三 使用CASE WHEN 和 GROUP BY将数据分组

> 为了计算更复杂的业务指标，下面将练习如何根据自己的标准对业务对象进行“分类和计数”

学习目标

- 熟练掌握CASE WHEN 的使用方法

1 使用CASE WHEN自定义分组

- 需求：我们要在报表中显示每种产品的库存量，但我们不想简单地将“ units_in_stock”列放在报表中。报表中只需要一个总体级别，例如低，高：

```
SELECT
  product_id,
  product_name,
  units_in_stock,
  CASE
    WHEN units_in_stock > 100 THEN 'high'
    WHEN units_in_stock > 50 THEN 'moderate'
    WHEN units_in_stock > 0 THEN 'low'
    WHEN units_in_stock = 0 THEN 'none'
  END AS availability
FROM products;
```

- 上面的SQL查询结果中，我们创建了一个新列 `availability` ， 通过 `CASE WHEN` 语句来对这一列赋值
- `CASE WHEN` 语法回顾
- 上面的查询中，通过 `units_in_stock` 列的值来判断库存的可用性
  - 库存大于100 的可用性为高(high)

    - 50到100的可用性为中等(moderate)
    - 小于50的为低(low)
    - 零库存 为 (none)

练习21

运行上面的SQL，比较 `units_in_stock` 和 `availability` 两列的结果

product_id	product_name	units_in_stock	availability
1	Chai	39	low
2	Chang	17	low
3	Aniseed Syrup	13	low
4	Chef Anton's Cajun Seasoning	53	moderate
5	Chef Anton's Gumbo Mix	0	none
6	Grandma's Boysenberry Spread	120	high
7	Uncle Bob's Organic Dried Pears	15	low
8	Northwoods Cranberry Sauce	6	low
9	Mishi Kobe Niku	29	low
10	Ikura	31	low
11	Queso Cabrales	22	low
12	Queso Manchego La Pastora	86	moderate
13	Konbu	24	low
14	Tofu	35	low
15	Genen Shouyu	39	low
16	Pavlova	29	low
17	Alice Mutton	0	none
18	Carnarvon Tigers	42	low
19	Teatime Chocolate Biscuits	25	low
20	Sir Rodney's Marmalade	40	low

练习22

- 需求：创建一个报表，统计员工的经验水平
- 显示字段：`first_name`，`last_name`，`hire_date`，和 `experience`
- 经验字段（`experience`）：
    - `'junior'` 2014年1月1日以后雇用的员工
    - `'middle'` 在2013年1月1日之后至2014年1月1日之前雇用的员工
    - `'senior'` 2013年1月1日或之前雇用的员工

```sql
SELECT
  first_name,
  last_name,
  hire_date,
  CASE
    WHEN hire_date > '2014-01-01' THEN 'junior'
    WHEN hire_date > '2013-01-01' THEN 'middle'
    WHEN hire_date <= '2013-01-01' THEN 'senior'
  END AS experience
FROM employees;
```

结果

first_name	last_name	hire_date	experience
Nancy	Davolio	2012-05-01	senior
Andrew	Fuller	2012-08-14	senior
John	Smith	2012-04-01	senior
Margaret	Peacock	2013-05-03	middle
Steven	Buchanan	2013-10-17	middle
Michael	Suyama	2013-10-17	middle
Robert	King	2014-01-02	junior
Laura	Callahan	2014-03-05	junior
Anne	Dodsworth	2014-11-15	junior
John	Smith	2017-03-21	junior

2 CASE WHEN中ELSE的使用

- 我们的商店要针对北美地区的用户做促销活动：任何运送到北美地区（美国，加拿大）的包裹免运费。
- 创建报表，查询订单编号为10720~10730 活动后的运费价格

```
SELECT
  order_id,
  customer_id,
  ship_country,
  CASE
    WHEN ship_country = 'USA' OR ship_country = 'Canada' THEN
0.0
  END AS shipping_cost
FROM orders
WHERE order_id BETWEEN 10720 AND 10730;
```

- 上面的SQL中，只定义了美国和加拿大的运费，并没有处理其他目的地的运费信息

练习23

- 运行上面的SQL 观察 `ship_country` 和 `shipping_cost` 列，除了美国和加拿大之外，其他行的 `shipping_cost` 的值为NULL

order_id	customer_id	ship_country	shipping_cost
10720	QUEDE	Brazil	null
10721	QUICK	Germany	null
10722	SAVEA	USA	0.0
10723	WHITC	USA	0.0
10724	MEREP	Canada	0.0
10725	FAMIA	Brazil	null
10726	EASTC	UK	null
10727	REGGC	Italy	null
10728	QUEEN	Brazil	null
10729	LINOD	Venezuela	null
10730	BONAP	France	null

- 在上面的案例中，除了北美地区的以外的订单，运费统计为NULL,如果将其他地区的运费设置为10美元，那么可以用如下方式处理：

```sql
SELECT
  order_id,
  customer_id,
  ship_country,
  CASE
    WHEN ship_country = 'USA' OR ship_country = 'Canada' THEN
0.0
    ELSE 10.0
  END AS shipping_cost
FROM orders
WHERE order_id BETWEEN 10720 AND 10730;
```

- 我们在 `CASE WHEN` 结构中添加了 `ELSE`

- 如果不满足其他条件，则执行 `ELSE` 。 因此，所有其他国家/地区的 `shipping_cost` 都将变为“ 10.0”，而不是 `NULL` 。

练习24

- 需求：创建客户基本信息报表
- 包含字段：
    - 客户id `customer_id`
    - 公司名字 `company_name`
    - 所在国家 `country`
    - 使用语言 `language`
- 使用语言 `language` 的取值按如下规则
    - Germany, Switzerland, and Austria 语言为德语 `'German'`
    - UK, Canada, the USA, and Ireland 语言为英语 `'English'`
    - 其他所有国家 `'Other'`

```sql
SELECT
  customer_id,
  company_name,
  country,
  CASE
    WHEN country IN ('Germany', 'Switzerland', 'Austria') THEN 'German'
    WHEN country IN ('UK', 'Canada', 'USA', 'Ireland') THEN 'English'
    ELSE 'Other'
  END AS language
FROM customers;
```

查询结果

customer_id	company_name	country	language
ALFKI	Alfreds Futterkiste	Germany	German
ANATR	Ana Trujillo Emparedados y helados	Mexico	Other
ANTON	Antonio Moreno Taquería	Mexico	Other
AROUT	Around the Horn	UK	English
BERGS	Berglunds snabbköp	Sweden	Other
BLAUS	Blauer See Delikatessen	Germany	German
BLONP	Blondesddsl père et fils	France	Other
BOLID	Bólido Comidas preparadas	Spain	Other
BONAP	Bon app'	France	Other
BOTTM	Bottom-Dollar Markets	Canada	English
BSBEV	B's Beverages	UK	English
CACTU	Cactus Comidas para llevar	Argentina	Other
CENTC	Centro comercial Moctezuma	Mexico	Other
CHOPS	Chop-suey Chinese	Switzerland	German
COMMI	Comércio Mineiro	Brazil	Other
CONSH	Consolidated Holdings	UK	English
DRACD	Drachenblut Delikatessen	Germany	German
DUMON	Du monde entier	France	Other
EASTC	Eastern Connection	UK	English
ERNSH	Ernst Handel	Austria	German
FAMIA	Familia Arquibaldo	Brazil	Other
FISSA	FISSA Fabrica Inter. Salchichas S.A.	Spain	Other
FOLIG	Folies gourmandes	France	Other

customer_id	company_name	country	language
FOLKO	Folk och fä HB	Sweden	Other
FRANK	Frankenversand	Germany	German
FRANR	France restauration	France	Other
FRANS	Franchi S.p.A.	Italy	Other
FURIB	Furia Bacalhau e Frutos do Mar	Portugal	Other
GALED	Galería del gastrónomo	Spain	Other
GODOS	Godos Cocina Típica	Spain	Other
GOURL	Gourmet Lanchonetes	Brazil	Other
GREAL	Great Lakes Food Market	USA	English
GROSR	GROSELLA-Restaurante	Venezuela	Other
HANAR	Hanari Carnes	Brazil	Other
HILAA	HILARION-Abastos	Venezuela	Other
HUNGC	Hungry Coyote Import Store	USA	English
HUNGO	Hungry Owl All-Night Grocers	Ireland	English
ISLAT	Island Trading	UK	English
KOENE	Königlich Essen	Germany	German
LACOR	La corne d'abondance	France	Other
LAMAI	La maison d'Asie	France	Other
LAUGB	Laughing Bacchus Wine Cellars	Canada	English
LAZYK	Lazy K Kountry Store	USA	English
LEHMS	Lehmanns Marktstand	Germany	German
LETSS	Let's Stop N Shop	USA	English
LILAS	LILA-Supermercado	Venezuela	Other

customer_id	company_name	country	language
LINOD	LINO-Delicateses	Venezuela	Other
LONEP	Lonesome Pine Restaurant	USA	English
MAGAA	Magazzini Alimentari Riuniti	Italy	Other
MAISD	Maison Dewey	Belgium	Other
MEREP	Mère Paillarde	Canada	English
MORGK	Morgenstern Gesundkost	Germany	German
NORTS	North/South	UK	English
OCEAN	Océano Atlántico Ltda.	Argentina	Other
OLDWO	Old World Delicatessen	USA	English
OTTIK	Ottilies Käseladen	Germany	German
PARIS	Paris spécialités	France	Other
PERIC	Pericles Comidas clásicas	Mexico	Other
PICCO	Piccolo und mehr	Austria	German
PRINI	Princesa Isabel Vinhos	Portugal	Other
QUEDE	Que Delícia	Brazil	Other
QUEEN	Queen Cozinha	Brazil	Other
QUICK	QUICK-Stop	Germany	German
RANCH	Rancho grande	Argentina	Other
RATTC	Rattlesnake Canyon Grocery	USA	English
REGGC	Reggiani Caseifici	Italy	Other
RICAR	Ricardo Adocicados	Brazil	Other
RICSU	Richter Supermarkt	Switzerland	German
ROMEY	Romero y tomillo	Spain	Other
SANTG	Santé Gourmet	Norway	Other

customer_id	company_name	country	language
SAVEA	Save-a-lot Markets	USA	English
SEVES	Seven Seas Imports	UK	English
SIMOB	Simons bistro	Denmark	Other
SPECD	Spécialités du monde	France	Other
SPLIR	Split Rail Beer & Ale	USA	English
SUPRD	Suprêmes délices	Belgium	Other
THEBI	The Big Cheese	USA	English
THECR	The Cracker Box	USA	English
TOMSP	Toms Spezialitäten	Germany	German
TORTU	Tortuga Restaurante	Mexico	Other
TRADH	Tradição Hipermercados	Brazil	Other
TRAIH	Trail's Head Gourmet Provisioners	USA	English
VAFFE	Vaffeljernet	Denmark	Other
VICTE	Victuailles en stock	France	Other
VINET	Vins et alcools Chevalier	France	Other
WANDK	Die Wandernde Kuh	Germany	German
WARTH	Wartian Herkku	Finland	Other
WELLI	Wellington Importadora	Brazil	Other
WHITC	White Clover Markets	USA	English
WILMK	Wilman Kala	Finland	Other
WOLZA	Wolski Zajazd	Poland	Other

练习25

- 需求：创建报表将所有产品划分为素食和非素食两类
- 报表中包含如下字段：
  - 产品名字 `product_name`
  - 类别名称 `category_name`
  - 膳食类型 `diet_type`：
    - 非素食 `'Non-vegetarian'` 商品类别字段的值为 `'Meat/Poultry'` 和 `'Seafood'`.
    - 素食

```sql
SELECT
  product_name,
  category_name,
  CASE
    WHEN category_name IN ('Meat/Poultry', 'Seafood') THEN
'Non-vegetarian'
    ELSE 'Vegetarian'
  END AS diet_type
FROM categories c
JOIN products p
  ON c.category_id = p.category_id;
```

查询结果

product_name	category_name	diet_type
Chai	Beverages	Vegetarian
Ikura	Seafood	Non-vegetarian
Queso Cabrales	Dairy Products	Vegetarian
Queso Manchego La Pastora	Dairy Products	Vegetarian
Konbu	Seafood	Non-vegetarian
Tofu	Produce	Vegetarian
Genen Shouyu	Condiments	Vegetarian
Pavlova	Confections	Vegetarian
Alice Mutton	Meat/Poultry	Non-vegetarian
Carnarvon Tigers	Seafood	Non-vegetarian
Teatime Chocolate Biscuits	Confections	Vegetarian
Chang	Beverages	Vegetarian
Sir Rodney's Marmalade	Confections	Vegetarian
Sir Rodney's Scones	Confections	Vegetarian
Gustaf's Knäckebröd	Grains/Cereals	Vegetarian
Tunnbröd	Grains/Cereals	Vegetarian
Guaraná Fantástica	Beverages	Vegetarian
NuNuCa Nuß-Nougat-Creme	Confections	Vegetarian
Gumbär Gummibärchen	Confections	Vegetarian
Schoggi Schokolade	Confections	Vegetarian
Rössle Sauerkraut	Produce	Vegetarian
Thüringer Rostbratwurst	Meat/Poultry	Non-vegetarian
Aniseed Syrup	Condiments	Vegetarian
Nord-Ost Matjeshering	Seafood	Non-vegetarian

product_name	category_name	diet_type
Gorgonzola Telino	Dairy Products	Vegetarian
Mascarpone Fabioli	Dairy Products	Vegetarian
Geitost	Dairy Products	Vegetarian
Sasquatch Ale	Beverages	Vegetarian
Steeleye Stout	Beverages	Vegetarian
Inlagd Sill	Seafood	Non-vegetarian
Gravad lax	Seafood	Non-vegetarian
Côte de Blaye	Beverages	Vegetarian
Chartreuse verte	Beverages	Vegetarian
Chef Anton's Cajun Seasoning	Condiments	Vegetarian
Boston Crab Meat	Seafood	Non-vegetarian
Jack's New England Clam Chowder	Seafood	Non-vegetarian
Singaporean Hokkien Fried Mee	Grains/Cereals	Vegetarian
Ipoh Coffee	Beverages	Vegetarian
Gula Malacca	Condiments	Vegetarian
Rogede sild	Seafood	Non-vegetarian
Spegesild	Seafood	Non-vegetarian
Zaanse koeken	Confections	Vegetarian
Chocolade	Confections	Vegetarian
Maxilaku	Confections	Vegetarian
Chef Anton's Gumbo Mix	Condiments	Vegetarian
Valkoinen suklaa	Confections	Vegetarian
Manjimup Dried Apples	Produce	Vegetarian
Filo Mix	Grains/Cereals	Vegetarian

product_name	category_name	diet_type
Perth Pasties	Meat/Poultry	Non-vegetarian
Tourtière	Meat/Poultry	Non-vegetarian
Pâté chinois	Meat/Poultry	Non-vegetarian
Gnocchi di nonna Alice	Grains/Cereals	Vegetarian
Ravioli Angelo	Grains/Cereals	Vegetarian
Escargots de Bourgogne	Seafood	Non-vegetarian
Raclette Courdavault	Dairy Products	Vegetarian
Grandma's Boysenberry Spread	Condiments	Vegetarian
Camembert Pierrot	Dairy Products	Vegetarian
Sirop d'érable	Condiments	Vegetarian
Tarte au sucre	Confections	Vegetarian
Vegie-spread	Condiments	Vegetarian
Wimmers gute Semmelknödel	Grains/Cereals	Vegetarian
Louisiana Fiery Hot Pepper Sauce	Condiments	Vegetarian
Louisiana Hot Spiced Okra	Condiments	Vegetarian
Laughing Lumberjack Lager	Beverages	Vegetarian
Scottish Longbreads	Confections	Vegetarian
Gudbrandsdalsost	Dairy Products	Vegetarian
Uncle Bob's Organic Dried Pears	Produce	Vegetarian
Outback Lager	Beverages	Vegetarian
Flotemysost	Dairy Products	Vegetarian
Mozzarella di Giovanni	Dairy Products	Vegetarian
Röd Kaviar	Seafood	Non-vegetarian
Longlife Tofu	Produce	Vegetarian

product_name	category_name	diet_type
Rhönbräu Klosterbier	Beverages	Vegetarian
Lakkalikööri	Beverages	Vegetarian
Original Frankfurter grüne Soße	Condiments	Vegetarian
Northwoods Cranberry Sauce	Condiments	Vegetarian
Mishi Kobe Niku	Meat/Poultry	Non-vegetarian

3 在GROUP BY中使用CASE WHEN

- 在引入北美地区免运费的促销策略时，我们也想知道运送到北美地区和其它国家地区的订单数量

```
SELECT
  CASE
    WHEN ship_country = 'USA' OR ship_country = 'Canada' THEN
0.0
    ELSE 10.0
  END AS shipping_cost,
  COUNT(*) AS order_count
FROM orders
GROUP BY
  CASE
    WHEN ship_country = 'USA' OR ship_country = 'Canada' THEN
0.0
    ELSE 10.0
  END;
```

- 在 `SELECT` 子句和 `GROUP BY` 子句中，有相同的 `CASE WHEN` 出现在 `GROUP BY` 子句中
- 这里并没有使用别名 `shipping_cost` 。虽然在SELECT子句中指定了别名（shipping_cost），但标准SQL不允许在GROUP BY子句中引用别名，所以这里 `CASE WHEN` 写了两次
- MySQL允许 在 `GROUP BY` 中使用列别名，在本案例中两种写法都可以

shipping_cost	order_count
10.0	678
0.0	152

- **注意：** `CASE WHEN` 语句在 `GROUP BY` 和 `SELECT` 子句中，写法必须相同

练习26

- 需求：创建报表统计供应商来自那个大洲
- 报表中包含两个字段：供应商来自哪个大洲（`supplier_continent`）和 供应产品种类数量（`product_count`）
- 供应商来自哪个大洲（`supplier_continent`）包含如下取值：
    - `'North America'` (供应商来自 `'USA'` 和 `'Canada'`.)
    - `'Asia'` (供应商来自 `'Japan'` 和 `'Singapore'`)
    - `'Other'` (其它国家)

```
SELECT
  CASE
    WHEN country IN ('USA', 'Canada') THEN 'North America'
    WHEN country IN ('Japan', 'Singapore') THEN 'Asia'
    ELSE 'Other'
  END AS supplier_continent,
  COUNT(*) AS product_count
FROM products p
JOIN suppliers s
  ON p.supplier_id = s.supplier_id
GROUP BY
  CASE
    WHEN country IN ('USA', 'Canada') THEN 'North America'
    WHEN country IN ('Japan', 'Singapore') THEN 'Asia'
    ELSE 'Other'
  END;
```

查询结果

supplier_continent	product_count
Other	49
Asia	9
North America	19

练习27

- 需求：创建一个简单的报表来统计员工的年龄情况
- 报表中包含如下字段
  - 年龄（`age`）：生日大于1980年1月1日 `'young'`，其余`'old'`
  - 员工数量（`employee_count`）

```
SELECT
  CASE
    WHEN birth_date > '1980-01-01' THEN 'young'
    ELSE 'old'
  END AS age,
  COUNT(*) AS employee_count
FROM employees
GROUP BY
  CASE
    WHEN birth_date > '1980-01-01' THEN 'young'
    ELSE 'old'
  END;
```

查询结果

age	employee_count
young	5
old	5

4 CASE WHEN 和 COUNT

- 可以将 `CASE WHEN` 和 `COUNT` 结合使用，自定义分组并统计每组数据数量

```
SELECT
  COUNT(CASE
    WHEN ship_country = 'USA' OR ship_country = 'Canada' THEN
order_id
  END) AS free_shipping,
  COUNT(CASE
    WHEN ship_country != 'USA' AND ship_country != 'Canada'
THEN order_id
  END) AS paid_shipping
FROM orders;
```

- 查询结果显示如下：

free_shipping	paid_shipping
152	678

- 在上面的查询中，在 `COUNT（）` 函数中包含了一个 `CASE WHEN` 子句。
    - 对于每一行， `CASE WHEN` 子句会检查 `ship_country` 中的值。 如果是“ USA”或“ Canada”，则将order_id传递给 `COUNT（）` 并进行计数。
    - 如果 `ship_country` 中的值不同，则 `CASE WHEN` 将返回 `NULL`， `COUNT（）` 不会统计 `NULL` 值。 `free_shipping` 列将**仅计算运往美国或加拿大的订单**”
    - `paid_shipping` 列的构建方式与上述方式类似

练习28

- 需求：统计客户的contact_title 字段值为 ’Owner' 的客户数量
- 查询结果有两个字段： `represented_by_owner` 和 `not_represented_by_owner`

```
SELECT
  COUNT(CASE
    WHEN contact_title = 'Owner' THEN customer_id
  END) AS represented_by_owner,
  COUNT(CASE
    WHEN contact_title != 'Owner' THEN customer_id
  END) AS not_represented_by_owner
FROM customers;
```

查询结果

represented_by_owner	not_represented_by_owner
17	74

练习29

- 需求：Washington (WA) 是 Northwind的主要运营地区，统计有多少订单是由华盛顿地区的员工处理的，多少订单是有其它地区的员工处理的
- 结果字段： `orders_wa_employees` 和 `orders_not_wa_employees`

```
SELECT
  COUNT(CASE
    WHEN region = 'WA' THEN order_id
  END) AS orders_wa_employees,
  COUNT(CASE
    WHEN region != 'WA' THEN order_id
  END) AS orders_not_wa_employees
FROM employees e
JOIN orders o
  ON e.employee_id = o.employee_id;
```

查询结果

orders_wa_employees	orders_not_wa_employees
605	225

5 GROUP BY 和 CASE WHEN组合使用

- 先看下面的SQL

```sql
SELECT
  ship_country,
  COUNT(CASE
    WHEN freight < 40.0 THEN order_id
  END) AS low_freight,
  COUNT(CASE
    WHEN freight >= 40.0 AND freight < 80.0 THEN order_id
  END) AS avg_freight,
  COUNT(CASE
    WHEN freight >= 80.0 THEN order_id
  END) AS high_freight
FROM orders
GROUP BY ship_country;
```

- 查询结果:

ship_country	low_freight	avg_freight	high_freight
France	46	14	17
Germany	50	34	38
Brazil	45	22	16
Belgium	10	5	4
Switzerland	7	3	8
Venezuela	23	10	13
Austria	6	7	27
Mexico	16	9	3
USA	53	22	47
Swede	15	5	17
Finland	16	3	3
Italy	20	6	2
Spain	16	3	4
UK	35	11	10
Ireland	4	5	10
Portugal	7	3	3
Canada	13	11	6
Denmark	8	6	4
Poland	6	0	1
Norway	3	2	1
Argentina	12	2	2

- 将 `COUNT(CASE WHEN...)` 和 `GROUP BY` 组合使用，可以创建更复杂的报表，在报表中，我们将运输到不同国家的订单根据运费高低进一步分成三组，并统计每组数量

练习30

- 需求：创建报表，统计不同类别产品的库存量，将库存量分成两类 >30 和 <=30 两档分别统计数量
- 报表包含三个字段
  - 类别名称 `category_name`，
  - 库存充足 `high_availability`
  - 库存紧张 `low_availability`

```
SELECT
  c.category_name,
  COUNT(CASE
    WHEN units_in_stock > 30 THEN product_id
  END) AS high_availability,
  COUNT(CASE
    WHEN units_in_stock <= 30 THEN product_id
  END) AS low_availability
FROM products p
JOIN categories c
  ON p.category_id = c.category_id
GROUP BY c.category_id,
  c.category_name;
```

结果

category_name	high_availability	low_availability
Grains/Cereals	4	3
Dairy Products	3	7
Meat/Poultry	1	5
Condiments	6	6
Produce	1	4
Beverages	6	6
Seafood	8	4
Confections	5	8

6 SUM中使用CASE WHEN

- 上面通过我们通过 `COUNT()` 函数和 `CASE WHEN` 子句联合使用来创建的报表，也可以通过 `SUM()` 来替代 `COUNT()`

```
SELECT
  SUM(CASE
    WHEN ship_country = 'USA' OR ship_country = 'Canada' THEN
1
  END) AS free_shipping,
  SUM(CASE
    WHEN ship_country != 'USA' AND ship_country != 'Canada'
THEN 1
  END) AS paid_shipping
FROM orders;
```

- 在上面的查询中，我们将 `SUM（）` 与 `CASE WHEN` 一起使用，结果与使用 `COUNT()` 相同

练习31

```
SELECT
  COUNT(CASE
    WHEN region = 'WA' THEN order_id
  END) AS orders_wa_employees,
  COUNT(CASE
    WHEN region != 'WA' THEN order_id
  END) AS orders_not_wa_employees
FROM employees e
JOIN orders o
  ON e.employee_id = o.employee_id;
```

- 将上面的SQL修改成用 SUM（） 函数实现

```
SELECT
  SUM(CASE
    WHEN region = 'WA' THEN 1
  END) AS orders_wa_employees,
  SUM(CASE
    WHEN region != 'WA' THEN 1
  END) AS orders_not_wa_employees
FROM employees e
JOIN orders o
  ON e.employee_id = o.employee_id;
```

查询结果

orders_wa_employees	orders_not_wa_employees
605	225

练习32

- 需求：创建报表统计运输到法国的的订单中，打折和未打折订单的总数量
- 结果包含两个字段：`full_price` （原价）和 `discounted_price` （打折）

```
SELECT
  SUM(CASE
    WHEN discount = 0 THEN 1
  END) AS full_price,
  SUM(CASE
    WHEN discount != 0 THEN 1
  END) AS discounted_price
FROM orders o
JOIN order_items oi
  ON o.order_id = oi.order_id
WHERE ship_country = 'France';
```

查询结果

full_price	discounted_price
107	77

7 SUM中使用CASE WHEN进行复杂计算

- 我们现在要统计每个订单的总付款额以及非素食产品的总付款额。

> 注: 非素食产品的产品ID （ `category_id` ） 是 6 和 8

```
SELECT
  o.order_id,
  SUM(oi.quantity * oi.unit_price * (1 - oi.discount)) AS total_price,
  SUM(CASE
    WHEN p.category_id in (6, 8) THEN oi.quantity * oi.unit_price * (1 - oi.discount)
    ELSE 0
  END) AS non_vegetarian_price
FROM orders o
JOIN order_items oi
  ON o.order_id = oi.order_id
JOIN products p
  ON p.product_id = oi.product_id
GROUP BY o.order_id;
```

- 之前的场景中，我们可以通过 `SUM(CASE WHEN...)` 来替换 `COUNT(CASE WHEN...)` ，但在上面的例子中，我们只能使用 `SUM(CASE WHEN...)` ，因为涉及到不同值的累加，不能通过COUNT计数替代

练习33

- 需求：输出报表，统计不同供应商供应商品的总库存量，以及高价值商品的库存量（单价超过40定义为高价值）
- 结果显示四列：
  - 供应商ID `supplier_id`
  - 供应商公司名 `company_name`
  - 由该供应商提供的总库存 `all_units`
  - 由该供应商提供的高价值商品库存 `expensive_units`

```sql
SELECT
  s.supplier_id,
  s.company_name,
  SUM(units_in_stock) AS all_units,
  SUM(CASE
    WHEN unit_price > 40.0 THEN units_in_stock
    ELSE 0
  END) AS expensive_units
FROM products p
JOIN suppliers s
  ON p.supplier_id = s.supplier_id
GROUP BY s.supplier_id,
  s.company_name;
```

显示结果

supplier_id	company_name	all_units	expensive_units
1	Exotic Liquids	69	0
4	Tokyo Traders	64	29
5	Cooperativa de Quesos 'Las Cabras'	108	0
6	Mayumi's	98	0
7	Pavlova, Ltd.	110	66
8	Specialty Biscuits, Ltd.	74	40
9	PB Knäckebröd AB	165	0
10	Refrescos Americanas LTDA	20	0
11	Heli Süßwaren GmbH & Co. KG	140	49
12	Plutzer Lebensmittelgroßmärkte AG	205	26
13	Nord-Ost-Fisch Handelsgesellschaft mbH	10	0
14	Formaggi Fortini s.r.l.	23	0
16	Bigfoot Breweries	347	0
17	Svensk Sjöföda AB	224	0
18	Aux joyeux ecclésiastiques	86	17
2	New Orleans Cajun Delights	133	0
19	New England Seafood Cannery	208	0
20	Leka Trading	70	17
21	Lyngbysild	100	0

supplier_id	company_name	all_units	expensive_units
22	Zaanse Snoepfabriek	51	0
23	Karkki Oy	132	0
24	G'day, Mate	58	20
25	Ma Maison	136	0
26	Pasta Buttini s.r.l.	57	0
27	Escargots Nouveaux	62	0
28	Gai pâturage	98	79
3	Grandma Kelly's Homestead	141	0
29	Forêts d'érables	130	17

小结

1. CASE WHEN语句检查一个或多个条件，并在找到第一个匹配条件时返回一个值。 如果没有 `ELSE` 子句并且没有匹配条件，则 `CASE WHEN` 返回 `NULL` 。

```
CASE
  WHEN condition_1 THEN result_1
  WHEN condition_2 THEN result_2
  ...
  ELSE result
END
```

2. 要添加新列，从而对业务数据进行**自定义分组**，可以在 `SELECT` 子句中使用 `CASE WHEN` ：

```
SELECT
  CASE
    WHEN ... THEN ...
  END AS sample_column
FROM table;
```

3. 可以在“ GROUP BY”子句中使用“ CASE WHEN”来创建自己的分组。 同样的 `CASE WHEN` 子句也必须出现在 `SELECT` 子句中：

```
SELECT
  CASE
    WHEN ... THEN ...
  END AS sample_column,
  COUNT(*) AS sample_count
FROM table
  ...
GROUP BY
  CASE WHEN ... THEN ...
  END;
```

4. 可以在 `COUNT()` 或 `SUM()` 函数内使用 `CASE WHEN` 来创建业务对象的自定义计数：

```
SELECT
  COUNT(CASE
    WHEN ... THEN column_name
  END) AS count_column
FROM table;
```

```
SELECT
  SUM(CASE
    WHEN ... THEN 1
  END) AS count_column
FROM table;
```

练习34

- 需求：创建报表来为每种商品添加价格标签，贵、中等、便宜
- 结果包含如下字段： `product_id` , `product_name` , `unit_price` ,和 `price_level`
- 价格等级 `price_level` 的取值说明：
  - `'expensive'` 单价高于100的产品
  - `'average'` 单价高于40但不超过100的产品
  - `'cheap'` 其他产品

```
SELECT
  product_id,
  product_name,
  unit_price,
  CASE
    WHEN unit_price > 100 THEN 'expensive'
    WHEN unit_price > 40 THEN 'average'
    ELSE 'cheap'
  END AS price_level
FROM products;
```

查询结果

product_id	product_name	unit_price	price_level
1	Chai	18	cheap
10	Ikura	31	cheap
11	Queso Cabrales	21	cheap
12	Queso Manchego La Pastora	38	cheap
13	Konbu	6	cheap
14	Tofu	23.25	cheap
15	Genen Shouyu	15.5	cheap
16	Pavlova	17.45	cheap
17	Alice Mutton	39	cheap
18	Carnarvon Tigers	62.5	average
19	Teatime Chocolate Biscuits	9.2	cheap
2	Chang	19	cheap
20	Sir Rodney's Marmalade	81	average
21	Sir Rodney's Scones	10	cheap
22	Gustaf's Knäckebröd	21	cheap
23	Tunnbröd	9	cheap
24	Guaraná Fantástica	4.5	cheap
25	NuNuCa Nuß-Nougat-Creme	14	cheap
26	Gumbär Gummibärchen	31.23	cheap
27	Schoggi Schokolade	43.9	average
28	Rössle Sauerkraut	45.6	average
29	Thüringer Rostbratwurst	123.79	expensive
3	Aniseed Syrup	10	cheap
30	Nord-Ost Matjeshering	25.89	cheap

product_id	product_name	unit_price	price_level
31	Gorgonzola Telino	12.5	cheap
32	Mascarpone Fabioli	32	cheap
33	Geitost	2.5	cheap
34	Sasquatch Ale	14	cheap
35	Steeleye Stout	18	cheap
36	Inlagd Sill	19	cheap
37	Gravad lax	26	cheap
38	Côte de Blaye	263.5	expensive
39	Chartreuse verte	18	cheap
4	Chef Anton's Cajun Seasoning	22	cheap
40	Boston Crab Meat	18.4	cheap
41	Jack's New England Clam Chowder	9.65	cheap
42	Singaporean Hokkien Fried Mee	14	cheap
43	Ipoh Coffee	46	average
44	Gula Malacca	19.45	cheap
45	Rogede sild	9.5	cheap
46	Spegesild	12	cheap
47	Zaanse koeken	9.5	cheap
48	Chocolade	12.75	cheap
49	Maxilaku	20	cheap
5	Chef Anton's Gumbo Mix	21.35	cheap
50	Valkoinen suklaa	16.25	cheap
51	Manjimup Dried Apples	53	average

product_id	product_name	unit_price	price_level
52	Filo Mix	7	cheap
53	Perth Pasties	32.8	cheap
54	Tourtière	7.45	cheap
55	Pâté chinois	24	cheap
56	Gnocchi di nonna Alice	38	cheap
57	Ravioli Angelo	19.5	cheap
58	Escargots de Bourgogne	13.25	cheap
59	Raclette Courdavault	55	average
6	Grandma's Boysenberry Spread	25	cheap
60	Camembert Pierrot	34	cheap
61	Sirop d'érable	28.5	cheap
62	Tarte au sucre	49.3	average
63	Vegie-spread	43.9	average
64	Wimmers gute Semmelknödel	33.25	cheap
65	Louisiana Fiery Hot Pepper Sauce	21.05	cheap
66	Louisiana Hot Spiced Okra	17	cheap
67	Laughing Lumberjack Lager	14	cheap
68	Scottish Longbreads	12.5	cheap
69	Gudbrandsdalsost	36	cheap
7	Uncle Bob's Organic Dried Pears	30	cheap
70	Outback Lager	15	cheap
71	Flotemysost	21.5	cheap

product_id	product_name	unit_price	price_level
72	Mozzarella di Giovanni	34.8	cheap
73	Röd Kaviar	15	cheap
74	Longlife Tofu	10	cheap
75	Rhönbräu Klosterbier	7.75	cheap
76	Lakkalikööri	18	cheap
77	Original Frankfurter grüne Soße	13	cheap
8	Northwoods Cranberry Sauce	40	cheap
9	Mishi Kobe Niku	97	average

练习35

- 需求：制作报表统计所有订单的总价（不计任何折扣）对它们进行分类。
- 包含一下字段：
  - `order_id`
  - `total_price`（折扣前）
  - `price_group`
- 字段 price_group 取值说明：
  - 总价超过2000美元
  - `'average'`，总价在$ 600到$ 2,000之间，包括两端
  - `'low'` 总价低于$ 600

```
SELECT
  order_id,
  SUM(unit_price * quantity) AS total_price,
  CASE
    WHEN SUM(unit_price * quantity) > 2000 THEN 'high'
    WHEN SUM(unit_price * quantity) > 600 THEN 'average'
    ELSE 'low'
  END AS price_group
FROM order_items
GROUP BY order_id;
```

查询结果(部分)

order_id	total_price	price_group
11038	751.00	average
10782	12.50	low
10725	287.80	low
10423	1020.00	average
10518	4150.05	high
10356	1106.40	average
10963	68.00	low
10596	1476.10	average
10282	155.40	low
10658	4668.00	high
10283	1414.80	average
10579	317.75	low
10693	2334.00	high
10896	750.50	average
10660	1701.00	average
10253	1444.80	average
10425	480.00	low
10774	875.00	average
10615	120.00	low
10514	8623.45	high

练习36

- 需求：统计所有订单的运费，将运费高低分为三档
- 报表中包含三个字段
  - `low_freight` `freight` 值小于“ 40.0”的订单数

    - `avg_freight` `freight` 值大于或等于“ 40.0”但小于“ 80.0”的订单数
    - `high_freight` `freight` 值大于或等于“ 80.0”的订单数

```
SELECT
  COUNT(CASE
    WHEN freight >= 80.0 THEN order_id
  END) AS high_freight,
  COUNT(CASE
    WHEN freight < 40.0 THEN order_id
  END) AS low_freight,
  COUNT(CASE
    WHEN freight >= 40.0 AND freight < 80.0 THEN order_id
  END) AS avg_freight
FROM orders;
```

查询结果

high_freight	low_freight	avg_freight
236	411	183