

PROC SQL SAS

Outline dalam praktikum ini sebagai berikut:

1. Import Data dan Create Tabel
 2. Perintah Select
 3. Filtering Data
 4. Grouping dan Sorting Data
 5. Perintah Update dan Delete
-

Dataset

Dataset yang digunakan dalam praktikum ini adalah data chinook.db yang dapat didownload pada link berikut: <https://cdn.sqlitetutorial.net/wp-content/uploads/2018/03/chinook.zip>. Untuk menggunakan data tersebut, silakan ekstrak dari file zip kemudian simpan masing – masing table ke dalam file berekstensi .csv.

Dataset memuat sebanyak 11 data/ tabel dengan deskripsi masing-masing tabel sebagai berikut:

- employees: menyimpan data employees seperti employee id, last name, first name, address, dll.
- customers: menyimpan data customer
- Artists: menyimpan data artist yaitu artist id dan name
- Albums: menyimpan data list of tracks. Setiap album milik satu artist, tetapi satu artist mungkin memiliki beberapa album
- Invoices dan invoice_items: Tabel invoices menyimpan data invoice header sedangkan table invoice_items menyimpan data invoice line items
- Media_types: menyimpan tipe media seperti MPED audio dan AAC audio files
- Genres: menyimpan jenis music misalnya jazz, rock, metal, dll
- Tracks: menyimpan data lagu, setiap lagu masuk dalam satu album
- Playlists dan playlist_track: table playlists menyimpan data terkait playlist. Setiap playlist memuat suatu track list. Masing – masing track bisa masuk kedalam beberapa playlist. Tabel playlist_track digunakan untuk mengakomodasi relasi antara table playlists dan table tracks

Import Data dan Create Tabel

Data hasil ekstraksi yang telah disimpan dalam ekstensi .csv upload ke dalam penyimpanan Files (Home) SAS Studio. Selanjutnya import dataset ke dalam SAS system dengan PROC IMPORT.

```
PROC IMPORT DATAFILE='/home/u49735076/albums.csv'  
  DBMS=CSV  
  OUT=WORK.albums  
  REPLACE;  
  GETNAMES=YES;  
RUN;  
  
PROC IMPORT DATAFILE='/home/u49735076/artists.csv'  
  DBMS=CSV  
  OUT=WORK.artists  
  REPLACE;  
  GETNAMES=YES;  
RUN;  
  
PROC IMPORT DATAFILE='/home/u49735076/customers.csv'  
  DBMS=CSV  
  OUT=WORK.customers  
  REPLACE;  
  GETNAMES=YES;  
RUN;
```

Lokasi dan nama file dalam pernyataan Datafile disesuaikan dengan lokasi dan nama file pada masing – masing pengguna. DBMS=CSV menyatakan data source identifier, OUT menyatakan nama gugus data SAS, REPLACE berfungsi untuk mengganti data yang sudah ada, GETNAMES untuk mendapatkan variabel dalam data. Selanjutnya untuk membangun table dari gugus data SAS menggunakan perintah CREATE TABEL yang memiliki bentuk umum sebagai berikut:

```
PROC SQL;  
  
CREATE TABLE table_name AS  
  
SELECT * FROM gugusdataSAS;  
  
QUIT;
```

Gugus data SAS hasil import di atas dapat dibuat table SQL dengan syntax berikut:

```
PROC SQL;  
CREATE TABLE Album AS  
SELECT *  
FROM WORK.albums;  
QUIT;  
  
PROC SQL;  
CREATE TABLE Artist AS  
SELECT *  
FROM WORK.artists;  
QUIT;  
  
PROC SQL;  
CREATE TABLE Customer AS  
SELECT *  
FROM WORK.customers;  
QUIT;
```

Untuk membuat table dalam SAS dapat dilakukan juga menggunakan syntax SQL seperti pada umumnya, yaitu melalui pendefinisian struktur table seperti berikut:

```
PROC SQL;
CREATE TABLE customers
(CustomerId Integer PRIMARY KEY,
 FirstName CHAR (40),
 LastName CHAR (20),
 Company CHAR (80),
 Address CHAR (70),
 City CHAR (40),
 State CHAR (40),
 Country CHAR (40),
 PostalCode CHAR (10),
 Phone CHAR (24),
 Fax CHAR (24),
 Email CHAR (60),
 SupportRepId INTEGER
);
QUIT;

PROC SQL;
CREATE TABLE Media
(MediaTypeId INTEGER NOT NULL, Name CHAR (120)
);
QUIT;
```

Perintah di atas menghasilkan table kosong tanpa record. Untuk menambahkan record dapat menggunakan perintah INSERT INTO.

```
PROC SQL;
INSERT INTO Media
VALUES (1,'MPEG audio file')
VALUES (2,'Protected AAC audio file')
VALUES (3,'Protected MPEG-4 video file')
VALUES (4,NULL)
VALUES (5,'AAC audio file');
QUIT;

PROC PRINT DATA=Media; run;

proc sql;
insert into customer (customerid, firstname, lastname, supportrepid)
values (58,'Muhammad','Ali',4)
values (59,'Akbar','Maulana',1);
quit;

proc print data=customer; run;
```

Perintah SELECT

Perintah SELECT berfungsi untuk mengambil keseluruhan atau sebagian (subsetting) data. Bentuk umum perintah SELECT adalah:

```
PROC SQL;  
SELECT columns  
FROM tables  
WHERE expression  
GROUP BY columns  
HAVING expression  
ORDER BY columns  
QUIT;
```

Jika ingin mengambil semua kolom yang ada dalam table maka argument *columns* setelah SELECT dapat diisi dengan tanda bintang (*). Dalam PROC SQL penggunaan perintah SELECT harus mengikuti urutan bentuk umum di atas.

Contoh penggunaan perintah SELECT:

- Mengambil semua kolom dari suatu tabel

Misalkan ingin mengambil data semua kolom dari table album. Syntax yang digunakan adalah:

```
proc sql outobs=15;  
select *  
from album;  
quit;
```

Argument outobs=15 berfungsi untuk membatasi jumlah data yang diambil yaitu hanya 15 data di urutan atas.

- Mengambil sebagian kolom dari suatu table dan mengubah output kolom dengan nama baru

Misalkan ingin mengambil kolom country, kolom firstname dan kolom lastname kemudian kedua kolom digabung menjadi satu dengan nama name. Syntax yang digunakan adalah:

```
proc sql;  
select firstname || ' ' || lastname as name, country  
from customer;  
quit;
```

Tanda || ' ' berfungsi untuk menggabungkan dua kolom dan menampilkannya dalam satu kolom. Argumen AS berfungsi untuk mengganti output nama kolom menjadi nama yang lain.

- Menampilkan kolom baru dari kolom yang sudah ada dan menggunakan kolom baru untuk membuat kolom baru lainnya

Misalkan ingin mengambil kolom invoiceid dan kolom baru bernama newunitprice yang merupakan hasil operasi unitprice dikali 100. Selanjutnya kolom newunitprice digunakan untuk menampilkan kolom price yang merupakan hasil perkalian kolom newunitprice dan kolom quantity. Syntax yang digunakan adalah:

```
proc sql;  
select invoiceid, (unitprice*100) as newunitprice,  
calculated newunitprice*quantity as price  
from Invoice Item;  
quit;
```

Perintah CALCULATED berfungsi untuk mengambil kolom/ variabel baru yang merupakan hasil operasi dari variabel yang sudah ada dalam table.

- Mengambil kolom dari suatu table dan membuat kolom baru yang nilainya berasal dari *conditional clause*

Misalkan ingin mengambil kolom billingcity dari table invoice dan membuat satu kolom baru bernama NewTotal yang nilainya 'High' jika data dalam kolom Total lebih dari sama dengan 10, selain itu kolom NewTotal bernilai 'Low'. Syntax yang digunakan adalah:

```
proc sql outobs=20;  
select billingcity,  
case  
when Total GE 10 THEN 'High'  
else 'Low'  
end as NewTotal  
from invoice;  
quit;
```

Ketika menggunakan perintah CASE diakhir *conditional clause* harus ditambah END.

Filtering Data

Filtering merupakan perintah untuk mengambil sebagian data yang memenuhi kriteria tertentu. Filtering data dapat dilakukan menggunakan klausa WHERE setelah perintah SELECT – FROM.

Contoh penggunaan prosedur Filtering data:

Misalkan ingin mengambil data customerid dan nama customer yang merupakan gabungan firstname dan lastname yang berasal dari negara Argentina. Syntax yang bisa digunakan adalah:

```
proc sql;  
select customerid, firstname||' '||lastname as name  
from customer  
where country='Argentina';  
quit;
```

Atau misalkan ingin mengambil data judul album dimana nama artisnya diawali dengan Billy. Syntax yang bisa digunakan adalah:

```
proc sql;  
select title from album  
where artistid in (select artistid from artist where name like 'Billy%');  
quit;
```

Contoh lain misalkan ingin mengambil data customerid, address dan supportrepid dari table customer dimana data companynya kosong. Syntax yang digunakan adalah:

```
proc sql;  
select customerid, address, supportrepid  
from customer  
where company is missing;  
quit;
```

Grouping dan Sorting Data

Grouping data berfungsi untuk mengelompokkan data yang memiliki nilai yang sama menjadi sutau ringkasan/ summary. Dalam mengelompokkan data biasanya digunakan suatu fungsi agregasi (COUNT, SUM, AVERAGE) untuk masing – masing kelompok. Dalam PROC SQL grouping data dilakukan dengan perintah GROUP BY. Sementara Sorting data berfungsi untuk mengurutkan data. Sorting data dapat dilakukan menggunakan perintah ORDER BY. Data akan diurutkan berdasarkan variabel/ kolom yang ditulis setelah perintah ORDER BY.

Contoh penggunaan prosedur Grouping dan Sorting Data:

- Grouping Data

Misalkan ingin melihat jumlah barang untuk setiap invoiceid. Syntax yang digunakan adalah:

```
proc sql;  
select invoiceid, sum(quantity) as total  
from invoice_item  
group by invoiceid;  
quit;
```

- Sorting Data

Misalkan ingin mengambil data invoiceid, customerid dan total dari data yang billingcountry dari negara Ireland. Data diurutkan berdasarkan nilai total dengan urutan dari terbesar ke terkecil. Syntax yang digunakan adalah:

```
proc sql;  
select invoiceid, customerid, total from invoice  
where billingcountry='Ireland'  
order by total desc;  
quit;
```

Perintah UPDATE dan DELETE

Perintah UPDATE digunakan untuk melakukan perubahan atau *updating* pada data yang sudah ada dalam record. Bentuk umum perintah UPDATE adalah:

```
PROC SQL;  
  
UPDATE table  
  
SET column=new_value  
  
QUIT;
```

```
PROC SQL;  
  
UPDATE table  
  
SET column=new_value  
  
WHERE expression  
  
QUIT;
```

Sementara perintah DELETE digunakan untuk menghapus record yang ada dalam data. Bentuk umum perintah DELETE adalah:

```
PROC SQL;  
  
DELETE FROM table  
  
WHERE expression  
  
QUIT;
```

Contoh penggunaan perintah UPDATE dan DELETE:

- UPDATE Data

Misalkan ingin mengubah nilai billingstate menjadi Dublin untuk data yang billingstate nya bernilai kosong. Syntax yang digunakan adalah:

```
proc sql;  
update invoice  
set billingstate='Dublin'  
where billingstate is missing;  
quit;  
  
proc sql outobs=20;  
select * from invoice;  
quit;
```


- DELETE data

Misalkan ingin menghapus data customer yang data fax nya kosong. Syntax yang digunakan adalah:

```
PROC SQL;  
DELETE FROM customer  
WHERE fax is missing;  
QUIT;  
PROC PRINT data = customer;  
RUN;
```