

二 使用SQL进行数据汇总

上一小节中，我们介绍了数据库中的表关系，了解了基本数据情况，在这一小节中，我们将通过SQL来创建简单数据报表。

学习目标

- 掌握GROUP BY 分组和 COUNT SUM等聚合函数的使用

1 详细报告

- 将一个或者多个业务对象的详细信息汇总到一张表中是一种比较常见的报表形式
- 我们需要的信息可能分散在多张表中，在写SQL时可以通过一个或者多个 JOIN 子句将信息进行汇总

```
SELECT
  c.company_name AS customer_company_name,
  e.first_name AS employee_first_name,
  e.last_name AS employee_last_name,
  o.order_date,
  o.shipped_date,
  o.ship_country
FROM orders o
JOIN employees e
  ON o.employee_id = e.employee_id
JOIN customers c
  ON o.customer_id = c.customer_id
WHERE o.ship_country = 'France';
```

- 在上面的SQL查询中，我们想收集运输到法国的订单的相关信息，包括订单涉及的顾客和员工信息，下单和发货日期等
- 由于相关数据保存在不同的表中，所以需要将 `orders` 表，`employees` 表和 `customers` 表连接在一起

- 注意在写SQL时，我们可以为每一张表都起了一个别名，可以减少输入的字符数

练习6

- 需求：提供订单编号为10248的相关信息，包括product name, unit price (在 `order_items` 表中), quantity (数量), company_name (供应商公司名字，起别名 `supplier_name`).

```
SELECT
    product_name,
    oi.unit_price,
    oi.quantity,
    company_name AS supplier_name
FROM order_items oi
JOIN products p
    ON oi.product_id = p.product_id
JOIN suppliers s
    ON s.supplier_id = p.supplier_id
WHERE oi.order_id = 10248;
```

- 查询结果

product_name	unit_price	quantity	supplier_name
Queso Cabrales	14.00	12	Cooperativa de Quesos 'Las Cabras'
Singaporean Hokkien Fried Mee	9.80	10	Leka Trading
Mozzarella di Giovanni	34.80	5	Formaggi Fortini s.r.l.

练习7

- 需求：提取每件商品的详细信息，包括 商品名称 (`product_name`) , 供应商的公司名称 (`company_name` , 在 `suppliers` 表中), 类别名称 `category_name` , 商品单价 `unit_price` , 和每单位商品数量 `quantity per unit`

```

SELECT
    p.product_name,
    s.company_name,
    c.category_name,
    p.unit_price,
    p.quantity_per_unit
FROM products p
JOIN suppliers s
    ON p.supplier_id = s.supplier_id
JOIN categories c
    ON c.category_id = p.category_id;

```

2 带时间限制的报表

- 另一种常见的报表需求是查询某段时间内的业务指标

```

SELECT
    COUNT(*)
FROM orders
WHERE order_date >= '2016-07-01' AND order_date < '2016-08-01';

```

- 在上面的查询中，我们统计了2016年7月的订单数量，
- 需要注意SQL中的日期总是放在单引号内，格式通常为“YYYY-MM-DD”（年-月-日）

练习8

- 统计2013年入职的员工，统计字段起别名 `number_of_employees`

```

SELECT
    COUNT(*) AS number_of_employees
FROM employees
WHERE hire_date >= '2013-01-01' AND hire_date < '2014-01-01';

```

- 查询结果

number_of_employees
3

3 计算多个对象

- 在业务报表中，我们通常希望同时计算多个业务对象的某些指标。

```
SELECT
    o.order_id,
    COUNT(*) AS order_items_count
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
WHERE o.order_id BETWEEN 10200 AND 10300
GROUP BY o.order_id;
```

- 在上面的查询中，我们统计了指定范围内的 `order_id`，计算每个订单中的商品数量
 - 通过连接 `orders` 和 `order_items` 表，在同一行显示单个订单商品及其父订单的信息
 - 按照父顺序对所有行进行分组
 - 使用 `COUNT (*)` 统计每个订单的商品数量

练习9

- 需求：统计每个供应商供应的商品种类数量
 - 结果返回供应商ID `supplier_id`，公司名字 `company_name`，商品种类数量（起别名 `products_count`）
 - 使用 `products` 和 `suppliers` 表。

```
SELECT
    s.supplier_id,
    company_name,
    COUNT(*) AS products_count
FROM products p
JOIN suppliers s
    ON p.supplier_id = s.supplier_id
GROUP BY s.supplier_id, company_name;
```

- 查询结果

supplier_id	company_name	products_count
1	Exotic Liquids	3
4	Tokyo Traders	3
5	Cooperativa de Quesos 'Las Cabras'	2
6	Mayumi's	3
7	Pavlova, Ltd.	5
8	Specialty Biscuits, Ltd.	4
9	PB Knäckebröd AB	2
10	Refrescos Americanas LTDA	1
11	Heli Süßwaren GmbH & Co. KG	3
12	Plutzer Lebensmittelgroßmärkte AG	5
13	Nord-Ost-Fisch Handelsgesellschaft mbH	1
14	Formaggi Fortini s.r.l.	3
16	Bigfoot Breweries	6
17	Svensk Sjöföda AB	3
18	Aux joyeux ecclésiastiques	2
2	New Orleans Cajun Delights	4
19	New England Seafood Cannery	2
20	Leka Trading	3
21	Lyngbysild	2
22	Zaanse Snoepfabriek	2
23	Karkki Oy	3
24	G'day, Mate	3
25	Ma Maison	2

supplier_id	company_name	products_count
26	Pasta Buttini s.r.l.	2
27	Escargots Nouveaux	1
28	Gai pâturage	2
3	Grandma Kelly's Homestead	3
29	Forêts d'érables	2

4 总订单金额

- 在销售报表中,我们经常需要计算订单的总付款额。

```
SELECT
    SUM(unit_price * quantity) AS total_price
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
WHERE o.order_id = 10250;
```

- 我们要查找ID为10250的订单的总价（折扣前），`SUM(unit_price * quantity)`

练习10

- Northwind商店某些产品会不定期做打折促销
- 每个商品的折扣都存储在 `order_items` 表的 `discount` 列中
- 例如，“0.20”折扣意味着客户支付原始价格的“ $1-0.2=0.8$ ”
- 在下面的代码中添加第二个名为 `total_price_after_discount` 的列，计算打折后的商品价格

```
SELECT
    SUM(unit_price * quantity) AS total_price
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
WHERE o.order_id = 10250;
```

答案

```

SELECT
    SUM(unit_price * quantity) AS total_price,
    SUM(unit_price * quantity * (1 - discount)) AS
total_price_after_discount
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
WHERE o.order_id = 10250;

```

- 查询结果

total_price	total_price_after_discount
1813	1552.6

5 计算多个订单的订单金额

- 上面的案例中我们计算了单个订单的订单总金额，接下来我们统计多个订单的总金额

```

SELECT
    o.order_id,
    c.company_name AS customer_company_name,
    SUM(unit_price * quantity) AS total_price
FROM orders o
JOIN customers c
    ON o.customer_id = c.customer_id
JOIN order_items oi
    ON o.order_id = oi.order_id
WHERE o.ship_country = 'France'
GROUP BY o.order_id, c.company_name;

```

- 我们想计算运输到法国的所有订单的总订单金额
- 在结果中，我们想保留订单ID和订单公司名字，可以通过 **GROUP BY** 实现
- 注意：通过 **GROUP BY** 我们只需要对 **order_id** 进行分组就可以了，但MySQL 5.7之后要求，在使用 **GROUP BY** 分组时，SELECT 后的字段，如果没有在聚合函数中使用，就必须在GROUP BY 后出现

练习11

- 统计每个员工处理的订单总数
- 结果包含员工ID `employee_id` , 姓名 `first_name` 和 `last_name` , 处理的订单总数(别名 `orders_count`)

```
SELECT
    e.employee_id,
    e.first_name,
    e.last_name,
    COUNT(*) AS orders_count
FROM orders o
JOIN employees e
    ON e.employee_id = o.employee_id
GROUP BY e.employee_id,
    e.first_name,
    e.last_name;
```

- 查询结果

employee_id	first_name	last_name	orders_count
5	Steven	Buchanan	42
6	Michael	Suyama	67
4	Margaret	Peacock	156
3	John	Smith	127
9	Anne	Dodsworth	43
1	Nancy	Davolio	123
8	Laura	Callahan	104
2	Andrew	Fuller	96
7	Robert	King	72

6 不同类别商品的库存

- 统计每个类别中的库存产品值多少钱？
 - 显示三列： `category_id` , `category_name` , 和 `category_total_value`
 - 如何计算库存商品总价： `SUM(unit_price * units_in_stock)` 。

```
SELECT
  c.category_id,
  c.category_name,
  SUM(unit_price * units_in_stock) AS category_total_value
FROM products p
JOIN categories c
  ON p.category_id = c.category_id
GROUP BY c.category_id,
  c.category_name;
```

- 查询结果

category_id	category_name	category_total_value
5	Grains/Cereals	5594.50
4	Dairy Products	11271.20
6	Meat/Poultry	5729.45
2	Condiments	12023.55
7	Produce	3549.35
1	Beverages	12480.25
8	Seafood	13010.35
3	Confections	10392.20

7 Group by分组

- 接下来，我们来了解每个员工的业绩：计算每个员工的订单数量
- 看下面的SQL是否有问题

```

SELECT
    e.first_name,
    e.last_name,
    COUNT(*) AS orders_count
FROM orders o
JOIN employees e
    ON o.employee_id = e.employee_id
GROUP BY e.first_name,
    e.last_name;

```

- 上面的SQL貌似正确，但是没有考虑到员工重名的问题，所以需要做一个小调整：

```

SELECT
    e.employee_id,
    e.first_name,
    e.last_name,
    COUNT(*) AS orders_count
FROM orders o
JOIN employees e
    ON o.employee_id = e.employee_id
GROUP BY e.employee_id,
    e.first_name,
    e.last_name;

```

- 在 `SELECT` 和 `GROUP BY` 中添加了员工ID `employee_id` 字段后,重名的问题就可以解决了
- 注意，在使用GROUP BY进行分组聚合统计时，需要考虑分组字段中的相同值的业务含义是否相同

练习12

- 需求：计算每个客户的下订单数
- 结果包含：用户id、用户公司名称、订单数量 (`customer_id`, `company_name`, `orders_count`)

```
SELECT
  c.customer_id,
  c.company_name,
  COUNT(*) AS orders_count
FROM orders o
JOIN customers c
  ON o.customer_id = c.customer_id
GROUP BY c.customer_id,
  c.company_name;
```

- 查询结果

customer_id	company_name	orders_count
VINET	Vins et alcools Chevalier	5
TOMSP	Toms Spezialitäten	6
HANAR	Hanari Carnes	14
VICTE	Victuailles en stock	10
SUPRD	Suprêmes délices	12
CHOPS	Chop-suey Chinese	8
RICSU	Richter Supermarkt	10
WELLI	Wellington Importadora	9
HILAA	HILARION-Abastos	18
ERNSH	Ernst Handel	30
CENTC	Centro comercial Moctezuma	1
OTTIK	Ottilies Käseladen	10
QUEDE	Que Delícia	9
RATTC	Rattlesnake Canyon Grocery	18
FOLKO	Folk och fä HB	19
BLONP	Blondesddsl père et fils	11
WARTH	Wartian Herkku	15
FRANK	Frankenversand	15
GROSR	GROSELLA-Restaurante	2
WHITC	White Clover Markets	14
SPLIR	Split Rail Beer & Ale	9
QUICK	QUICK-Stop	28
MAGAA	Magazzini Alimentari Riuniti	10
TORTU	Tortuga Restaurante	10

customer_id	company_name	orders_count
MORGK	Morgenstern Gesundkost	5
BERGS	Berglunds snabbköp	18
LEHMS	Lehmanns Marktstand	15
ROMEY	Romero y tomilló	5
LILAS	LILA-Supermercado	14
RICAR	Ricardo Adocicados	11
REGGC	Reggiani Caseifici	12
BSBEV	B's Beverages	10
COMMI	Comércio Mineiro	5
TRADH	Tradição Hipermercados	6
HUNGO	Hungry Owl All-Night Grocers	19
WANDK	Die Wandernde Kuh	10
GODOS	Godos Cocina Típica	10
OLDWO	Old World Delicatessen	10
LONEP	Lonesome Pine Restaurant	8
ANATR	Ana Trujillo Emparedados y helados	4
THEBI	The Big Cheese	4
ISLAT	Island Trading	10
PERIC	Pericles Comidas clásicas	6
KOENE	Königlich Essen	14
SAVEA	Save-a-lot Markets	31
BOLID	Bólido Comidas preparadas	3
FURIB	Furia Bacalhau e Frutos do Mar	8
BONAP	Bon app'	17

customer_id	company_name	orders_count
MEREP	Mère Paillard	13
PRINI	Princesa Isabel Vinhos	5
SIMOB	Simons bistro	7
FAMIA	Familia Arquibaldo	7
LAMAI	La maison d'Asie	14
PICCO	Piccolo und mehr	10
AROUT	Around the Horn	13
SEVES	Seven Seas Imports	9
DRACD	Drachenblut Delikatessen	6
EASTC	Eastern Connection	8
GALED	Galería del gastrónomo	5
VAFFE	Vaffeljernet	11
WOLZA	Wolski Zajazd	7
HUNGC	Hungry Coyote Import Store	5
SANTG	Santé Gourmet	6
BOTTM	Bottom-Dollar Markets	14
LINOD	LINO-Delicateses	12
FOLIG	Folies gourmandes	5
FRANS	Franchi S.p.A.	6
GOURL	Gourmet Lanchonetes	9
CONSH	Consolidated Holdings	3
RANCH	Rancho grande	5
LAZYK	Lazy K Kountry Store	2
LAUGB	Laughing Bacchus Wine Cellars	3

customer_id	company_name	orders_count
BLAUS	Blauer See Delikatessen	7
NORTS	North/South	3
CACTU	Cactus Comidas para llevar	6
GREAL	Great Lakes Food Market	11
MAISD	Maison Dewey	7
TRAIH	Trail's Head Gourmet Provisioners	3
LETSS	Let's Stop N Shop	4
WILMK	Wilman Kala	7
THECR	The Cracker Box	3
ALFKI	Alfreds Futterkiste	6
FRANR	France restauration	3
SPECD	Spécialités du monde	4
LACOR	La corne d'abondance	4

8 选择显示部分信息

- 再看一下上面的例子

```

SELECT
  e.employee_id,
  e.first_name,
  e.last_name,
  COUNT(*) AS orders_count
FROM orders o
JOIN employees e
  ON o.employee_id = e.employee_id
GROUP BY e.employee_id,
  e.first_name,
  e.last_name;

```

- 我们通过 `employee_id` 进行分组, 但是 `GROUP BY` 中的字段, 不一定在 `SELECT` 中出现, 例如下面的SQL:

```
SELECT
  e.first_name,
  e.last_name,
  COUNT(*) AS orders_count
FROM orders o
JOIN employees e
  ON o.employee_id = e.employee_id
GROUP BY e.employee_id,
  e.first_name,
  e.last_name;
```

- 之前我们强调过, `SELECT` 中的字段, 如果没在聚合函数中使用, 就一定更要在 `GROUP BY` 子句中出现
- 但是, `GROUP BY` 子句中的字段, 可以不用都出现在 `SELECT` 中

练习13

- 需求: 统计2016年6月到2016年7月用户的总下单金额并按金额从高到低排序
- 结果包含: 顾客公司名称 `company_name` 和总下单金额 (折后实付金额) `total_paid`
- 提示:
 - 计算实际总付款金额: `SUM(unit_price * quantity * (1 - discount))`
 - 日期过滤 `WHERE order_date >= '2016-06-01' AND order_date < '2016-08-01'`


```
SELECT
    c.company_name,
    SUM(unit_price * quantity * (1 - discount)) AS total_paid
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
JOIN customers c
    ON o.customer_id = c.customer_id
WHERE order_date >= '2016-06-01' AND order_date < '2016-08-01'
GROUP BY c.customer_id,
    c.company_name
ORDER BY total_paid DESC;
```

- 查询结果

company_name	total_paid
Suprêmes délices	3597.9
Frankenversand	3536.6
Ernst Handel	3488.68
Hanari Carnes	2997.4
Richter Supermarkt	2490.5
Toms Spezialitäten	1863.4
Otilies Käseladen	1504.65
Blondesddsl père et fils	1176
HILARION-Abastos	1119.9
GROSELLA-Restaurante	1101.2
Folk och fä HB	695.62
Victuailles en stock	654.06
White Clover Markets	642.2
Rattlesnake Canyon Grocery	584
Chop-suey Chinese	556.62
Wellington Importadora	517.8
Que Delícia	448
Vins et alcools Chevalier	440
Wartian Herkku	346.56
Centro comercial Moctezuma	100.8

9 COUNT()函数回顾

- 当创建业务报表的时候，需要注意 `COUNT(*)` 和 `COUNT(列名)` 之间的区别
- 假设我们要统计发货到不同国家/地区的订单数量以及已经发货的订单数量

```
SELECT
    ship_country,
    COUNT(*) AS all_orders,
    COUNT(shipped_date) AS shipped_orders
FROM orders
GROUP BY ship_country;
```

- `COUNT (*)` 将计算ship_country中的所有订单
- `COUNT (shipped_date)` 将仅计算shipped_date列值不为NULL的行
- 在我们的数据库中, `shipped_date` 列中的 `NULL` 表示尚未发货, `COUNT (shipped_date)` 仅计算已经发货的订单。

练习14

- 需求: 统计客户总数和带有传真号码的客户数量
- 需要字段: `all_customers_count` 和 `customers_with_fax_count`

```
SELECT
    COUNT(*) AS all_customers_count,
    COUNT(fax) AS customers_with_fax_count
FROM customers;
```

- 显示结果

all_customers_count	customers_with_fax_count
91	69

10 计数统计

- 使用SQL出报表时, 必须记住某些对象可能不存在
- 下面的练习中, 我们讨论在将“`COUNT ()`”与“`LEFT JOIN`”一起使用时的一个非常常见的错误。
- 我们要创建报表, 计算三个不同客户ID的订单: “ALFKI”, “FISSA”和“PARIS”:

```

SELECT
    c.customer_id,
    COUNT(o.order_id) AS order_count
FROM customers c
LEFT JOIN orders o
    ON o.customer_id = c.customer_id
WHERE c.customer_id IN ('ALFKI', 'FISSA', 'PARIS')
GROUP BY c.customer_id;

```

需要注意如下几点：

1. 这里使用了 `LEFT JOIN` 来确保报表中看到**全部三个客户ID**。如果使用 `JOIN`，任何没有下订单的客户将不会显示在报表中
2. 这里使用了 `COUNT(o.order_id)` 而不是 `COUNT(*)`
 - 只有当 `order_id` 列的值不为 `NULL` 的时候才会被计数。
 - 如客户未订购任何商品，会有一行具有给定的 `customer_id` 和一个 `NULL` 值在 `order_id` 列中，此时 `COUNT(*)` 将返回 `1` 而不是 `0`

练习15

- 需求：查找每个供应商提供的产品总数
 - 显示 `company_name` 和 `products_count`（提供的产品数量）列
 - 包括没有提供任何产品的供应商。

```

SELECT
    s.company_name,
    COUNT(p.product_id) AS products_count
FROM suppliers s
LEFT JOIN products p
    ON p.supplier_id = s.supplier_id
GROUP BY
    s.supplier_id,
    s.company_name;

```

- 查询结果

company_name	products_count
Exotic Liquids	3
New Orleans Cajun Delights	4
Grandma Kelly's Homestead	3
Tokyo Traders	3
Cooperativa de Quesos 'Las Cabras'	2
Mayumi's	3
Pavlova, Ltd.	5
Specialty Biscuits, Ltd.	4
PB Knäckebröd AB	2
Refrescos Americanas LTDA	1
Heli Süßwaren GmbH & Co. KG	3
Plutzer Lebensmittelgroßmärkte AG	5
Nord-Ost-Fisch Handelsgesellschaft mbH	1
Formaggi Fortini s.r.l.	3
Norske Meierier	0
Bigfoot Breweries	6
Svensk Sjöföda AB	3
Aux joyeux ecclésiastiques	2
New England Seafood Cannery	2
Leka Trading	3
Lyngbysild	2
Zaanse Snoepfabriek	2
Karkki Oy	3
G'day, Mate	3

company_name	products_count
Ma Maison	2
Pasta Buttini s.r.l.	2
Escargots Nouveaux	1
Gai pâturage	2
Forêts d'érables	2

11 统计不同的对象数量

- 假设我们要查找每个订单中的商品数量，但是同一商品可能在一个订单中出现多次。在这种情况下，我们需要每个订单中唯一商品的数量。以下查询将为我们提供总计数和唯一项目数：

```
SELECT
  order_id,
  COUNT(product_id) AS products_count,
  COUNT(DISTINCT product_id) AS unique_products_count
FROM order_items
GROUP BY order_id;
```

- 这里需要区分 `COUNT(product_id)` 和 `COUNT(DISTINCT product_id)` 之间的区别：
 - `COUNT(DISTINCT product_id)` 回去重，重复的product_id只会统计一次

练习16

- 查询订单运送到西班牙的客户数量（去重）别名 `number_of_companies`

```
SELECT COUNT(DISTINCT customer_id) AS number_of_companies
FROM orders
WHERE ship_country = 'Spain';
```

- 查询结果

number_of_companies
4

小结

1. 分组聚合查询中，SELECT 选中的列，如果在聚合函数中没有使用，就一定要出现在GROUP BY子句中
2. GROUP BY 分组 中的clause must appear in the SELECT clause.
3. 使用 COUNT() 的时候需要注意

4.	COUNT()函数使用方式	统计内容
	COUNT(*)	所有行数量
	COUNT(column_name)	统计 column_name 这列
	COUNT(DISTINCT column_name)	只统计 column_name 列中非空唯一值的数量

5. 避免将 LEFT JOIN s 和 COUNT(*) 在一起使用，使用 COUNT(column_name)

练习17

- 需求：查找每个供应商提供的产品总数。
- 显示以下列：
 - “supplier_id”，“company_name”和“products_supplied_count”（该公司提供的产品数量）
 - 未提供产品的供应商也要查询出来

```
SELECT
  s.supplier_id,
  s.company_name,
  COUNT(product_id) AS products_supplied_count
FROM products p
RIGHT JOIN suppliers s
  ON p.supplier_id = s.supplier_id
GROUP BY s.supplier_id,
  s.company_name;
```

- 查询结果

supplier_id	company_name	products_supplied_count
1	Exotic Liquids	3
2	New Orleans Cajun Delights	4
3	Grandma Kelly's Homestead	3
4	Tokyo Traders	3
5	Cooperativa de Quesos 'Las Cabras'	2
6	Mayumi's	3
7	Pavlova, Ltd.	5
8	Specialty Biscuits, Ltd.	4
9	PB Knäckebröd AB	2
10	Refrescos Americanas LTDA	1
11	Heli Süßwaren GmbH & Co. KG	3
12	Plutzer Lebensmittelgroßmärkte AG	5
13	Nord-Ost-Fisch Handelsgesellschaft mbH	1
14	Formaggi Fortini s.r.l.	3
15	Norske Meierier	0
16	Bigfoot Breweries	6
17	Svensk Sjöföda AB	3
18	Aux joyeux ecclésiastiques	2
19	New England Seafood Cannery	2
20	Leka Trading	3
21	Lyngbysild	2

supplier_id	company_name	products_supplied_count
22	Zaanse Snoepfabriek	2
23	Karkki Oy	3
24	G'day, Mate	3
25	Ma Maison	2
26	Pasta Buttini s.r.l.	2
27	Escargots Nouveaux	1
28	Gai pâturage	2
29	Forêts d'érables	2

练习18

- 运往法国的所有订单中有多少种不同的产品？将结果命名为“distinct_products”。

```
SELECT
    COUNT(DISTINCT oi.product_id) AS distinct_products
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
WHERE ship_country = 'France';
```

- 显示结果

distinct_products
65

练习19

显示有关产品供应商的三种信息：

1. `all_suppliers` （供应商总数）
2. `suppliers_region_assigned` （分配了地区的供应商数量）
3. `unique_supplier_regions` （多少不同个区域）

```

SELECT
    COUNT(*) AS all_suppliers,
    COUNT(region) AS suppliers_region_assigned,
    COUNT(DISTINCT region) AS unique_supplier_regions
FROM suppliers;

```

- 查询结果

all_suppliers	suppliers_region_assigned	unique_supplier_regions
29	9	8

练习20

- 需求：统计2016年7月5日至2016年7月31日间，每位员工处理的所有订单，折扣前的总订单值
- 结果包含如下列：
 - first_name, last_name和sum_orders
 - 按“sum_orders”对结果进行降序排序。

```

SELECT
    e.first_name,
    e.last_name,
    SUM(unit_price * quantity) AS sum_orders
FROM orders o
JOIN order_items oi
    ON o.order_id = oi.order_id
JOIN employees e
    ON e.employee_id = o.employee_id
WHERE order_date >= '2016-07-05' AND order_date < '2016-08-01'
GROUP BY e.employee_id,
    e.first_name,
    e.last_name
ORDER BY sum_orders DESC;

```

- 查询结果

first_name	last_name	sum_orders
Margaret	Peacock	12540.9
Anne	Dodsworth	4955.3
John	Smith	2998.2
Michael	Suyama	2587.9
Nancy	Davolio	2018.6
Laura	Callahan	1726
Steven	Buchanan	1301.2
Andrew	Fuller	1176
John	Smith	448