1. Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form (BCNF) adalah suatu relasi dalam basis data harus dirancang sedemikian rupa sehingga mereka tidak memiliki ketergantungan sebagian (*partial dependecy*), maupun ketergantungan transitif (*transitive dependecy*). Sebuah relasi memenuhi syarat BCNF jika dan hanya jika setiap **determinan** dalam relasi tersebut adalah *candidate key*. Ini berarti setiap determinan dalam relasi BCNF bernilai unik.

Relasi Boyce-Codd Normal Form (BCNF) tidak mengharuskan sudah dalam bentuk normal ketiga (3NF), baru bisa di buatkan ke dalam BCNF. Oleh karena itu untuk melakukan uji BCNF hanya dengan mengidentifaksi seluruh determinan yang ada pada suatu relasi, lalu pastikan determinan-determinan tersebut adalah *candidate key*. Dengan demikian, bisa dikatakan bahwa BCNF lebih baik dari bentuk normal ketiga (3NF), sehingga setiap relasi di dalam BCNF juga merupakan relasi dalam 3NF, tetapi tidak sebaliknya, suatu relasi di dalam 3NF belum tentu merupakan relasi di dalam BCNF

BCNF merupakan bentuk normal sebagian perbaikan terhadap 3NF. BCNF memiliki ketentuan yaitu :

- Masing-masing atribut utama bergantung fungsional penuh pada masing-masing kunci dimana kunci tersebut bukan bagiannya.
- Relasi adalah BCNF (optimal) jika setiap determinan atribut-atribut relasi adalah kunci relasi.
- Relasi adalah BCNF (optimal) jika kapanpun fakta-fakta disimpan mengenai beberapa atribut, maka atribut-atribut ini merupakan satu kunci relasi.
- BCNF dapat memiliki lebih dari satu kunci.

Properti penting BCNF adalah relasi tidak memiliki informasi yang redundan.

Contoh Boyce-Codd Normal Form (BCNF) :

Relasi kehadiran tidak BCNF

Dosen	Semester	Kuliah	Sesi	Kehadiran
Joe	1/88	COBOL	Sesi1	35
Jeni	1/88	MATH	Sesi1	40
Garin	2/88	UNIX	Sesi2	33
Jeni	1/88	MATH	Sesi2	42
Garin	2/88	UNIX	Sesi1	47
Joe	1/88	COBOL	Sesi2	50
Joe	1/88	COBOL	Sesi3	12

Relasi kehadiran BCNF

Semester	Kuliah	Sesi	Kehadiran
1/88	COBOL	Sesi1	35
1/88	MATH	Sesi1	40
2/88	UNIX	Sesi2	33
1/88	MATH	Sesi2	42
2/88	UNIX	Sesi1	47
1/88	COBOL	Sesi2	50
1/88	COBOL	Sesi3	12

Relasi dosen BCNF

Dosen	Semester	Kuliah
Joe	1/88	COBOL
Jeni	1/88	MATH
Garin	2/88	UNIX

- Buatlah query untuk 3 skenario berikut:
  "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."

```
select c.id customer_id , t.total_transaction
from datasource_sql_ds11.customer c
left join
(select customer_id , count(id) total_transaction
from datasource_sql_ds11."transaction"
group by customer_id)
t on c.id = t.customer_id
where c.gender = 'Female' and c.email = 'Gmail' and c.city = 'Jakarta' and
t.total_transaction >10
```

- "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."

```
select product_id , sum(quantity) total
from datasource_sql_ds11."transaction"
where created_at between '2018-10-01' and '2018-12-31'
and quantity > (select avg(quantity) from datasource_sql_ds11."transaction" )
group by product_id
order by total desc
limit 5
```

- "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."

```
select distinct s.id, s.type, c.gender, c.city, count(t.id) transaction_count
from datasource_sql_ds11."transaction" t
left join datasource_sql_ds11.customer c on t.customer_id = c.id
left join datasource_sql_ds11.store s on t.store_id = s.id
group by s.id, s.type, c.city, c.gender
order by transaction_count desc
```