



## เรื่อง การออกแบบคลังข้อมูล (Data Warehouse Design)

จัดทำโดย

นาย กิตติพงศ์ พวงสินธ์ รหัสนักศึกษา 66070016

นาย ศุภณัฐ จันทรสาขา รหัสนักศึกษา 66070196

นาย สิริภพ สรรค์ศิลา รหัสนักศึกษา 66070204

นาย อีรศานต์ ชูเชิด รหัสนักศึกษา 66070274

นาย วรวิทย์ มหาทอง รหัสนักศึกษา 66070307

เสนอ

อาจารย์ เฉลิมพล ศิริกายน

คณะเทคโนโลยีสารสนเทศ สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ

รายงานนี้เป็นส่วนหนึ่งของวิชา 06026212 DATA WAREHOUSING

ภาคเรียนที่ 1 ปีการศึกษา 2568

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## สารบัญ

เรื่อง	หน้า
<b>1. บทนำ</b>	
- อธิบายบริบทของหน่วยงาน	1
- ปัญหาทางข้อมูลที่พบ	1
- วัตถุประสงค์ของระบบคลังข้อมูล	2
<b>2. การวิเคราะห์ความต้องการ</b>	
- แหล่งข้อมูลต้นทาง (Source Systems)	3
- ผู้ใช้งานเป้าหมาย	3
- ตาราง BUS Matrix	4
- ข้อจำกัดหรือเงื่อนไขเฉพาะของหน่วยงาน	4
<b>3. การออกแบบคลังข้อมูล</b>	
- ฐานข้อมูลที่ใช้	5
- โครงสร้าง Star Schema	5
- รายละเอียด Fact Table และ Dimension	6
- การจัดการ Slowly Changing Dimension	13
- ER Diagram และ Data Dictionary	15
<b>4. การออกกระบวนการ ETL/ELT ด้วย Apache NiFi</b>	
- แผนภาพ Flow ของ NiFi และรายละเอียดการกำหนดค่าต่างๆ	21
<b>5. การสร้าง Dashboard</b>	
- เครื่องมือที่ใช้	44
- ตัวอย่าง Dash board	44
- การเชื่อมโยงกับ KPI	45
<b>6. บทสรุปและข้อเสนอแนะ</b>	
- บทสรุปของการทำ และ ส่วนเสนอแนะ	47

## บทนำ

### บทเกริ่นนำ

ปัจจุบัน Dairy King มีการเติบโตอย่างต่อเนื่อง แต่การบริหารจัดการข้อมูลยังคงจัดกระจายอยู่ในระบบปฏิบัติการ (OLTP) ทำให้การวิเคราะห์เชิงลึกและภาพรวมองค์กรเป็นไปอย่างล่าช้าและขาดความสอดคล้อง โครงการคลังข้อมูล (Data Warehouse) มีวัตถุประสงค์เพื่อรวมศูนย์และจัดระเบียบข้อมูลเพื่อสนับสนุนการตัดสินใจเชิงกลยุทธ์และการวิเคราะห์พฤติกรรมลูกค้าได้อย่างรวดเร็วและแม่นยำ

### บริบทของหน่วยงาน

**ชื่อหน่วยงาน:** Dairy King

**ลักษณะธุรกิจ:** ธุรกิจค้าปลีกอาหาร เน้นผลิตภัณฑ์ไอศกรีมและเครื่องดื่ม

**โครงสร้างองค์กร:** มีการดำเนินงานแบบหลายสาขา โดยมีระบบงานสนับสนุนหลักแยกกัน

**แหล่งข้อมูลต้นทาง:** 1. ระบบ ณ จุดขาย 2. ระบบจัดการสินค้าคงคลัง 3. ระบบความสัมพันธ์ลูกค้า/การตลาด

**เป้าหมายทางธุรกิจ** เพิ่มยอดขายและกำไร, ปรับปรุงประสิทธิภาพการดำเนินงานในสาขา, สร้างความภักดีของลูกค้า

### ปัญหาของข้อมูลในปัจจุบัน

#### 1 ความไม่สอดคล้องและการจัดกระจายของข้อมูล

- 1.1 ข้อมูลธุรกรรมและข้อมูลสินค้าคงคลังถูกจัดเก็บแยกส่วน ทำให้ยากต่อการเชื่อมโยงเพื่อหาความสัมพันธ์ระหว่างการขายกับระดับสต็อก
- 1.2 นิยามข้อมูล ไม่เป็นมาตรฐานเดียวกัน เช่น การคำนวณ ต้นทุนสินค้า อาจแตกต่างกันระหว่างสาขาและฝ่ายบัญชี

## 2 ข้อจำกัดในการวิเคราะห์เชิงลึก

- 2.1 ระบบ POS เน้นการประมวลผลธุรกรรม ทำให้ไม่สามารถตอบคำถามเชิงธุรกิจที่ซับซ้อน เช่น การวิเคราะห์ตะกร้าสินค้า หรือการวิเคราะห์ยอดขายตามช่วงเวลาและภูมิศาสตร์
- 2.2 การดึงรายงานขนาดใหญ่ เช่น ข้อมูลย้อนหลัง 3 ปีรวมทุกสาขา ทำให้ระบบปฏิบัติการทำงานช้า

## 3 ปัญหาด้านคุณภาพและความน่าเชื่อถือ

พบข้อมูลซ้ำซ้อนหรือข้อผิดพลาดในการบันทึกข้อมูลลูกค้า ทำให้การทำ Segmentation และการตลาดแบบส่วนบุคคลทำได้ยาก

## วัตถุประสงค์ของการทำคลังข้อมูล

การพัฒนา DW มุ่งเน้นไปที่การตอบสนองความต้องการด้าน การวิเคราะห์เชิงธุรกิจ เพื่อให้ Dairy King สามารถแข่งขันในตลาดได้ดียิ่งขึ้น

### 1 วัตถุประสงค์หลัก

- 1.1 Single Source of Truth ข้อมูลรายงานทางการตลาดและบัญชีมีความสอดคล้อง
- 1.2 สนับสนุนการตัดสินใจ ลดเวลาจัดทำรายงานสำคัญจาก 1 วัน เหลือไม่เกิน 1 ชั่วโมง

### 2 วัตถุประสงค์เชิงกลยุทธ์

- 2.1 การเข้าใจลูกค้า วิเคราะห์พฤติกรรมการซื้อ เพื่อสร้างแคมเปญการตลาดแบบเจาะจงบุคคล และเพิ่มมูลค่าตลอดช่วงชีวิตลูกค้า
- 2.2 การเพิ่มประสิทธิภาพการดำเนินงาน วิเคราะห์ความแม่นยำของการพยากรณ์ความต้องการ เพื่อปรับปรุงการจัดการสินค้าคงคลังและลดการสูญเสีย
- 2.3 การวิเคราะห์ประสิทธิภาพสาขา เปรียบเทียบผลการดำเนินงานของแต่ละสาขาและภูมิภาค เพื่อระบุแนวทางปฏิบัติที่ดีที่สุด

## การวิเคราะห์ความต้องการ

### แหล่งข้อมูลต้นทาง (Source Systems)

ระบบต้นทาง	ลักษณะข้อมูล	คอลัมน์/ข้อมูลสำคัญในไฟล์ที่นำเข้า DW
1. ระบบ ณ จุดขาย (POS System)	ธุรกรรมการขาย	sale_id, order_date, quantity, amount, product_price, PaymentMethod, SatisfactionScore
2. ระบบความสัมพันธ์ลูกค้า (CRM/Loyalty System)	ข้อมูลลูกค้า	customer_id, customer_name, customer_gender, customer_age, customer_city, customer_membership_level, CouponUsed
3. ข้อมูลสินค้าและสาขา	ข้อมูลหลัก	product_name, product_category, product_size, store_name, store_province, store_manager_name, PromoID, PromoName
4. ข้อมูลพนักงาน (HR)	ข้อมูลหลัก	employee_id, employee_name, employee_position, employee_hire_date, employee_salary

### ผู้ใช้งานเป้าหมาย (Target Users)

1. ผู้บริหารระดับสูง ต้องการDashboard สรุปภาพรวมของ Subject Area การขาย และการเงิน เพื่อวัดผลภาพรวมองค์กร
2. ผู้จัดการฝ่ายการตลาด เน้นการวิเคราะห์ Subject Area ลูกค้า และ โปรโมชั่น เพื่อสร้างแคมเปญแบบเจาะจง
3. ผู้จัดการฝ่ายปฏิบัติการ/สาขา เน้นการวิเคราะห์ Subject Area สาขา และ สินค้าคงคลัง เพื่อเพิ่มประสิทธิภาพรายวัน/รายสัปดาห์
4. นักวิเคราะห์ข้อมูล ต้องการเข้าถึงข้อมูลในทุก Subject Area เพื่อสร้างแบบจำลองพยากรณ์และการวิเคราะห์ที่ซับซ้อน

## ตาราง BUS Matrix

Subject Area	Business Process	Fact Table หลักที่เกี่ยวข้อง	KPI ที่สำคัญในการวิเคราะห์
1. การขาย	การทำธุรกรรมการขาย	Sales Transaction Fact	ยอดขายรวม, ยอดขายเฉลี่ยต่อบิล, ปริมาณการขาย, ยอดขายต่อชั่วโมง/วัน
2. ลูกค้า	การจัดการลูกค้าสัมพันธ์	Sales Transaction Fact, Customer Fact	มูลค่าตลอดช่วงชีวิตลูกค้า, อัตราลูกค้าซ้ำซ้ำ, การวิเคราะห์ลูกค้าตามระดับสมาชิก
3. สินค้า	การวิเคราะห์สินค้า	Sales Transaction Fact	ยอดขายตามรสชาติ/ประเภท, การวิเคราะห์ตะกร้าสินค้า, สินค้าขายดี
4. สาขา	การบริหารประสิทธิภาพสาขา	Sales Transaction Fact, Monthly P&L Fact	ยอดขายต่อพื้นที่, ยอดขายตามภูมิภาค/จังหวัด, คะแนนความพึงพอใจของลูกค้า
5. โปรโมชั่น	การตลาดและการส่งเสริมการขาย	Promotion Fact, Sales Transaction Fact	ผลตอบแทนจากการลงทุน, อัตราการใช้คูปอง, อัตราการเพิ่มของยอดขาย
6. พนักงาน	การจัดการทรัพยากรบุคคล	Sales Transaction Fact	ยอดขายต่อพนักงาน, ประสิทธิภาพการให้บริการตามคะแนนความพึงพอใจ

## ข้อจำกัดหรือเงื่อนไขเฉพาะของหน่วยงาน

### 1. ข้อจำกัดด้านความครบถ้วนของข้อมูล

- ขาดข้อมูลสต็อก/ต้นทุน ไฟล์ธุรกรรมที่ได้รับขาดข้อมูลโดยตรงเกี่ยวกับวัตถุดิบ ต้นทุนขาย และการสูญเสีย ซึ่งเป็นหัวใจสำคัญของ Subject Area สินค้าคงคลัง ต้องมีการหาแหล่งข้อมูลเพิ่มเติมหรือประมาณการ
- ขาดข้อมูลสภาพอากาศ ธุรกิจไอศกรีมต้องพึ่งพาข้อมูลสภาพอากาศภายนอกเพื่อการวิเคราะห์พยากรณ์ความต้องการ

### 2. การจัดการข้อมูลคุณภาพ

- ข้อมูลลูกค้าบางส่วน อาจมีข้อผิดพลาดหรือซ้ำซ้อน จำเป็นต้องมีการทำความสะอาดข้อมูล ในขั้นตอน ETL อย่างเข้มงวด

## การออกแบบคลังข้อมูล

### ฐานข้อมูลที่ใช้

PostgreSQL รองรับ OLAP ขนาดเล็ก-กลาง ดั้งเดิม ใช้งานง่าย มีฟีเจอร์เวลา/transaction ดี



### โครงสร้าง Star Schema

การออกแบบคลังข้อมูลของ Dairy King ใช้ **Star Schema** เพราะเข้าใจง่าย ประสิทธิภาพสูง เหมาะกับการวิเคราะห์ข้อมูลการขายและลูกค้า โดยมีตารางธุรกรรมการขายเป็นศูนย์กลาง เชื่อมโยงกับมิติต่าง ๆ เช่น ลูกค้า พนักงาน สาขา สินค้า และโปรโมชั่น เพื่อสนับสนุนการตัดสินใจเชิงกลยุทธ์ของธุรกิจได้อย่างรวดเร็วและยืดหยุ่น

## รายละเอียด Fact Table และ Dimension Table

Fact table:

Fact: fact\_sales มีคอลัมน์ ดังนี้

- Sales\_Key (PK): คีย์หลักของตาราง Fact
- Date\_Key (FK): คีย์เชื่อมไปยัง Dim\_Date
- Store\_Key (FK): คีย์เชื่อมไปยัง Dim\_Store
- Employee\_Key (FK): คีย์เชื่อมไปยัง Dim\_Employee
- Customer\_Key (FK): คีย์เชื่อมไปยัง Dim\_Customer
- Product\_Key (FK): คีย์เชื่อมไปยัง Dim\_Product
- Payment\_Key (FK): คีย์เชื่อมไปยัง Dim\_Payment
- Promotion\_Key (FK): คีย์เชื่อมไปยัง Dim\_Promo
- OrderType\_Key (FK): คีย์เชื่อมไปยัง Dim\_Order\_Type
- TransactionID: รหัสธุรกรรม (Degenerate Dimension)
- Amount: ยอดขาย
- Quantity: จำนวนสินค้าที่ขาย
- Satisfaction\_Score: คะแนนความพึงพอใจ
- Coupon\_Used\_Flag: ตัวนับการใช้คูปอง (1=ใช้, 0=ไม่ใช้)



Dimension tables มีดังนี้

**dim\_date** เก็บข้อมูลคุณลักษณะของเวลา

- date\_key (PK): คีย์หลักมิติเวลา
- full\_date: วันที่ (รูปแบบเต็ม)
- Day: วันที่ (ตัวเลข)
- Month: เดือน (ตัวเลข)
- Year: ปี
- Weekday: วันในสัปดาห์

dim\_date

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

date\_key

integer

full\_date

date

day

integer

month

integer

year

integer

weekday

text

Length/Precision

Scale

Not NULL?

Primary key?

Default

dim\_customer เก็บข้อมูลคุณลักษณะของลูกค้า

- Customer\_Key (PK): คีย์หลักมิติลูกค้า
- Customer\_ID: รหัสลูกค้าเดิม
- Customer\_Name: ชื่อลูกค้า
- Gender: เพศ
- Customer\_Age: อายุลูกค้า
- Customer\_Phone: เบอร์โทรศัพท์
- Customer\_City: เมืองของลูกค้า
- Membership\_Level: ระดับสมาชิก

dim\_customer

General

Columns

Advanced

Constraints

Partitions

Parameters

Security

SQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	customer_sk	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	customer_id	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	customer_name	text			<input type="checkbox"/>	<input type="checkbox"/>	
	gender	text			<input type="checkbox"/>	<input type="checkbox"/>	
	customer_age	integer			<input type="checkbox"/>	<input type="checkbox"/>	
	customer_phone	character varying	20		<input type="checkbox"/>	<input type="checkbox"/>	
	customer_city	text			<input type="checkbox"/>	<input type="checkbox"/>	
	membership_level	text			<input type="checkbox"/>	<input type="checkbox"/>	
	effective_from	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	effective_to	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>	'9999-12-31':d
	is_current	boolean			<input checked="" type="checkbox"/>	<input type="checkbox"/>	true

**dim\_employee** เก็บข้อมูลคุณลักษณะของพนักงาน

- Employee\_Key (PK): คีย์หลักมิติพนักงาน
- Employee\_ID: รหัสพนักงานเดิม
- Employee\_Name: ชื่อพนักงาน
- Position: ตำแหน่งงาน
- Phone: เบอร์โทรศัพท์
- Hire\_Date: วันที่เริ่มจ้าง
- Salary: เงินเดือน

























dim\_employee

GeneralColumnsAdvancedConstraintsPartitionsParametersSecuritySQL

Inherited from table(s)

Select to inherit from...

Columns

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	employee_key	bigint <div></div>			<div></div>	<div></div>	nextval('dim_em
 	employee_code	text <div></div>			<div></div>	<div></div>	
 	employee_name	text <div></div>			<div></div>	<div></div>	
 	position	text <div></div>			<div></div>	<div></div>	
 	phone	text <div></div>			<div></div>	<div></div>	
 	hire_date	date <div></div>			<div></div>	<div></div>	
 	salary	numeric <div></div>	12	2	<div></div>	<div></div>	
 	current_store_id	integer <div></div>			<div></div>	<div></div>	
 	prev_store_id	integer <div></div>			<div></div>	<div></div>	
 	store_changed_at	date <div></div>			<div></div>	<div></div>	
 	created_at	timestamp without time zone <div></div>			<div></div>	<div></div>	now()
 	updated_at	timestamp without time zone <div></div>			<div></div>	<div></div>	

### dim\_store เก็บข้อมูลคุณลักษณะของสาขา

- Store\_Key (PK): รหัสสาขา
- Store\_Name: ชื่อสาขา
- Store\_City: เมืองที่สาขาตั้งอยู่
- Store\_Address: ที่อยู่สาขา
- Store\_Manager\_ID: รหัสผู้จัดการสาขา
- Store\_Manager\_Name: ชื่อผู้จัดการสาขา

dim\_store

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns							
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	store_id	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	store_name	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	store_city	text			<input type="checkbox"/>	<input type="checkbox"/>	
	store_address	text			<input type="checkbox"/>	<input type="checkbox"/>	
	store_manager_id	integer			<input type="checkbox"/>	<input type="checkbox"/>	
	store_manager_name	text			<input type="checkbox"/>	<input type="checkbox"/>	

### dim\_product เก็บข้อมูลคุณลักษณะของสินค้า

- Product\_Key (PK): คีย์หลักมิติสินค้า
- Product\_ID: รหัสสินค้าเดิม
- Product\_Name: ชื่อสินค้า
- Product\_Category: หมวดหมู่สินค้า
- Product\_Size: ขนาดสินค้า
- Product\_Price: ราคาต่อหน่วย

dim\_product

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns							
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
	product_key	bigint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('dim_pi')
	product_id	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	product_name	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	product_category	text			<input type="checkbox"/>	<input type="checkbox"/>	
	product_size	text			<input type="checkbox"/>	<input type="checkbox"/>	
	product_price	numeric	10	2	<input type="checkbox"/>	<input type="checkbox"/>	
	created_at	timestamp without time zone			<input type="checkbox"/>	<input type="checkbox"/>	now()
	updated_at	timestamp without time zone			<input type="checkbox"/>	<input type="checkbox"/>	

dim\_promo เก็บข้อมูลคุณลักษณะของโปรโมชั่น













- Promo\_Key (PK): คีย์หลักมิติโปรโมชั่น
- Promo\_ID: รหัสโปรโมชั่น
- Promo\_Name: ชื่อโปรโมชั่น
- Discount\_Type ประเภทส่วนลด
- Start\_Date วันเริ่มต้นโปรโมชั่น
- End\_Date วันสิ้นสุดโปรโมชั่น

dim\_promo

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns +

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	promo_key	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('dim_
 	promo_id	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	promo_name	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 	discount_type	text			<input type="checkbox"/>	<input type="checkbox"/>	
 	start_date	date			<input type="checkbox"/>	<input type="checkbox"/>	
 	end_date	date			<input type="checkbox"/>	<input type="checkbox"/>	

dim\_payment\_method เก็บข้อมูลช่องทางการชำระเงิน





- Payment\_Key (PK): คีย์หลักมิติการชำระเงิน
- Payment\_Method: ช่องทางการชำระเงิน (เช่น Cash, QR, Card)

dim\_payment\_method

General Columns Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns +

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	payment_method_key	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('dim_
 	payment_method	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

dim\_order\_type ตารางนี้เป็นการสร้างขึ้นใหม่ให้สอดคล้องกับ Schema และข้อมูลที่มี

- Order\_Type\_Key (PK): คีย์หลักมิติประเภทการสั่งซื้อ
- Order\_Type: ประเภทการสั่ง (เช่น Walk-in, Delivery)

dim\_order\_type

General

Columns

Advanced

Constraints

Partitions

Parameters

Security





SQL

Inherited from table(s)

Select to inherit from...

Columns

+

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
 	order_type_key	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nextval('c
 	order_type	text			<input checked="" type="checkbox"/>	<input type="checkbox"/>	

## การจัดการ Slowly Changing

การจัดการข้อมูลมิติที่มีการเปลี่ยนแปลงช้า (Slowly Changing Dimensions หรือ SCD) เป็นแนวทางสำคัญในการออกแบบคลังข้อมูลของ Dairy King เพื่อให้สามารถติดตามและเก็บประวัติของข้อมูลเชิงมิติที่เปลี่ยนแปลงตามช่วงเวลาได้อย่างถูกต้อง ข้อมูลบางประเภทมีการเปลี่ยนแปลงบ่อย ในขณะที่บางประเภทควรคงที่ ดังนั้นจึงเลือกใช้ SCD หลายประเภทให้เหมาะสมกับลักษณะของแต่ละ Dimension ดังนี้

### 1. SCD Type 0 – Fixed Dimension (Immutable Data) ข้อมูล dim\_store

ตาราง **dim\_store** เก็บข้อมูลสาขา เช่น ชื่อ จังหวัด และ ผู้จัดการสาขา ซึ่งเป็นข้อมูลที่ไม่ควรเปลี่ยนแปลง จึงถูกกำหนดให้เป็น **SCD Type 0** โดยข้อมูลที่บันทึกแล้วจะถือเป็นข้อมูลถาวร หากมีการเปลี่ยนแปลงจะต้องออก **Store ID** ใหม่แทนการแก้ไขข้อมูลเดิม

**ประโยชน์:** ช่วยรักษาความถูกต้องของ Foreign Key ระหว่าง Fact กับ Dimension และป้องกันความคลาดเคลื่อนของข้อมูลทางภูมิศาสตร์

### 2. SCD Type 1 – Overwrite (No History) ข้อมูล dim\_product

มิติ **สินค้า (dim\_product)** อาจมีการปรับเปลี่ยนข้อมูล เช่น ชื่อสินค้า หรือ ราคาขาย แต่ไม่จำเป็นต้องเก็บข้อมูลย้อนหลัง เมื่อมีการเปลี่ยนแปลงจะอัปเดตทับค่าเดิมโดยตรง

**ตัวอย่าง:** เมื่อราคาของ “Vanilla Ice Cream” เปลี่ยนจาก 45 บาท เป็น 49 บาท ข้อมูลใน **dim\_product** จะถูกเขียนทับด้วยราคาล่าสุดทันที

**ประโยชน์:** ลดขนาดฐานข้อมูล และ เหมาะสำหรับรายงานที่ต้องการข้อมูลปัจจุบัน เช่น ยอดขายล่าสุด หรือ สินค้าขายดีรายเดือน

### 3. SCD Type 2 – Add New Row (Full History) : dim\_customer

มิติ ลูกค้า (dim\_customer) มีข้อมูลที่เปลี่ยนแปลงตามช่วงเวลา เช่น ระดับสมาชิก (Member Level) หรือ ที่อยู่ เพื่อให้สามารถวิเคราะห์พฤติกรรมของลูกค้าในแต่ละช่วงเวลาได้ จึงใช้รูปแบบ SCD Type 2 โดยจะสร้างแถวใหม่ทุกครั้งที่มีการเปลี่ยนแปลงคุณลักษณะสำคัญ

- แถวเก่าจะถูกปิดด้วยค่า is\_current = false และ valid\_to เท่ากับวันที่สิ้นสุด
- แถวใหม่จะมี is\_current = true และ valid\_from เป็นวันที่เริ่มต้นเวอร์ชันใหม่

**ประโยชน์:** สามารถวิเคราะห์แนวโน้มของลูกค้าได้ เช่น การเลื่อนระดับสมาชิก หรือ การย้ายพื้นที่อาศัย ช่วยให้การวิเคราะห์ Customer Lifetime Value มีความถูกต้องมากขึ้น

### 4. SCD Type 3 – Limited History (Previous vs Current) : dim\_employee

มิติ พนักงาน (dim\_employee) มีการเปลี่ยนแปลงบางส่วน เช่น สาขาที่ประจำอยู่ แต่ไม่จำเป็นต้องเก็บข้อมูลย้อนหลังหลายช่วง จึงใช้รูปแบบ SCD Type 3 ซึ่งจะเก็บเพียง “ค่าปัจจุบัน” และ “ค่าก่อนหน้า” ในแถวเดียวกัน

เช่น เมื่อพนักงานย้ายจากสาขา CentralWorld ไป Siam Paragon ค่า prev\_store\_id จะบันทึก CentralWorld และ current\_store\_id จะเปลี่ยนเป็น Siam Paragon

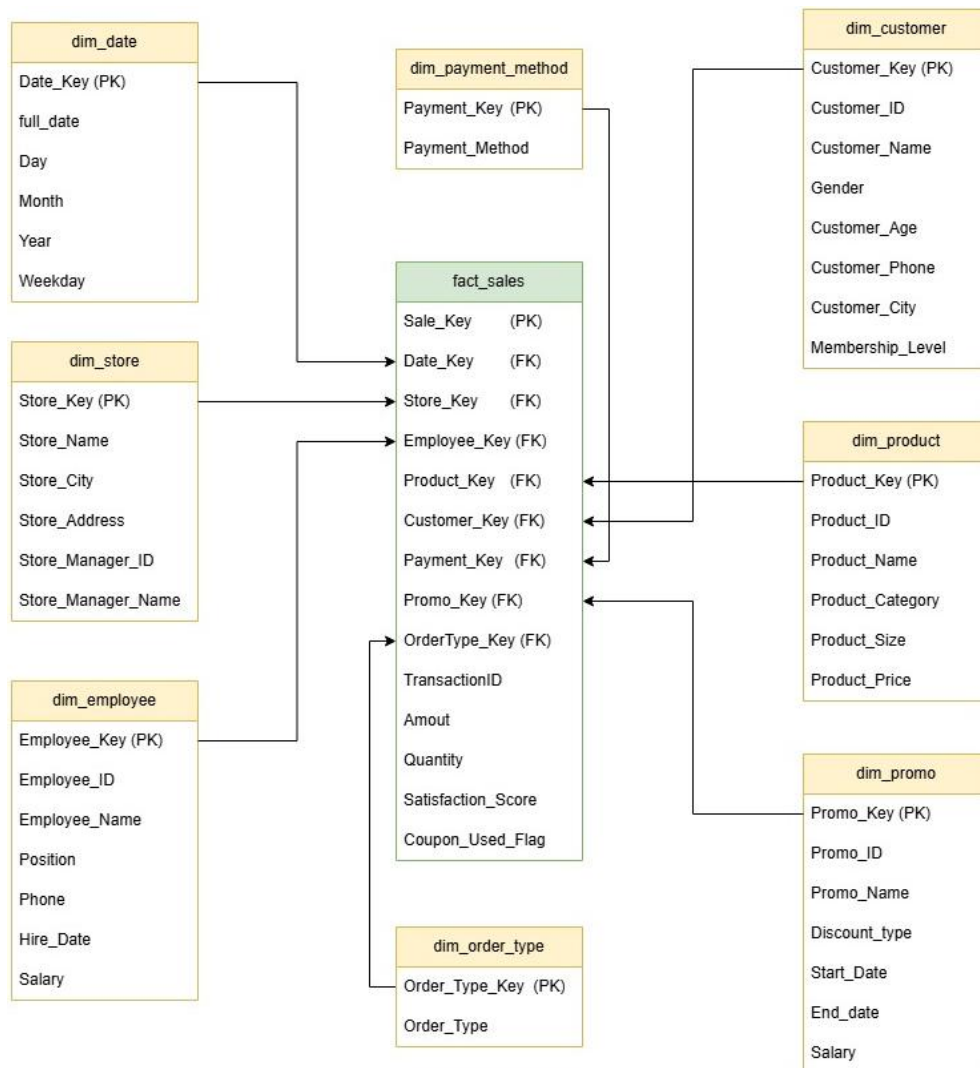
**ประโยชน์:** ประหยัดพื้นที่จัดเก็บ และสามารถเปรียบเทียบผลงานของพนักงานก่อน-หลังการย้ายสาขาได้โดยไม่ต้องสร้างหลายเวอร์ชัน



## ER Diagram และ Data Dictionary

### ER Diagram

โครงสร้าง Star Schema โดยมี fact\_sales เป็นศูนย์กลาง เชื่อมกับมิติหลัก: dim\_date, dim\_store, dim\_employee, dim\_customer, dim\_product, dim\_promo, dim\_payment\_method, dim\_order\_type และมี Degenerate Dimension คือ TransactionID อยู่ใน Fact



## Data Dictionary

### 1) Fact Table

fact\_sales

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
Sales_Key	BIGSERIAL	PK	รหัสหลักของรายการขาย
Date_Key	INT	FK	เชื่อมกับ dim_date
Store_Key	INT	FK	เชื่อมกับ dim_store
Employee_Key	INT	FK	เชื่อมกับ dim_employee
Customer_Key	INT	FK	เชื่อมกับ dim_customer
Product_Key	INT	FK	เชื่อมกับ dim_product
Payment_Key	INT	FK	เชื่อมกับ dim_payment_method
Promotion_Key	INT	FK	เชื่อมกับ dim_promo
OrderType_Key	INT	FK	เชื่อมกับ dim_order_type
TransactionID	TEXT	–	รหัสธุรกรรมจากระบบ POS
Amount	NUMERIC(12,2)	–	ยอดขายรวมของรายการ
Quantity	INT	–	จำนวนสินค้าที่ขาย
Satisfaction_Score	NUMERIC(3,2)	–	คะแนนความพึงพอใจ
Coupon_Used_Flag	SMALLINT	–	1 = ใช้คูปอง / 0 = ไม่ใช้

## 2) Dimension Tables

dim\_date

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
date_key	INT	PK	คีย์ประจำวัน (รูปแบบ YYYYMMDD)
full_date	DATE	–	วันที่
day	SMALLINT	–	วันที่ (1–31)
month	SMALLINT	–	เดือน
year	SMALLINT	–	ปี
weekday	SMALLINT	–	วันในสัปดาห์ (1–7)

dim\_customer

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
customer_key	BIGSERIAL	PK	คีย์หลักของลูกค้า
customer_id	TEXT	–	รหัสลูกค้าจาก CRM
customer_name	TEXT	–	ชื่อลูกค้า
gender	TEXT	–	เพศ
customer_age	SMALLINT	–	อายุ
customer_phone	TEXT	–	เบอร์โทรศัพท์
customer_city	TEXT	–	จังหวัดหรือเมืองที่อยู่
membership_level	TEXT	–	ระดับสมาชิก

dim\_employee

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
employee_key	BIGSERIAL	PK	คีย์หลักของพนักงาน
employee_id	TEXT	–	รหัสพนักงานจาก HR
employee_name	TEXT	–	ชื่อพนักงาน
position	TEXT	–	ตำแหน่งงาน
phone	TEXT	–	เบอร์โทรศัพท์
hire_date	DATE	–	วันที่เริ่มงาน
salary	NUMERIC(12,2)	–	เงินเดือน

dim\_store

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
store_key	BIGSERIAL	PK	คีย์หลักของสาขา
store_name	TEXT	–	ชื่อสาขา
store_city	TEXT	–	จังหวัด
store_address	TEXT	–	ที่อยู่สาขา
store_manager_id	TEXT	–	รหัสผู้จัดการสาขา
store_manager_name	TEXT	–	ชื่อผู้จัดการ

dim\_product

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
product_key	BIGSERIAL	PK	คีย์หลักของสินค้า
product_id	TEXT	–	รหัสสินค้า
product_name	TEXT	–	ชื่อสินค้า
product_category	TEXT	–	หมวดหมู่สินค้า
product_size	TEXT	–	ขนาดบรรจุภัณฑ์
product_price	NUMERIC(12,2)	–	ราคาต่อหน่วย

dim\_promo

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
promo_key	BIGSERIAL	PK	คีย์หลักของโปรโมชั่น
promo_id	TEXT	–	รหัสโปรโมชั่น
promo_name	TEXT	–	ชื่อโปรโมชั่น
discount_type	TEXT	–	ประเภทส่วนลด
start_date	DATE	–	วันเริ่มต้น
end_date	DATE	–	วันสิ้นสุด

dim\_payment\_method

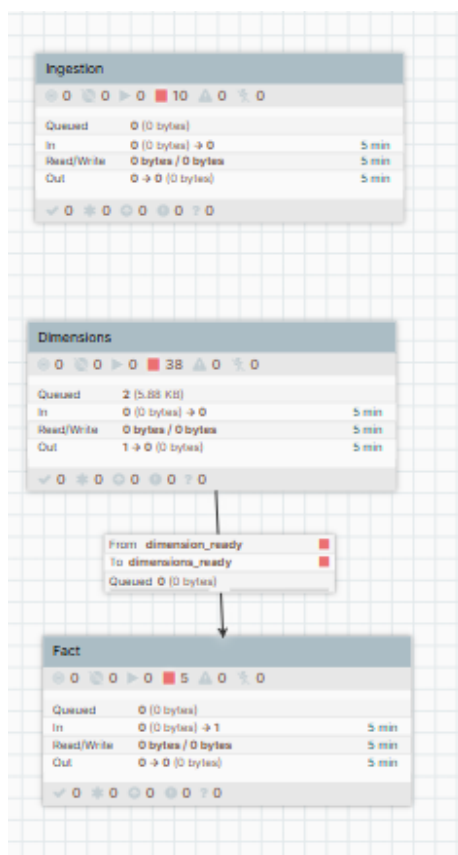
คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
payment_key	BIGSERIAL	PK	คีย์หลักของช่องทางการชำระเงิน
payment_method	TEXT	–	ประเภทการชำระเงิน (Cash, QR, Card)

dim\_order\_type

คอลัมน์	ชนิดข้อมูล	คีย์	คำอธิบาย
order_type_key	BIGSERIAL	PK	คีย์หลักของประเภทคำสั่งซื้อ
order_type	TEXT	–	ประเภทการสั่ง (Walk-in, Delivery)

## การออกแบบกระบวนการ ETL/ELT ด้วย Apache NiFi

แผนภาพ Flow ของ NiFi และรายละเอียดการกำหนดค่าต่างๆ



โครงสร้างการประมวลผลข้อมูลในระบบ Data Warehouse ถูกแบ่งออกเป็น 3 กลุ่มหลัก ได้แก่ Ingestion, Dimensions และ Fact ซึ่งทำงานแบบลำดับขั้น (Sequential Pipeline) เพื่อควบคุมความถูกต้องของข้อมูลและความสัมพันธ์ระหว่างตารางโดยมีรายละเอียดดังนี้:

## 1. Ingestion Process Group

กลุ่มนี้มีหน้าที่รับข้อมูลจากแหล่งต้นทาง เช่น ไฟล์ CSV หรือข้อมูลระบบปฏิบัติการ แล้วนำมาผ่านขั้นตอนต่างๆ ได้แก่:

- การอ่านไฟล์และตรวจสอบรูปแบบข้อมูล
- การแปลง schema ให้เป็นมาตรฐาน
- การคัดกรองข้อมูลที่ซ้ำกับฐานข้อมูลเดิม
- การบันทึกข้อมูลใหม่ลงใน Staging Layer

Output ของส่วนนี้คือข้อมูลที่ถูกเตรียมพร้อมในสถานะ `stage_ready` ซึ่งพร้อมสำหรับกระบวนการตรวจสอบด้านคุณภาพ (Validation) และโหลดเข้าสู่ Dimension

### วัตถุประสงค์

- ปรับข้อมูลให้อยู่ในรูปแบบเดียวกัน
- ป้องกันข้อมูลซ้ำซ้อน
- ทำให้การประมวลผลในส่วนถัดไปมีความถูกต้อง

## 2. Dimensions Process Group

ส่วนนี้รับข้อมูลที่ถูกเตรียมแล้ว และนำไปบันทึกลง Dimension Tables ตามกลไก Slowly Changing Dimension (SCD) โดยแบ่งตามคุณสมบัติข้อมูล เช่น:

- SCD Type 0 สำหรับข้อมูลไม่อนุญาตให้มีการเปลี่ยนแปลงเช่น Store
- SCD Type 1 สำหรับข้อมูลที่ไม่ต้องเก็บประวัติ เช่น Product
- SCD Type 2 สำหรับข้อมูลที่ต้องเก็บประวัติเชิงเวลา เช่น สถานะสมาชิกของลูกค้า
- SCD Type 3 สำหรับข้อมูลที่ติดตามค่าเก่าและใหม่ใน record เดียว เช่น การย้ายสาขาของพนักงาน

ในส่วนนี้มีการใช้ Wait และ Notify เพื่อชิงโครโนซีให้ Dimension ทั้งหมดทำงานเสร็จก่อนส่งข้อมูลต่อไปยัง Fact



### ผลลัพธ์ของขั้นตอนนี้

- ได้ Dimension ที่เป็นปัจจุบันและน่าเชื่อถือ
- มีประวัติย้อนหลัง (History Tracking) ในกรณีที่ทำเป็น
- พร้อมสำหรับการ Join กับ Fact

### 3. Fact Process Group

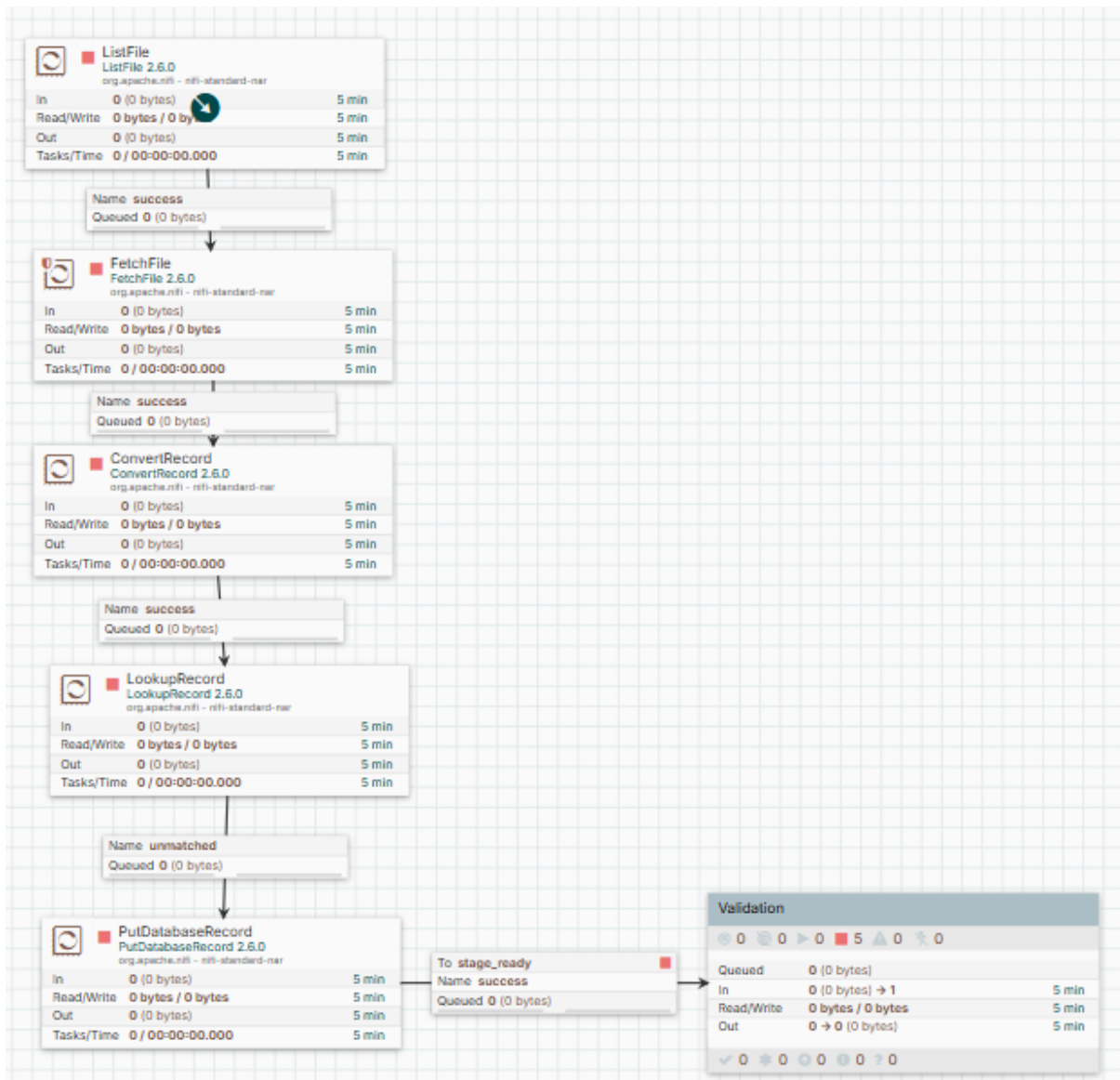
กลุ่มนี้คือขั้นตอนสุดท้ายของ Pipeline โดยรับสัญญาณ dimension\_ready ซึ่งรับรองว่า Dimension ทั้งหมดถูกโหลดเรียบร้อยแล้ว จากนั้น Fact Loader จะ:

- ทำการ Join ระหว่างข้อมูลใน Staging กับ Dimension Keys
- แมป Surrogate Keys แทน Business Keys
- คำนวณค่ามาตรวัด (Measures) เช่นจำนวนยอดขาย ยอดชำระเงิน
- ทำการ Insert ข้อมูลลง Fact Table แบบ Batch

### วัตถุประสงค์

- ป้องกัน Foreign Key Missing
- เก็บข้อมูลธุรกรรม (Transactional Facts)
- รองรับการวิเคราะห์เชิง BI และ Data Mining

## Data Ingestion Flow (Staging Layer)



กระบวนการนี้เป็นขั้นตอนการนำเข้าข้อมูล (Ingestion) จากไฟล์ต้นทางเข้าสู่ระบบ Staging ของ Data Warehouse โดยใช้ Apache NiFi เพื่อจัดการกระบวนการอ่านไฟล์ ตรวจสอบความถูกต้องของข้อมูล แปลงรูปแบบ และบันทึกลงฐานข้อมูลเพื่อเตรียมสำหรับการประมวลผลหลังจากนั้น

ลำดับการทำงานมีดังนี้:

#### 1. ListFile

Processor นี้ทำหน้าที่ค้นหาไฟล์ข้อมูลจากโพลเดอร์ต้นทางตามกำหนด เช่น raw transaction หรือข้อมูลมิติอื่น ๆ เมื่อพบไฟล์ใหม่ ระบบจะสร้าง FlowFile แทนเมตาเดตาของไฟล์นั้นโดยยังไม่ทำการดึงเนื้อหาจริงในขั้นตอนนี้

#### 2. FetchFile

โปรเซสเซอร์นี้จะโหลดเนื้อหาจริงของไฟล์ที่ค้นพบก่อนหน้านี้เข้าไปใน FlowFile เพื่อให้สามารถประมวลผลข้อมูลภายในได้ อนุญาตให้ NiFi จัดการข้อมูลแบบ Stream โดยไม่ต้องโหลดไฟล์ทั้งหมดไว้ในหน่วยความจำมากเกินไป

#### 3. ConvertRecord

ใช้สำหรับแปลงรูปแบบข้อมูลให้อยู่ในโครงสร้างมาตรฐานที่สอดคล้องกับ Schema Registry เช่น การแปลง CSV ไปเป็น JSON หรือ Avro รวมถึงตรวจสอบชนิดข้อมูล (datatype) และความถูกต้องของฟิลด์ ช่วยให้ข้อมูลอยู่ในรูปแบบที่พร้อมใช้งานในขั้นตอนถัดไป

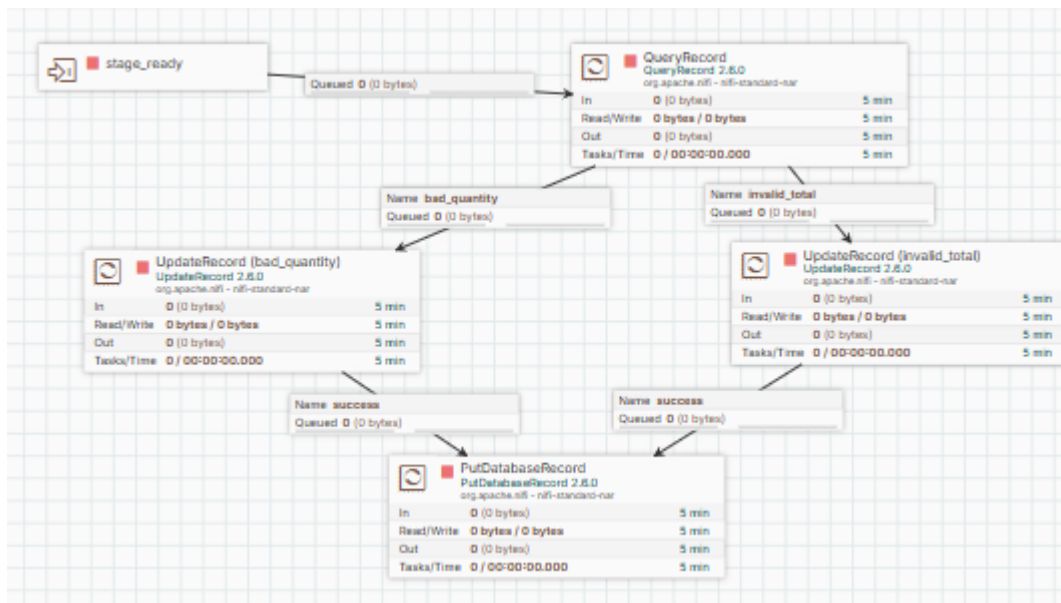
#### 4. LookupRecord

Processor นี้จะตรวจสอบว่าข้อมูลที่เข้ามามีอยู่แล้วในฐานข้อมูลหรือไม่ผ่านคีย์ที่กำหนด เช่น รหัสลูกค้า รหัสสินค้า หรือรหัสพนักงาน หากข้อมูลพบอยู่แล้วจะถือว่าเป็นข้อมูลที่ไม่ต้องโหลดซ้ำ ส่วนข้อมูลที่ยังไม่เคยมีในระบบจะถูกส่งต่อไปยังสาขา “unmatched”

#### 5. PutDatabaseRecord

สำหรับข้อมูลที่พบว่าเป็น “unmatched” จะถูกบันทึกลงในตาราง Staging เพื่อจัดเตรียมข้อมูลสำหรับกระบวนการ Dimension Loading หรือ Fact Loading ในลำดับถัดไป Processor นี้รองรับการ insert แบบ batch ทำให้สามารถจัดการข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพ

## 6. Validation



ขั้นตอนสุดท้ายหลังจากข้อมูลถูกโหลดเข้าสู่พื้นที่ Staging แล้ว กระบวนการนี้มีหน้าที่ตรวจสอบคุณภาพข้อมูล (Data Quality) ก่อนที่จะถูกนำไปประมวลผลต่อในขั้นตอน Dimension หรือ Fact Loading โดยใช้ Apache NiFi เพื่อแยกข้อมูลผิดปกติ อัปเดตสถานะ และบันทึกข้อมูลที่ถูก correction ลงฐานข้อมูลอีกครั้ง

### ลำดับการทำงานของ Validation ประกอบด้วย:

#### 6.1. stage\_ready

ข้อมูลทุกชุดที่ถูกตรวจสอบเบื้องต้นใน Staging จะถูกส่งเข้าสู่กระบวนการนี้ เพื่อยืนยันว่าผ่านการตรวจสอบ schema และพร้อมสำหรับการตรวจสอบเชิงเงื่อนไข (Logical Validation)

#### 6.2. QueryRecord

Processor นี้ทำหน้าที่ประเมินกฎธุรกิจ (Business Rules) เพื่อหา record ที่มีค่าผิดปกติ เช่น

- ปริมาณสินค้า (quantity)  $\leq 0$
- ยอดรวมการชำระเงิน (total\_amount)  $\leq 0$

โดยจะแยกข้อมูลออกเป็นสองเส้นทาง ได้แก่

- bad\_quantity สำหรับปริมาณไม่สมเหตุสมผล
- invalid\_total สำหรับยอดรวมที่ไม่ถูกต้อง

การแยกเส้นทางนี้ช่วยให้สามารถกำหนดวิธีแก้ไขเฉพาะกรณีได้

### 6.3. UpdateRecord (bad\_quantity / invalid\_total)

เมื่อพบข้อมูลผิดปกติ ระบบจะปรับปรุงค่า field เพิ่มเติม เช่น

- การตั้ง flag แจ้งเตือนความผิดปกติ
- การเปลี่ยนสถานะ record ให้ระบบ downstream สามารถกรองได้

ปรับข้อมูลให้ถูกต้องตามกฎหมาย

### 6.4. PutDatabaseRecord

ข้อมูลที่ถูกปรับปรุงสถานะหรือค่า attribute จะถูกบันทึกกลับเข้าสู่ Staging เพื่อนำไปใช้งานต่อ โดยจะเก็บทั้งข้อมูลในรูป corrected และ flagged เพื่อบริการวิเคราะห์ภายหลัง

เมื่อจบกระบวนการ **Data Validation Flow** นี้:

- ข้อมูลผิดปกติถูกกำกับด้วยสถานะชัดเจน
- ไม่มีการส่งข้อมูลที่ไม่สมเหตุผลเข้าสู่ Fact
- ผู้ใช้สามารถ audit และ debug ได้ง่ายขึ้น

### ผลลัพธ์ของ Flow

- ข้อมูลถูกจัดรูปแบบ (Normalize) และตรวจสอบแล้ว
- คัดกรองข้อมูลซ้ำไม่ให้เข้าสู่ระบบอีก
- ลดปัญหาความผิดพลาดด้าน schema และ Data Type
- เตรียมข้อมูลเข้าสู่ขั้นตอน SCD และ Fact Loading ได้อย่างปลอดภัย
- ทำให้การประมวลผลใน Data Warehouse มีความถูกต้องและน่าเชื่อถือ

## ตัวอย่างข้อมูลใน pdAdmin4

	transaction_id text	order_date date	customer_id text	employee_id integer	product_id text	quantity integer	amount numeric (10,2)	employee_name text	customer_name text	gender text	customer_age integer	customer_phone text	customer_city text
1	1	2025-04-03	C092	5	P007	0	90.00	EM-005	customer_92	Female	37	881709884	Nakhon Ratchasi...
2	2	2025-01-30	C130	9	P004	0	225.00	EM-009	customer_130	Female	25	820495300	Bangkok
3	3	2025-06-10	C028	12	P016	1	130.00	EM-012	customer_28	Female	56	839093907	Chiang Mai
4	4	2025-03-06	C124	6	P020	1	75.00	Warinya Chaisiri	customer_124	Male	20	875732489	Phuket
5	5	2025-05-26	C155	2	P006	1	65.00	EM-002	customer_155	Male	20	840185054	Khon Kaen
6	6	2025-02-01	C085	8	P002	5	275.00	EM-008	customer_85	Male	20	830840174	Bangkok
7	7	2025-09-26	C059	3	P015	1	55.00	EM-003	customer_59	Male	54	839549535	Phuket
8	8	2024-06-18	C089	8	P020	5	375.00	EM-008	customer_89	Female	43	826953831	Chiang Mai
9	9	2024-02-16	C047	4	P011	3	180.00	EM-004	customer_47	Female	41	811079600	Chiang Mai
10	10	2025-04-02	C050	4	P009	5	325.00	EM-004	customer_50	Female	56	853381573	Nakhon Ratchasi...

membership_level text	product_name text	product_category text	product_size text	product_price numeric (10,2)	store_id integer	store_name text	store_address text
Silver	Strawberry Ice Cream - Small	Ice Cream	Small	45.00	2	Dairy King - CentralWorld	4,4/1-4/2,4/4 Rajdamri Rd, Pathum Wan, Bangkok 10330
Silver	Chocolate Ice Cream - Small	Ice Cream	Small	45.00	5	Dairy King - Terminal 21	2,88 Sukhumvit Rd, Khlong Toei Nuea, Watthana, Bangkok 10110
Silver	Caramel Shake - Medium	Shake	Medium	65.00	8	Dairy King - Emporium	622 Sukhumvit Rd, Khlong Tan, Khlong Toei, Bangkok 10110
Platinum	Chocolate Shake - Large	Shake	Large	75.00	1	Dairy King - Siam Paragon	991 Rama 1 Rd, Pathum Wan, Bangkok 10330
Silver	Chocolate Ice Cream - Large	Ice Cream	Large	65.00	4	Dairy King - Mega Bangna	39 Moo 6 Bangna-Trad Rd, Bang Kaeo, Bang Phli, Samut Prakan 10...
Silver	Vanilla Ice Cream - Medium	Ice Cream	Medium	55.00	5	Dairy King - Terminal 21	2,88 Sukhumvit Rd, Khlong Toei Nuea, Watthana, Bangkok 10110
Silver	Caramel Shake - Small	Shake	Small	55.00	7	Dairy King - The Mall Bangk...	3522 Lat Phrao Rd, Khlong Chan, Bang Kapi, Bangkok 10240
Silver	Chocolate Shake - Large	Shake	Large	75.00	1	Dairy King - Siam Paragon	991 Rama 1 Rd, Pathum Wan, Bangkok 10330
Silver	Matcha Ice Cream - Medium	Ice Cream	Medium	60.00	3	Dairy King - IconSiam	299 Charoen Nakhon Rd, Khlong Ton Sai, Khlong San, Bangkok 106...
Silver	Strawberry Ice Cream - Large	Ice Cream	Large	65.00	2	Dairy King - CentralWorld	4,4/1-4/2,4/4 Rajdamri Rd, Pathum Wan, Bangkok 10330

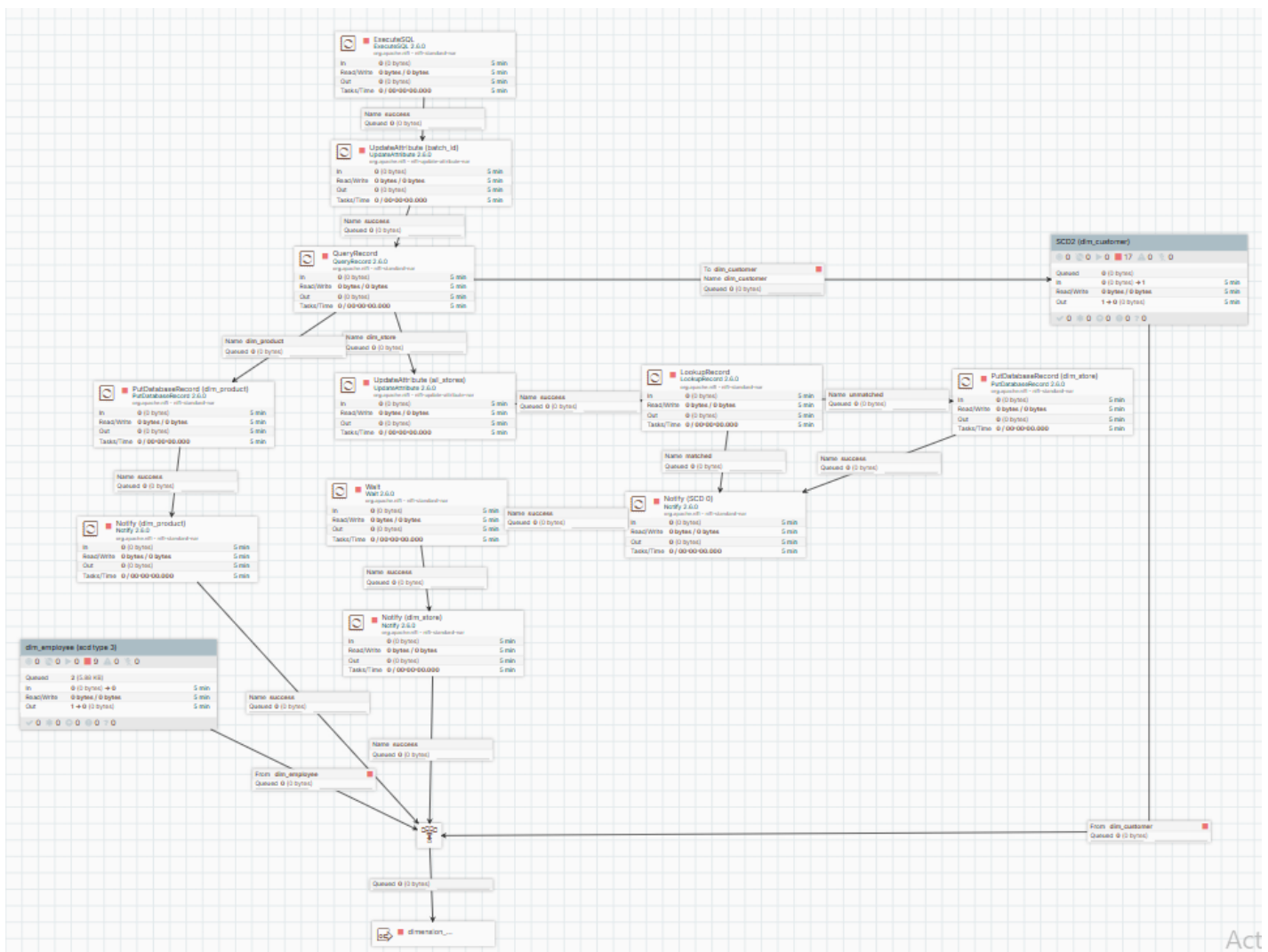
  

store_city text	store_manager_id integer	store_manager_name text	order_type text	payment_method text	coupon_used boolean	satisfaction_score integer	promo_id text	promo_name text
Bangkok	6	Warinya Chaisiri	Delivery	QR	true	[null]	PR004	New Menu Launch
Bangkok	21	Pongsathorn Jirasuk	Walk-in	Card	false	1	[null]	[null]
Bangkok	36	Nattapong Suriyakul	Delivery	Card	true	[null]	PR004	New Menu Launch
Bangkok	1	Somchai Prasert	Delivery	QR	true	3	PR005	Happy Hour 1+1
Samut Prak...	16	Kamonwan Thepsiri	Walk-in	QR	false	[null]	[null]	[null]
Bangkok	21	Pongsathorn Jirasuk	Delivery	QR	true	[null]	PR005	Happy Hour 1+1
Bangkok	31	Thanawat Meechai	Delivery	Cash	false	1	[null]	[null]
Bangkok	1	Somchai Prasert	Delivery	Card	true	5	PR004	New Menu Launch
Bangkok	11	Anucha Wongchai	Walk-in	Cash	false	[null]	[null]	[null]
Bangkok	6	Warinya Chaisiri	Walk-in	Cash	false	2	[null]	[null]

## Dimension Loading Flow

กระบวนการในหน้านี้เป็นส่วนของการโหลดข้อมูลเข้าสู่ตาราง Dimension ภายใน Data Warehouse โดยออกแบบให้รองรับกรณีการเปลี่ยนแปลงข้อมูลที่แตกต่างกันตามลักษณะเชิงธุรกิจ (Slowly Changing Dimensions) โดยใช้ Apache NiFi ในการควบคุมการไหลของข้อมูล การตรวจจับการเปลี่ยนแปลง และบันทึกข้อมูลลงในฐานข้อมูลอย่างเป็นระบบ

Flow นี้ประกอบด้วย Dimension หลัก 4 กลุ่ม ได้แก่ Product, Store และ Customer รวมถึง Employee ซึ่งใช้ SCD หลายประเภทไม่เหมือนกัน



## 1) dim\_store — SCD Type 0

ข้อมูลเกี่ยวกับสาขา เช่นรหัสสาขา ชื่อสาขา หรือสถานที่ตั้งเป็นข้อมูลที่ไม่ควรเปลี่ยนแปลงตลอดอายุสาขา ดังนั้นจึงใช้ SCD Type 0 ซึ่งมีลักษณะสำคัญคือ:

- ไม่อนุญาตให้อัปเดตข้อมูล
- หากข้อมูลแตกต่างจากเดิม ถือว่า source ผิด หรือธุรกิจต้องออก store\_id ใหม่
- ประวัติของสาขาปัจจุบันถือเป็น immutable

การออกแบบนี้มีข้อดี:

- ป้องกันข้อมูลผิดพลาดจาก human update
- รักษาความถูกต้องของ referential keys
- เหมาะกับข้อมูล master ระดับสถานที่

เครื่องมือ NiFi จะ:

- ตรวจสอบความซ้ำซ้อน (Duplicate Check)
- ปล่อยให้ผ่านเฉพาะข้อมูลที่ไม่เคยมี

ไม่มีการ overwrite หรือสร้าง version ใหม่

- ตัวอย่างข้อมูลใน pdAdmin4

store_id [PK] integer	store_name text	store_city text	store_address text	store_manager_id integer	store_manager_name text
1	Dairy King - Siam Paragon	Bangkok	991 Rama I Rd, Pathum Wan, Bangkok 10330	1	Somchai Prasert
2	Dairy King - CentralWorld	Bangkok	4,4/1-4/2,4/4 Rajdamri Rd, Pathum Wan, Bangkok 10330	6	Warinya Chaisiri
3	Dairy King - IconSiam	Bangkok	299 Charoen Nakhon Rd, Khlong Ton Sai, Khlong San, Bangkok 106...	11	Anucha Wongchai
4	Dairy King - Mega Bangna	Samut Prak...	39 Moo 6 Bangna-Trad Rd, Bang Kaeo, Bang Phli, Samut Prakan 10...	16	Kamonwan Thepsiri
5	Dairy King - Terminal 21	Bangkok	2,88 Sukhumvit Rd, Khlong Toei Nuea, Watthana, Bangkok 10110	21	Pongsathorn Jirasuk
6	Dairy King - Central Ladprao	Bangkok	1691 Phahonyothin Rd, Chatuchak, Bangkok 10900	26	Ratchanok Wattana...
7	Dairy King - The Mall Bangk...	Bangkok	3522 Lat Phrao Rd, Khlong Chan, Bang Kapi, Bangkok 10240	31	Thanawat Meechai
8	Dairy King - Emporium	Bangkok	622 Sukhumvit Rd, Khlong Tan, Khlong Toei, Bangkok 10110	36	Nattapong Suriyakul



## 2) dim\_product — SCD Type 1

ข้อมูลสินค้า เช่นชื่อและราคา อาจมีการแก้ไขได้ตลอดเวลา แต่ไม่มีความจำเป็นต้องเก็บประวัติย้อนหลัง ระบบจึงใช้รูปแบบ SCD Type 1 โดยมีหลักการคือ:

- เมื่อข้อมูลสินค้าเปลี่ยน ระบบจะ update ทับค่าเดิมทันที (overwrite)
- ไม่มีการสร้างแถวใหม่

Processor ที่ใช้หลักคือ:

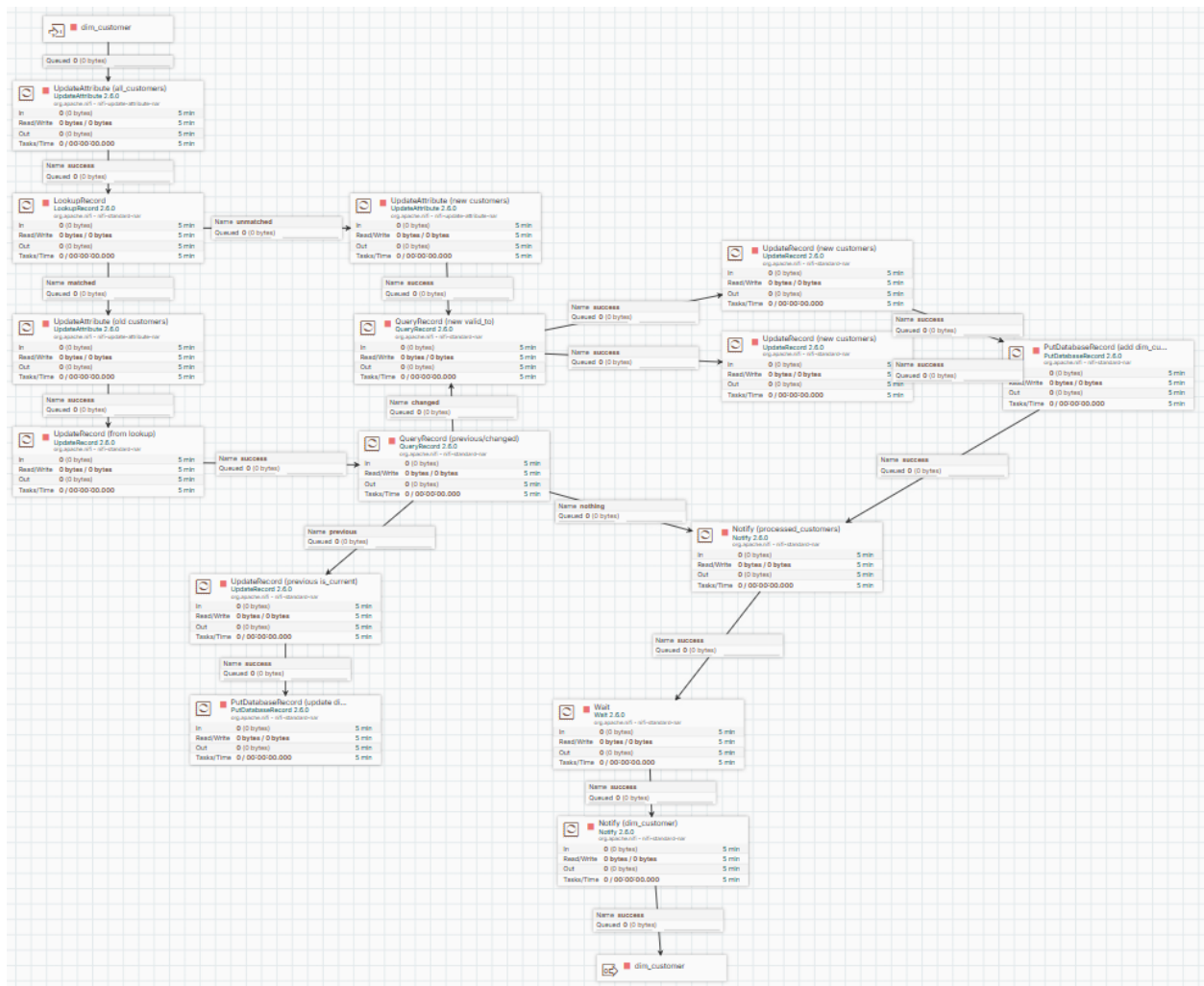
- LookupRecord ตรวจสอบสินค้าเดิม
- QueryRecord ตรวจสอบ attribute ที่เปลี่ยน
- UpdateRecord ปรับค่าใหม่
- PutDatabaseRecord บันทึกลงตาราง

ผลลัพธ์คือข้อมูลสินค้าเป็นข้อมูลล่าสุดเสมอ

- ตัวอย่างข้อมูลใน pdAdmin4

	product_key [PK] bigint	product_id text	product_name text	product_category text	product_size text	product_price numeric (10,2)	created_at timestamp without time zone	updated_at timestamp without time zone
1	321	P002	Vanilla Ice Cream - Medium	Ice Cream	Medium	55.00	2025-10-31 16:42:30.26717	[null]
2	322	P015	Caramel Shake - Small	Shake	Small	55.00	2025-10-31 16:42:30.26717	[null]
3	323	P011	Matcha Ice Cream - Medium	Ice Cream	Medium	60.00	2025-10-31 16:42:30.26717	[null]
4	324	P003	Vanilla Ice Cream - Large	Ice Cream	Large	65.00	2025-10-31 16:42:30.26717	[null]
5	325	P008	Strawberry Ice Cream - Medi...	Ice Cream	Medium	55.00	2025-10-31 16:42:30.26717	[null]
6	326	P016	Caramel Shake - Medium	Shake	Medium	65.00	2025-10-31 16:42:30.26717	[null]
7	327	P013	Banana Split Sundae	Sundae	[null]	85.00	2025-10-31 16:42:30.26717	[null]
8	328	P014	Brownie Sundae	Sundae	[null]	90.00	2025-10-31 16:42:30.26717	[null]
9	329	P005	Chocolate Ice Cream - Mediu...	Ice Cream	Medium	55.00	2025-10-31 16:42:30.26717	[null]
10	330	P009	Strawberry Ice Cream - Large	Ice Cream	Large	65.00	2025-10-31 16:42:30.26717	[null]
11	331	P007	Strawberry Ice Cream - Small	Ice Cream	Small	45.00	2025-10-31 16:42:30.26717	[null]
12	332	P001	Vanilla Ice Cream - Small	Ice Cream	Small	45.00	2025-10-31 16:42:30.26717	[null]
13	333	P020	Chocolate Shake - Large	Shake	Large	75.00	2025-10-31 16:42:30.26717	[null]
14	334	P017	Caramel Shake - Large	Shake	Large	75.00	2025-10-31 16:42:30.26717	[null]
15	335	P018	Chocolate Shake - Small	Shake	Small	55.00	2025-10-31 16:42:30.26717	[null]
16	336	P010	Matcha Ice Cream - Small	Ice Cream	Small	50.00	2025-10-31 16:42:30.26717	[null]
17	337	P004	Chocolate Ice Cream - Small	Ice Cream	Small	45.00	2025-10-31 16:42:30.26717	[null]
18	338	P019	Chocolate Shake - Medium	Shake	Medium	65.00	2025-10-31 16:42:30.26717	[null]
19	339	P012	Matcha Ice Cream - Large	Ice Cream	Large	70.00	2025-10-31 16:42:30.26717	[null]
20	340	P006	Chocolate Ice Cream - Large	Ice Cream	Large	65.00	2025-10-31 16:42:30.26717	[null]

### 3) dim\_customer — SCD Type 2



กระบวนการนี้ถูกออกแบบเพื่อรองรับ Slowly Changing Dimension ประเภทที่ 2 (SCD Type 2) สำหรับจัดการข้อมูลลูกค้าที่มีการเปลี่ยนแปลงเชิงคุณลักษณะ (attributes) ตามช่วงเวลา โดยยังคงเก็บประวัติย้อนหลัง เพื่อรองรับการวิเคราะห์พฤติกรรมลูกค้าในแต่ละช่วง

Flow นี้ถูกดำเนินการผ่าน Apache NiFi ด้วยชุด Processor ที่ร่วมกันตรวจจับการเปลี่ยนแปลง สร้างเวอร์ชันข้อมูลใหม่ และปิดเวอร์ชันเก่าอย่างเป็นระบบ

เป้าหมายของ SCD Type 2 ใน Customer

- เก็บประวัติการเปลี่ยนแปลงของลูกค้า (เช่น ระดับสมาชิก)
- ทำให้สามารถวิเคราะห์ย้อนหลังตามช่วงเวลาได้
- รักษาความถูกต้องของข้อมูลในเชิง Business Context

## 1. LookupRecord

เมื่อข้อมูลลูกค้าใหม่เข้ามา ระบบจะใช้ตัวประมวลผลนี้เพื่อตรวจสอบว่า **customer\_id** มีอยู่แล้วในตาราง **dim\_customer** หรือไม่

- หากไม่พบ → ถือเป็นลูกค้าใหม่
- หากพบ → ตรวจสอบข้อมูล attribute ปัจจุบัน

## 2. UpdateAttribute

ขั้นตอนนี้ใช้เพื่อกำหนดค่าช่วยเหลือใน Flow เช่น:

- การนับจำนวน record
- การกำหนด flag สำหรับเส้นทางข้อมูล

## 3. QueryRecord (detect\_valid\_id)

ใช้เกณฑ์ตรวจสอบความถูกต้องของข้อมูล เช่น:

- มีรหัสลูกค้า
- มีระดับสมาชิก
- ไม่มีค่า null สำคัญ

ฟิลเตอร์นี้ช่วยกันข้อมูลผิดรูปแบบไม่ให้เข้าสู่ตารางมิติ

## 4. QueryRecord (detect\_change)

เป็นหัวใจของ SCD2 โดยตรวจจับว่าค่าหลัก ๆ เปลี่ยนแปลงหรือไม่ เช่น:

`membership_level != previous_membership_level`

หากไม่มีการเปลี่ยนแปลง → เข้าสาย nothing แล้วจบ Flow ทันที

## 5. Previous Record Path (ปิดเวอร์ชันเก่า)

เมื่อพบว่า attribute มีการเปลี่ยน ระบบจะส่งข้อมูลเข้าสู่เส้นทาง previous, และดำเนินการ:

**UpdateRecord (previous\_is\_current)**

- ตั้ง is\_current = false
- กำหนด valid\_to = current\_date

จากนั้น PutDatabaseRecord จะอัปเดตแถวเก่าใน dim\_customer ให้เป็น historical snapshot ผลลัพธ์:

- แถวเดิมหยุดมีผลสำหรับการใช้งาน analytics ปัจจุบัน

## 6. New Version Path (เพิ่มแถวใหม่)

พร้อมกัน ระบบจะสร้าง แถวใหม่ เพื่อแทนค่าปัจจุบันด้วย:

- is\_current = true
- valid\_from = current\_date
- valid\_to = NULL
- ระดับสมาชิกล่าสุด

กระบวนการนี้ถูกทำผ่าน:

- UpdateRecord (new customer version)
- PutDatabaseRecord (insert into dim\_customer)

ผลลัพธ์:

- ลูกค้า 1 คนอาจมีหลายแถว (หลายเวอร์ชันตามเวลา)

## 7. กรณีลูกค้าใหม่

สำหรับข้อมูลที่ไม่พบใน LookupRecord:

- จะเข้าสู่ UpdateRecord (new customer)
- จากนั้น Insert ลงฐานโดยตรง

ไม่ต้องปิดเวอร์ชันใด

## 8. Notify → Wait Synchronization

Flow จะส่งสัญญาณด้วย Notify เพื่อแจ้งว่า:

มี customer version ใหม่ถูกประมวลผลสำเร็จ

Wait จะรอจนข้อความ notification ครบตามจำนวนที่กำหนดก่อนส่งข้อมูลออกจาก Process Group ไปยังขั้นตอนถัดไป เช่น Fact Loading เพื่อความถูกต้องของ Foreign Key mapping

- ตัวอย่างข้อมูลใน pdAdmin4

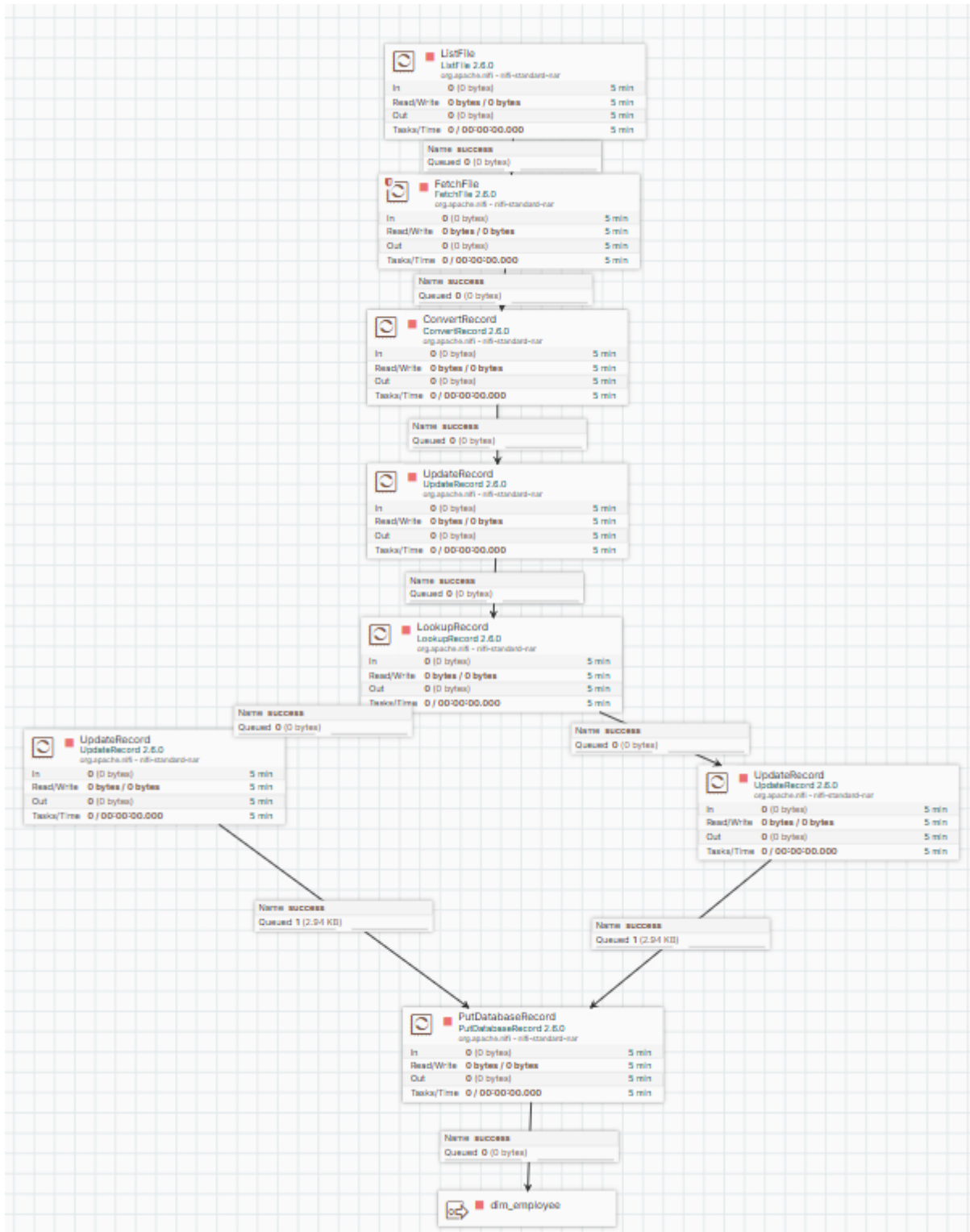
customer_sk [PK] bigint	customer_id text	customer_name text	gender text	customer_age integer	customer_phone character varying (20)	customer_city text	membership_level text	valid_from date	valid_to date	is_current boolean
1985	C129	customer_129	Female	28	844907127	Khon Kaen	Gold	2024-01-26	2024-10-...	true
1986	C129	customer_129	Female	28	844907127	Khon Kaen	Gold	2024-10-11	2025-03-...	true
1987	C129	customer_129	Female	28	844907127	Khon Kaen	Gold	2025-03-28	2262-04-...	true
1988	C008	customer_8	Male	56	870303849	Bangkok	Gold	2024-01-15	2024-03-...	true
1989	C008	customer_8	Male	56	870303849	Bangkok	Gold	2024-03-22	2024-11-...	true
1990	C008	customer_8	Male	56	870303849	Bangkok	Gold	2024-11-18	2024-12-...	true
1991	C008	customer_8	Male	56	870303849	Bangkok	Gold	2024-12-04	2025-07-...	true
1992	C008	customer_8	Male	56	870303849	Bangkok	Gold	2025-07-14	2262-04-...	true
1993	C128	customer_128	Male	41	864717184	Phuket	Gold	2024-05-16	2025-05-...	true
1994	C128	customer_128	Male	41	864717184	Phuket	Gold	2025-05-13	2025-07-...	true

## 4) dim\_employee — SCD Type 3

กระบวนการนี้เป็นส่วนหนึ่งของการโหลดข้อมูลเข้าสู่ตารางมิติเชิงบุคคลากร (dim\_employee)

ภายในระบบ Data Warehouse ซึ่งใช้รูปแบบ Slowly Changing Dimension ประเภทที่ 3 (SCD Type 3) เพื่อจัดการข้อมูลที่มีความจำเป็นต้องเก็บประวัติการเปลี่ยนแปลงแบบจำกัด โดยเฉพาะกรณี "การย้ายสาขา" ของพนักงาน

ในการดำเนินการนี้ Apache NiFi ถูกนำมาใช้เป็นเครื่องมือจัดการการไหลของข้อมูล การตรวจจับข้อมูลที่เปลี่ยนแปลง และการบันทึกหลักฐานข้อมูลตามเงื่อนไขที่กำหนด



### 1. ListFile

ขั้นตอนเริ่มต้น ทำหน้าที่ตรวจสอบไฟล์ข้อมูลพนักงานจากโฟลเดอร์ต้นทางเมื่อไฟล์ใหม่เข้ามา จะสร้าง FlowFile ตัวแทน metadata โดยยังไม่โหลดข้อมูล

### 2. FetchFile

หลังจากพบไฟล์ Processor นี้จะโหลดข้อมูลจริงเข้ามาใน FlowFile เพื่อให้พร้อมประมวลผล

### 3. ConvertRecord

แปลงรูปแบบข้อมูลให้อยู่ใน Schema ที่กำหนด เช่นจาก CSV → JSON รวมถึงทำ normalization data type เพื่อให้พร้อมใช้งาน

### 4. UpdateRecord (Data Cleansing / Restructure)

จัดรูปแบบฟิลด์ เช่น:

- แปลงชนิดข้อมูล (int, text)
- Trim whitespace
- จัดการค่า null

ช่วยลดปัญหาในขั้นตอน Lookup และ Update

### 5. LookupRecord

ตรวจสอบว่าพนักงานมีอยู่แล้วในตารางมิติหรือไม่โดยใช้ employee\_code เป็น Business Key

- หากไม่พบ → ถือเป็นพนักงานใหม่
- หากพบ → ตรวจสอบค่าที่เปลี่ยนแปลง เช่น store\_id

### 6. UpdateRecord (เมื่อพบข้อมูลใหม่)

กรณีเป็นพนักงานใหม่:

- ตั้งค่า current\_store\_id ด้วยค่าจาก source

- ตั้งค่า prev\_store\_id = NULL
- บันทึกลงฐานข้อมูลทันที (Insert)

เหมาะกับกรณีพนักงานเพิ่งเริ่มทำงาน

#### 7. UpdateRecord (เมื่อพบการย้ายสาขา)

หาก current\_store\_id != store\_id ใหม่

ระบบจะ:

- ย้ายค่าเดิม current\_store\_id ไปเก็บใน prev\_store\_id
- อัปเดตค่าใหม่ใส่ current\_store\_id
- ไม่สร้างแถวใหม่

Output = ข้อมูลเก่าคงอยู่ แต่ลดทอนประวัติย้อนหลังเหลือ 1 ชั้น

#### 8. PutDatabaseRecord

โปรเซสเซอร์นี้รับผิดชอบการบันทึกข้อมูลที่ถูกปรับปรุงแล้วกลับไปยัง dim\_employee แบบอัปเดตทับ  
รองรับการโหลดจำนวนมากแบบ Batch

#### เป้าหมายของ SCD Type 3 สำหรับ Employee

- ติดตามข้อมูลการย้ายสาขาของพนักงานย้อนหลัง 1 ชั้น
- วิเคราะห์ความสัมพันธ์ระหว่างเวลาทำงานกับสาขาประจำ
- ใช้พื้นที่จัดเก็บอย่างมีประสิทธิภาพกว่าการสร้างหลายเวอร์ชัน (เหมือน SCD2)

#### ข้อดีของ SCD Type 3

- ประหยัด storage
- Query ข้อมูลง่าย ไม่ต้อง join หลายเวอร์ชัน
- ดูข้อมูลก่อน-หลังของการย้ายสาขาได้ทันที

#### ข้อจำกัด

- เก็บประวัติย้อนหลังได้เพียง 1 เวอร์ชัน



- ไม่เหมาะหากมีการเปลี่ยนแปลงหลายครั้ง

ตัวอย่างข้อมูลใน pdAdmin4

employee_key [PK] bigint	employee_code integer	employee_name text	position text	phone text	hire_date date	salary numeric (12,2)	store_id integer	prev_store_id integer	store_changed_at date	created_at timestamp without time zone	updated_at timestamp without time zone
763	1	Somchai Prasert	Manager	9858612...	2020-06-06	35000.00	1	[null]	2025-11-03	2025-11-03 11:14:33.364889	2025-11-03 11:14:21
764	2	EM-002	Ice Cream Mak...	9229706...	2024-09-14	21046.00	1	[null]	2025-11-03	2025-11-03 11:14:33.364889	2025-11-03 11:14:21
765	3	EM-003	Barieta	9985915...	2021-12-25	20943.00	1	[null]	2025-11-03	2025-11-03 11:14:33.364889	2025-11-03 11:14:21
766	4	EM-004	Ice Cream Mak...	9546515...	2022-12-09	18816.00	1	[null]	2025-11-03	2025-11-03 11:14:33.364889	2025-11-03 11:14:21
767	5	EM-005	Ice Cream Mak...	9234545...	2024-04-15	21676.00	1	[null]	2025-11-03	2025-11-03 11:14:33.364889	2025-11-03 11:14:21

## 5) Synchronization via Wait / Notify

เพื่อควบคุมลำดับการทำงานของ Pipeline ใช้:

- Notify ส่งสัญญาณจากแต่ละ Dimension เมื่อโหลดเสร็จ
- Wait รวมสัญญาณทั้งหมดเพื่อให้แน่ใจว่ามีติทุกตัวถูกประมวลผลเรียบร้อย

สิ่งนี้ป้องกัน:

- Fact Table โหลดก่อนที่ Dimension จะพร้อม
- Foreign Key Missing
- Data Integrity Issues

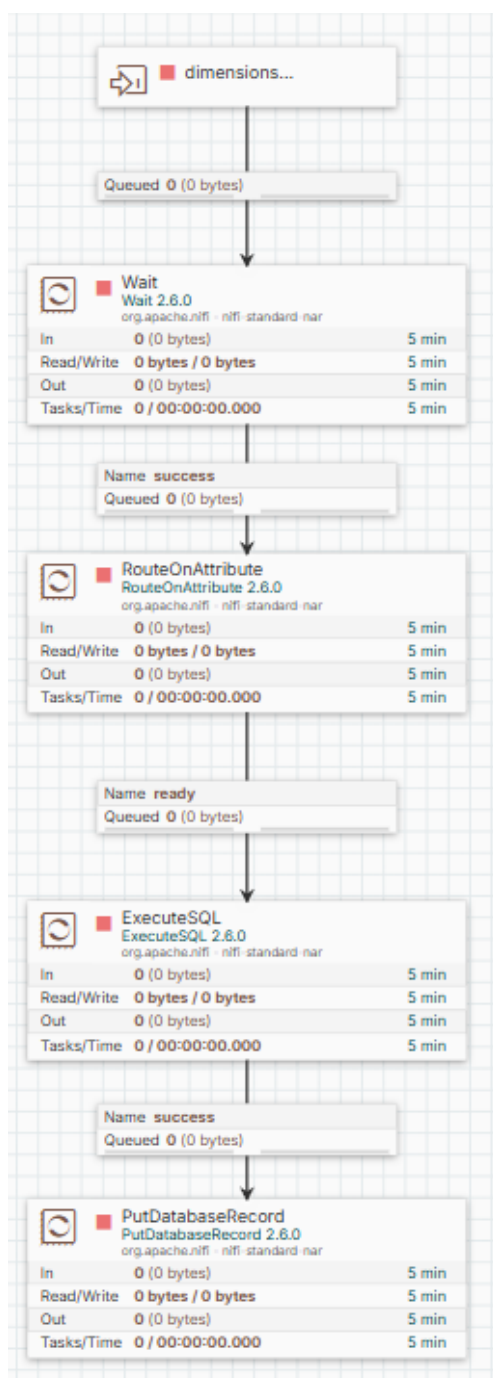
## 6) Output

เมื่อแต่ละ Dimension โหลดเสร็จ ระบบจะส่งต่อ FlowFile ไปที่ dimension\_ready ซึ่งเป็น trigger สำหรับ Fact Loader ในขั้นตอนถัดไปภายในสถาปัตยกรรม Data Warehouse

## สรุป

Flow หน้านี้เป็นโครงสร้างสำคัญของ Data Warehouse Layer ในส่วนของ Dimension Loading ออกแบบให้รองรับรูปแบบ Slowly Changing Dimension แบบหลายประเภท เพื่อสร้างข้อมูลมิติที่ถูกต้อง มีประวัติย้อนหลังตามความจำเป็น และพร้อมใช้งานในการ Join กับ Fact Table อย่างสมบูรณ์ ซึ่งช่วยให้การวิเคราะห์เชิงธุรกิจมีความแม่นยำและเชื่อถือได้

## Fact Loading Flow – Fact\_Sale



กระบวนการนี้เป็นขั้นตอนการโหลดข้อมูลเชิงธุรกรรม (Transaction Data) ลงสู่ตาราง Fact หลักของระบบ Data Warehouse คือ fact\_sale ซึ่งใช้สำหรับวิเคราะห์ยอดขายตามมิติต่าง ๆ เช่น สินค้า ลูกค้า สาขา และพนักงาน

Flow นี้ถูกออกแบบตามหลัก Data Warehouse Architecture เพื่อให้มั่นใจได้ว่า Dimension ทั้งหมดพร้อมและมี Surrogate Key ครบก่อนที่จะทำการโหลดธุรกรรม ซึ่งช่วยให้การอ้างอิงข้อมูล (Referential Integrity) ถูกต้องและสมบูรณ์

## 1. Trigger จาก Dimension Group (dimensions...)

Fact Flow เริ่มได้ก็ต่อเมื่อ:

- ทุก Dimension ประมวลผลเสร็จ
- Surrogate Key ถูกสร้างใน dim\_customer, dim\_product, dim\_employee, dim\_store

กระบวนการนี้ช่วยป้องกันปัญหา Fact orphan row (ธุรกรรมไม่มี Dimension อ้างอิง)

## 2. Wait

Processor นี้รอรับสัญญาณจาก **Notify** ของแต่ละ Dimension

Flow จะไม่ทำงานจนกว่า:

- จำนวนการแจ้งเตือนครบตามที่กำหนด
- ทุก Dimension Ready

ผลลัพธ์: ข้อมูลทุกมิติมีความสมบูรณ์ก่อนโหลด Fact

## 3. RouteOnAttribute

หลังจาก Wait ปล่อยข้อมูล FlowFile Processor นี้จะตรวจสอบ attribute เฉพาะ เช่น:

- สถานะ ready = true
- batch flag
- dimension\_sync\_complete

หากข้อมูลพร้อมจริง → ส่งต่อไปสู่ขั้นตอน ExecuteSQL

## 4. ExecuteSQL

Processor นี้ทำหน้าที่ ดึงข้อมูลธุรกรรมที่ผ่านการ Prepare แล้วจาก Staging เช่น:

- mapping surrogate key
- cleaning

- validation ผ่าน pipeline ก่อนหน้า

คำสั่ง SQL ประกอบด้วย:

- Join กับ dimension (surrogate key lookup)
- Aggregate (หากมี)
- Filter เฉพาะ record ที่ยังไม่ถูกโหลด

ผลลัพธ์คือชุดข้อมูล Fact ที่พร้อมบันทึกลง fact\_sale

## 5. PutDatabaseRecord

ขั้นตอนสุดท้ายของ Flow:

- บันทึก record ลงตาราง fact\_sale
- ใช้โหมด Insert แบบ Batch เพื่อเพิ่มประสิทธิภาพ
- รองรับโหลดจำนวนมาก (High Throughput)

ข้อมูลที่อยู่ใน Fact ประกอบด้วย:

- surrogate keys ของทุก Dimension
- measure เช่น quantity, amount, discount, satisfaction\_score
- time key

## คุณลักษณะสำคัญของกระบวนการนี้

- ป้องกันข้อมูลอ้างอิงไม่สมบูรณ์
- รองรับ Incremental Loading
- ออกแบบแบบ Batch Control
- ทำงานสัมพันธ์กับ SCD Logic ของ Dimension

## ประโยชน์เชิงวิเคราะห์

เมื่อ Fact สำเร็จ จะสามารถตอบคำถาม BI เช่น:

- ยอดขายต่อสินค้า/ช่วงเวลา
- ลูกค้ายอดนิยมสร้างรายได้สูงสุด
- พนักงานหรือสาขาใดมีประสิทธิภาพสูง
- Promotion ไหนมีผลเชิงบวกต่อยอดขาย

## สรุป Fact sale

Flow สำหรับ fact\_sale เป็นกระบวนการที่เชื่อมโยงข้อมูลธุรกรรมกับข้อมูลมิติต่าง ๆ โดยใช้กลไก

Synchronization ผ่าน Wait/Notify เพื่อป้องกันปัญหาด้านความถูกต้องเชิงสถาปัตยกรรม ทำให้ข้อมูล Fact ถูกโหลดอย่างครบถ้วน ถูกต้อง และพร้อมสำหรับการวิเคราะห์เชิงธุรกิจในระดับองค์กร

## สรุปภาพรวมการทำงานของระบบ Data Warehouse และ ETL Flow

ระบบนี้ถูกออกแบบเพื่อรองรับการโหลดข้อมูลจากต้นทางเข้าสู่ Data Warehouse ด้วยกระบวนการ ETL ผ่าน Apache NiFi พร้อมการจัดการความเปลี่ยนแปลงของข้อมูลมิติตามหลักการ Slowly Changing Dimension (SCD) เพื่อให้ข้อมูลมีความถูกต้อง สมบูรณ์ และวิเคราะห์ได้ตามช่วงเวลา

## การสร้าง Dashboard

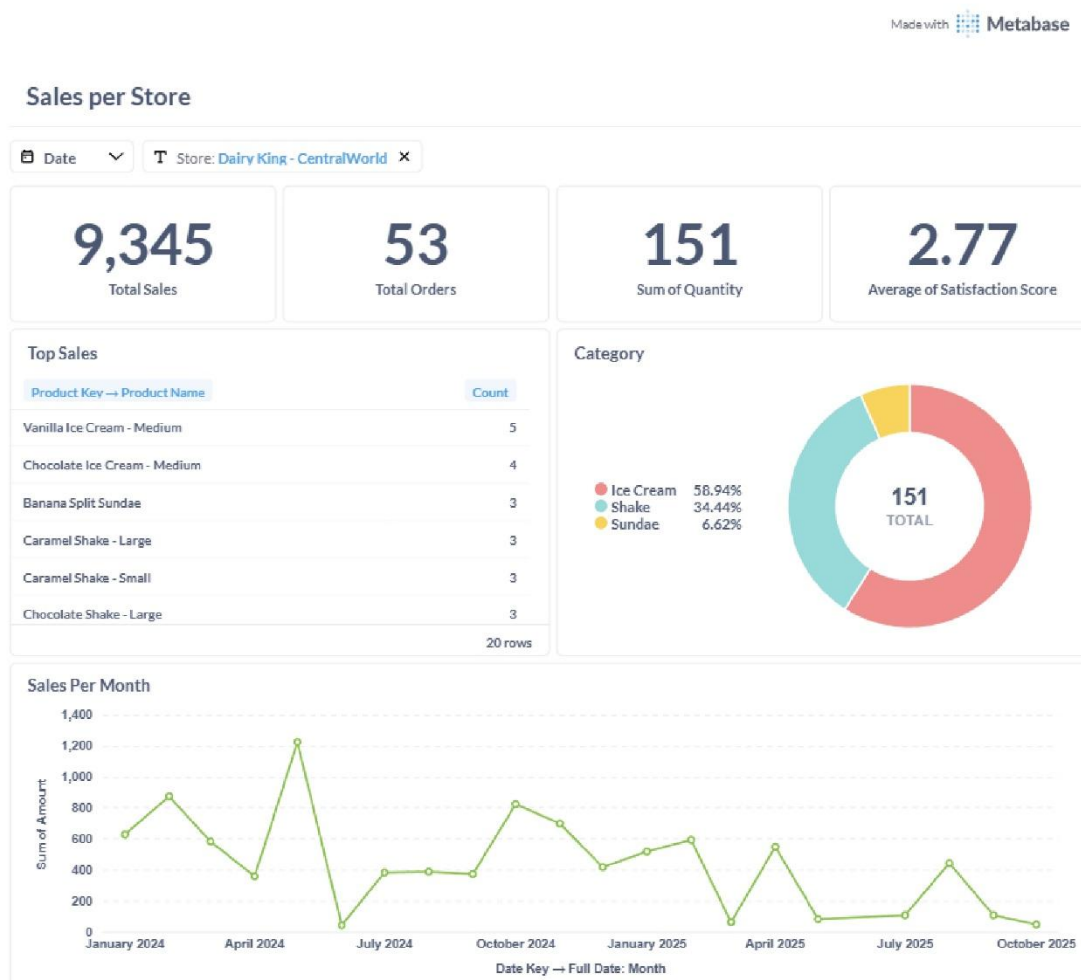
### เครื่องมือที่ใช้

Metabase ใช้งานง่าย รองรับหลายฐานข้อมูล มี filter / drill down / parameter



### ตัวอย่าง Dashboard

Dashboard แสดงยอดขายต่อสาขา



## การเชื่อมโยงกับ KPI

Dashboard “Sales per Store” ที่ยกตัวอย่างมา เชื่อมโยงกับตัวชี้วัดทางธุรกิจ (KPI) หลายด้าน

### 1. Subject Area: การขาย (Sales)

- KPI: ยอดขายรวม, ปริมาณการขาย, ยอดขายต่อบิล/วัน

Dashboard Component ที่เกี่ยวข้อง:

- Metric: *Total Sales, Total Orders, Sum of Quantity*
- Graph: *Sales per Month*

การเชื่อมโยง:

ข้อมูลยอดขายและจำนวนคำสั่งซื้อ มาจากตารางธุรกรรมการขายใช้ในการคำนวณ KPI เช่น “ยอดขายรวมต่อเดือน” และ “ยอดขายเฉลี่ยต่อวัน”

### 2. Subject Area: ลูกค้า (Customer)

- KPI: อัตราความพึงพอใจลูกค้า, การวิเคราะห์พฤติกรรมลูกค้า

Dashboard Component ที่เกี่ยวข้อง:

- Metric: *Average Satisfaction Score (2.77)*

การเชื่อมโยง:

คะแนนความพึงพอใจสะท้อนคุณภาพการบริการของสาขาซึ่งเป็น KPI สำคัญใน Business Process “การจัดการลูกค้าสัมพันธ์ (CRM)”

### 3. Subject Area: สินค้า (Product)

- KPI: ยอดขายตามประเภทสินค้า, การวิเคราะห์สินค้าขายดี

Dashboard Component ที่เกี่ยวข้อง:

- Table: *Top Sales* (เช่น Vanilla Ice Cream, Chocolate Ice Cream)
- Pie Chart: *Category* (Ice Cream 58.94%, Shake 34.44%, Sundae 6.62%)

การเชื่อมโยง:

กราฟเหล่านี้ใช้วัด KPI ด้าน “ยอดขายตามประเภทสินค้า” เพื่อระบุว่าสินค้ากลุ่มใดทำรายได้หลักให้สาขา

### 4. Subject Area: สาขา (Branch)

- KPI: ยอดขายของแต่ละสาขา, คะแนนพึงพอใจรายสาขา

Dashboard Component ที่เกี่ยวข้อง:

- Filter: *Store: Dairy King – CentralWorld*

การเชื่อมโยง:

Dashboard แสดงเฉพาะข้อมูลของสาขา CentralWorld เพื่อเปรียบเทียบผลการดำเนินงานระหว่างสาขาในภาพรวมขององค์กร



## บทสรุปและข้อเสนอแนะ

### บทสรุปของการทำ และ ส่วนเสนอแนะ

#### บทสรุปของการทำ

โครงการ การออกแบบคลังข้อมูล (Data Warehouse Design) สำหรับองค์กรจำลอง Dairy King มีจุดประสงค์เพื่อรวบรวมและจัดระเบียบข้อมูลจากหลายระบบ ได้แก่ ระบบขาย (POS), ระบบลูกค้าสัมพันธ์ (CRM), ระบบสินค้าคงคลัง และข้อมูลพนักงาน ให้รวมศูนย์อยู่ในฐานข้อมูลเดียว เพื่อสนับสนุนการวิเคราะห์เชิงธุรกิจและการตัดสินใจของผู้บริหารได้อย่างมีประสิทธิภาพ

การดำเนินงานประกอบด้วย 5 ส่วนหลัก ได้แก่

#### 1. การวิเคราะห์ความต้องการ

มีการระบุแหล่งข้อมูลต้นทาง ผู้ใช้งานเป้าหมาย และกำหนด Subject Area – KPI – Business Process ผ่าน BUS Matrix เพื่อให้เข้าใจบริบทของข้อมูลและแนวทางการออกแบบที่ตอบโจทย์เป้าหมายทางธุรกิจของ Dairy King เช่น การเพิ่มยอดขาย การรักษฐานลูกค้า และการเพิ่มประสิทธิภาพการดำเนินงานในแต่ละสาขา

#### 2. การออกแบบคลังข้อมูล

ใช้โครงสร้างแบบ Star Schema โดยมี fact\_sales เป็นตารางศูนย์กลาง เชื่อมโยงกับมิติต่าง ๆ เช่น dim\_customer, dim\_product, dim\_store, dim\_employee, dim\_promo เป็นต้น เพื่อให้สามารถวิเคราะห์ได้ในหลายมิติ ทั้งเชิงเวลา พื้นที่ และลูกค้า

#### 3. การจัดการข้อมูลมิติ (Slowly Changing Dimensions)

มีการกำหนดประเภทของ SCD ที่เหมาะสมกับลักษณะข้อมูลแต่ละมิติ เช่น ลูกค้าใช้ SCD Type 2 เพื่อเก็บประวัติย้อนหลัง ส่วนสินค้าใช้ SCD Type 1 สำหรับข้อมูลที่ต้องการเพียงค่าปัจจุบัน เพื่อให้โครงสร้างข้อมูลรองรับการเปลี่ยนแปลงตามธรรมชาติของธุรกิจได้อย่างถูกต้อง

#### 4. การออกแบบกระบวนการ ETL ด้วย Apache NiFi

ได้สร้าง Flow ครอบคลุมตั้งแต่การรับข้อมูลจากต้นทาง (Ingestion), การตรวจสอบคุณภาพข้อมูล (Validation), การโหลดข้อมูลเข้าสู่มิติ (Dimension Loading) และการสร้าง Fact Table (Fact Loading) โดยใช้กลไก Wait / Notify เพื่อควบคุมลำดับการทำงานให้มีความถูกต้องและครบถ้วน

## 5. การสร้าง Dashboard ด้วย Metabase

มีการออกแบบ Dashboard แสดงผลตัวชี้วัดสำคัญ (KPI) เช่น ยอดขายรวมต่อสาขา, ยอดขายเฉลี่ยต่อวัน, อัตราการใช้คุปอง และคะแนนความพึงพอใจของลูกค้า เพื่อให้ผู้บริหารสามารถติดตามผลการดำเนินงานของธุรกิจได้อย่างรวดเร็วและเข้าใจง่าย

จากการดำเนินงานทั้งหมด ระบบคลังข้อมูลของ Dairy King สามารถรวมศูนย์ข้อมูลจากหลายแหล่งได้สำเร็จ มีความถูกต้อง น่าเชื่อถือ และพร้อมต่อการวิเคราะห์เชิงกลยุทธ์ ซึ่งช่วยเพิ่มประสิทธิภาพในการตัดสินใจทางธุรกิจขององค์กรได้อย่างชัดเจน

## ข้อเสนอแนะ

1. เพิ่มแหล่งข้อมูลภายนอกเพื่อเพิ่มความแม่นยำของการวิเคราะห์  
เช่น การนำเข้าข้อมูลสภาพอากาศ หรือเทศกาล เพื่อช่วยพยากรณ์ยอดขายของผลิตภัณฑ์ไอศกรีมตามฤดูกาลได้ดียิ่งขึ้น
2. ปรับปรุง Data Quality ให้ครอบคลุมทุกขั้นตอน  
แม้ว่าระบบจะมีการตรวจสอบข้อมูลในขั้นตอน ETL แล้ว แต่ควรมีการติดตามคุณภาพข้อมูลอย่างต่อเนื่อง เช่น การตรวจหาค่าผิดปกติ (Outlier) หรือข้อมูลสูญหาย (Missing Value) ในตารางมิติหลัก
3. พัฒนา Dashboard ให้รองรับการวิเคราะห์เชิงลึกมากขึ้น  
ควรเพิ่มฟังก์ชัน Drill Down และ Filter ขั้นสูงใน Metabase เพื่อให้ผู้ใช้สามารถดูข้อมูลเชิงลึกในระดับสินค้า ลูกค้า หรือสาขาได้ในมุมมองที่เฉพาะเจาะจงมากขึ้น
4. ขยายขอบเขตของคลังข้อมูลในอนาคต  
เช่น การเพิ่ม Subject Area ใหม่ด้านต้นทุน (Cost) และการเงิน (Finance) เพื่อสนับสนุนการวิเคราะห์กำไรสุทธิและผลตอบแทนจากโปรโมชั่น (ROI) ได้ครบวงจรมากขึ้น
5. วางแผนการสำรองข้อมูลและการรักษาความปลอดภัย  
เนื่องจากข้อมูลลูกค้าและพนักงานเป็นข้อมูลสำคัญขององค์กร ควรกำหนดแนวทางสำรองข้อมูลอัตโนมัติ (Backup Policy) และกำหนดสิทธิ์การเข้าถึงข้อมูลในระดับผู้ใช้ เพื่อป้องกันการรั่วไหลของข้อมูลส่วนบุคคล