

# ADVANCED SQL

Homework 1 - Week 3

## **Kelompok 7** **Citizen Data Scientist**

- Diaz Jubiary
- Hermulia Hadie
- Desi Sulistyowati
- Farahul Jannah



# BOYCE-CODD NORMAL FORM (BCNF)

- BCNF adalah teknik normalisasi database agar hubungan antar tabel yang tidak memiliki *partial* dan *transitive dependency*.
- BCNF berfungsi untuk menangani anomali dan *overlooping* yang tidak dapat ditangani dalam bentuk 3NF

Syarat BCNF adalah:

- Tabel harus dalam 3NF. Tabel 3NF didekomposisi menjadi beberapa tabel yang masing-masing memenuhi BCNF.
- Jika dan hanya jika setiap determinan yang ada pada relasi tersebut adalah *candidate key*. Definisi yang lain: jika untuk setiap ketergantungan penuh (*functional dependency*) nontrivial :  $X \rightarrow A$  atribut X adalah superkey.

# PERBEDAAN 3NF DAN BCNF

## PERBANDINGAN ANTARA BCNF DAN 3NF

- Contoh kasus redundansi pada 3NF

Jadwal = (Nim, Modul, Dosen)

FD = {Dosen → Modul}


Relasi ini memenuhi 3NF, karena tidak ada ketergantungan transitif.

Tetapi tidak memenuhi BCNF karena dari Dosen → Modul maka Dosen bukan candidate key.

Alternatif yang dilakukan adalah dekomposisi tabel menjadi :

<u>NIM</u>	<u>Modul</u>	Dosen
P11.2004.0129	VB.Net	Ajib
P11.2004.0130	Prolog	Aris
P11.2004.0129	VB Net	Ajib
P11.2004.0201	VB Net	Budi
P11.2004.0250	Prolog	Jono
P11.2004.0260	VB.Net	Budi

**NOT BCNF**



<u>NIM</u>	Dosen
P11.2004.0129	Ajib
P11.2004.0130	Aris
P11.2004.0129	Ajib
P11.2004.0201	Budi
P11.2004.0250	Jono
P11.2004.0260	Budi

**BCNF**

Dosen	<u>Modul</u>
Ajib	VB.Net
Aris	Prolog
Jono	Prolog
Budi	VB.Net

# PERBEDAAN 3NF DAN BCNF

No.	3NF	BCNF
1	Dalam 3NF seharusnya tidak ada ketergantungan transitif yaitu tidak ada atribut non prima yang harus bergantung secara transitif pada kunci kandidat.	Dalam BCNF untuk sembarang relasi $A \rightarrow B$ , A harus menjadi super key dari relasi.
2	Relatif lebih lemah dibanding BCNF	Relatif lebih kuat dari 3NF.
3	Dalam 3NF ketergantungan fungsional sudah ada dalam 1NF dan 2NF.	Dalam BCNF ketergantungan fungsional sudah ada dalam 1NF, 2NF dan 3NF
4	Redudansi relatif lebih tinggi pada 3NF.	Redudansi relatif lebih rendah di BCNF



# PERBEDAAN 3NF DAN BCNF

No.	3NF	BCNF
5	3NF dapat diperoleh tanpa mengorbankan ketergantungan fungsional.	Ketergantungan fungsional mungkin tidak dapat dipertahankan saat menggunakan metode BCNF.
6	3NF dapat dicapai tanpa kehilangan informasi dari tabel lama.	BCNF dapat menghilangkan beberapa informasi dari tabel lama.
7	3NF relatif lebih mudah untuk dicapai.	BCNF relatif lebih sulit untuk dicapai.
8	dapat dicapai dengan 3NF.	sulit dicapai dengan 3NF.

# SQL QUERY

- **Case 1**

- The marketing team wants to measure the audience for our product ad. They asked you about, how many female customers do you think in Jakarta have Gmail accounts? Also, if possible, they want to push the ad to potential users only. So they think it's better if you filter them by those who have transactions at least 10 times

- **Case 2**

- The product team wants to add some new products to our marketplace this week. Can you advise currently which product has the maximum performance in Q4 (Oct 2018 - Dec 2018)? Let say give them the top 5 products that have total transactions (quantity) above average.

- **Case 3**

- Our CEO asked what type of store currently gets the most transaction-specific in quantity from the Jakarta region? Is it the same between males and females? For comparison, please share outside the Jakarta region as well



# CASE 1

## SQL Query

```
select count (*)
from
(
  select customer_id,
  count(*) total_transaction
  from "transaction" t
  join customer c
  on customer_id =c.id
  where
    c.city = 'Jakarta' and
    c.email = 'Gmail' and
    c.gender = 'Female'
  group by 1
) A
```

Total customer wanita di Jakarta yang memiliki akun Gmail sebanyak 2913 orang.

```
select count (*) from
(select customer_id,
count(*) total_transaction from "transaction" t
join customer c
on customer_id =c.id
where
c.city = 'Jakarta' and
c.email = 'Gmail' and
c.gender = 'Female'
group by customer_id
--having count(*) >=10
) A
<
```

Results 1 x

count (*)
2,913

# CASE 1

## SQL Query

```
select count (*)
from
(
  select customer_id,
  count(*) total_transaction
  from "transaction" t
  join customer c
  on customer_id =c.id
  where
    c.city = 'Jakarta' and
    c.email = 'Gmail' and
    c.gender = 'Female'
  group by 1
  having count(*) >=10
) A
```

Total customer wanita di Jakarta yang memiliki akun Gmail dengan jumlah transaksi paling sedikit 10 kali adalah sebanyak 395 orang

```
select count (*) from
(select customer_id,
count(*) total_transaction from "transaction" t
join customer c
on customer_id =c.id
where
c.city = 'Jakarta' and
c.email = 'Gmail' and
c.gender = 'Female'
group by customer_id
having count(*) >=10
) A
```

Results 1 x

select count (\*) from (select customer\_id, count(\*) total\_transaction

count	
123	
395	



# CASE 2

## SQL Query








```
with
  pool as (
    select product_id,
           sum(quantity) as sum_quantity
    from   datasource_sql_ds11.transaction
    where  extract ('year' from created_at) in (2018) and
           extract ('month' from created_at) in (10,11,12)
    group by 1
    order by 2 desc
  )
```

```
select  trsct.product_id,
        trsct.created_at,
        store."type",
        row_number() over(partition by trsct.product_id) as transactions_order,
        sum(trsct.quantity) over(partition by trsct.product_id) as total_quantity,
        (select avg(sum_quantity) from pool) as average_total_quantity
from    datasource_sql_ds11."transaction" as trsct
left join datasource_sql_ds11.store as store on trsct.store_id = store.id
where  extract ('year' from created_at) in (2018) and
        extract ('month' from created_at) in (10,11,12) and
        store."type" = 'Online store'
order by 4, 5 desc
limit 5
```

**Top 5 produk dengan performance terbaik di *marketplace* pada periode Q4 2018 yang total transaksi diatas rata-rata adalah produk\_id 49, 39, 38, 50, dan 58**

# CASE 2

## Hasil

	123 product_id 	 created_at 	ABC type 	123 transactions_order 	123 total_quantity 	123 average_total_quantity 
1	49	2018-10-27	Online store	1	892,462	155,293.4651162791
2	39	2018-11-11	Online store	1	834,085	155,293.4651162791
3	38	2018-12-03	Online store	1	816,780	155,293.4651162791
4	50	2018-12-01	Online store	1	758,775	155,293.4651162791
5	58	2018-12-05	Online store	1	370,041	155,293.4651162791

Top 5 produk dengan performance terbaik periode Q4 2018 di *marketplace* yang total transaksi diatas rata-rata adalah produk\_id 49, 39, 38, 50, dan 58



# CASE 3

## SQL Query

```
select s."type",
       sum(quantity) total_quantity,
       c.city,
       c.gender from "transaction" t
join store s on store_id = s.id
join customer c on customer_id = c.id
where c.city = 'Jakarta'
group by s."type" , c.city, c.gender
order by total_quantity desc
```

Tipe toko dengan transaksi terbanyak di Jakarta adalah online store. Pembeli wanita membeli total 749.269 barang, pembeli pria 720.870 pada toko tersebut.

```
select s."type" , sum(quantity) total_quantity, c.city, c.gender from "transaction" t
join store s
on store_id = s.id
join customer c
on customer_id = c.id
where c.city = 'Jakarta'
group by s."type" , c.city, c.gender
order by total_quantity desc
```

	type	total_quantity	city	gender
1	Online store	749,269	Jakarta	Male
2	Online store	720,870	Jakarta	Female
3	Offline store	163,001	Jakarta	Female
4	Offline store	158,253	Jakarta	Male
5	Partnership	11,010	Jakarta	Female
6	Event	10,916	Jakarta	Female
7	Event	10,176	Jakarta	Male
8	Partnership	7,969	Jakarta	Male

# CASE 3

## SQL Query

```
select
  s."type",
  sum(quantity),
  case when city = 'Jakarta' then 'Jakarta'
  else 'Luar Jakarta' end Citylocation,
  c.gender
from "transaction" t
join customer c on customer_id = c.id
join store s on s.id = store_id
group by store_id,
  Citylocation,
  c.gender,
  s."type"
ORDER BY sum(quantity) desc
```

```
select S."type",
       sum(quantity),
       case when city = 'Jakarta' then 'Jakarta'
       else 'Luar Jakarta' end Citylocation,
       C.gender
from "transaction" t
join customer C
on customer_id = C.id
join store S
on S.id = store_id
group by store_id, Citylocation, C.gender, S."type"
ORDER BY sum(quantity) desc
```

	ABC type	123 sum	ABC citylocation	ABC gender
1	Online store	2,962,155	Luar Jakarta	Male
2	Online store	2,940,683	Luar Jakarta	Female
3	Online store	749,269	Jakarta	Male
4	Online store	720,870	Jakarta	Female
5	Offline store	645,756	Luar Jakarta	Female
6	Offline store	624,408	Luar Jakarta	Male
7	Offline store	163,001	Jakarta	Female
8	Offline store	158,253	Jakarta	Male
9	Event	46,229	Luar Jakarta	Female
10	Event	42,558	Luar Jakarta	Male
11	Partnership	39,468	Luar Jakarta	Male
12	Partnership	32,069	Luar Jakarta	Female
13	Partnership	11,010	Jakarta	Female
14	Event	10,916	Jakarta	Female
15	Event	10,176	Jakarta	Male
16	Partnership	7,969	Jakarta	Male

Perbandingan antara pembelian Jakarta dengan Luar Jakarta ditunjukkan oleh gambar disamping kiri ini



# CASE 3

## SQL Query

```
select
  s."type",
  sum(quantity) total_quantity,
  c.city,
  c.gender from "transaction" t
join store s
  on store_id = s.id
join customer c
  on customer_id =c.id
where c.city <>'Jakarta'
group by s."type" , c.city, c.gender
order by city desc
```

Jika ingin melihat detail region selain Jakarta dapat dilihat pada gambar di bawah ini

```
select s."type" , sum(quantity) total_quantity, c.city, c.gender from "transaction" t
join store s
on store_id = s.id
join customer c
on customer_id =c.id
where c.city <>'Jakarta'
group by s."type" , c.city, c.gender
order by city desc
```

	ABC type	123 total_quantity	ABC city	ABC gender
1	Event	11,845	Tangerang	Female
2	Event	11,412	Tangerang	Male
3	Offline store	165,004	Tangerang	Female
4	Offline store	149,742	Tangerang	Male
5	Online store	730,557	Tangerang	Female
6	Online store	749,926	Tangerang	Male
7	Partnership	9,112	Tangerang	Female
8	Partnership	7,543	Tangerang	Male
9	Event	24,509	Depok	Female
10	Event	20,454	Depok	Male
11	Offline store	322,125	Depok	Female
12	Offline store	321,558	Depok	Male
13	Online store	1,487,399	Depok	Female
14	Online store	1,481,058	Depok	Male
15	Partnership	14,498	Depok	Female
16	Partnership	23,643	Depok	Male
17	Event	9,875	Bogor	Female
18	Event	10,692	Bogor	Male
19	Offline store	158,627	Bogor	Female
20	Offline store	153,108	Bogor	Male
21	Online store	722,727	Bogor	Female
22	Online store	731,171	Bogor	Male
23	Partnership	8,459	Bogor	Female
24	Partnership	8,282	Bogor	Male

# **Terima Kasih**

Kelompok 7 - Citizen Data Scientist