

ฐานข้อมูลธุรกิจร้าน With it
(With it store's Database)

จัดทำโดย

นางสาวชิตวรี	หอมผล	รหัสனிสิต 6142752026
นางสาวสุพิชชา	สวัสดีอักษรชื่น	รหัสனிสิต 6142923036

เสนอ

อาจารย์ประเสริฐ คณาวัฒน์ไชย

รายงานนี้เป็นส่วนหนึ่งของการเรียนรายวิชา
การใช้ฐานข้อมูลเพื่อการจัดการ
(Database Applications for Management)

จุฬาลงกรณ์มหาวิทยาลัย
ภาคการศึกษาต้น ปีการศึกษา 2563

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อประกอบการเรียนวิชา การใช้งานข้อมูลเพื่อการจัดการ หรือ Database Applications for Management โดยมีจุดประสงค์เพื่อการศึกษาความรู้ที่ได้จากเรื่องการใช้งานข้อมูลเพื่อการจัดการ ซึ่งรายงานนี้มีเนื้อหาเกี่ยวกับการเขียน SQL การทำ Primary Key และ Foreign Key การทำ Index การทำ Normalization ตลอดจนการประยุกต์ใช้ ODBC เพื่อเชื่อมต่อฐานข้อมูลกับ Microsoft Excel

ผู้จัดทำได้เลือกธุรกิจเสื้อผ้าแฟชั่น With it store เนื่องจากเป็นธุรกิจที่น่าสนใจ ผู้จัดทำขอขอบคุณอาจารย์ประเสริฐ คณาวัฒน์ไชย ผู้ให้ความรู้ และแนวทางการศึกษา รวมถึงทุกคนที่ให้ความช่วยเหลือมาโดยตลอด ผู้จัดทำหวังว่ารายงานฉบับนี้จะให้ความรู้ และเป็นประโยชน์แก่ผู้อ่านทุก ๆ ท่าน

คณะผู้จัดทำ

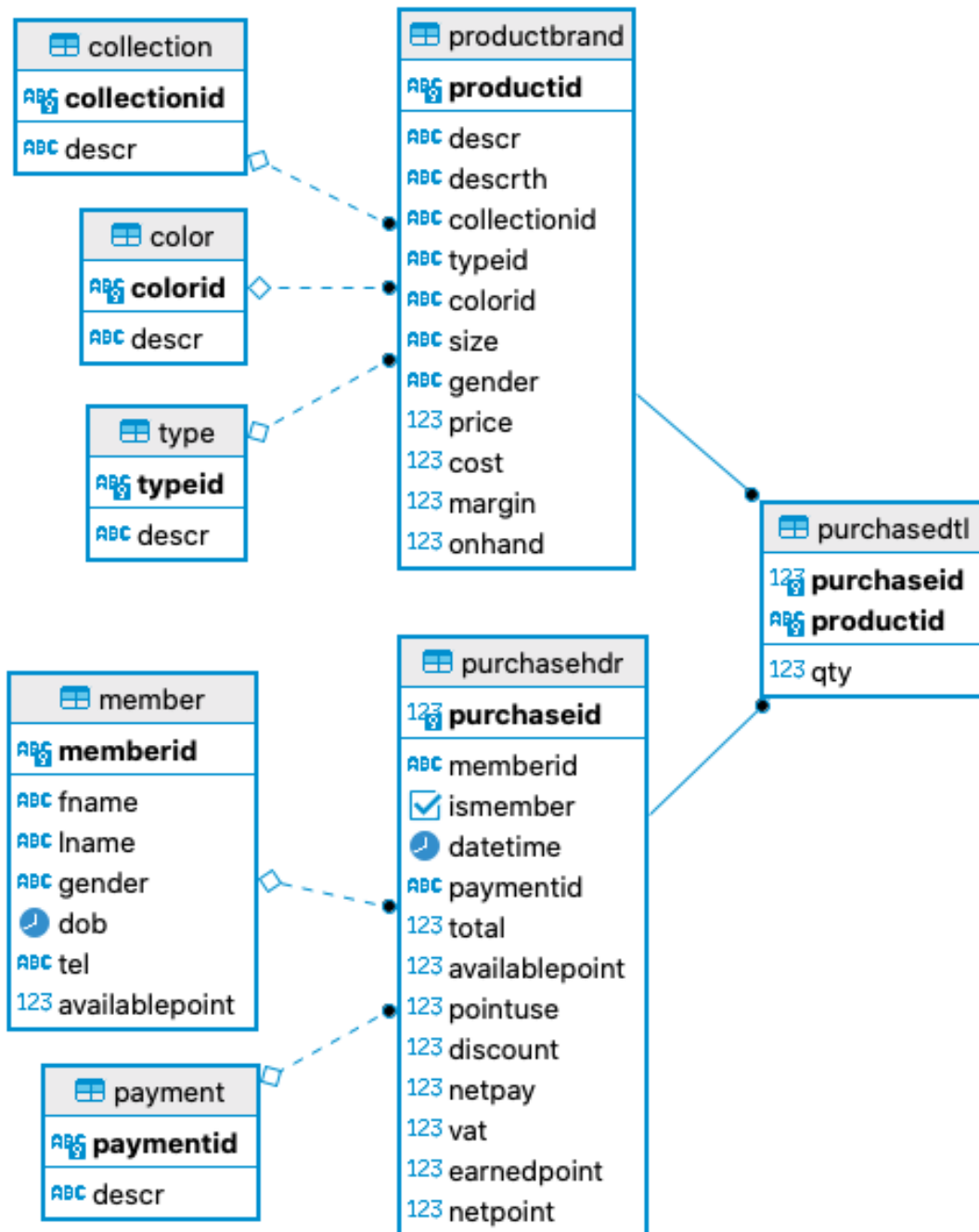
นางสาวชิตวรี หอมผล

นางสาวสุพิชชา สวัสดิ์อักษรชื่น

สารบัญ

ER Diagram.....	1
รายละเอียดตาราง ER Diagram.....	2
ตารางที่ 1 Member.....	2
ตารางที่ 2 Collection	2
ตารางที่ 3 Type.....	2
ตารางที่ 4 Color	3
ตารางที่ 5 Productbrand	3
ตารางที่ 6 Payment.....	3
ตารางที่ 7 Purchasedtl	4
ตารางที่ 8 Purchasehdr	4
การประยุกต์ใช้ฐานข้อมูล	5
รายงานที่ 1 สินค้าที่สร้างผลกำไรให้น้อยที่สุด 3 อันดับแรก.....	5
รายงานที่ 2 สินค้าที่ขายได้น้อยที่สุด 3 อันดับแรกของแต่ละคอลเล็กชัน	6
รายงานที่ 3 เสื้อผ้าแต่ละประเภท สีใดขายดีที่สุด.....	7
รายงานที่ 4 หาจำนวนครั้งที่ลูกค้าใช้จ่ายเกิน X,XXX บาท โดยแยกตามเดือน และปี.....	8
รายงานที่ 5 สรุปประเภทการชำระเงิน	9
รายงานที่ 6 คะแนนที่สมาชิกได้รับทั้งหมด	10
รายงานที่ 7 ลูกค้าที่เป็นสมาชิกอยู่ในช่วงวัยใด	11
รายงานที่ 8 รายได้ต่อเดือนในปี 2019 และ 2020	12
รายงานที่ 9 เปรียบเทียบรายได้ในแต่ละเดือนของปี 2019 และปี 2020.....	13
รายงานที่ 10 เปรียบเทียบรายได้ของแต่ละเดือนกับรายได้เฉลี่ยของปี 20XX.....	14
บรรณานุกรม.....	15

ER Diagram



รายละเอียดตาราง ER Diagram

ตารางที่ 1 Member

เก็บข้อมูลของลูกค้าที่เป็นสมาชิกกับทางร้าน โดยมี memberid เป็น Primary key

member
memberid
fname
lname
gender
dob
tel
availablepoint

คำอธิบายชื่อคอลัมน์

- memberid คือ รหัสสมาชิก
- fname (firstname) คือ ชื่อ
- lname (lastname) คือ นามสกุล
- gender คือ เพศ
- dob (date of birth) คือ วันเดือนปีเกิด
- tel (telephone number) คือ หมายเลขโทรศัพท์
- availablepoint คือ คะแนนหรือแต้มที่สมาชิกมีอยู่

ตารางที่ 2 Collection

เก็บข้อมูลชื่อคอลเล็กชันสินค้าของทางร้าน โดยมี collectionid เป็น Primary key

collection
collectionid
descr

คำอธิบายชื่อคอลัมน์

- collectionid คือ รหัสคอลเล็กชันของสินค้า
- descr (description) คือ คำอธิบาย collectionid

ตารางที่ 3 Type

เก็บข้อมูลชื่อประเภทสินค้าของทางร้าน โดยมี typeid เป็น Primary key

type
typeid
descr

คำอธิบายชื่อคอลัมน์

- typeid คือ รหัสประเภทของสินค้า
- descr (description) คือ คำอธิบาย typeid

ตารางที่ 4 Color

เก็บข้อมูลสีของสินค้าโดยมี Colorid เป็น Primary key

 color
 colorid
 descr

คำอธิบายชื่อคอลัมน์

- colorid คือ รหัสสีของสินค้า
- descr (description) คือ คำอธิบาย colorid

ตารางที่ 5 Productbrand

เก็บข้อมูลรายละเอียดของสินค้าภายในร้านทั้งหมด โดยมี productid เป็น Primary key มี collectionid จากตาราง Collection, typeid จากตาราง Type และ colorid จากตาราง Color เป็น Foreign key และมี descr และ descrth เป็น Unique index

 productbrand
 productid
 descr
 descrth
 collectionid
 typeid
 colorid
 size
 gender
 price
 cost
 margin
 onhand

คำอธิบายชื่อคอลัมน์

- productid คือ รหัสสินค้า
- descr (description) คือ ชื่อสินค้าเป็นภาษาอังกฤษ
- descrth (description in Thai) คือ ชื่อสินค้าเป็นภาษาไทย
- size คือ ขนาดของสินค้า
- gender คือ เสื้อผ้าสำหรับเพศใด
- price คือ ราคาสินค้า
- cost คือ ต้นทุนสินค้า
- margin คือ กำไรสินค้า
- onhand คือ สินค้าคงเหลือ

ตารางที่ 6 Payment

เก็บข้อมูลรูปแบบช่องทางการชำระเงินของทางร้าน มี paymentid เป็น Primary key

 payment
 paymentid
 descr

คำอธิบายชื่อคอลัมน์

- paymentid คือ รหัสรูปแบบช่องทางการชำระเงิน
- descr (description) คือ คำอธิบาย paymentid

ตารางที่ 7 Purchasedtl

เก็บข้อมูลรายละเอียดของสินค้าที่ซื้อในแต่ละคำสั่งซื้อ โดยมี purchaseid เป็น Primary key และมี productid จากตาราง Productbrand เป็น Foreign key















 purchasedtl
 purchaseid
 productid
 qty

คำอธิบายชื่อคอลัมน์

- purchaseid คือ รหัสคำสั่งซื้อ
- qty (quantity) คือ จำนวนการซื้อ

ตารางที่ 8 Purchasehdr

เก็บข้อมูลรายละเอียดคำสั่งซื้อ โดยมี purchaseid เป็น Primary key และมี memberid จากตาราง member และ paymentid จากตาราง payment ทำหน้าที่เป็น Foreign key

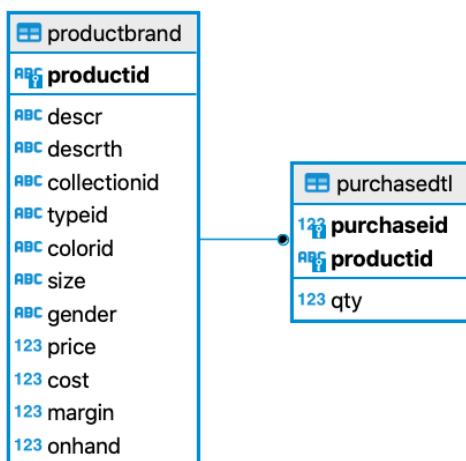
 purchasehdr
 purchaseid
 memberid
 ismember
 datetime
 paymentid
 total
 availablepoint
 pointuse
 discount
 netpay
 vat
 earnedpoint
 netpoint

คำอธิบายชื่อคอลัมน์

- ismember คือ เป็นสมาชิกหรือไม่
- datetime คือ วันและเวลาที่เกิดคำสั่งซื้อ
- total คือ ยอดการชำระเงิน
- availablepoint คือ คะแนนก่อนหน้าที่สมาชิกมีอยู่
- pointuse คือ คะแนนที่ถูกค่าใช้จ่ายเพื่อเป็นส่วนลด
- discount คือ ส่วนลดจากการใช้แต้ม
- netpay คือ ยอดการชำระเงินหลังหัก discount แล้ว
- vat คือ ภาษีมูลค่าเพิ่ม 7% ของราคาสินค้า
- earnedpoint คือ คะแนนที่ถูกค่าจะได้รับหลังชำระเงิน
- netpoint คือ คะแนนสุทธิหักหัก pointuse และรวมกับ earnedpoint แล้ว

การประยุกต์ใช้ฐานข้อมูล

รายงานที่ 1 สินค้าที่สร้างผลกำไรให้น้อยที่สุด 3 อันดับแรก



```

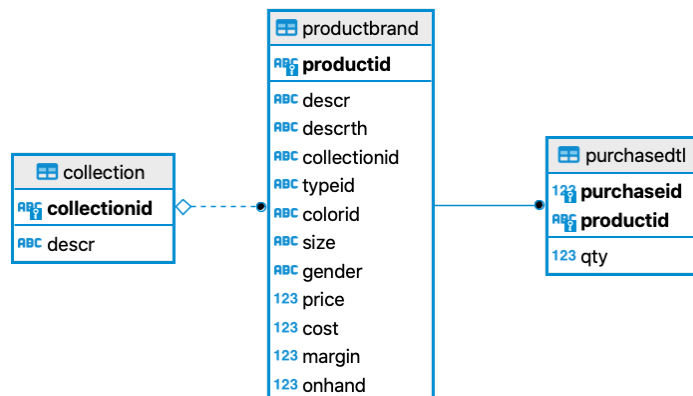
1 %%sql
2 select productid,b.descrth, sum(qty) "qty_sale" , b.margin
3   , to_char(b.margin*sum(qty),'999G999D99') total_margin
4   from purchasedtl a inner join productbrand b using(productid)
5  group by productid,b.descrth,b.margin
6  order by total_margin asc
7  limit 3
  
```

	productid	descrth	qty_sale	margin	total_margin
0	TP4061ORF	เสื้อยืดชั้นเซท (สีส้ม)	16	80	1,280.00
1	TP4057WHF	เสื้อยืดhappy money (สีขาว)	20	80	1,600.00
2	TP4058PIF	เสื้อยืด HEY GIRL (สีชมพู)	24	80	1,920.00

เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์ว่าสินค้าใดที่สร้างผลกำไรให้น้อยที่สุด 3 อันดับแรก และจากคำสั่งซื้อทั้งหมดลูกค้าซื้อสินค้า 3 อันดับนั้นเป็นเท่าใด

จากผลลัพธ์ข้างต้น มีการเชื่อม 2 ตารางเข้าด้วยกัน โดยตาราง Purchasedtl และ Productbrand จะถูกเชื่อมกันผ่านทางคอลัมน์ Productid เพื่อให้ทราบว่าสินค้าใดถูกซื้อเท่าไร และมี margin เป็นเท่าไร

รายงานที่ 2 สินค้าที่ขายได้น้อยที่สุด 3 อันดับแรกของแต่ละคอลเล็กชัน



```

1  %%sql
2  with cte as (
3  select c.descr "collection",b.descr "product_name", sum(p.qty) quantity
4  from purchasedtl p inner join productbrand b using(productid)
5       inner join collection c using(collectionid)
6  group by c.descr,b.descr
7  )
8  ,cte2 as (
9  select collection,product_name,quantity,
10         dense_rank() over (partition by collection order by quantity asc) "rank"
11         from cte
12 )
13 select * from cte2
14 where rank<=3
15 order by collection,rank
  
```

collection	product_name	quantity	rank
Jeans	Meyou Jean Shirt (Sky)	25	1
Jeans	Massy Jean Short Pant (Sky)	26	2
Jeans	Mori Jean Pant (Sky)	26	2
Jeans	Minno Jean Shirt (Sky)	26	2
Jeans	Massy Jean Short Pant (Navy)	37	3
Let's Go Camping	Grace Pants (Caramel Brown)	26	1
Let's Go Camping	Daily Pocket Line (White)	27	2
Let's Go Camping	Kelly Dress (Caramel Brown)	28	3
Let's Go Camping	Line Side Basic Pant (Cream)	28	3
Morning Garden	Lalinda Shirt (Cream)	22	1
Morning Garden	Amelai Taffy Dress (Pink)	23	2
Morning Garden	Matilda Butterfly (White)	23	2
Morning Garden	Matilda Butterfly (Cream)	24	3
T-Shirt	Sunset T-Shirt (Orange)	16	1
T-Shirt	Happy Money T-Shirt (White)	20	2
T-Shirt	Hey Girl T-Shirt (Pink)	24	3
Working	V Working Shirt (Brown)	21	1
Working	Cotton Working Shirt (Pink)	22	2
Working	Half Ruffle Working Blouse (Orange)	25	3

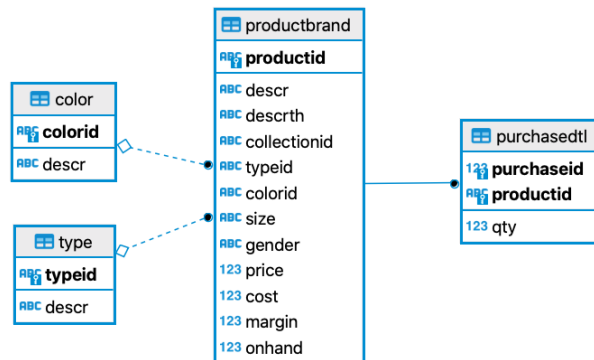
เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์ของสินค้าที่ขายได้น้อยที่สุดเป็น 3 อันดับแรกของแต่ละคอลเล็กชัน เพื่อพิจารณาว่าควรผลิตต่อในจำนวนเท่าใด หากไม่คุ้มการผลิตทางร้านจะอย่างไรต่อไป

จากผลลัพธ์ข้างต้น มีการเชื่อมโยง 3 ตารางเข้าด้วยกันโดย

- ตาราง Productbrand และ Collection จะถูกเชื่อมกันผ่านทางคอลัมน์ Collectionid เพื่อให้ทราบว่าสินค้าที่ถูกค้าซื้อเป็นเสื้อผ้าคอลเล็กชันใด และสามารถแสดงชื่อเต็มของคอลเล็กชันเสื้อผ้าได้
- ตาราง Purchasedtl และ Productbrand จะถูกเชื่อมกันผ่านทางคอลัมน์ Productid เพื่อให้ทราบว่าในแต่ละ

ละ Purchaseid มีการซื้อสินค้าสินค้าใด เป็นจำนวนเท่าใด และสามารถแสดงชื่อเต็มของสินค้าได้

รายงานที่ 3 เสื้อผ้าแต่ละประเภท สีใดขายดีที่สุด



```

1  %%sql
2  with cte as (
3  select t.descr "type", c.descr "color", sum(p.qty) "quantity"
4  from purchasedtl p inner join productbrand using(productid)
5       inner join type t using(typeid) inner join color c using(colorid)
6  group by t.descr, c.descr
7  )
8  , cte2 as (
9  select type, color, quantity
10         , rank() over (partition by type order by quantity desc)
11         from cte
12 order by type, quantity desc
13 )
14 select type, color, quantity from cte2 where rank = 1 order by quantity desc
  
```

	type	color	quantity
0	Top	Cream	573
1	Pants	Cream	469
2	Dress	Cream	176
3	Skirt	Cream	40
4	Jumpsuit	Cream	37
5	Jumpsuit	Ash Brown	37

เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์ว่าเสื้อผ้าแต่ละประเภท สีใดขายดีที่สุด เพื่อจะได้รู้ความต้องการอย่างตรงจุดว่าลูกค้าส่วนมากชอบสีใดในประเภทสินค้านั้น ๆ ในคอลเล็กชันต่อไปหรือของล็อตถัดไปจะได้ออกแบบหรือผลิตให้ตรงต่อความต้องการของลูกค้า

จากผลลัพธ์ข้างต้น มีการเชื่อม 4 ตารางเข้าด้วยกัน โดย

- ตาราง Productbrand และ Type จะถูกเชื่อมกันผ่านทางคอลัมน์ Typeid เพื่อให้ทราบว่าสินค้าที่ลูกค้าซื้อเป็นเสื้อผ้าประเภทใด และสามารถแสดงชื่อเต็มของประเภทเสื้อผ้าได้
- ตาราง Type และ Color จะถูกเชื่อมกันผ่านทางคอลัมน์ Colorid เพื่อให้สามารถแสดงชื่อเต็มของสีเสื้อผ้าที่ลูกค้าได้ทำการซื้อ
- ตาราง Purchasedtl และ Productbrand จะถูกเชื่อมกันผ่านทางคอลัมน์ Productid เพื่อให้ทราบว่าในแต่ละ Purchaseid มีการซื้อสินค้าสินค้าใด และเป็นจำนวนเท่าใด

รายงานที่ 4 หาจำนวนครั้งที่ลูกค้าใช้จ่ายเงิน X,XXX บาท โดยแยกตามเดือน และปี

```

1  %%sql
2  drop function if exists yc;
3  CREATE FUNCTION yc(year_ int, netpay_ int)
4  RETURNS TABLE (
5      _year_ int,
6      _month_ text,
7      _count_ bigint
8  )
9  AS $$
10 BEGIN RETURN QUERY
11 select extract(year from datetime)::int "year"
12        ,to_char(datetime,'Mon') "month"
13        ,count(netpay) as Count
14        from purchasehdr
15        where netpay>netpay_ and extract(year from datetime)::int=year_
16        group by year,month,extract(month from datetime)
17        order by year,extract(month from datetime);
18 END; $$
19 LANGUAGE 'plpgsql';

```

```
1  %%sql select * from yc (2020,1000)
```

(ฟังก์ชันเพื่อให้ผู้ใช้ระบุปีและค่าที่ต้องการทราบ)

year	_month_	_count_
2020	Jan	4
2020	Feb	4
2020	Mar	10
2020	Apr	9
2020	May	5
2020	Jun	5
2020	Jul	11
2020	Aug	9
2020	Sep	6
2020	Oct	10
2020	Nov	7
2020	Dec	5

เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์ว่าลูกค้าที่ใช้จ่ายเงินค่าที่เรากำหนด ในที่นี้คือ 1,000 บาท เกินกี่คนในปีที่เรากำหนด ในที่นี้คือปี 2020 เพื่อดูแนวโน้มความเป็นไปได้เมื่อทางร้านจะจัดโปรโมชั่นต่าง ๆ

จากผลลัพธ์ มีการทำฟังก์ชันให้ผู้ใช้งานระบุปีและค่าที่ต้องการทราบ โดยใช้ตาราง Purchasehdr ในการดึงข้อมูลการใช้จ่ายลูกค้าในแต่ละวันของปีที่เราต้องการจะนำข้อมูลมาวิเคราะห์

รายงานที่ 5 สรุปประเภทการชำระเงิน

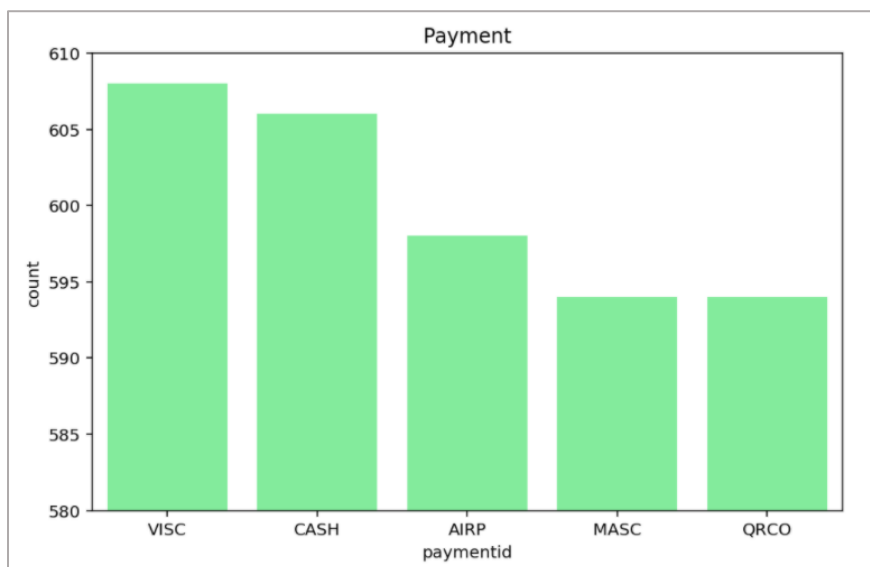
```
%sql
payment_df <<
select paymentid from purchasehdr
```

```
* postgresql+psycpg2://postgres:***@localhost:5432/withit
3000 rows affected.
Returning data to local variable payment_df
```

payment_df.head()

	purchaseid	paymentid
0	1	QRCO
1	2	MASC
2	3	VISC
3	4	AIRP
4	5	CASH

```
1 plt.figure(figsize=(8,5))
2 plt.title('Payment')
3 plt.ylim(580,610)
4 sns.countplot(x='paymentid',data=payment_df,color='lightgreen'
5               ,order = payment_df['paymentid'].value_counts().index)
```



เป็นกราฟแท่งที่สรุปประเภทช่องทางการชำระเงินว่าลูกค้าชำระจากช่องทางใดมากที่สุดไปน้อยที่สุด จากผลลัพธ์ ใช้ตาราง Purchasehdr เพื่อดึงข้อมูลว่าลูกค้าแต่ละคนใช้ช่องทางใดในการชำระเงิน

รายงานที่ 6 คะแนนที่สมาชิกได้รับทั้งหมด

```
1 %%sql
2 select memberid, sum(earnedpoint) "earned_point"
3   from purchasehdr
4  group by memberid
5  order by earned_point desc
6  limit 10
```

memberid	earned_point
WI38	325
WI16	295
WI47	290
WI19	285
WI22	285
WI12	280
WI35	260
WI11	255
WI36	250
WI30	245

เป็นตารางที่แสดงคะแนนที่สมาชิกได้รับทั้งหมด โดยจากผลลัพธ์ ใช้ตาราง Purchasehdr เพื่อดึงข้อมูลว่าสมาชิกคนใด ได้รับคะแนนไปเท่าไร

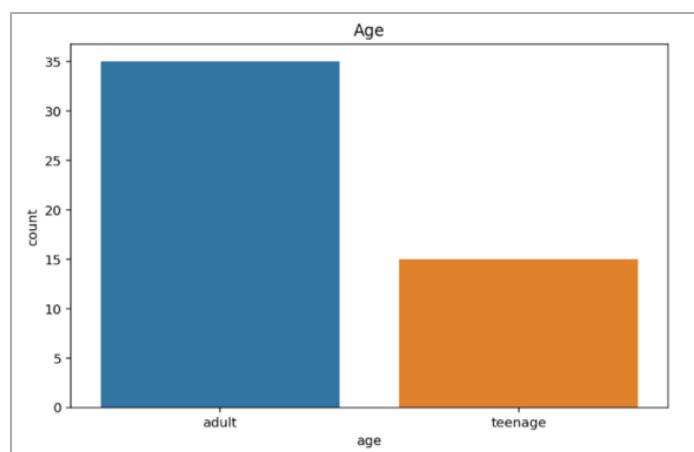
รายงานที่ 7 ลูกค้าที่เป็นสมาชิกอยู่ในช่วงวัยใด

```

1 %%sql
2 age_df <<
3 select case
4     when extract(year from dob)::int between 2008 and 2018 then 'child'
5     when extract(year from dob)::int between 2000 and 2007 then 'teenage'
6     when extract(year from dob)::int between 1960 and 1999 then 'adult'
7     when extract(year from dob)::int > 1960 then 'elder'
8 end as age
9     ,count(*)
10    from member
11   group by age

```

1	age_df
	age count
0	adult 35
1	teenage 15

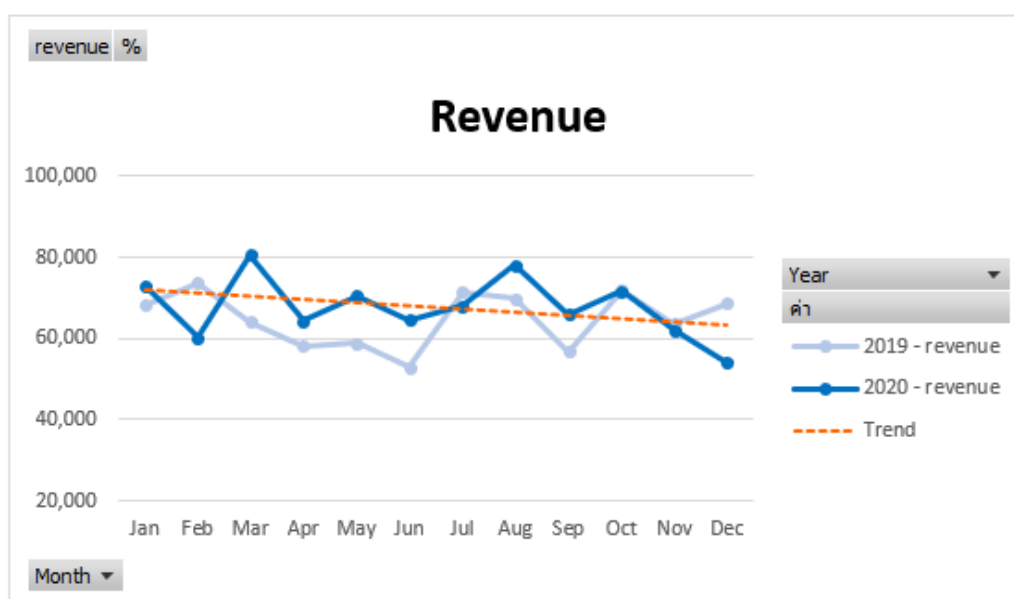


เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์ว่าสมาชิกของร้านอยู่ในช่วงวัยใดบ้าง เพื่อจะได้รู้ว่าควรออกแบบเสื้อผ้าสไตล์ใดเพิ่ม เช่น หากมีสมาชิกที่อยู่ในวัยทำงานเยอะ ทางร้านควรออกเสื้อผ้าแบบใหม่ ๆ เพิ่มในคอลเล็กชัน working

โดยจากผลลัพธ์ ใช้ตาราง Member เพื่อดึงข้อมูลวันเกิดของสมาชิกว่าอยู่ในช่วงปีใดบ้าง

รายงานที่ 8 รายได้ต่อเดือนในปี 2019 และ 2020

Year ▾							
2019		2020					
Month ▾	revenue %	revenue %	revenue %	revenue %	ผลรวม revenue	ผลรวม %	
Jan	68,190 8.76%	72,960 8.97%			141,150	8.87%	
Feb	73,820 9.48%	60,280 7.41%			134,100	8.43%	
Mar	64,100 8.23%	80,570 9.91%			144,670	9.09%	
Apr	58,080 7.46%	64,300 7.91%			122,380	7.69%	
May	58,950 7.57%	70,610 8.68%			129,560	8.14%	
Jun	52,940 6.80%	64,690 7.95%			117,630	7.39%	
Jul	71,380 9.17%	68,140 8.38%			139,520	8.77%	
Aug	69,850 8.97%	77,980 9.59%			147,830	9.29%	
Sep	56,750 7.29%	65,890 8.10%			122,640	7.71%	
Oct	71,980 9.25%	71,599 8.80%			143,579	9.02%	
Nov	63,660 8.18%	62,020 7.63%			125,680	7.90%	
Dec	68,760 8.83%	54,180 6.66%			122,940	7.72%	
Total	778,460 100.00%	813,219 100.00%			1,591,679	100.00%	



เป็นตารางและกราฟที่แสดงให้เห็นรายได้ต่อเดือนในปี 2019 และปี 2020 และแนวโน้ม(Trend)ของรายได้ในปี2020

จากผลลัพธ์ใช้ตาราง Purchasehdr เพื่อดึงข้อมูลรายได้(Netpay)แต่ละเดือนของทั้ง 2 ปี โดยใช้ ODBC (Open Database Connectivity) เพื่อเชื่อมข้อมูลเข้า Microsoft Excel แล้วนำข้อมูลไปวิเคราะห์ Pivot Table

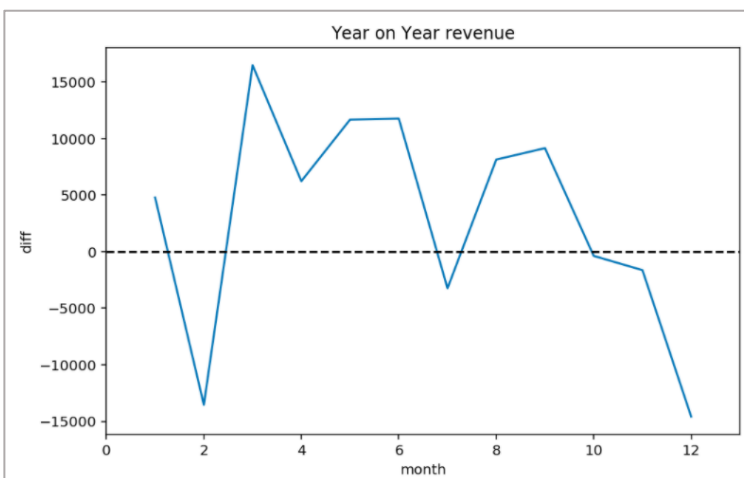
รายงานที่ 9 เปรียบเทียบรายได้ในแต่ละเดือนของปี 2019 และปี 2020

```

1 %%sql
2 revenue_df <-
3 with cte as (
4 select extract (month from datetime)::int "month" ,sum(netpay) "revenue2019" from purchasehdr
5   where extract (year from datetime)=2019
6   group by month
7 ), cte2 as (
8 select extract (month from datetime)::int "month" ,sum(netpay) "revenue2020" from purchasehdr
9   where extract (year from datetime)=2020
10  group by month
11 ), cte3 as (
12 select a.month, revenue2019
13        ,revenue2020
14        ,revenue2020-revenue2019 "diff"
15   from cte a inner join cte2 b using(month)
16   group by a.month, revenue2019, revenue2020
17   order by a.month
18 )
19 select * from cte3

```

	month	revenue2019	revenue2020	diff
0	1	68190.00	72960.00	4770.00
1	2	73820.00	60280.00	-13540.00
2	3	64100.00	80570.00	16470.00
3	4	58080.00	64300.00	6220.00
4	5	58950.00	70610.00	11660.00
5	6	52940.00	64690.00	11750.00
6	7	71380.00	68140.00	-3240.00
7	8	69850.00	77980.00	8130.00
8	9	56750.00	65890.00	9140.00
9	10	71980.00	71599.00	-381.00
10	11	63660.00	62020.00	-1640.00
11	12	68760.00	54180.00	-14580.00



เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์การเปรียบเทียบรายได้ในแต่ละเดือนของปี 2019 และปี 2020 และนำผลการเปรียบเทียบไปแสดงผลในรูปแบบกราฟเส้น ข้อมูลที่อยู่บนเส้นประ คือ รายได้ในปี 2020 สูงกว่าปี 2019 ในทางตรงกันข้าม หากข้อมูลอยู่ใต้เส้นประ คือ รายได้ในปี 2020 ต่ำกว่าปี 2019

จากตารางและกราฟสามารถดูได้ว่าในเดือนนั้น ๆ มีเหตุการณ์ใดที่ทำให้ลูกค้ามาซื้อมากกว่าหรือน้อยกว่าเดือนอื่น ๆ นำไปสู่การพิจารณาถึงปีถัดไปว่าควรจัดโปรโมชั่นหรือจัดอีเวนต์ใดเพื่อรักษายอดขายให้ไม่ต่ำไปมากกว่าเดิม

จากผลลัพธ์ ใช้ตาราง Purchasehdr เพื่อดึงข้อมูลว่าในแต่ละเดือนของทั้ง 2 ปี มียอดขายเป็นเท่าไร

รายงานที่ 10 เปรียบเทียบรายได้ของแต่ละเดือนกับรายได้เฉลี่ยของปี 20XX

```

1  %sql
2  avg_df <<
3  drop function if exists cavg;
4  CREATE FUNCTION cavg(year_ int)
5  RETURNS TABLE (
6      month_text,
7      total_revenue numeric(10,2),
8      average_revenue numeric(10,2)
9  )
10 AS $$
11 BEGIN RETURN QUERY
12 with snp as (
13     select extract(year from datetime)::int "yearno", extract(month from datetime)::int "monthno"
14           ,to_char(datetime,'Mon') "month",sum(netpay) revenue
15       from purchasehdr
16      group by month,monthno,yearno
17 )
18 select month,revenue
19       ,(select avg(revenue) from snp where yearno=year_)::numeric(10,2) "avg_revenue"
20   from snp
21  where yearno=year_
22  order by monthno;
23 END; $$
24 LANGUAGE 'plpgsql';

```

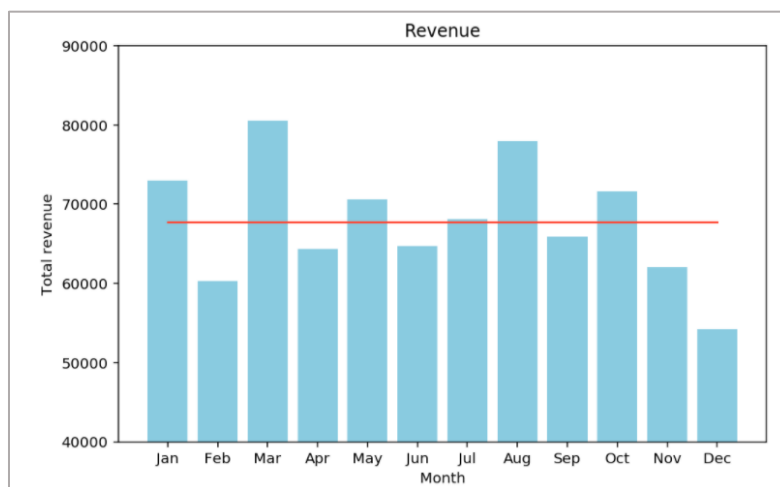
```

1  %sql
2  avg_df <<
3  select * from cavg (2020)

```

(ฟังก์ชันเพื่อให้ผู้ใช้ระบุปีที่ต้องการทราบ)

1	avg_df
	month_ total_revenue average_revenue
0	Jan 72960.00 67768.25
1	Feb 60280.00 67768.25
2	Mar 80570.00 67768.25
3	Apr 64300.00 67768.25
4	May 70610.00 67768.25
5	Jun 64690.00 67768.25
6	Jul 68140.00 67768.25
7	Aug 77980.00 67768.25
8	Sep 65890.00 67768.25
9	Oct 71599.00 67768.25
10	Nov 62020.00 67768.25
11	Dec 54180.00 67768.25



เป็นตารางที่แสดงให้เห็นถึงผลลัพธ์ว่ารายได้ในแต่ละเดือนของปีที่เราระบุ ในที่นี้คือ 2020 คือเท่าไร และกราฟแท่งแสดงให้เห็นว่ารายได้ในแต่ละเดือนนั้นต่ำหรือสูงกว่ารายได้เฉลี่ยของปีที่ระบุ โดยเส้นสีแดงคือรายได้เฉลี่ย

จากผลลัพธ์ มีการทำฟังก์ชันให้ผู้ใช้งานระบุปีที่ต้องการทราบ โดยใช้ตาราง Purchasehdr ในการดึงข้อมูลรายได้(Netpay)ในแต่ละเดือนของปีที่เราต้องการนำมาวิเคราะห์

บรรณานุกรม

Shopee. With.it.store. จาก <https://shopee.co.th/with.it.store?page=0&shopCollection=38932770>

Youtube. (9 มกราคม 2562). การสร้างกราฟด้วย seaborn: สร้างกราฟนับความถี่ข้อมูลแบบตัวอักษร category และ discrete ด้วย countplot [ไฟล์วิดีโอ]. จาก <https://youtu.be/TJ2xK3AV5RQ>

Youtube. (12 มกราคม 2562). สร้างกราฟด้วย seaborn: การสร้างกราฟแท่งด้วย barplot [ไฟล์วิดีโอ]. จาก https://youtu.be/A9U_IRSRHyE

Youtube. (28 กุมภาพันธ์ 2562). การสร้างกราฟด้วย seaborn: การสร้างกราฟเส้นด้วย lineplot และ relplot [ไฟล์วิดีโอ]. จาก <https://youtu.be/MgL4aJA0li0>