

[Task 5] Data Modeling, Warehouse Foundation and SQL Operation

Kalbe Nutritionals - DBA

Presented by
Raja Alamsyah Tahir



Raja Alamsyah Tahir

As a recent graduate in Computer Engineering with a deep interest in Data Analysis and Database Administration, I am actively seeking opportunities to further develop my skills and knowledge in this field.

My Experience

ETL Developer Intern – PT Bank Tabungan Negera (Persero) Tbk

- Revamped the Extract, Transform, Load (ETL) infrastructure by leveraging IBM DataStage data integration for seamless integration of heterogeneous data sources.

Data Analyst Intern – ICT IPB University

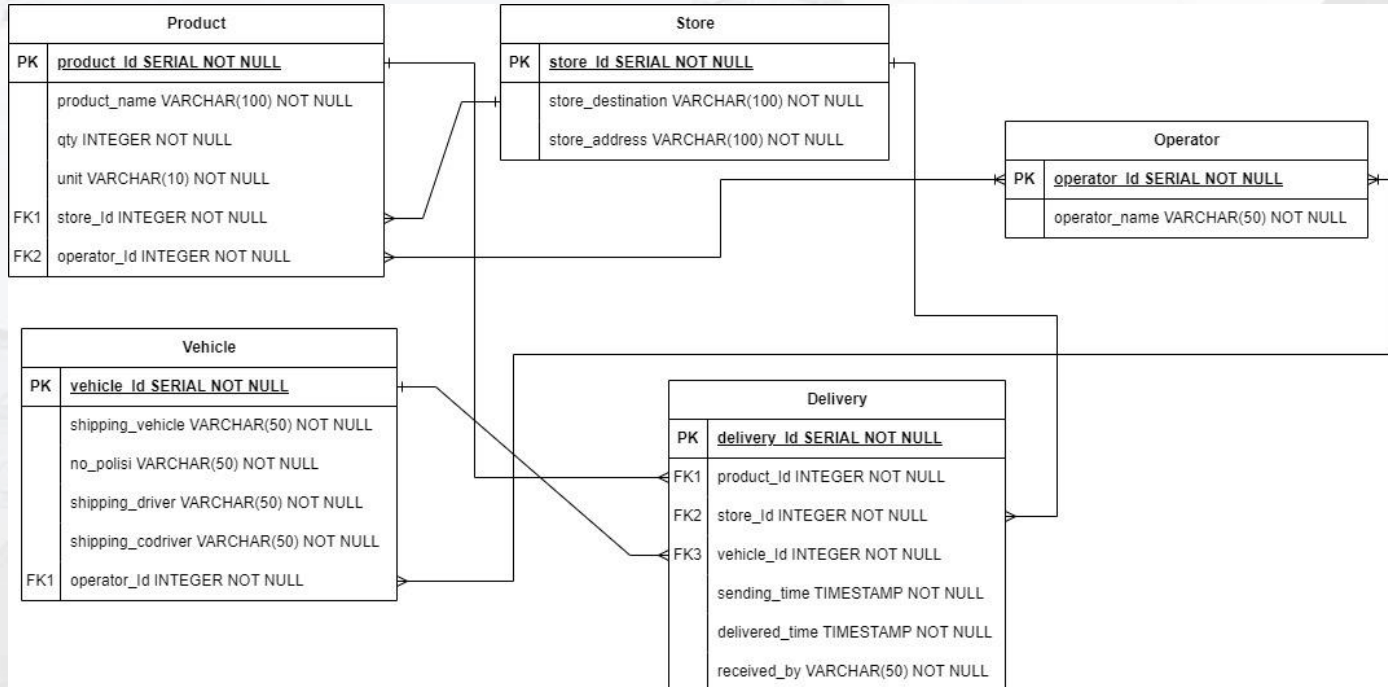
- Collaborated proactively with the operations team to identify and implement potential improvements that simplified the administration process and increased efficiency.

Case Study

1. Lakukanlah normalisasi dari hasil pencatatan distribusi barang tersebut
2. Buatlah ERD berdasarkan hasil dari normalisasi data distribusi barang
 - Dapat menggunakan notasi Chen atau pun Crow's foot
3. Buatlah database dan struktur table & relasi menggunakan RDBMS PostgreSQL
 - Buatlah struktur table tersebut di dalam schema yang bernama app
 - Untuk data produk, silahkan import dari file lampiran kedua
 - Buatlah sebuah user yang akan digunakan oleh backend programmer untuk melakukan operasi database dengan akses hanya dapat melakukan DML (INSERT, UPDATE, DELETE, SELECT)
 - Buatlah index di dalam table sesuai kebutuhan untuk mengoptimalkan query

Result

Membuat Normalisasi dan ERD Crow's Foot pada Distribusi Barang



Membuat tabel distribusi barang dan import data menggunakan file csv

```
CREATE TABLE IF NOT EXISTS public.distribusi_barang (  
  No INTEGER,  
  Product_Name VARCHAR(100),  
  Qty INTEGER,  
  Unit VARCHAR(10),  
  Store_Destination VARCHAR(100),  
  Store_Address VARCHAR(100),  
  Operator_Name VARCHAR(50),  
  Shipping_Vehicle VARCHAR(50),  
  No_Polisi VARCHAR(50),  
  Shipping_Driver VARCHAR(50),  
  Shipping_codriver VARCHAR(50),  
  Sending_Time TIMESTAMP WITHOUT TIME ZONE,  
  Delivered_Time TIMESTAMP WITHOUT TIME ZONE,  
  Received_By VARCHAR(50)  
);
```

| | no integer | product_name character varying (100) | qty integer | unit character varying (10) | store_destination character varying (100) |
|---|---------------|---|----------------|--------------------------------|--|
| 1 | 1 | Hydro Coco 250ml | 5 | box | Apotek Agus Sari |
| 2 | 2 | Hydro Coco Vita-D 330ml | 5 | box | Apotek Agus Sari |
| 3 | 3 | Milna Biskuit Bayi Apel | 10 | box | Apotek Agus Sari |
| 4 | 4 | Hydro Coco 250ml | 3 | box | Toko Maju Bersama |
| 5 | 5 | Hydro Coco 330ml | 2 | box | Toko Maju Bersama |

```
copy distribusi_barang from  
'E:\Daftar Kerja\Magang\Rakamin\Database Administrator - Kalbe\distribusi_barang.csv'  
(format csv, null "NULL", DELIMITER ';', HEADER);
```

```
SELECT *  
FROM distribusi_barang;
```


Membuat tabel data_produk dan import data menggunakan file csv

```
CREATE TABLE IF NOT EXISTS public.data_produk (  
  No INTEGER,  
  Name VARCHAR(100)  
);  
  
copy data_produk from  
'E:\Daftar Kerja\Magang\Rakamin\Database Administrator - Kalbe\data_produk.csv'  
(format csv, null "NULL", DELIMITER ';', HEADER);  
  
SELECT *  
FROM data_produk;
```

| | no integer | name character varying (100) |
|---|---------------|---------------------------------|
| 1 | 1 | Hydro Coco 250ml |
| 2 | 2 | Hydro Coco 330ml |
| 3 | 3 | Hydro Coco 500ml |
| 4 | 4 | Hydro Coco 1 liter |
| 5 | 5 | Hydro Coco Vita-D 330ml |
| 6 | 6 | Milna Biskuit Bayi Original |
| 7 | 7 | Milna Biskuit Bayi Beras Merah |
| 8 | 8 | Milna Biskuit Bayi Kacang Hijau |

Membuat schema app dan tabel hasil normalisasi distribusi barang

```
CREATE TABLE app.product (  
  product_id SERIAL PRIMARY KEY,  
  product_name VARCHAR(100) NOT NULL,  
  qty INTEGER NOT NULL,  
  unit VARCHAR(10) NOT NULL,  
  store_id INTEGER NOT NULL,  
  operator_id INTEGER NOT NULL  
);  
  
CREATE TABLE app.store (  
  store_id SERIAL PRIMARY KEY,  
  store_destination VARCHAR(100) NOT NULL,  
  store_address VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE app.operator (  
  operator_id SERIAL PRIMARY KEY,  
  operator_name VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE app.vehicle (  
  vehicle_id SERIAL PRIMARY KEY,  
  shipping_vehicle VARCHAR(50) NOT NULL,  
  no_polisi VARCHAR(50) NOT NULL,  
  shipping_driver VARCHAR(50) NOT NULL,  
  shipping_codriver VARCHAR(50) NOT NULL,  
  operator_id INTEGER NOT NULL  
);  
  
CREATE TABLE app.delivery (  
  delivery_id SERIAL PRIMARY KEY,  
  product_id INTEGER NOT NULL,  
  store_id INTEGER NOT NULL,  
  vehicle_id INTEGER NOT NULL,  
  sending_time TIMESTAMP WITHOUT TIME ZONE NOT NULL,  
  delivered_time TIMESTAMP WITHOUT TIME ZONE NOT NULL,  
  received_by VARCHAR(50) NOT NULL  
);
```

Membuat Foreign Key pada tiap tabel

```
ALTER TABLE app.product  
ADD CONSTRAINT FK_product_storeid  
FOREIGN KEY (store_id)  
REFERENCES app.store(store_id);
```

```
ALTER TABLE app.product  
ADD CONSTRAINT FK_product_operatorid  
FOREIGN KEY (operator_id)  
REFERENCES app.operator(operator_id);
```

```
ALTER TABLE app.vehicle  
ADD CONSTRAINT FK_vehicle_operatorid  
FOREIGN KEY (operator_id)  
REFERENCES app.operator(operator_id);
```

```
ALTER TABLE app.delivery  
ADD CONSTRAINT FK_delivery_product  
FOREIGN KEY (product_id)  
REFERENCES app.product(product_id);
```

```
ALTER TABLE app.delivery  
ADD CONSTRAINT FK_delivery_storeid  
FOREIGN KEY (store_id)  
REFERENCES app.store(store_id);
```

```
ALTER TABLE app.delivery  
ADD CONSTRAINT FK_delivery_vehicleid  
FOREIGN KEY (vehicle_id)  
REFERENCES app.vehicle(vehicle_id);
```


Mengambil data dari public.distribusi_barang

```
INSERT INTO app.store (store_destination, store_address)
SELECT DISTINCT store_destination, store_address
FROM public.distribusi_barang
ORDER BY store_destination;
```

```
SELECT *
FROM app.store;
```

```
INSERT INTO app.operator (operator_name)
SELECT DISTINCT operator_name
FROM public.distribusi_barang
ORDER BY operator_name;
```

```
SELECT *
FROM app.operator;
```

| | store_id [PK] integer | store_destination character varying (100) | store_address character varying (100) |
|---|--------------------------|--|--|
| 1 | 1 | Apotek Agung | Pasar Senen no 301 |
| 2 | 2 | Apotek Agus Sari | Jln Angga Jaya no 21 |
| 3 | 3 | Toko Anak Sehat | Jln Imam Bonjol no 33 |
| 4 | 4 | Toko Maju Bersama | Jln Agus Salim no 22 |

| | operator_id [PK] integer | operator_name character varying (50) |
|---|-----------------------------|---|
| 1 | 1 | Ahmad Agus |
| 2 | 2 | Fitrianto |

Mengambil data dari public.distribusi_barang dan tabel lain

```
INSERT INTO app.product (product_name, qty, unit, store_id, operator_id)
SELECT DISTINCT db.product_name, db.qty, db.unit, s.store_id, o.operator_id
FROM public.distribusi_barang db
INNER JOIN app.store s
    ON db.store_destination = s.store_destination
INNER JOIN app.operator o
    ON db.operator_name = o.operator_name
ORDER BY s.store_id, o.operator_id;

SELECT *
FROM app.product;
```

| | product_id [PK] integer | product_name character varying (100) | qty integer | unit character varying (10) | store_id integer | operator_id integer |
|----|----------------------------|---|----------------|--------------------------------|---------------------|------------------------|
| 1 | 1 | Entrasol Active Vanilla latte | 5 | box | 1 | 2 |
| 2 | 2 | Entrasol Gold Chocolate | 6 | box | 1 | 2 |
| 3 | 3 | Entrasol Gold Original | 4 | box | 1 | 2 |
| 4 | 4 | Hydro Coco 250ml | 5 | box | 2 | 1 |
| 5 | 5 | Hydro Coco Vita-D 330ml | 5 | box | 2 | 1 |
| 6 | 6 | Milna Biskuit Bayi Apel | 10 | box | 2 | 1 |
| 7 | 7 | Milna Bubur Organik Multigrain | 10 | box | 3 | 2 |
| 8 | 8 | Milna Nature Delight Apel Labu Wortel | 5 | box | 3 | 2 |
| 9 | 9 | Milna Nature Delight Apel Peach | 10 | box | 3 | 2 |
| 10 | 10 | Milna Rice Crackers Apple Orange | 12 | box | 3 | 2 |
| 11 | 11 | Entrasol Gold Original | 8 | box | 4 | 1 |
| 12 | 12 | Hydro Coco 250ml | 3 | box | 4 | 1 |
| 13 | 13 | Hydro Coco 330ml | 2 | box | 4 | 1 |

Mengambil data dari public.distribusi_barang dan tabel lain

```
INSERT INTO app.vehicle (shipping_vehicle, no_polisi, shipping_driver, shipping_codriver, operator_id)
SELECT DISTINCT db.shipping_vehicle, db.no_polisi, db.shipping_driver, db.shipping_codriver, o.operator_id
FROM public.distribusi_barang db
INNER JOIN app.operator o
    ON db.operator_name = o.operator_name
ORDER BY db.shipping_vehicle, operator_id;

SELECT *
FROM app.vehicle;
```

| | vehicle_id [PK] integer | shipping_vehicle character varying (50) | no_polisi character varying (50) | shipping_driver character varying (50) | shipping_codriver character varying (50) | operator_id integer |
|---|----------------------------|--|-------------------------------------|---|---|------------------------|
| 1 | 1 | Box A001 | B 1234 GA | Dimas Ahmad | Andi Wahyu | 1 |
| 2 | 2 | Box A001 | B 1234 GA | Ginanjari | Hari Saputra | 2 |
| 3 | 3 | Box A002 | B 3214 JS | Hari Saputra | Dadang Bima | 1 |

Mengambil data dari public.distribusi_barang dan tabel lain

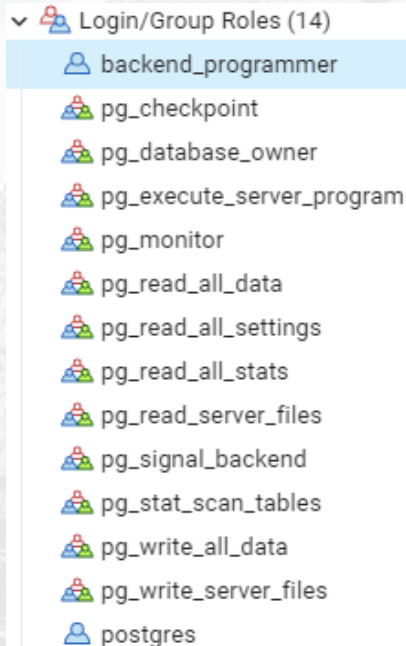
```
INSERT INTO app.delivery (product_id, store_id, vehicle_id, sending_time, delivered_time, received_by)
SELECT p.product_id, s.store_id, v.vehicle_id, db.sending_time, db.delivered_time, db.received_by
FROM public.distribusi_barang db
INNER JOIN app.product p
    ON db.product_name = p.product_name
INNER JOIN app.store s
    ON db.store_destination = s.store_destination
INNER JOIN app.vehicle v
    ON db.shipping_vehicle = v.shipping_vehicle
ORDER BY p.product_id, s.store_id, v.vehicle_id;

SELECT *
FROM app.delivery;
```

| | delivery_id [PK] integer | product_id integer | store_id integer | vehicle_id integer | sending_time timestamp without time zone | delivered_time timestamp without time zone | received_by character varying (50) |
|----|-----------------------------|-----------------------|---------------------|-----------------------|---|---|---------------------------------------|
| 1 | 1 | 1 | 1 | 1 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 2 | 2 | 1 | 1 | 1 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 3 | 3 | 1 | 1 | 2 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 4 | 4 | 1 | 1 | 2 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 5 | 5 | 2 | 1 | 1 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 6 | 6 | 2 | 1 | 1 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 7 | 7 | 2 | 1 | 2 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 8 | 8 | 2 | 1 | 2 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 9 | 9 | 3 | 1 | 1 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 10 | 10 | 3 | 1 | 1 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 11 | 11 | 3 | 1 | 2 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 12 | 12 | 3 | 1 | 2 | 2023-05-02 09:00:00 | 2023-05-02 14:00:00 | Jamal |
| 13 | 13 | 3 | 4 | 3 | 2023-05-01 11:00:00 | 2023-05-01 13:00:00 | Eriawan |
| 14 | 14 | 3 | 4 | 3 | 2023-05-01 11:00:00 | 2023-05-01 13:00:00 | Eriawan |
| 15 | 15 | 4 | 2 | 1 | 2023-05-01 10:00:00 | 2023-05-01 13:30:00 | Dian Ayu |

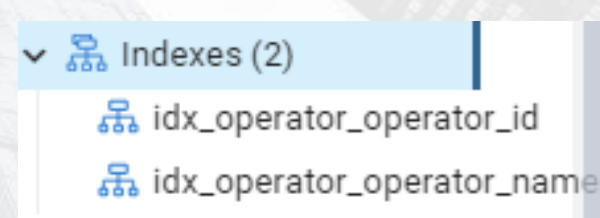
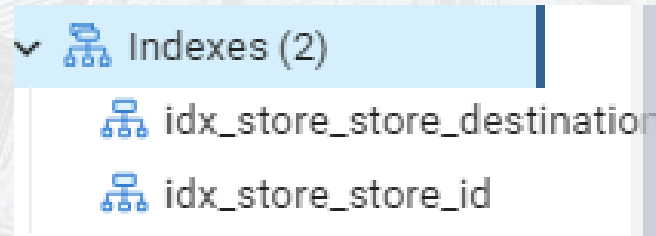
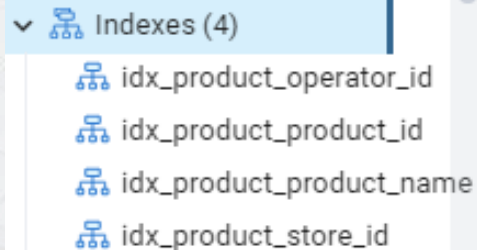
Membuat users untuk backend programmer

```
CREATE USER backend_programmer WITH ENCRYPTED PASSWORD 'kalbe123';  
GRANT CONNECT ON DATABASE app TO backend_programmer;  
GRANT USAGE ON SCHEMA app TO backend_programmer;  
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA app TO backend_programmer;
```



Membuat index pada tiap tabel

```
CREATE INDEX idx_product_product_id ON app.product (product_id);  
CREATE INDEX idx_product_product_name ON app.product (product_name);  
CREATE INDEX idx_product_store_id ON app.product (store_id);  
CREATE INDEX idx_product_operator_id ON app.product (operator_id);  
  
CREATE INDEX idx_store_store_id ON app.store (store_id);  
CREATE INDEX idx_store_store_destination ON app.store (store_destination);  
  
CREATE INDEX idx_operator_operator_id ON app.operator (operator_id);  
CREATE INDEX idx_operator_operator_name ON app.operator (operator_name);
```



Membuat index pada tiap tabel

```
CREATE INDEX idx_vehicle_vehicle_id ON app.vehicle (vehicle_id);  
CREATE INDEX idx_vehicle_shipping_vehicle ON app.vehicle (shipping_vehicle);  
CREATE INDEX idx_vehicle_operator_id ON app.vehicle (operator_id);  
  
CREATE INDEX idx_delivery_delivery_id ON app.delivery (delivery_id);  
CREATE INDEX idx_delivery_product_id ON app.delivery (product_id);  
CREATE INDEX idx_delivery_store_id ON app.delivery (store_id);  
CREATE INDEX idx_delivery_vehicle_id ON app.delivery (vehicle_id);  
CREATE INDEX idx_delivery_received_by ON app.delivery (received_by);
```

Indexes (3)

- idx_vehicle_operator_id
- idx_vehicle_shipping_vehicle
- idx_vehicle_vehicle_id

Indexes (5)

- idx_delivery_delivery_id
- idx_delivery_product_id
- idx_delivery_received_by
- idx_delivery_store_id
- idx_delivery_vehicle_id

Case Study


4. Buatlah query untuk kebutuhan - kebutuhan di bawah ini
 - Menampilkan 2 driver dengan pengiriman terbanyak bulan Mei 2023
 - Menampilkan 10 barang paling sering dikirim di bulan Mei 2023
 - Menampilkan semua pengiriman yang belum selesai
5. Buatlah sebuah user defined function
 - Untuk membuat ID Shipment dengan format yymmddxxx (contoh: 230519001, 230519002)
6. Buatlah 2 buah stored procedure
 - Untuk membuat shipment baru
 - Untuk menambahkan product ke dalam shipment
7. Buatlah Daily Backup
 - Buatlah task / job untuk melakukan backup database pukul 23:00 setiap hari Daftarkan database tersebut ke dalam SolarWinds DPA
 - Buatlah alert ketika suatu query memiliki total waiting time sebesar 10 detik atau lebih

Result

Menampilkan 2 driver dengan pengiriman terbanyak bulan Mei 2023

```
--Showing the 2 drivers with the most deliveries of May 2023
```

```
SELECT shipping_driver, COUNT(*) AS total_shipment  
FROM public.distribusi_barang  
WHERE EXTRACT(MONTH FROM delivered_time) = 5 AND EXTRACT(YEAR FROM delivered_time) = 2023  
GROUP BY shipping_driver  
ORDER BY total_shipment DESC  
LIMIT 2;
```

| | shipping_driver  | total_shipment  |
|---|---|--|
| | character varying (50) | bigint |
| 1 | GINANJAR | 14 |
| 2 | DIMAS AHMAD | 6 |

Menampilkan 10 barang paling sering dikirim di bulan Mei 2023

```
SELECT product_name, SUM(qty) AS total_qty  
FROM public.distribusi_barang  
WHERE EXTRACT(MONTH FROM delivered_time) = 5 AND EXTRACT(YEAR FROM delivered_time) = 2023  
GROUP BY product_name  
ORDER BY total_qty DESC  
LIMIT 10;
```

| | product_name character varying (100) | total_qty bigint |
|----|---|---------------------|
| 1 | Entrasol Gold Original | 24 |
| 2 | Milna Rice Crackers Apple Orange | 24 |
| 3 | Milna Biskuit Bayi Apel | 20 |
| 4 | Milna Nature Delight Apel Peach | 20 |
| 5 | Milna Bubur Organik Multigrain | 20 |
| 6 | Hydro Coco 250ml | 16 |
| 7 | Entrasol Gold Chocolate | 12 |
| 8 | Hydro Coco Vita-D 330ml | 10 |
| 9 | Entrasol Active Vanilla latte | 10 |
| 10 | Milna Nature Delight Apel Labu Wortel | 10 |


Menampilkan 10 barang paling sering dikirim di bulan Mei 2023

```
SELECT *  
FROM public.distribusi_barang  
WHERE delivered_time IS NULL;
```

| | | | | | | |
|---------------|---|----------------|--------------------------------|--|--|---|
| no integer | product_name character varying (100) | qty integer | unit character varying (10) | store_destination character varying (100) | store_address character varying (100) | operator_name character varying (50) |
|---------------|---|----------------|--------------------------------|--|--|---|

Membuat ID Shipment dengan format yymmddxxx (contoh: 230519001, 230519002)

```
SELECT TO_CHAR(sending_time, 'YYMMDD') || LPAD(ROW_NUMBER() OVER  
      (ORDER BY sending_time)::TEXT, 3, '0') AS shipment_id  
FROM public.distribusi_barang;
```

| | shipment_id  |
|----|---|
| 1 | 230501001 |
| 2 | 230501002 |
| 3 | 230501003 |
| 4 | 230501004 |
| 5 | 230501005 |
| 6 | 230501006 |
| 7 | 230501007 |
| 8 | 230501008 |
| 9 | 230501009 |
| 10 | 230501010 |

Membuat shipment baru dan menambahkan product ke dalam shipment

```
CREATE OR REPLACE PROCEDURE sp_create_shipment(  
    IN product_name VARCHAR(100),  
    IN qty INTEGER,  
    IN unit VARCHAR(10),  
    IN store_destination VARCHAR(100),  
    IN store_address VARCHAR(100),  
    IN operator_name VARCHAR(50),  
    IN shipping_vehicle VARCHAR(50),  
    IN no_polisi VARCHAR(50),  
    IN shipping_driver VARCHAR(50),  
    IN shipping_codriver VARCHAR(50),  
    IN sending_time TIMESTAMP WITHOUT TIME ZONE,  
    IN delivered_time TIMESTAMP WITHOUT TIME ZONE,  
    IN received_by VARCHAR(50)  
)  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    INSERT INTO public.distribusi_barang ("product_name", "qty", "unit", "store_destination",  
        "store_address", "operator_name", "shipping_vehicle", "no_polisi", "shipping_driver",  
        "shipping_codriver", "sending_time", "delivered_time", "received_by")  
    VALUES (product_name, qty, unit, store_destination, store_address, operator_name,  
        shipping_vehicle, no_polisi, shipping_driver, shipping_codriver, sending_time,  
        delivered_time, received_by);  
END;  
$$;
```

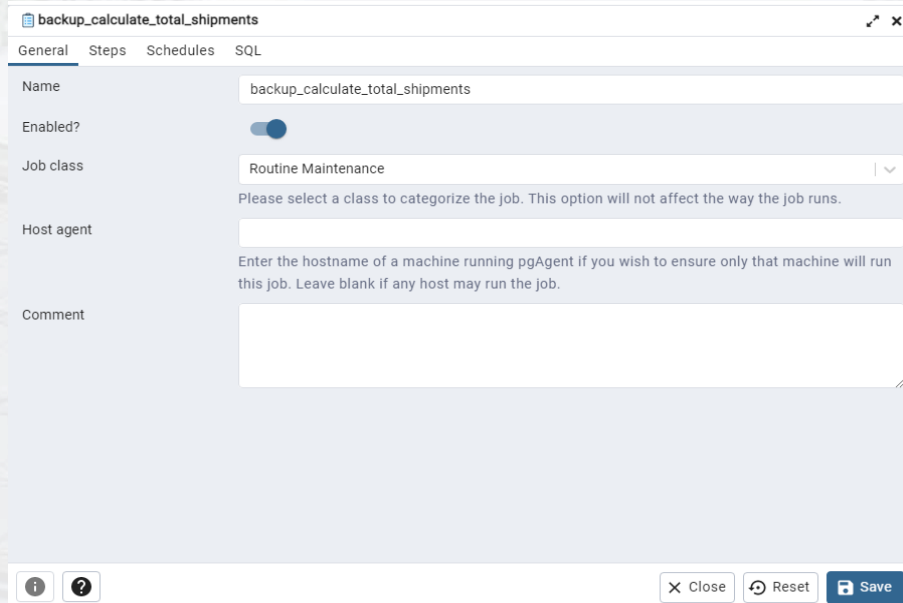
```
CREATE OR REPLACE PROCEDURE sp_add_product_to_shipment(  
    IN shipment_id bigint,  
    IN product_name character varying,  
    IN qty integer,  
    IN unit character varying  
)  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    INSERT INTO app.distribusi_barang ("shipment_id", "product_name", "qty", "unit")  
    VALUES (shipment_id, product_name, qty, unit);  
END;  
$$;
```

Procedures (2)

{ } sp_add_product_to_shipment(IN shipment

{ } sp_create_shipment(IN product_name cha

Buatlah task / job untuk melakukan backup database pukul 23:00 setiap hari Daftarkan database tersebut ke dalam SolarWinds DPA

A screenshot of the SolarWinds DPA configuration window for a job named "backup_calculate_total_shipments". The window has tabs for General, Steps, Schedules, and SQL. The General tab is active, showing fields for Name, Enabled?, Job class, Host agent, and Comment. The Name field contains "backup_calculate_total_shipments". The Enabled? toggle is turned on. The Job class is set to "Routine Maintenance". The Host agent field is empty. The Comment field is a large text area. At the bottom, there are buttons for Close, Reset, and Save.

backup_calculate_total_shipments

General Steps Schedules SQL

Name: backup_calculate_total_shipments

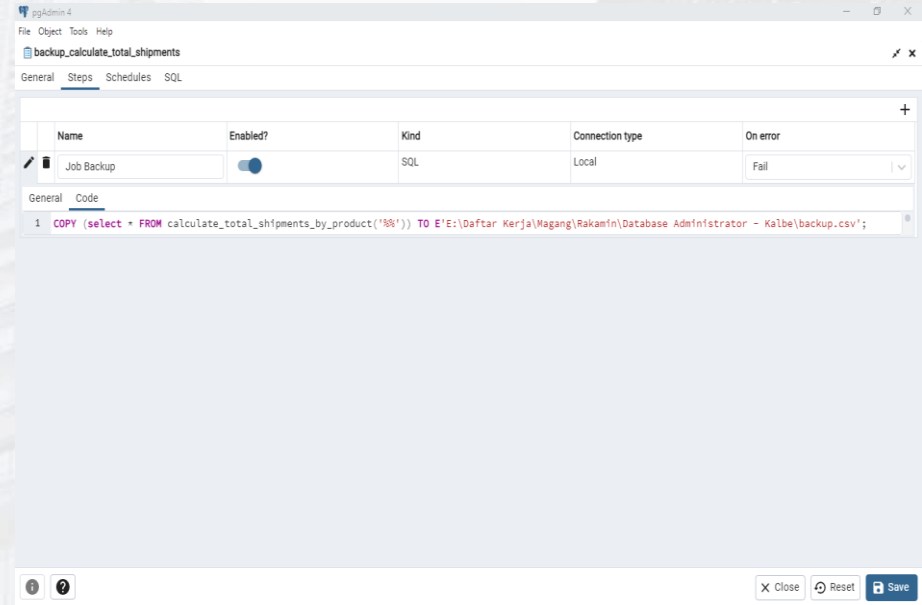
Enabled?: ☒

Job class: Routine Maintenance

Host agent:

Comment:

Close Reset Save

A screenshot of the pgAdmin 4 configuration window for a job named "backup_calculate_total_shipments". The window has tabs for General, Steps, Schedules, and SQL. The General tab is active, showing a table of jobs and a Code tab with a SQL query. The table has columns for Name, Enabled?, Kind, Connection type, and On error. The job "Job Backup" is listed with Kind "SQL", Connection type "Local", and On error "Fail". The Code tab shows a SQL query: "COPY (select * FROM calculate_total_shipments_by_product('%')) TO E:\Daftar Kerja\Wagang\Rakamin\Database Administrator - Kalbe\backup.csv;". At the bottom, there are buttons for Close, Reset, and Save.

pgAdmin 4

File Object Tools Help

backup_calculate_total_shipments

General Steps Schedules SQL

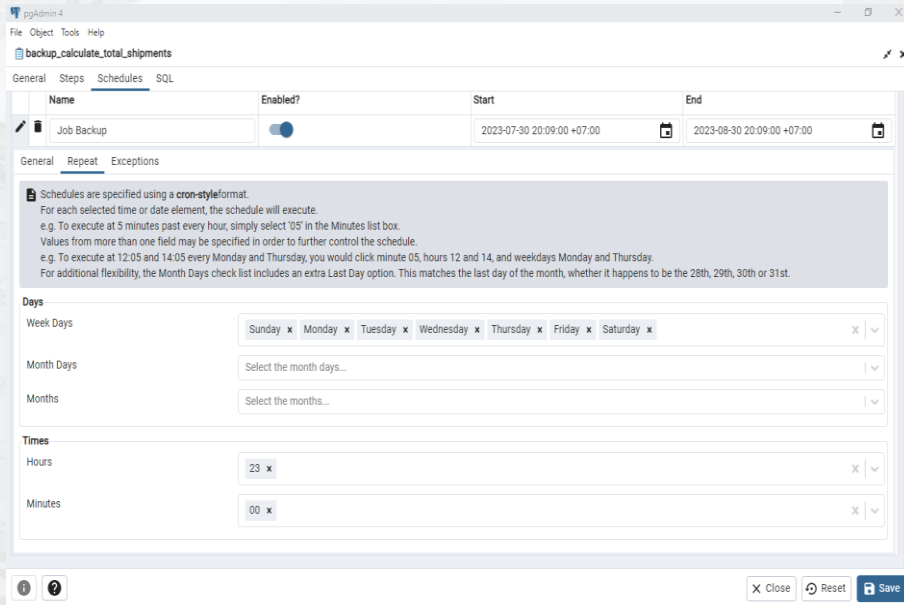
| Name | Enabled? | Kind | Connection type | On error |
|------------|-------------------------------------|------|-----------------|----------|
| Job Backup | <input checked="" type="checkbox"/> | SQL | Local | Fail |

General Code

```
1 COPY (select * FROM calculate_total_shipments_by_product('%')) TO E:\Daftar Kerja\Wagang\Rakamin\Database Administrator - Kalbe\backup.csv;
```

Close Reset Save

Buatlah task / job untuk melakukan backup database pukul 23:00 setiap hari menggunakan PG Agent



pgAdmin 4

File Object Tools Help

backup_calculate_total_shipments

General Steps Schedules SQL

| Name | Enabled? | Start | End |
|------------|-------------------------------------|----------------------------|----------------------------|
| Job Backup | <input checked="" type="checkbox"/> | 2023-07-30 20:09:00 +07:00 | 2023-08-30 20:09:00 +07:00 |

General Repeat Exceptions

Schedules are specified using a cron-style format.
For each selected time or date element, the schedule will execute.
e.g. To execute at 5 minutes past every hour, simply select '05' in the Minutes list box.
Values from more than one field may be specified in order to further control the schedule.
e.g. To execute at 12:05 and 14:05 every Monday and Thursday, you would click minute 05, hours 12 and 14, and weekdays Monday and Thursday.
For additional flexibility, the Month Days check list includes an extra Last Day option. This matches the last day of the month, whether it happens to be the 28th, 29th, 30th or 31st.

Days

Week Days: Sunday x Monday x Tuesday x Wednesday x Thursday x Friday x Saturday x

Month Days: Select the month days...

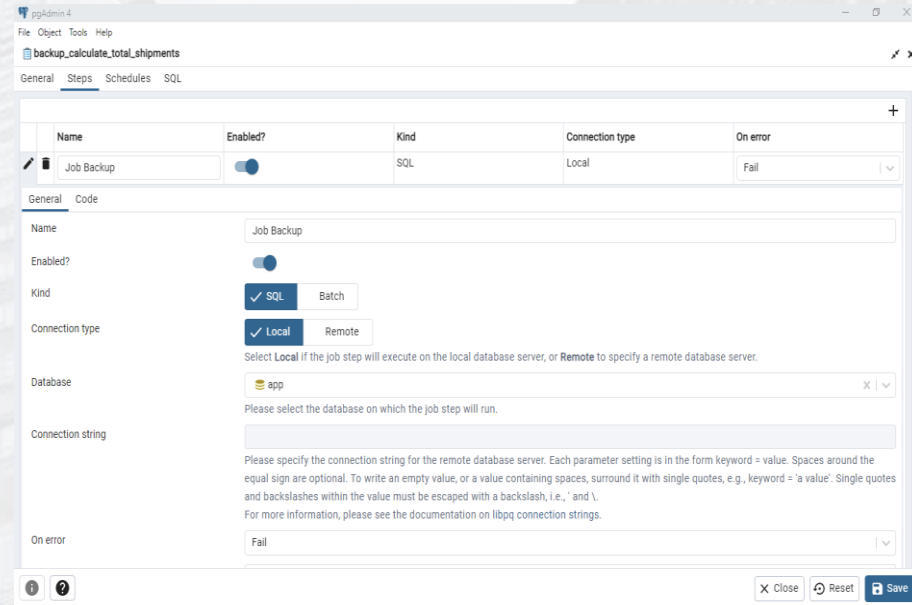
Months: Select the months...

Times

Hours: 23 x

Minutes: 00 x

X Close Reset Save



pgAdmin 4

File Object Tools Help

backup_calculate_total_shipments

General Steps Schedules SQL

| Name | Enabled? | Kind | Connection type | On error |
|------------|-------------------------------------|------|-----------------|----------|
| Job Backup | <input checked="" type="checkbox"/> | SQL | Local | Fail |

General Code

Name: Job Backup

Enabled?: ☒

Kind: ☒ SQL ☐ Batch

Connection type: ☒ Local ☐ Remote

Select Local if the job step will execute on the local database server, or Remote to specify a remote database server.

Database: app

Please select the database on which the job step will run.

Connection string:

Please specify the connection string for the remote database server. Each parameter setting is in the form keyword = value. Spaces around the equal sign are optional. To write an empty value, or a value containing spaces, surround it with single quotes, e.g., keyword = 'a value'. Single quotes and backslashes within the value must be escaped with a backslash, i.e., ' and \.

For more information, please see the documentation on libpq connection strings.

On error: Fail

X Close Reset Save

Close Reset Save

Link Github

https://github.com/rajaalamsyah85/FinalTask_Kalbe_DBA

Video Presentation

<https://drive.google.com/drive/folders/1U4ayb6m6JmITsZPEBUlyCjmHptAYAKa6?hl=id>

Thank You



Rakamin
Academy



KALBE
Nutritional