekusi

```
airport=# SELECT * FROM view_airplane LIMIT 2;
 airplane_id | capacity |     identifier
-------------+----------+---------------------
           1 |      150 | Airbus-A320-Familie
           1 |      150 | Airbus-A320-Familie
(2 rows)
```

3. view_station1

Query Definisi	`CREATE VIEW view_station1 AS` `SELECT *`

	FROM weatherdata WHERE station = 1;
Query Eksekusi (Limit 2)	SELECT * FROM view_station1 LIMIT 2;

SS Eksekusi Query Definisi

```
airport=# CREATE VIEW view_station1 AS
SELECT *
FROM weatherdata
WHERE station = 1;
CREATE VIEW
```

SS Eksekusi Query Eksekusi

```
airport=# SELECT * FROM view_station1 LIMIT 2;
  log_date  |   time   | station | temp | humidity | airpressure | wind  | weather | winddirection
------------+----------+---------+------+----------+-------------+-------+---------+---------------
 2015-12-01 | 00:00:00 |       1 |  1.0 |    100.0 |     1034.00 | 13.00 | Regen   |            83
 2015-12-01 | 00:05:00 |       1 |  1.0 |     99.0 |     1030.00 | 10.00 | Regen   |            80
(2 rows)
```

Note: Tambahkan sesuai kebutuhan

Daftar User

Jika pada soal disebutkan ada pegawai yang membutuhkan akses secara langsung, sertakan nama pegawai berikut kredensial dan role mereka pada AirportDB!

Pegawai	Username	Password	Role
Pak Tius	tius12	tius123	airport_manager
Bu Didot	didot12	didot123	marketing_manager
Pak Cello	cello12	cello123	admin_in_systems
Pak Jay	jay12	jay123	head_ticketing
Bu Rifi	rifi12	rifi123	trainee_ticketing
Pak Julala	julala12	julala123	head_atc
Madame Marie Antoinette	mma12	mma123	weather_expert

Note: Tambahkan baris sesuai kebutuhan

Query Implementasi Access Control

Tuliskan *query* pemberian akses!

Query untuk Pembentukan Role Airport Manager dan Pegawai yang Membutuhkan	`CREATE USER tius12 WITH PASSWORD 'tius123';`
	`CREATE ROLE airport_manager;`
	`GRANT INSERT, UPDATE ON airport, airline TO airport_manager;`
	`GRANT airport_manager TO tius12;`

Query untuk Pembentukan Role Manajer Marketing dan Pegawai yang membutuhkan	`CREATE USER didot12 WITH PASSWORD 'didot123';`
	`CREATE ROLE marketing_manager;`
	`GRANT SELECT ON employee_department TO marketing_manager;`
	`GRANT marketing_manager TO didot12;`

Query untuk Pembentukan Role Admin InSystems dan Pegawai yang membutuhkan	`CREATE USER cello12 WITH PASSWORD 'cello123';`
	`CREATE ROLE admin_in_systems;`
	`GRANT INSERT, UPDATE ON booking TO admin_in_systems;`
	`GRANT admin_in_systems TO cello12;`

Query untuk Pembentukan Role Divisi Ticketing dan Pegawai yang Membutuhkan	`CREATE USER rifi12 WITH PASSWORD 'rifi123';`
	`CREATE ROLE trainee_ticketing VALID UNTIL '2025-04-25';`
	`GRANT SELECT ON booking, passenger, passengerdetails TO trainee_ticketing;`
	`GRANT INSERT ON booking TO trainee_ticketing;`
	`GRANT UPDATE emailaddress, telephoneno ON passengerdetails TO trainee_ticketing;`
	`GRANT trainee_ticketing TO rifi12;`

Query untuk Pembentukan Role Air Traffic Control dan Pegawai yang Membutuhkan	`CREATE USER julala12 WITH PASSWORD 'julala123';`
	`CREATE ROLE head_atc;`
	`GRANT SELECT, UPDATE ON flight TO head_atc;`
	`GRANT SELECT ON view_airplane TO head_atc;`
	`GRANT head_atc TO julala12;`

Query untuk Pembentukan Role Weather Expert dan Pegawai yang Membutuhkan	`CREATE USER mma12 WITH PASSWORD 'mma123';`
	`CREATE ROLE weather_expert;`
	`GRANT INSERT, DELETE ON weatherdata TO weather_expert;`
	`GRANT SELECT ON view_station1 TO weather_expert;`
	`GRANT weather_expert TO mma12;`

Screenshot semua role yang terbentuk! Query untuk mendapatkan semua role adalah sebagai berikut (silakan dimodifikasi untuk mempermudah screenshot)

```
SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'probation_ticketing'
```

```
ORDER BY grantee, table_name;
```

Screenshot semua Role

```
airport=# SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'airport_manager'
ORDER BY grantee, table_name;
    role_name    | table_name | privilege_type
-----------------+------------+----------------
 airport_manager | airline    | INSERT
 airport_manager | airline    | UPDATE
 airport_manager | airport    | INSERT
 airport_manager | airport    | UPDATE
(4 rows)
```

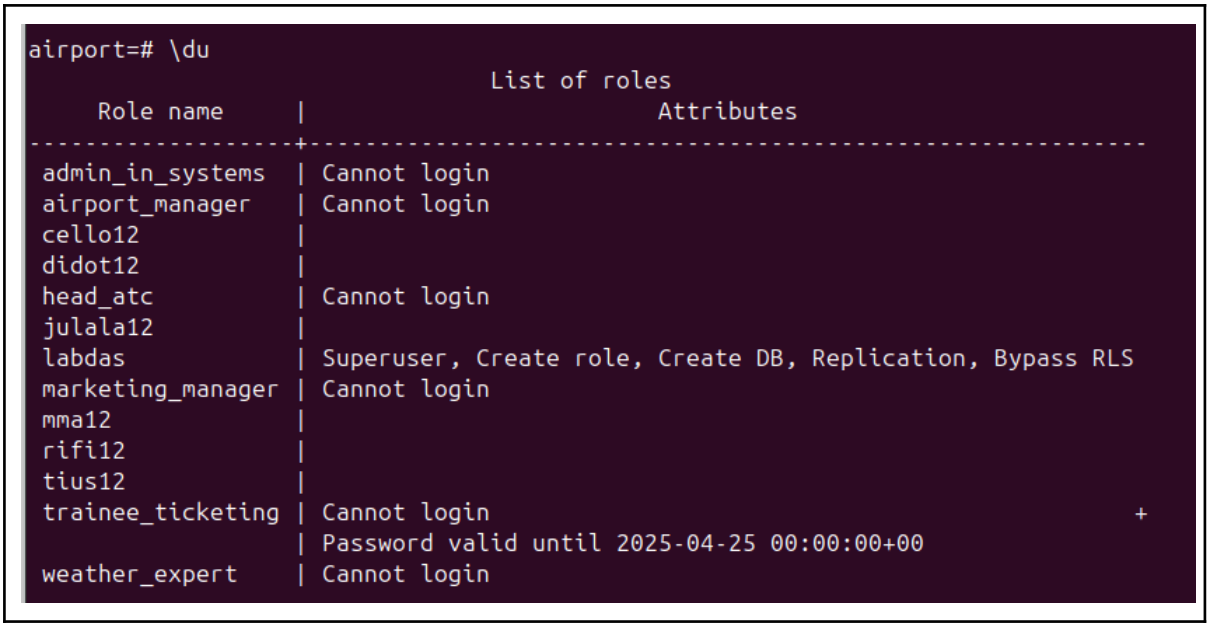
```
airport=# SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'marketing_manager'
ORDER BY grantee, table_name;
     role_name     |     table_name      | privilege_type
-------------------+---------------------+----------------
 marketing_manager | employee_department | SELECT
(1 row)
```

```
airport=# SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'admin_in_systems'
ORDER BY grantee, table_name;
    role_name     | table_name | privilege_type
------------------+------------+----------------
 admin_in_systems | booking    | INSERT
 admin_in_systems | booking    | UPDATE
(2 rows)
```

```
airport=# SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'trainee_ticketing'
ORDER BY grantee, table_name;
     role_name     |    table_name    | privilege_type
-------------------+------------------+----------------
 trainee_ticketing | booking          | INSERT
 trainee_ticketing | booking          | SELECT
 trainee_ticketing | passenger        | SELECT
 trainee_ticketing | passengerdetails | SELECT
(4 rows)
```

```
airport=# SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'head_atc'
ORDER BY grantee, table_name;
 role_name |  table_name   | privilege_type
-----------+---------------+----------------
 head_atc  | flight        | SELECT
 head_atc  | flight        | UPDATE
 head_atc  | view_airplane | SELECT
(3 rows)
```

```
airport=# SELECT grantee AS role_name,
       table_name,
       privilege_type
FROM information_schema.table_privileges
WHERE grantee LIKE 'weather_expert'
ORDER BY grantee, table_name;
   role_name    |  table_name   | privilege_type
----------------+---------------+----------------
 weather_expert | view_station1 | SELECT
 weather_expert | weatherdata   | INSERT
 weather_expert | weatherdata   | DELETE
(3 rows)
```



```
airport=# \du
                                   List of roles
     Role name     |                         Attributes
-------------------+------------------------------------------------------------
 admin_in_systems  | Cannot login
 airport_manager   | Cannot login
 cello12           |
 didot12           |
 head_atc          | Cannot login
 julala12          |
 labdas            | Superuser, Create role, Create DB, Replication, Bypass RLS
 marketing_manager | Cannot login
 mma12             |
 rifi12            |
 tius12            |
 trainee_ticketing | Cannot login                                               +
                   | Password valid until 2025-04-25 00:00:00+00
 weather_expert    | Cannot login
```

[BONUS] Madame Antoinette yang Baik Hati

Untuk soal (f), implementasikan sebuah *trigger* yang melakukan verifikasi data yang masuk pada pemasukan dan penghapusan data yang dilakukan oleh Madame Antoinette.

HINT : untuk mengetahui user yang aktif saat ini dapat menggunakan global variable current_user yang mengembalikan nama dari user saat ini. Contoh penggunaan:

```
SELECT current_user
SELECT * FROM user_data WHERE username = current_user;
```

Pada *trigger* ini, lakukan verifikasi apakah *user* saat ini adalah Madame Antoinette.

- Jika benar, pastikan bahwa data yang ingin dimasukkan/dihapus Madame Antoinette adalah benar data yang terkait dengan station 1.
    - Jika benar, maka lanjutkan operasi manipulasi.
    - Jika tidak, tampilkan pesan error.
- Jika tidak, lanjutkan operasi manipulasi

Query Definisi Trigger Insert	CREATE TRIGGER verif_station
	BEFORE INSERT ON weatherdata
	FOR EACH ROW
	WHEN (current_user = ‘mma12’)
	EXECUTE FUNCTION check_station();
	
	CREATE FUNCTION check_station() RETURNS TRIGGER
	AS $$
	BEGIN
	IF NEW.station = 1 THEN RETURN NEW;
	ELSE
	RAISE EXCEPTION ‘Station tidak dapat diakses oleh Madam Antionette’;
	END IF;
	END;
	$$ LANGUAGE plpgsql;

Query *Testing 1*	INSERT INTO weatherdata VALUES (‘2020-10-03’, ‘08:09:00’, 1, 1.0, 99.00, 1020.00, 10.00, ‘Regen’, 80);
Query *Testing 2*	INSERT INTO weatherdata VALUES (‘2020-10-03’, ‘08:11:00’, 2, 1.0, 99.00, 1020.00, 10.00, ‘Regen’, 80);
Query Eksekusi Definisi Trigger	

```
airport=# CREATE FUNCTION check_station() RETURNS TRIGGER AS $$
BEGIN
IF NEW.station = 1 THEN RETURN NEW;
ELSE
RAISE EXCEPTION 'Station tidak dapat diakses oleh Madam Antionette';
END IF;
END;
$$ LANGUAGE plpgsql;
CREATE FUNCTION
```

```
airport=# CREATE TRIGGER verif_station
BEFORE INSERT ON weatherdata
FOR EACH ROW
WHEN (current_user = 'mma12')
EXECUTE FUNCTION check_station();
CREATE TRIGGER
```

Query Eksekusi Query Testing 1

User julala12

```
airport=> INSERT INTO weatherdata VALUES ('2020-10-03', '08:09:00', 1, 1.0, 99.00, 1020.00, 10.00, 'Regen', 80);
ERROR:  permission denied for table weatherdata
```

User mma12

```
airport=> INSERT INTO weatherdata VALUES ('2020-10-03', '08:09:00', 1, 1.0, 99.00, 1020.00, 10.00, 'Regen', 80);
INSERT 0 1
```

Query Eksekusi Query Testing 2

User julala12

```
airport=> INSERT INTO weatherdata VALUES ('2020-10-03', '08:11:00', 2, 1.0, 99.00, 1020.00, 10.00, 'Regen', 80);
ERROR:  permission denied for table weatherdata
```

User mma12

```
airport=> INSERT INTO weatherdata VALUES ('2020-10-03', '08:11:00', 2, 1.0, 99.00, 1020.00, 10.00, 'Regen', 80);
ERROR:  Station tidak dapat diakses oleh Madam Antionette
CONTEXT:  PL/pgSQL function check_station() line 5 at RAISE
```

Query Definisi Trigger Delete	CREATE TRIGGER verif_station
	BEFORE DELETE ON weatherdata
	FOR EACH ROW
	WHEN (current_user = ‘mma12’)
	EXECUTE FUNCTION check_station();
	
	CREATE FUNCTION check_station() RETURNS TRIGGER
	AS $$
	BEGIN
	IF NEW.station = 1 THEN RETURN NEW;
	ELSE
	RAISE EXCEPTION ‘Station tidak dapat diakses oleh Madam Antionette’;
	END IF;

	END; $$ LANGUAGE plpgsql;
Query *Testing 1*	SELECT * FROM weatherdata WHERE humidity = 99.0 AND station = 1; DELETE FROM weatherdata WHERE humidity = 99.0 AND station = 1;
Query *Testing 2*	SELECT * FROM weatherdata WHERE humidity = 99.0 AND station = 2; DELETE FROM weatherdata WHERE humidity = 99.0 AND station = 2;
Query Eksekusi Definisi Trigger	
	
Query Eksekusi Query Testing 1	
User julala12 User mma12	
Query Eksekusi Query Testing 2	
User julala12 User mma12	

Pembagian Kerja

NIM	Tugas
13522053	Semua
13522092	Semua