17장. Data Redaction

Data Redaction은 특정 사용자에게 표시되는 데이터를 동적으로 변경하여 민감한 데이터 노출을 제한합니다. CREATE REDACTION POLICY 명령을 사용하여 Data Redaction 정책을 등록 합니다.
이 명령은 정책이 적용되는 테이블, 열, 사용자를 결정 및 기타 옵션을 지정합니다.
SuperUser와 테이블 소유자는 데이터 편집을 건너뛰고 원본 데이터를 볼수 있습니다.
다른 모든 사용자는 수정 정책이 적용되어 제한된 데이터를 볼 수 있습니다.

17.1 CREATE REDACTION POLICY

CREATE REDACTION POLICY는 테이블에 대한 새로운 데이터 수정 정책을 정의합니다.

17.1.1 Syntax

```
CREATE REDACTION POLICY <name> ON <table_name>
  [ FOR ( <expression> ) ]
  [ ADD [ COLUMN ] <column_name> USING <funcname_clause> ]
```

CREATE REDACTION POLICY 명령은 열 데이터를 교정하여 테이블에 대한 Redaction 보안 정책을 정의합니다. 새로 생성된 Data Redaction 정책은 기본적으로
활성화됩니다.
FOR : 대상 User | Role 조건식.
ADD COLUMN : 대상 COLUMN
USING : Redaction 함수

17.1.2 Parameters

- name : Data Redaction Policy명
- table_name : Data Redaction Table명
- expression: 제한 User 정책 조건
- column_name : Data Redaction Column명
- funcname_clause : Data Redaction Function명

17.1.3 Examples

1) Data Redaction 사전 준비 (Table, Data, User, Function)

```
# employees Table 생성
CREATE TABLE employees (
 id         integer GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
 name       varchar(40) NOT NULL,
 ssn        varchar(11) NOT NULL,
 phone      varchar(10),
 birthday   date,
 salary     money,
 email      varchar(100)
);

# Data Insert
INSERT INTO employees (name, ssn, phone, birthday, salary, email)
VALUES
( 'Sally Sample', '020-78-9345', '5081234567', '1961-02-02', 51234.34,
'sally.sample@enterprisag.com'),
( 'Jane Doe', '123-33-9345', '6171234567', '1963-02-14', 62500.00,
'jane.doe@gmail.com'),
( 'Bill Foo', '123-89-9345', '9781234567','1963-02-14', 45350,
'william.foe@hotmail.com');

# User 생성 및 권한 부여 (hr, scott)
CREATE USER hr  with password '1234';
CREATE USER scott with password '1234';
GRANT ALL ON employees TO hr, scott;

# SSN Column Redaction 함수 생성
CREATE OR REPLACE FUNCTION redact_ssn (ssn varchar(11))
RETURNS varchar(11) AS $$
BEGIN
  /* replaces 020-12-9876 with xxx-xx-9876 */
  return overlay (ssn placing 'xxx-xx' from 1);
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION redact_ssn_new (ssn varchar(11))
RETURNS varchar(11) AS $$
BEGIN
  /* replaces 020-12-9876 with ***-**-9876 */
  return overlay (ssn placing '***-**' from 1);
END;
$$ LANGUAGE plpgsql;

# Salary Column Redaction 함수 생성
CREATE OR REPLACE FUNCTION redact_salary ()
RETURNS money AS $$
```

```
BEGIN
return 0::money;
END;
$$ LANGUAGE plpgsql;
```

2) Data Redaction POLICY 생성

```
CREATE REDACTION POLICY redact_policy_personal_info ON employees FOR (session_user !=
'hr')
ADD COLUMN ssn USING redact_ssn(ssn),
ADD COLUMN salary USING redact_salary();
```

- Table : employees
- 제한 조건 : hr User 제외
- Redaction Column : SSN, SALARY

3) hr User를 이용하여 employees Table 조회

```
# hr user 접속
\c postgres hr

# employees Table 조회
SELECT * FROM EMPLOYEES;
 id |     name     |     ssn     |   phone    | birthday  | salary |          email
----+--------------+-------------+------------+-----------+----------+-----------------------------
  1 | Sally Sample | 020-78-9345 | 5081234567 | 1961-02-02 | ₩51,234 |
sally.sample@enterprisag.com
  2 | Jane Doe     | 123-33-9345 | 6171234567 | 1963-02-14 | ₩62,500 | jane.doe@gmail.com
  3 | Bill Foo     | 123-89-9345 | 9781234567 | 1963-02-14 | ₩45,350 |
william.foe@hotmail.com
(3 rows)
```

4) scott User를 이용하여 employees Table 조회

```
# scott user 접속
\c postgres scott

# employees Table 조회
SELECT * FROM EMPLOYEES;
 id |     name     |     ssn     |   phone    | birthday  | salary |          email
----+--------------+-------------+------------+-----------+--------+-----------------------------
  1 | Sally Sample | xxx-xx-9345 | 5081234567 | 1961-02-02 |  ₩0| sally.sample@enterprisag.com
```

```
 2 | Jane Doe     | xxx-xx-9345 | 6171234567 | 1963-02-14 |    ₩0 | jane.doe@gmail.com
 3 | Bill Foo     | xxx-xx-9345 | 9781234567 | 1963-02-14 |    ₩0 | william.foe@hotmail.com
(3 rows)
```

17.2 ALTER REDACTION POLICY

ALTER REDACTION POLICY는 테이블에 대한 Data Redaction Policy를 변경합니다.

17.2.1 Syntax

```
ALTER REDACTION POLICY <name> ON <table_name> RENAME TO <new_name>

ALTER REDACTION POLICY <name> ON <table_name> FOR ( <expression> )

ALTER REDACTION POLICY <name> ON <table_name> { ENABLE | DISABLE}

ALTER REDACTION POLICY <name> ON <table_name>
  ADD [ COLUMN ] <column_name> USING <funcname_clause>

ALTER REDACTION POLICY <name> ON <table_name>
  MODIFY [ COLUMN ] <column_name> USING <funcname_clause>

ALTER REDACTION POLICY <name> ON <table_name>
  DROP [ COLUMN ] <column_name>
```

FOR : 대상 User | Role 조건식을 변경합니다.
ENABLE : Redaction 정책을 활성화 합니다.
DISABLE : Redaction 정책을 비활성화합니다.
ADD COLUMN : COLUMN에 Redaction 기능을 추가 합니다.
MODIFY COLUMN : COLUMN의 Redaction 기능을 변경 합니다.
DROP COLUMN : COLUMN을 Redaction기능을 제외 합니다.

17.2.2 Parameters

- name : Data Redaction Policy명
- table_name : Data Redaction Table명
- new name : 변경할 새로운 Data Redaction Policy명
- expression: 제한 User 정책 조건
- column_name : Data Redaction Column명
- funcname_clause : Data Redaction Function명

17.2.3 Examples

1) 제한 User 조건을 수정 합니다. (hr, manager 정책 해제)

```
ALTER REDACTION POLICY redact_policy_personal_info ON employees
FOR (session_user != 'hr' AND session_user != 'manager');
```

2) ssn에 대한 Redaction Function을 redact_ssn_new로 변경합니다.

```
ALTER REDACTION POLICY redact_policy_personal_info ON employees
MODIFY COLUMN ssn USING redact_ssn_new(ssn);
```

17.3 DROP REDACTION POLICY

DROP REDACTION POLICY는 테이블에서 Redaction 정책을 제거합니다.

17.3.1 Syntax

```
DROP REDACTION POLICY <name> ON <table_name>
```

17.3.2 Parameters

- name : Data Redaction Policy명
- table_name : Data Redaction Table명

17.3.3 Examples

1) employees 테이블에서 redact_policy_personal_info 라는 데이터 수정 정책을 삭제 합니다.

```
DROP REDACTION POLICY redact_policy_personal_info ON employees;
```