## Tables

**customer**

| customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London | | 5005 |

**orders**

| ord_no | purch_amt | ord_date | customer_id | salesman_id |
|---|---|---|---|---|
| 70001 | 150.5 | 2012-10-05 | 3005 | 5002 |
| 70009 | 270.65 | 2012-09-10 | 3001 | 5005 |
| 70002 | 65.26 | 2012-10-05 | 3002 | 5001 |
| 70004 | 110.5 | 2012-08-17 | 3009 | 5003 |
| 70007 | 948.5 | 2012-09-10 | 3005 | 5002 |
| 70005 | 2400.6 | 2012-07-27 | 3007 | 5001 |
| 70008 | 5760 | 2012-09-10 | 3002 | 5001 |
| 70010 | 1983.43 | 2012-10-10 | 3004 | 5006 |
| 70003 | 2480.4 | 2012-10-10 | 3009 | 5003 |
| 70012 | 250.45 | 2012-06-27 | 3008 | 5002 |
| 70011 | 75.29 | 2012-08-17 | 3003 | 5007 |
| 70013 | 3045.6 | 2012-04-25 | 3002 | 5001 |

**salesman**

| salesman_id | name | city | commission |
|---|---|---|---|
| 5001 | James Hoog | New York | 0.15 |
| 5002 | Nail Knite | Paris | 0.13 |
| 5005 | Pit Alex | London | 0.11 |
| 5006 | Mc Lyon | Paris | 0.14 |
| 5003 | Lauson Hen | San Jose | 0.12 |
| 5007 | Paul Adam | Rome | 0.13 |

**company_mast**

| COM_ID | COM_NAME |
|---|---|
| 11 | Samsung |
| 12 | iBall |
| 13 | Epsion |

```
    14 Zebronics
    15 Asus
    16 Frontech
```

**item_mast**

| PRO_ID | PRO_NAME | PRO_PRICE | PRO_COM |
|---|---|---|---|
| 101 | Mother Board | 3200.00 | 15 |
| 102 | Key Board | 450.00 | 16 |
| 103 | ZIP drive | 250.00 | 14 |
| 104 | Speaker | 550.00 | 16 |
| 105 | Monitor | 5000.00 | 11 |
| 106 | DVD drive | 900.00 | 12 |
| 107 | CD drive | 800.00 | 12 |
| 108 | Printer | 2600.00 | 13 |
| 109 | Refill cartridge | 350.00 | 13 |
| 110 | Mouse | 250.00 | 12 |

**emp_department**

| DPT_CODE | DPT_NAME | DPT_ALLOTMENT |
|---|---|---|
| 57 | IT | 65000 |
| 63 | Finance | 15000 |
| 47 | HR | 240000 |
| 27 | RD | 55000 |
| 89 | QC | 75000 |

**emp_details**

| EMP_IDNO | EMP_FNAME | EMP_LNAME | EMP_DEPT |
|---|---|---|---|
| 127323 | Michale | Robbin | 57 |
| 526689 | Carlos | Snares | 63 |
| 843795 | Enric | Dosio | 57 |
| 328717 | Jhon | Snares | 63 |
| 444527 | Joseph | Dosni | 47 |
| 659831 | Zanifer | Emily | 47 |
| 847674 | Kuleswar | Sitaraman | 57 |
| 748681 | Henrey | Gabriel | 47 |
| 555935 | Alex | Manuel | 57 |
| 539569 | George | Mardy | 27 |
| 733843 | Mario | Saule | 63 |
| 631548 | Alan | Snappy | 27 |
| 839139 | Maria | Foster | 57 |

## Queries

1. From the salesman and orders tables, find all the orders issued by the salesman ‘Paul Adam’. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

```
SELECT *
FROM orders
WHERE salesman_id = (SELECT salesman_id FROM salesman WHERE name =
‘Paul Adam’)
```

2. From the salesman and orders tables, find all the orders, which are generated by those salespeople, who live in the city of London. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.

```
SELECT *
FROM orders
WHERE salesman_id IN (SELECT salesman_id FROM salesman WHERE city =
‘London’)
```

3. From the orders table, find the orders generated by the salespeople who works for customers whose id is 3007. Return ord_no, purch_amt, ord_date, customer_id, salesman_id. A customer can works only with a salespeople.

```
SELECT *
FROM orders
WHERE salesman_id = (SELECT salesman_id FROM orders WHERE
customer_id = 3007)
```

**4.** From the orders table, find the order values greater than the average order value of 10th October 2012. Return ord_no, purch_amt, ord_date, customer_id, salesman_id.

```
SELECT *
FROM orders
WHERE purch_amt > (SELECT AVG(purch_amt) FROM orders WHERE ord_date
= '2012-10-10')
```

**5.** From the salesman and orders tables, find all the orders generated in New York city. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

```
SELECT *
FROM orders
WHERE salesman_id IN (SELECT salesman_id FROM salesman WHERE city =
'New York')
```

-

**6.** From the customer and salesman tables, find the commission of the salespeople work in Paris City. Return commission.

```
SELECT commission
FROM salesman
WHERE salesman_id IN (SELECT salesman_id FROM customer WHERE city =
'Paris')
```

**7.** Write a query to display all the customers whose id is 2001 bellow the salesman ID of Mc Lyon.

```
SELECT *
FROM customer
WHERE customer_id = (SELECT salesman_id — 2001 FROM salesman WHERE
name = 'Mc Lyon')
```

**8.** From the customer table, count number of customers with grades above the average grades of New York City. Return grade and count.

```
SELECT grade, COUNT(grade)
FROM customer
WHERE grade > (SELECT AVG(grade) FROM customer WHERE city = 'New
York')
GROUP BY grade
```

**9.** From the salesman and orders tables, find those salespeople who earned the maximum commission. Return ord_no, purch_amt, ord_date, and salesman_id.

```
SELECT ord_no, purch_amt, ord_date, salesman_id
FROM orders
WHERE salesman_id IN (SELECT salesman_id FROM salesman WHERE
commission = (SELECT MAX(commission) FROM salesman))
```

**10.** From the customer and orders tables, find the customers whose orders issued on 17th August, 2012. Return ord_no, purch_amt, ord_date, customer_id, salesman_id and cust_name.

```
SELECT o.*, c.cust_name
FROM orders AS o
JOIN customer AS c
ON o.customer_id = c.customer_id
AND o.salesman_id = c.salesman_id
WHERE o.customer_id IN (SELECT customer_id FROM orders WHERE
ord_date = '2012-08-17')
AND o.ord_date = '2012-08-17'
```

**11.** From the customer and salesman tables, find the salespeople who had more than one customer. Return salesman_id and name.

```
SELECT salesman_id, name
FROM salesman
WHERE salesman_id IN (SELECT salesman_id FROM customer GROUP BY
salesman_id HAVING COUNT(salesman_id) > 1)
```

**12.** From the orders table, find those orders, which amount is higher than the average amount of the related customer. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

```
SELECT o.* FROM
orders AS o
JOIN
(
SELECT customer_id, AVG(purch_amt) AS avg
FROM orders
GROUP BY customer_id
) AS c
ON o.customer_id = c.customer_id
WHERE o.purch_amt > c.avg
```

**13.** From the orders table, find those orders, which are equal or higher than average amount of the orders. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

```
SELECT *
FROM orders
WHERE purch_amt >= (SELECT AVG(purch_amt) FROM orders)
```

**14.** Write a query to find the sums of the amounts from the orders table, grouped by date, eliminating all those dates where the sum was not at least 1000.00 above the maximum order amount for that date.

```
SELECT s.*
FROM
(
SELECT ord_date, SUM(purch_amt) AS sum
FROM orders
```

```
GROUP BY ord_date
) AS s
JOIN
(
SELECT ord_date, MAX(purch_amt) + 1000 AS pmax
FROM orders
GROUP BY ord_date
) AS m
ON s.ord_date = m.ord_date
WHERE s.sum >= m.pmax
```

**15.** Write a query to extract all data from the customer table if and only if one or more of the customers in the customer table are located in London.

```
SELECT *
FROM customer
WHERE EXISTS (SELECT city FROM customer WHERE city = 'London')
```

**16.** From the customer and salesman tables, find the salespeople who deal multiple customers. Return salesman_id, name, city and commission.

```
SELECT *
FROM salesman
WHERE salesman_id IN (SELECT salesman_id FROM customer GROUP BY
salesman_id HAVING COUNT(salesman_id) > 1)
```

**17.** From the customer and salesman tables, find the salespeople who deal a single customer. Return salesman_id, name, city and commission.

```
SELECT *
FROM salesman
WHERE salesman_id IN (SELECT salesman_id FROM customer GROUP BY
salesman_id HAVING COUNT(salesman_id) = 1)
```

**18.** From the salesman and orders tables, find the salespeople who deal the customers with more than one order. Return salesman_id, name, city and commission.

```
SELECT *
FROM salesman
WHERE salesman_id IN (SELECT salesman_id FROM orders GROUP BY
salesman_id HAVING COUNT(salesman_id) > 1)
```

**19.** From the customer and salesman tables, find the salespeople who deals those customers who live in the same city. Return salesman_id, name, city and commission.

```
SELECT *
FROM salesman
WHERE salesman_id IN (SELECT s.salesman_id FROM salesman AS s JOIN
customer AS c ON s.salesman_id = c.salesman_id WHERE s.city =
c.city)
```

**20.** From the customer and salesman tables, find the salespeople whose place of living (city) matches with any of the city where customers live. Return salesman_id, name, city and commission.

```
SELECT *
FROM salesman
WHERE city IN (SELECT city FROM customer)
```