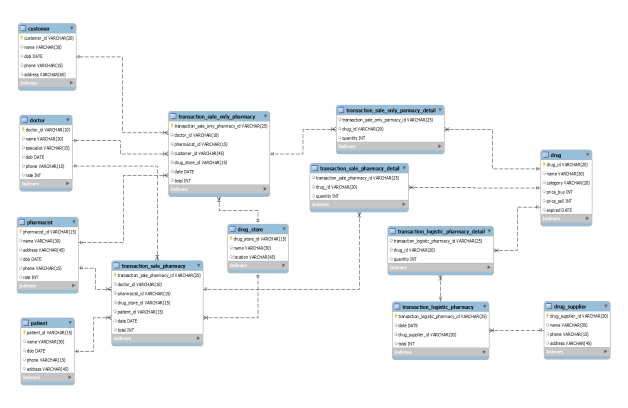
LAPORAN TUGAS BESAR DATA WAREHOUSE

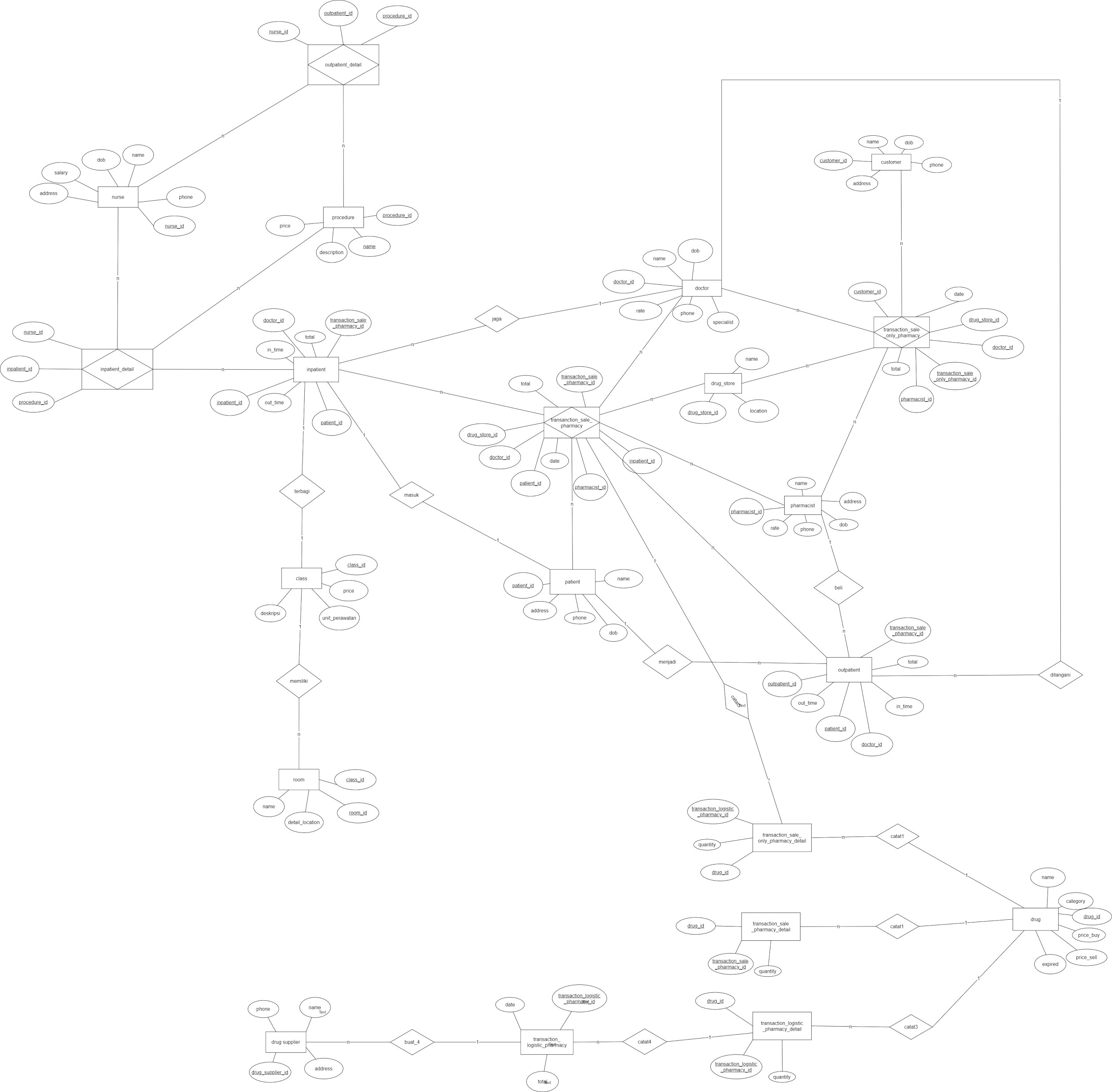
1. **Identitas Anggota Kelompok**

* Subkhan Ibnu Aji (1301172001)
* Ahmad Refi Juliano (1301174040)
* Novelly Naomi Lingga (1301173407)
* Muh. Alkahfi Khuzaimy Abdullah (1301174048)

1. **Skema Relasi Source Data**



1. **ER Diagram Reconcile Data**



link : https://ibb.co/tbrGZpb

1. **Analisis Proses Perubahan Source Data Menjadi Reconcile**

Analisis proses perubahan dari source data menjadi data reconcile adalah kami menggunakan kaitan entitas di setiap source data, terdapat 2 source data yaitu farmasi dan perawatan. Didalam source data farmasi memiliki kesamaan entitas dengan yang berada pada source data perawatan, seperti contohnya entitas doctor. untuk menghubungkan kedua source data tersebut kami menghubungkan entitas setiap source data yang terhubung dengan entitas doctor yang masing masing dimiliki source data maka dari itu akan didapatkan data yang reconcile. Kemudian kami juga melihat beberapa dataset merupakan data master dan sisa nya ialah data yang transactional, dari sanalah kami menentukan keterkaitan antar data dan juga menentukan primary key, foreign key, serta atrbibut masing-masing data, dan jadilah data reconcile.

1. **Analisis Fase User Requirement Analysis**

Berikut ini adalah urutan analisis kami dari beberapa fase URA yang telah dilalui:

1. Membuat Application Field Table, ini merupakan fase awal yang perlu dilakukan karena dalam data warehouse yang akan dibangun nanti perlu diketahui pada lingkup apa pengimplementasiannya, kemudian dari data source yang telah diberikan user, perlu ditentukan data mart dan berbagai fact atau event yang biasanya terjadi atau dilakukan oleh perusahaan terkait. 1 fact dapat mencangkup beberapa data, maka perlu diperhatikan dalam menentukan jumlah serta apa event pada fact itu sendiri.

2. Membuat User Requirement Glossary Table, Tentunya setiap fact haruslah dijabarkan apa saja yang menjadi dimension, measures, serta berapa lama history nya, karena bertujuan untuk membuat user mengerti terhadap setiap fact yang dirancang

3. Membuat Preliminary workload, sebelum bertemu user, perlu dipersiapkan workload ini untuk mencatat berbagai informasi atau data yang ingin diketahui oleh user, yang mana nantinya informasi ini akan didapatkan dengan menjalankan query.

1. **APPLICATION FIELD TABLE**

|  |  |  |
| --- | --- | --- |
| **Application Field** | **Data Mart** | **Facts** |
| HEALTH  SERVICE | Pharmacy | Transaction Sale Pharmacy, Transaction Sale Only Pharmacy, Transaction Logistic Pharmacy |
| Nursing | Inpatient, Outpatient |

1. **USER REQUIREMENT GLOSSARY TABLE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fact** | **Dimensions** | **Measures** | **History** |
| Transaction Sale Pharmacy | doctor, pharmacist, patient, drug, drug\_store, date, total, quantity | -Income and Outcome for all sales and purchase  -Quantity of Sales and Purchase for all date or specific date  -Quantity of Drugs, Drug Store, Doctor, Pharmacist, and Patient in each or all sales and purchase  -Total spending money for each patient | 1 Year |
| Transaction Sale Only Pharmacy | doctor, pharmacist, customer, drug, drug\_store, date, total, quantity | -Income and Outcome for all sales and purchase  -Quantity of Sales and Purchase for all date or specific date  -Quantity of Drugs, Drug Store, Doctor, Pharmacist, and Customer in each or all sales and purchase  -Total spending money for each customer | 1 Year |
| Transaction Logistic Pharmacy | drug, drug\_supplier, date, quantity, total | -Quantity of Drug, for each or all Drug Supplier in Logistic Transaction  -Quantity of Drug Supplier in Logistic Transaction  -Total Money from drug supply | 1 Year |
| Inpatient | doctor, pharmacist, patient, drug\_store, date, total, class, time | -Quantity of Patients, Pharmacists, Doctors, Drug Stores, for all or specific date, class, or time  -Total duration for each patients  -Total income from all inpatient | 1 Year |
| Outpatient | doctor, pharmacist, patient, drug\_store, date, total, time | -Quantity of Patients, Pharmacists, Doctors, Drug Stores, for all or specific wdate, or time  -Total duration for each patients  -Total income from all outpatient | 1 Year |

1. **PRELIMINARY WORKLOAD**

**Sebelum Wawancara**

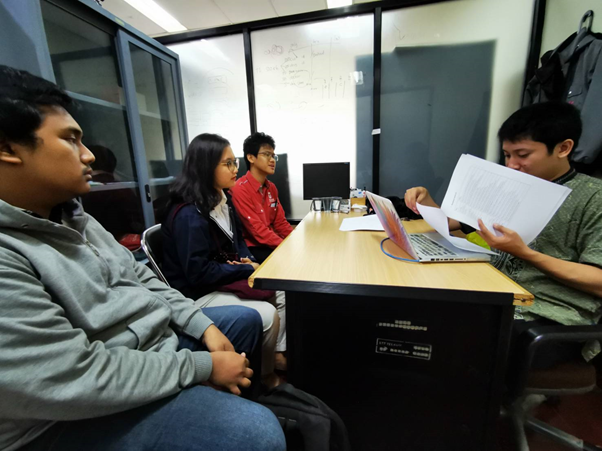
|  |  |
| --- | --- |
| **Fact** | **Query** |
| Transaction Sale Pharmacy | Berapa total gaji seluruh dokter?  Berapa banyak obat yang dibeli oleh semua pasien? |
| Transaction Sale Only Pharmacy | Berapa banyak jumlah obat yang dibeli bulan ini?  Berapa jumlah drug store dari semua lokasi?  Berapa banyak obat yang dipasok ke drug store? |
| Transaction Logistic Pharmacy | Berapa banyak nama obat yang terdaftar?  Berapa besar pengeluaran dalam pembelian obat bulan ini? |
| Inpatient | Berapa jumlah perawat di rumah sakit?  Berapa jumlah kelas ruangan di rumah sakit?  Berapa banyak pasien yang menjalankan rawat inap?  Berapa lama durasi pasien rawat inap menginap?  Berapa lama suster merawat seorang pasien?  Berapa banyak jumlah pasien di sebuah kamar?  Berapa banyak jumlah pasien di level 3?  Berapa banyak aturan di rumah sakit? |
| Outpatient | Berapa banyak pasien yang telah selesai rawat jalan?  Berapa jumlah kontrol yang disarankan oleh dokter? |

**Setelah Wawancara**

Keterangan : Tulisan berwarna merah adalah masukan dari user

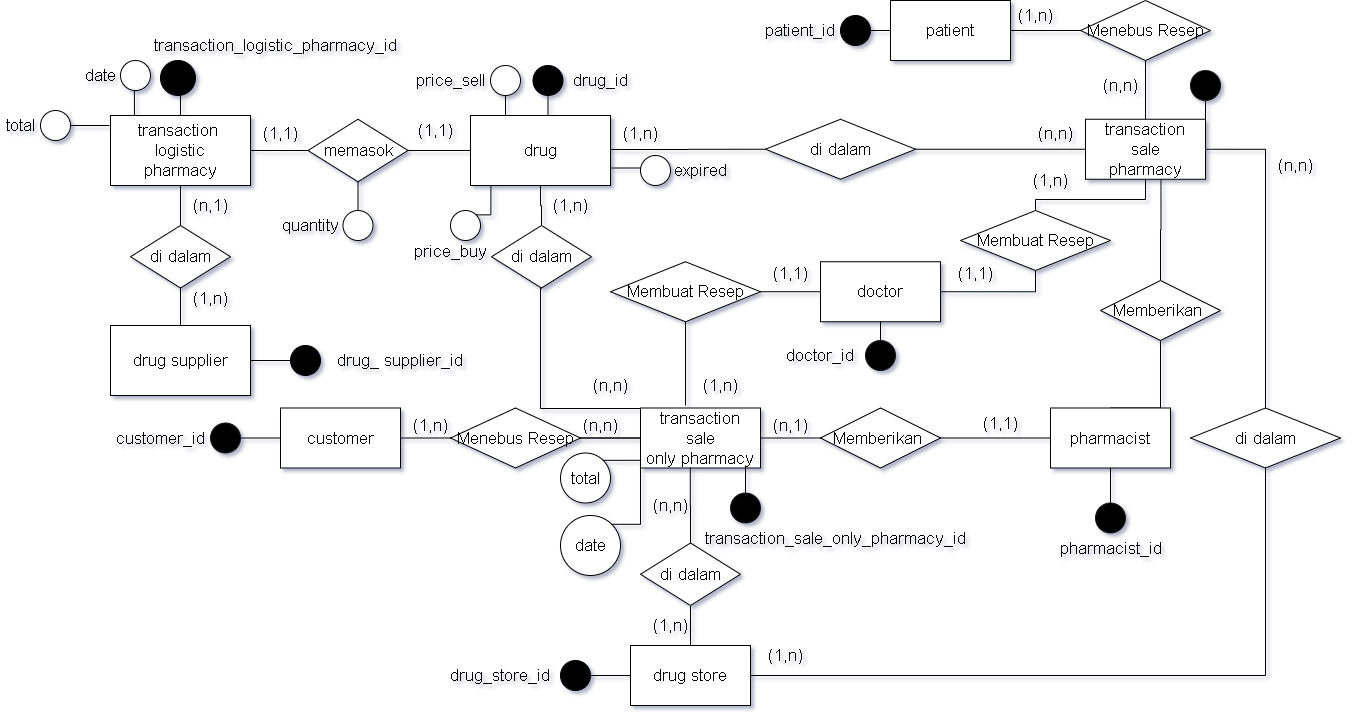
|  |  |
| --- | --- |
| **Fact** | **Query** |
| Transaction Sale Pharmacy | Berapa total gaji seluruh dokter?  Berapa banyak obat yang dibeli oleh semua pasien?  -5 Obat apa yang paling banyak dibeli dalam setiap bulan?  -Urutan Drug Store dari yang paling laris sampai yang tidak laris?  -Analisis 5 Kabupaten yang berobat paling banyak setiap bulannya? |
| Transaction Sale Only Pharmacy | Berapa banyak jumlah obat yang dibeli bulan ini?  Berapa jumlah drug store dari semua lokasi?  Berapa banyak obat yang dipasok ke drug store?  -Analisis 5 orang farmasi dengan gaji paling besar setiap bulan?  -Besar pemasukan setiap drug store (uang masuk) dan jumlah produk yang terjual atau keluar?  -Besar keuntungan setiap drug store?  -Berapa hari rata-rata obat akan expired saat dijual kepada pelanggan pada tiap bulannya?  -Besar pengeluaran setiap drug store (biaya) dan jumlah produk yang masuk? |
| Transaction Logistic Pharmacy | Berapa banyak nama obat yang terdaftar?  Berapa besar pengeluaran dalam pembelian obat bulan ini?  -5 drug supplier yang paling laris? |
| Inpatient | Berapa jumlah perawat di rumah sakit?  Berapa jumlah kelas ruangan di rumah sakit?  Berapa banyak pasien yang menjalankan rawat inap?  Berapa lama durasi pasien rawat inap menginap?  Berapa lama suster merawat seorang pasien?  Berapa banyak jumlah pasien di sebuah kamar?  Berapa banyak jumlah pasien di level 3?  Berapa banyak aturan di rumah sakit? |
| Outpatient | Berapa banyak pasien yang telah selesai rawat jalan?  Berapa jumlah kontrol yang disarankan oleh dokter? |

1. **LAMPIRAN WAWANCARA**

****

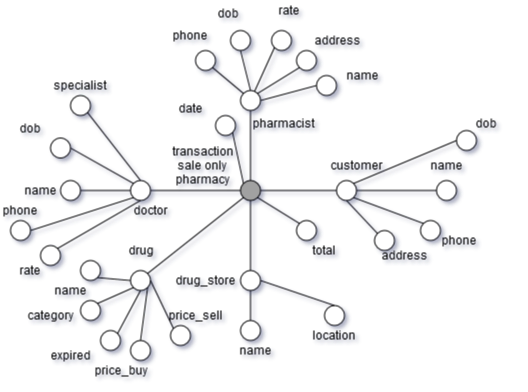
Wawancara dengan pak Doni Hari Senin, 02/03/2020

**6. Analisis Conceptual Design (Fact Schema)**

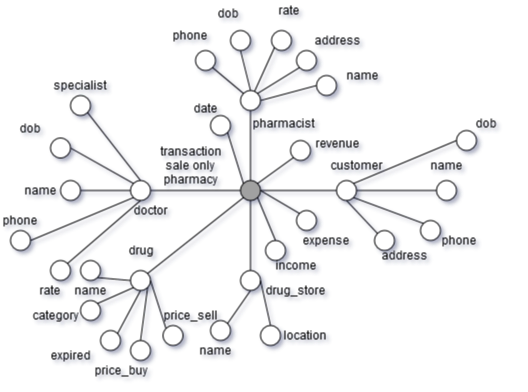


1. **Daftar Fact**
   1. Transaction Sale Only Pharmacy : Event ini melingkupi data penjualan obat dari toko obat yang dilakukan oleh apoteker kepada customer, disertai total biayanya, berdasarkan resep dokter.
   2. Transaction Sale Pharmacy : Event ini melingkupi data penjualan obat dari toko obat yang dilakukan oleh apoteker kepada pasien terdaftar, disertai total biayanya, berdasarkan resep dokter.
   3. Transaction Logistic Pharmacy : Event ini melingkupi data pemasukan stok obat dari pemasok obat kepada toko obat, disertai total biayanya.
2. **Atribut Tree, Prune, dan Graft**
   1. **Transaction Sale Only Pharmacy**

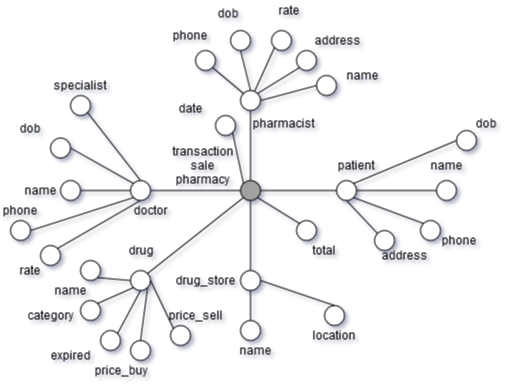
* ATRIBUT TREE



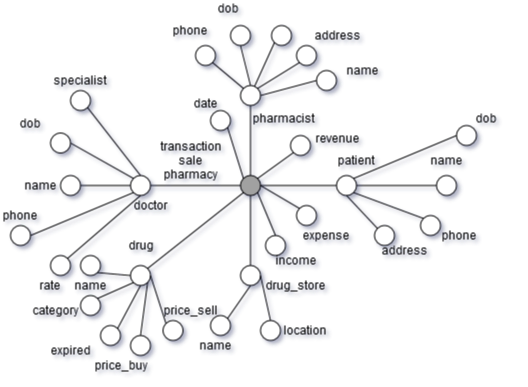
* PRUNE AND GRAFT



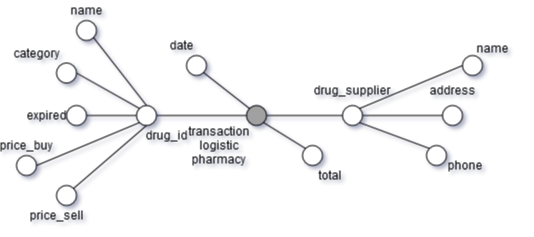
* 1. **Transaction Sale Pharmacy**
* ATRIBUT TREE



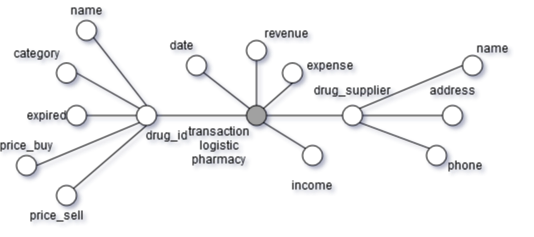
* PRUNE AND GRAFT



* 1. **Transaction Logistic Pharmacy**
* ATRIBUT TREE



* PRUNE AND GRAFT

****

1. **Daftar Dimensi**
   1. Transaction Sale Only Pharmacy : doctor, pharmacist, customer, drug\_store, date, drug, drug\_qty
   2. Transaction Sale Pharmacy : doctor, pharmacist, patient, drug\_store, date, drug, drug\_qty
   3. Transaction Logistic Pharmacy : date, drug\_supplier, drug, drug\_qty
2. **Daftar Measures**
3. Transaction Sale Only Pharmacy :

* Quantity of Transaction Sale Only Pharmacy
* Quantity of Doctors, Pharmacists, and Customers
* Revenue, Expense, Income

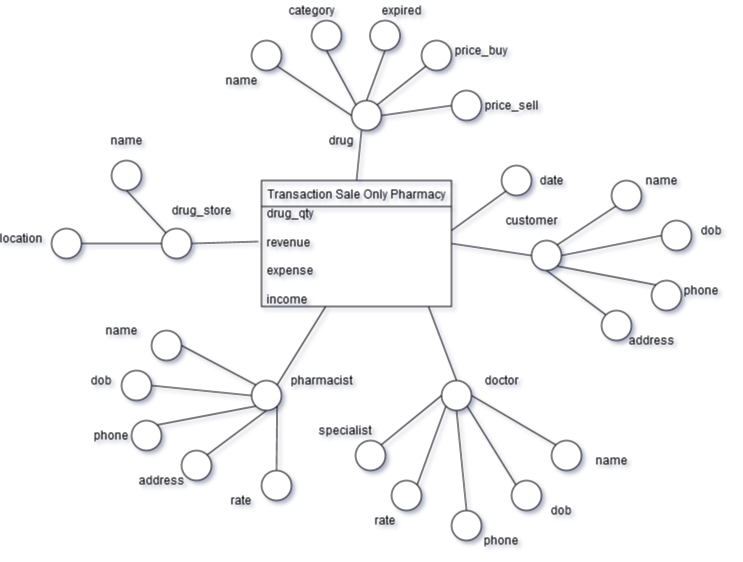
1. Transaction Sale Pharmacy :

* Quantity of Transaction Sale Pharmacy
* Quantity of Patients, Doctors, and Pharmacists
  + Revenue, Expense, Income

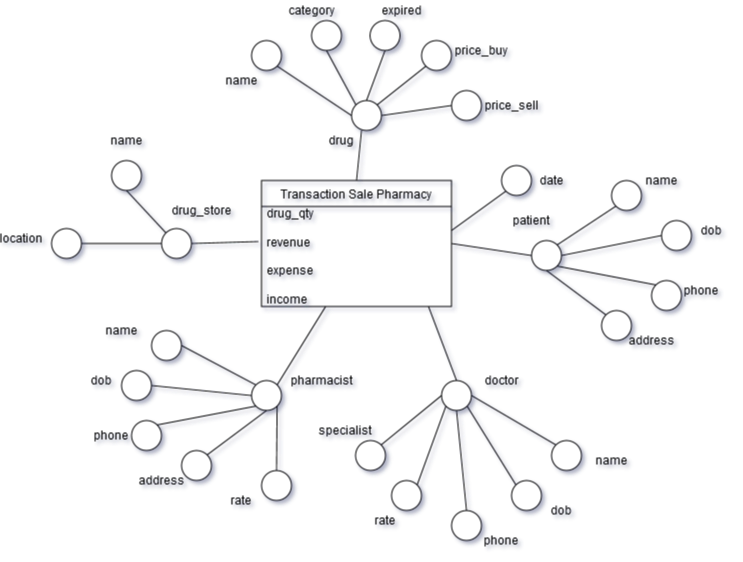
1. Transaction Logistic Pharmacy :

* Number of drug supplier
  + Revenue, Expense, Income

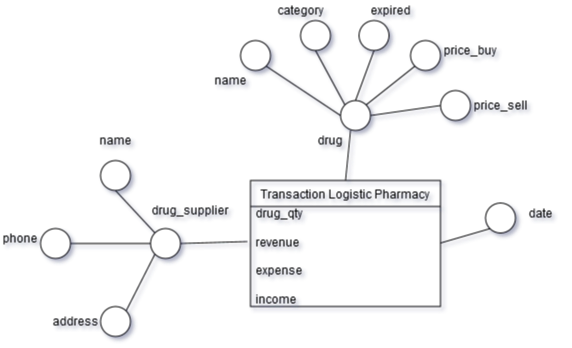
1. **Fact Schema**
2. **Transaction Sale Only Pharmacy**



1. **Transaction Sale Pharmacy**



1. **Transaction Logistic Pharmacy**

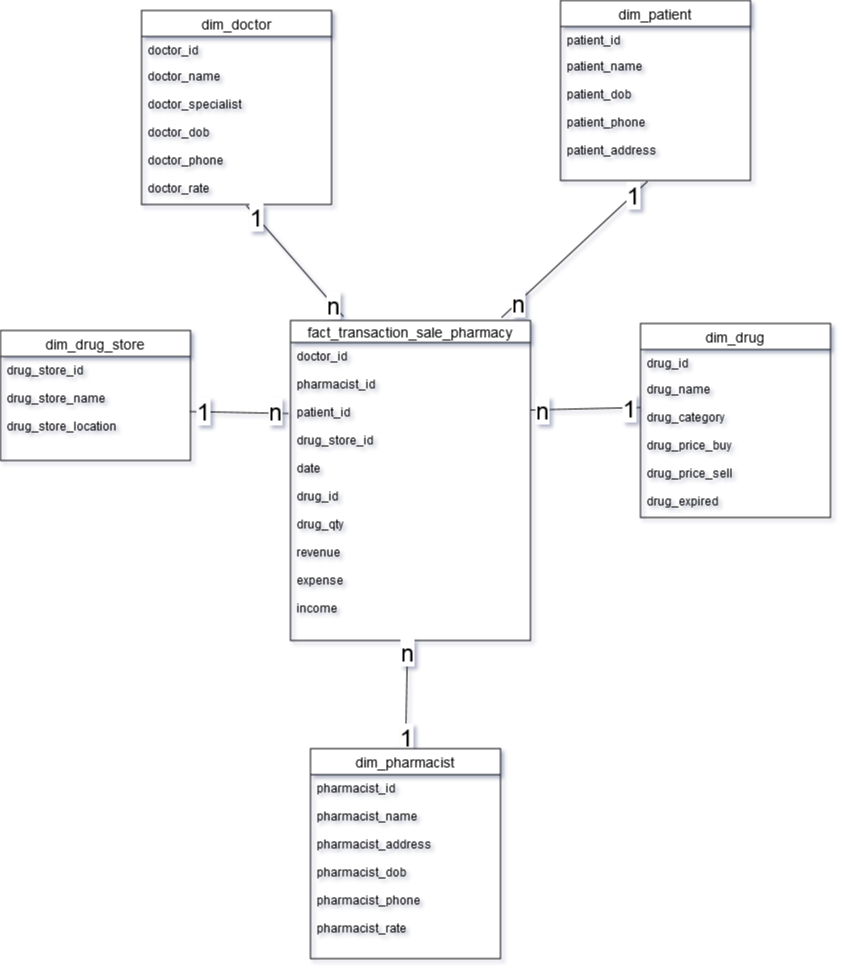


**7. Analisis Logical Design (Star Schema)**

1. **Fact Transaction Sale Pharmacy**

Proses:

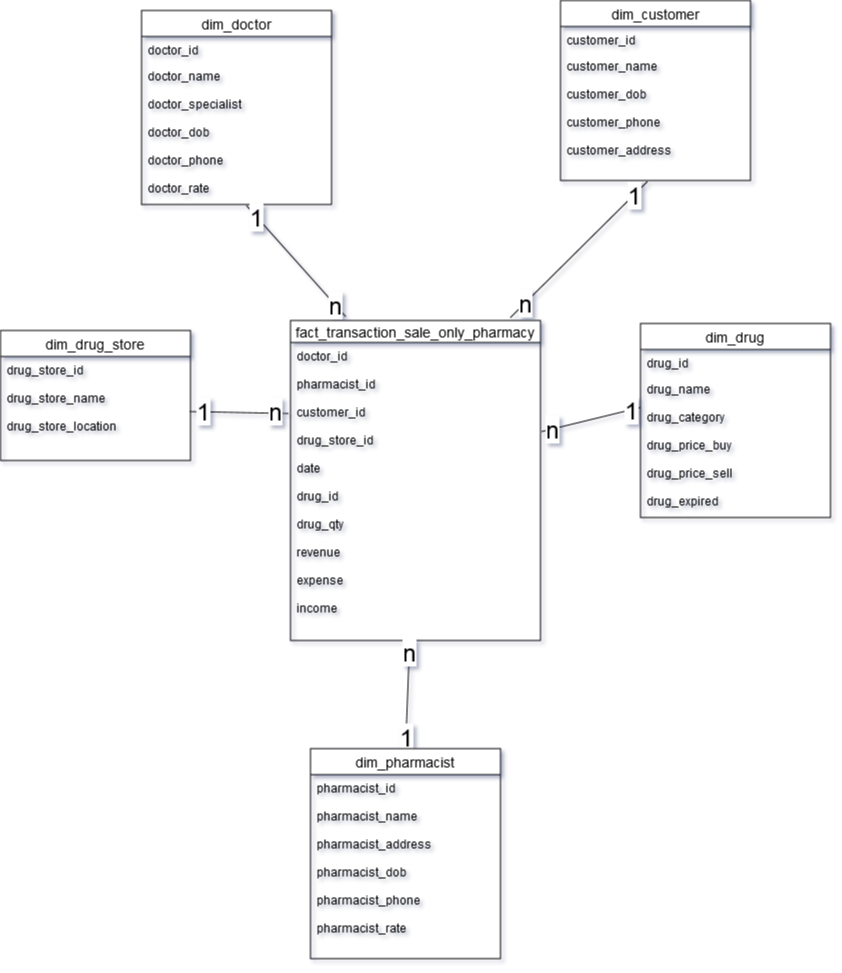
1. Menjadikan table transaction\_sale\_pharmacy menjadi fact (karena merupakan event yang dapat diobservasi) pada star schema.
2. Menjadikan seluruh atribut yang memiliki nomor unik atau id yang terhubung dengan table fact\_transaction\_sale\_pharmacy (menjadi foreign key-nya). sebagai dimension atau dim.
3. Melakukan repetisi nilai untuk setiap atribut ke dalam table fact\_transaction\_sale\_pharmacy
4. Mengatur kardinalitasnya



**2. Fact Transaction Sale Only Pharmacy**

Proses:

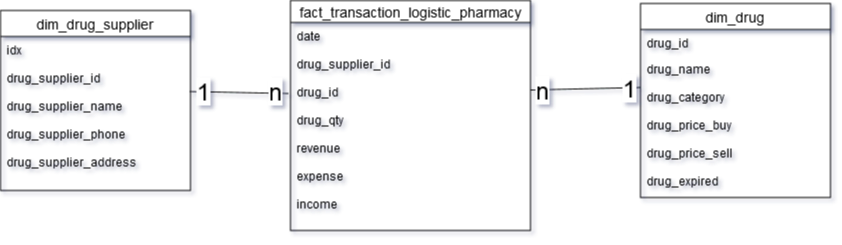
1. Menjadikan table Fact Transaction Sale Only sebagai fact (karena merupakan event yang dapat diobservasi) pada star schema
2. Menjadikan seluruh atribut yang memiliki nomor unik atau id yang terhubung dengan Fact Transaction Sale Only (menjadi foreign key nya), sebagai dimension atau dim
3. Melakukan repetisi nilai unik setiap atribut kedalam Fact Transaction Sale Only
4. Mengatur kardinalitasnya



**3. Fact Transaction Logistic Pharmacy**

Proses:

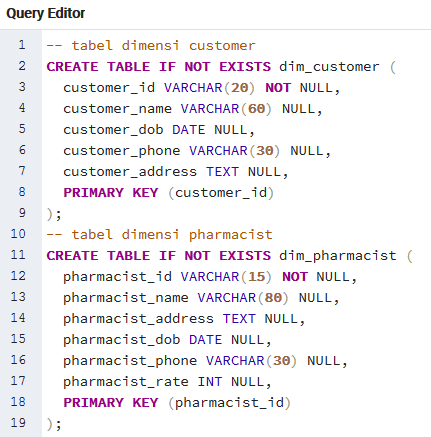
1. Menjadikan table transaction\_logistic\_pharmacy sebagai fact (karena merupakan event yang dapat diobservasi) pada star schema
2. Menjadikan seluruh atribut yang memiliki nomor unik atau id yang terhubung dengan fact transaction\_logistic\_pharmacy (menjadi foreign key nya), sebagai dimension atau dim
3. Melakukan repetisi nilai unik setiap atribut kedalam fact transaction\_logistic\_pharmacy
4. Mengatur kardinalitasnya

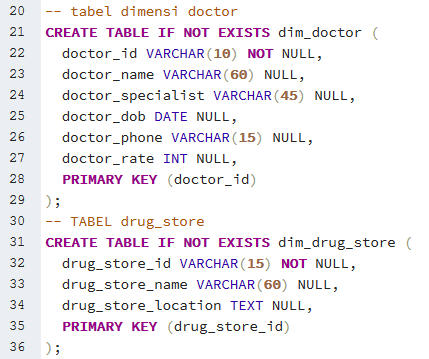


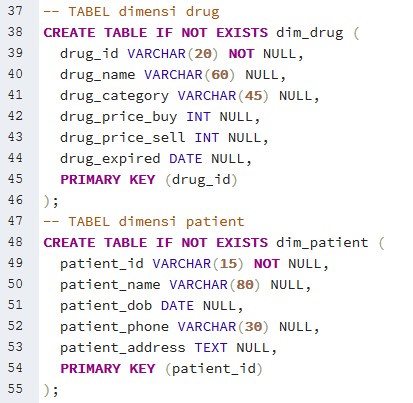
**8. Physical Design: DDL Fact Schema, Query Event, Strategi dan Query ETL**

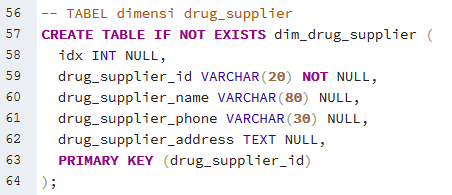
Prosedur proses ETL kami terdiri dari 3 tahap, yaitu Extract, Transform, dan Loading

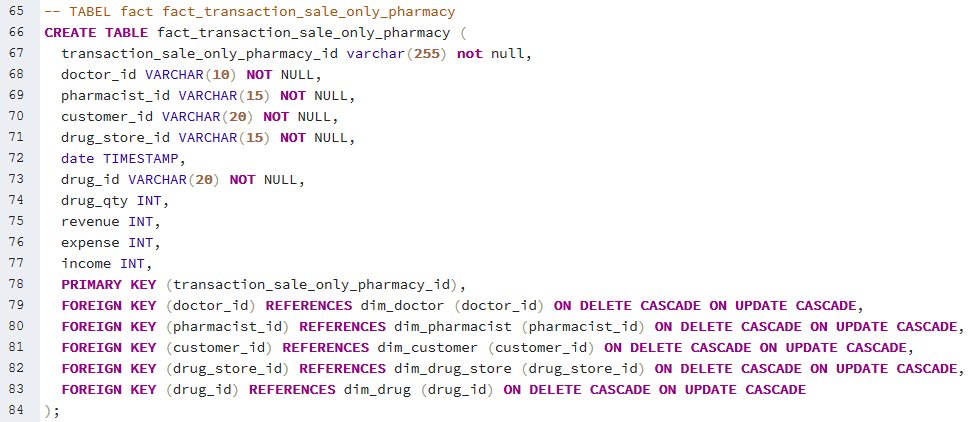
1. Extract : Sebelum kami melakukan extract terhadap data source, kami menyiapkan database nya terlebih dahulu, dengan mengeksekusi data definition language pada postgresql. Setelah database siap, maka kami lakukan extract terhadap data source, untuk mengambil semua recordnya
2. Transform : Transformasi data kami lakukan secara manual, yaitu dengan mengakses kedalam data source secara langsung, lalu kami memperbaiki id yang duplicate, data yang tidak konsisten, dan record yang berantakan.
3. Loading : dalam proses ini, data source yang telah diextract dan ditransform, diinsert kedalam database, insert data dimulai dari semua tabel dimension, dan setelah itu tabel fact. Insert dilakukan menggunakan Program Python, sehingga setiap pengisian tabel dapat dilakukan dengan mengeksekusi setiap program ETL yang telah ada.
4. **Data Definition Language Star Schema**

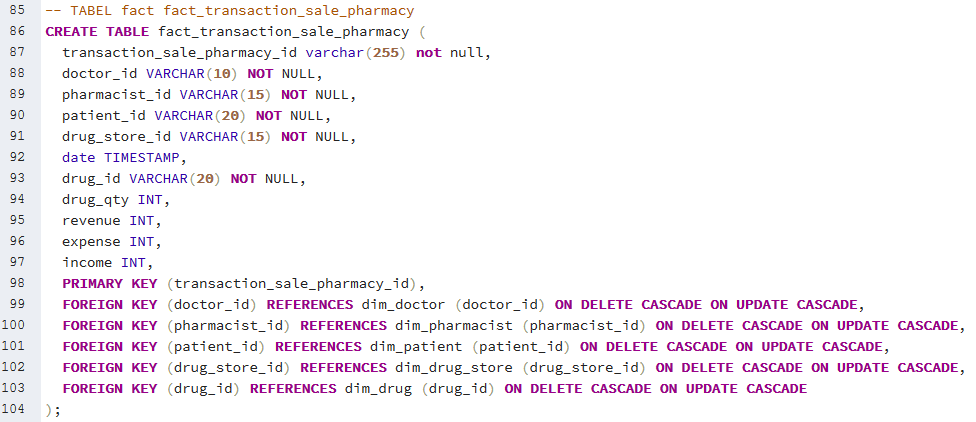
****

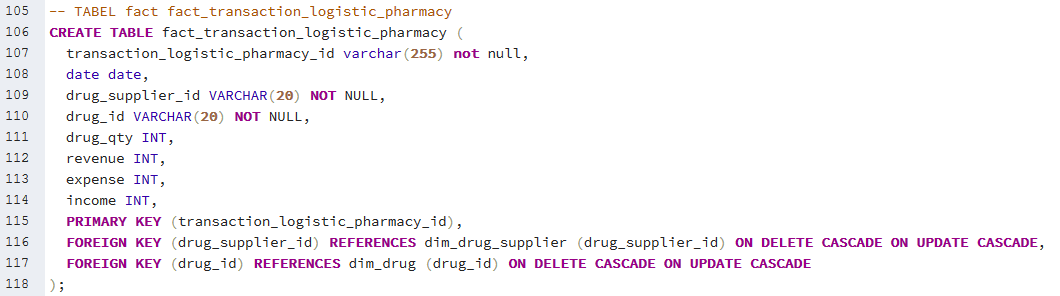
****

****

****

****

****

****

1. **Query Event**

1. Analisis 5 orang farmasi dengan gaji paling besar setiap bulan?

SELECT pharmacist\_name, pharmacist\_rate

FROM dim\_pharmacist

ORDER BY pharmacist\_rate DESC LIMIT 5;

2. 5 Obat apa yang paling banyak dibeli dalam setiap bulan?

SELECT drug\_name,COUNT(fact\_transaction\_sale\_only\_pharmacy.drug\_id)

AS totalsale FROM fact\_transaction\_sale\_only\_pharmacy

INNER JOIN dim\_drug

ON fact\_transaction\_sale\_only\_pharmacy.drug\_id=dim\_drug.drug\_id

GROUP BY drug\_name

ORDER BY totalsale DESC

LIMIT 5;

3. 5 Drug Supplier terlaris

SELECT drug\_supplier\_name,COUNT(recap.drug\_supplier\_id) AS penjualan FROM fact\_transaction\_logistic\_pharmacy AS recap

INNER JOIN dim\_drug\_supplier as ds

ON ds.drug\_supplier\_id = recap.drug\_supplier\_id

GROUP BY drug\_supplier\_name

ORDER BY penjualan DESC LIMIT 5;

4. Urutan Drug Store dari yang paling laris sampai yang tidak laris?

SELECT drug\_store\_name, COUNT(fact\_transaction\_sale\_pharmacy.drug\_store\_id) AS totalsale FROM dim\_drug\_store

JOIN fact\_transaction\_sale\_pharmacy

ON dim\_drug\_store.drug\_store\_id = fact\_transaction\_sale\_pharmacy.drug\_store\_id

GROUP BY drug\_store\_name

ORDER BY drug\_store\_name DESC;

5. Besar pengeluaran setiap drug store (biaya) dan jumlah produk yang masuk pada tahun 2018

SELECT DISTINCT drug\_store\_name, EXTRACT (YEAR FROM recap.date) AS year, SUM(revenue) AS revenue, SUM(drug\_qty) AS jmlobatmasuk

FROM fact\_transaction\_sale\_only\_pharmacy AS recap

INNER JOIN dim\_drug\_store as ds ON ds.drug\_store\_id = recap.drug\_store\_id

WHERE(SELECT EXTRACT (YEAR FROM recap.date)) = '2018'

GROUP BY drug\_store\_name,year

ORDER BY revenue DESC;

6. Besar pemasukan setiap drug store dan jumlah produk yang terjual pada tahun 2018

SELECT DISTINCT drug\_store\_name, EXTRACT (YEAR

FROM recap.date) AS year, SUM(expense) AS expense, SUM(drug\_qty) AS jmlobatterjual

FROM fact\_transaction\_sale\_only\_pharmacy AS recap

INNER JOIN dim\_drug\_store as ds ON ds.drug\_store\_id = recap.drug\_store\_id

WHERE(SELECT EXTRACT (YEAR FROM recap.date)) = '2018'

GROUP BY drug\_store\_name,year

ORDER BY expense DESC;

7. Besar keuntungan setiap drug store dan jumlah transaksi penjualan pada tahun 2018

SELECT DISTINCT drug\_store\_name, EXTRACT (YEAR FROM recap.date) AS year, SUM(income) AS income,

COUNT(recap.drug\_store\_id) AS jmlpenjualan

FROM fact\_transaction\_sale\_only\_pharmacy AS recap

INNER JOIN dim\_drug\_store AS ds ON ds.drug\_store\_id = recap.drug\_store\_id

WHERE (SELECT EXTRACT (YEAR FROM recap.date)) = '2018'

GROUP BY drug\_store\_name,year

ORDER BY income DESC;

8. Berapa hari rata-rata obat akan expired saat dijual kepada pelanggan pada tiap bulannya?

SELECT AVG(drug\_expired-date) AS expired FROM fact\_transaction\_sale\_only\_pharmacy AS recap

INNER JOIN dim\_drug as d

ON d.drug\_id=recap.drug\_id;

1. **Strategi dan Query ETL**

**Strategi**

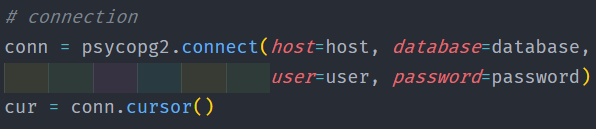
Untuk melakukan ETL (Extract, Transform, Load) dalam menghasilkan database dengan star schema sebagai acuan nya (terdapat fact dan dimension), maka terdapat beberapa strategi yang digunakan:

1. DDL Star Schema
2. CSV data reconcile
3. Procedure Query

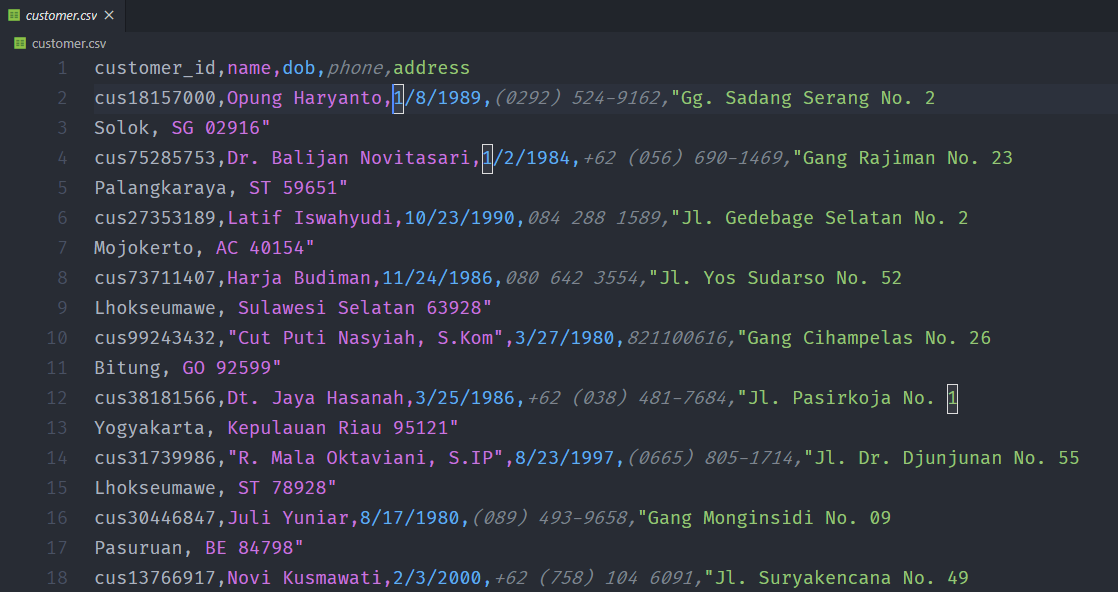
**Query ETL**

1. **Extract dari data source**

****

****

1. **Transform dengan mengubah record data source secara langsung**

****

1. **Load kedalam database**
2. Dimension Customer

sql = "INSERT INTO dim\_customer(customer\_id,customer\_name,customer\_dob,customer\_phone,customer\_address) VALUES (%s,%s,%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Dimension Doctor

sql = "INSERT INTO dim\_doctor(doctor\_id,doctor\_name,doctor\_specialist,doctor\_dob,doctor\_phone,doctor\_rate) VALUES (%s,%s,%s,%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=';')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Dimension Drug Store

sql = "INSERT INTO dim\_drug\_store(drug\_store\_id,drug\_store\_name,drug\_store\_location) VALUES (%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Dimension Drug Supplier

sql = "INSERT INTO dim\_drug\_supplier(idx,drug\_supplier\_id,drug\_supplier\_name,drug\_supplier\_phone,drug\_supplier\_address) VALUES (%s,%s,%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Dimension Drug

sql = "INSERT INTO dim\_drug(drug\_id,drug\_name,drug\_category,drug\_price\_buy,drug\_price\_sell,drug\_expired) VALUES (%s,%s,%s,%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Dimension Patient

sql = "INSERT INTO dim\_patient(patient\_id,patient\_name,patient\_dob,patient\_phone,patient\_address) VALUES (%s,%s,%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Dimension Pharmacist

sql = "INSERT INTO dim\_pharmacist(pharmacist\_id,pharmacist\_name,pharmacist\_address,pharmacist\_dob,pharmacist\_phone,pharmacist\_rate) VALUES (%s,%s,%s,%s,%s,%s)"

*with* open(csvfile) *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=';')

next(data)

*for* i in tqdm(data):

cur.execute(sql, i)

conn.commit()

1. Fact Transaction Sale Only Pharmacy

logging.debug('Start Read Data')

sale\_only = []

*with* open(csvfile, *mode*='r') *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

sale\_only.append(i)

logging.debug('Done Read Data')

logging.debug('Start Insert Data')

*for* row in tqdm(sale\_only):

t\_id = row[0]

d\_id = row[1]

p\_id = row[2]

c\_id = row[3]

ds\_id = row[4]

date = row[5]

drug\_id = row[6]

drug\_qty = row[7]

*# revenue*

cur\_revenue = conn.cursor()

cur\_revenue.execute(

"SELECT %s\*drug\_price\_sell FROM dim\_drug WHERE drug\_id = %s", [drug\_qty, drug\_id])

revenue = float(cur\_revenue.fetchone()[0])

*# expense*

cur\_expense = conn.cursor()

cur\_expense.execute(

"SELECT %s\*drug\_price\_buy FROM dim\_drug WHERE drug\_id = %s", [drug\_qty, drug\_id])

expense = float(cur\_expense.fetchone()[0])

*# income*

income = revenue - expense

sql = "INSERT INTO fact\_transaction\_sale\_only\_pharmacy(transaction\_sale\_only\_pharmacy\_id,doctor\_id,pharmacist\_id,customer\_id,drug\_store\_id,date,drug\_id,drug\_qty,revenue,expense,income) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"

value = (t\_id, d\_id, p\_id, c\_id, ds\_id, date,

drug\_id, drug\_qty, revenue, expense, income)

cur.execute(sql, value)

conn.commit()

logging.debug('Finish!')

1. Fact Transaction Sale Pharmacy

logging.debug('Start Read Data')

sale = []

*with* open(csvfile, *mode*='r') *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

sale.append(i)

logging.debug('Done Read Data')

logging.debug('Start Insert Data')

*for* row in tqdm(sale):

t\_id = row[0]

d\_id = row[1]

phar\_id = row[2]

ds\_id = row[3]

pat\_id = row[4]

date = row[5]

drug\_id = row[6]

drug\_qty = row[7]

*# revenue*

cur\_revenue = conn.cursor()

cur\_revenue.execute(

"SELECT %s\*drug\_price\_sell FROM dim\_drug WHERE drug\_id = %s", [drug\_qty, drug\_id])

revenue = float(cur\_revenue.fetchone()[0])

*# expense*

cur\_expense = conn.cursor()

cur\_expense.execute(

"SELECT %s\*drug\_price\_buy FROM dim\_drug WHERE drug\_id = %s", [drug\_qty, drug\_id])

expense = float(cur\_expense.fetchone()[0])

*# income*

income = revenue - expense

sql = "INSERT INTO fact\_transaction\_sale\_pharmacy(transaction\_sale\_pharmacy\_id,doctor\_id,pharmacist\_id,drug\_store\_id,patient\_id,date,drug\_id,drug\_qty,revenue,expense,income) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"

value = (t\_id, d\_id, phar\_id, ds\_id, pat\_id, date,

drug\_id, drug\_qty, revenue, expense, income)

cur.execute(sql, value)

conn.commit()

logging.debug('Finish!')

1. Fact Transaction Logistic Pharmacy

logging.debug('Start Read Data')

logistic = []

*with* open(csvfile, *mode*='r') *as* csv\_file:

data = csv.reader(csv\_file, *delimiter*=',')

next(data)

*for* i in tqdm(data):

logistic.append(i)

logging.debug('Done Read Data')

logging.debug('Start Insert Data')

*for* row in tqdm(logistic):

t\_id = row[0]

date = row[1]

drug\_supp = row[2]

drug\_id = row[3]

drug\_qty = row[4]

*# revenue*

cur\_revenue = conn.cursor()

cur\_revenue.execute(

"SELECT %s\*drug\_price\_sell FROM dim\_drug WHERE drug\_id = %s", [drug\_qty, drug\_id])

revenue = float(cur\_revenue.fetchone()[0])

*# expense*

cur\_expense = conn.cursor()

cur\_expense.execute(

"SELECT %s\*drug\_price\_buy FROM dim\_drug WHERE drug\_id = %s", [drug\_qty, drug\_id])

expense = float(cur\_expense.fetchone()[0])

*# income*

income = revenue - expense

sql = "INSERT INTO fact\_transaction\_logistic\_pharmacy(transaction\_logistic\_pharmacy\_id,date,drug\_supplier\_id,drug\_id,drug\_qty,revenue,expense,income) VALUES (%s,%s,%s,%s,%s,%s,%s,%s)"

value = (t\_id, date, drug\_supp, drug\_id,

drug\_qty, revenue, expense, income)

cur.execute(sql, value)

conn.commit()

logging.debug('Finish!')