Subqueries

October 2023

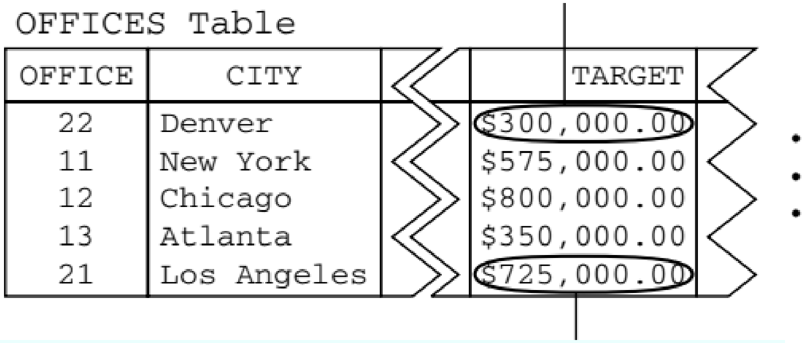
Subqueries

- A query within a query
- The results of the subquery are used by the DBMS to determine the results of the higher-level query that contains the subquery.
- Appears within the WHERE or HAVING clause of another SQL statement in the simplest forms .
- Provide an efficient and natural way to handle query requests that are themselves expressed in terms of the results of other queries.

Using Subqueries

- List the offices (city) where the sales target for the office exceeds the sum of the individual salespeople's quotas.

```
SELECT CITY   FROM OFFICES   WHERE TARGET > ???
```



OFFICES Table

OFFICE	CITY	TARGET
22	Denver	$300,000.00
11	New York	$575,000.00
12	Chicago	$800,000.00
13	Atlanta	$350,000.00
21	Los Angeles	$725,000.00

```
SELECT SUM(QUOTA)
  FROM SALESREPS
 WHERE REP_OFFICE = 22 ;
```

```
SELECT SUM(QUOTA)
  FROM SALESREPS
 WHERE REP_OFFICE = 11 ;
```

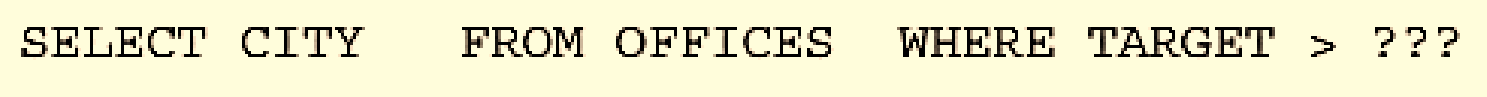
```
SELECT SUM(QUOTA)
  FROM SALESREPS
 WHERE REP_OFFICE = 12 ;
```

```
SELECT SUM(QUOTA)
  FROM SALESREPS
 WHERE REP_OFFICE = 13 ;
```

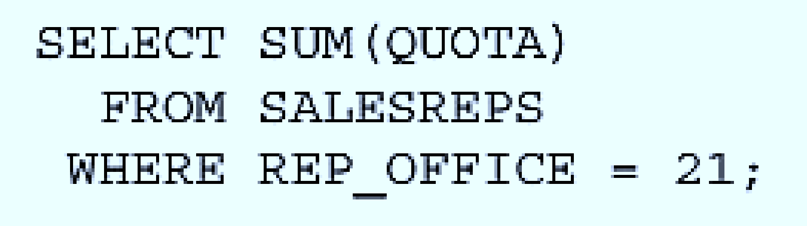
```
SELECT SUM(QUOTA)
  FROM SALESREPS
 WHERE REP_OFFICE = 21;
```

Using Subqueries

- List the offices (city) where the sales target for the office exceeds the sum of the individual salespeople’s quotas.

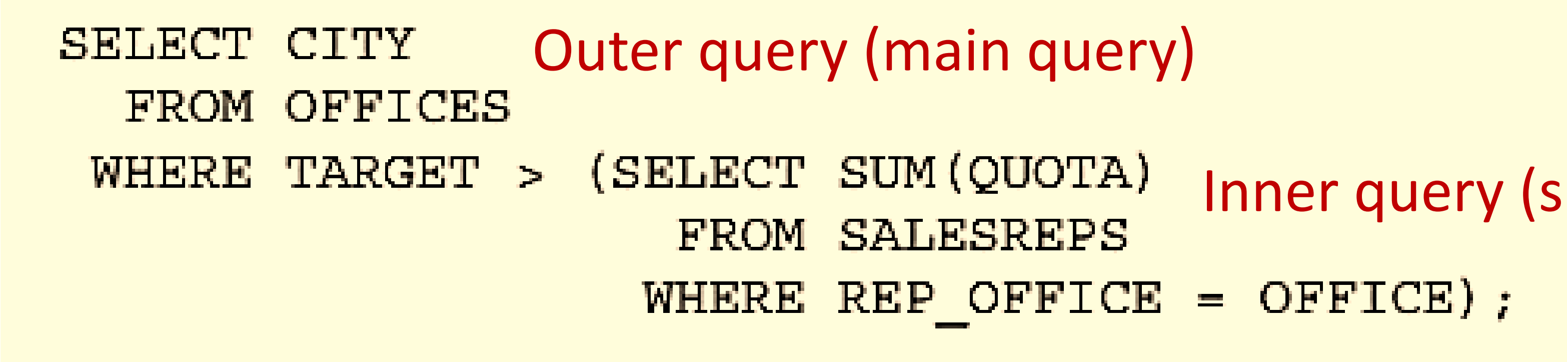


```
SELECT CITY   FROM OFFICES  WHERE TARGET > ???
```



```
SELECT SUM(QUOTA)
  FROM SALESREPS
 WHERE REP_OFFICE = 21;
```

No need to run this query for each of the offices to get the result



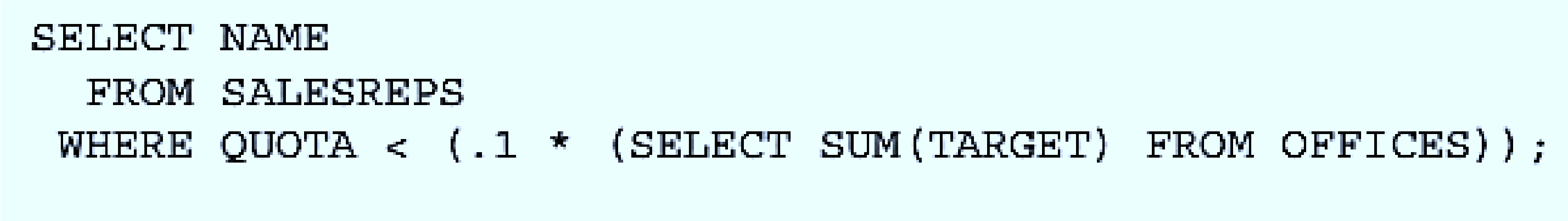
```
SELECT CITY
  FROM OFFICES
 WHERE TARGET > (SELECT SUM(QUOTA)
                   FROM SALESREPS
                  WHERE REP_OFFICE = OFFICE);
```

Outer query (main query)

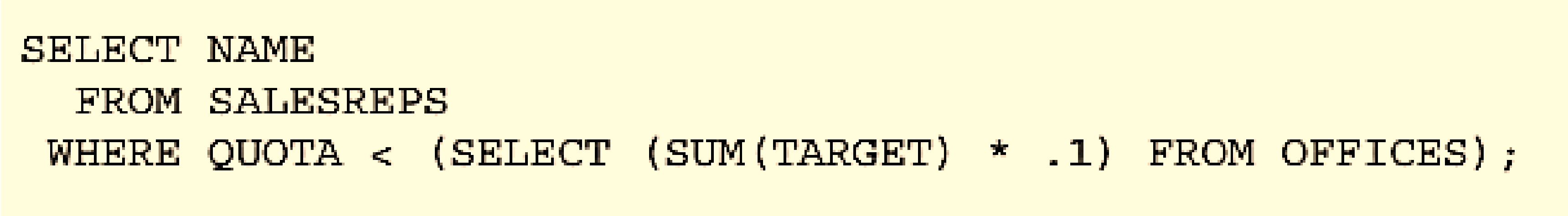
Inner query (subquery)

Subqueries in the WHERE Clause

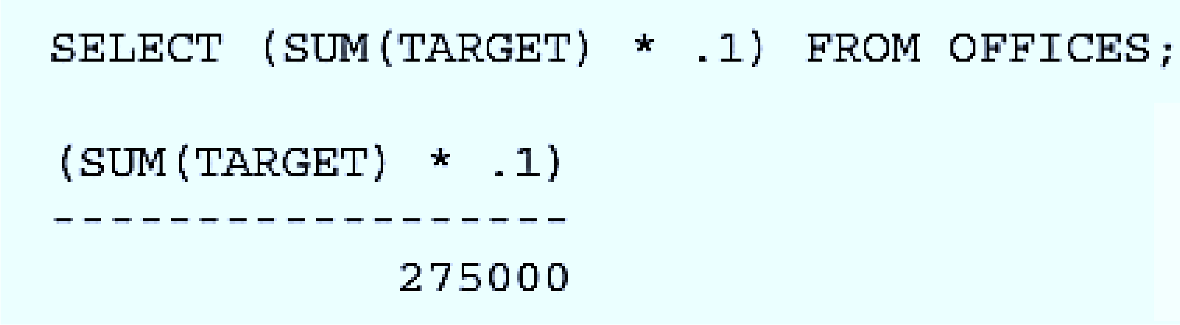
- List the salespeople (name) whose quota is less than 10% of the companywide sales target.



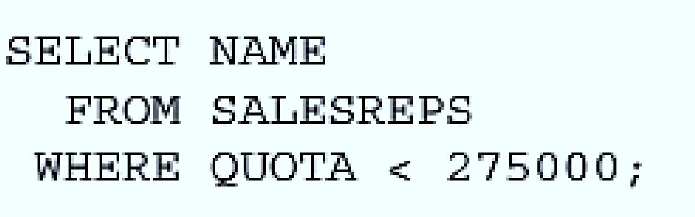
```
SELECT NAME
  FROM SALESREPS
 WHERE QUOTA < (.1 * (SELECT SUM(TARGET) FROM OFFICES));
```



```
SELECT NAME
  FROM SALESREPS
 WHERE QUOTA < (SELECT (SUM(TARGET) * .1) FROM OFFICES);
```



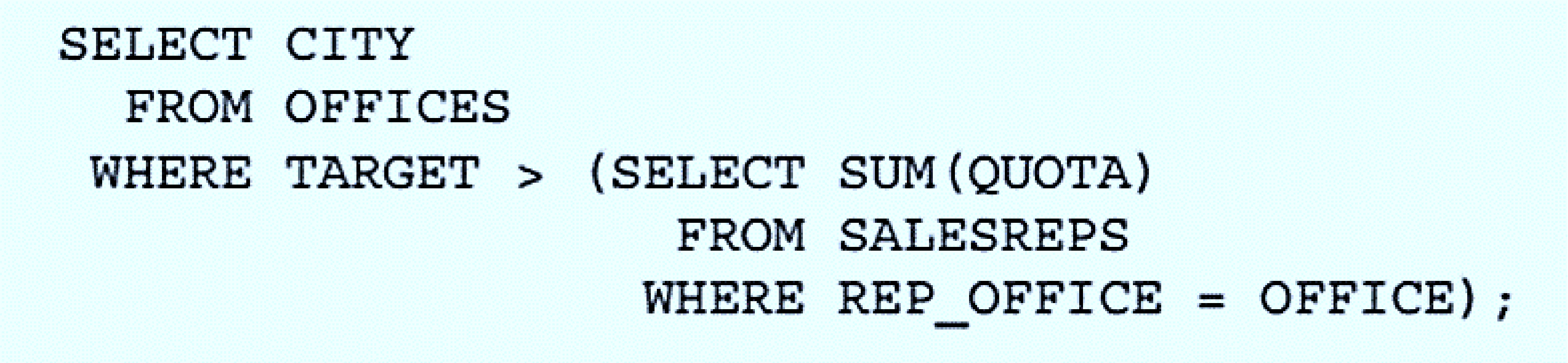
```
SELECT (SUM(TARGET) * .1) FROM OFFICES;

(SUM(TARGET) * .1)
------------------
            275000
```



```
SELECT NAME
  FROM SALESREPS
 WHERE QUOTA < 275000;
```

Subqueries in the WHERE Clause

- List the offices (city) where the sales target for the office exceeds the sum of the individual salespeople’s quotas.



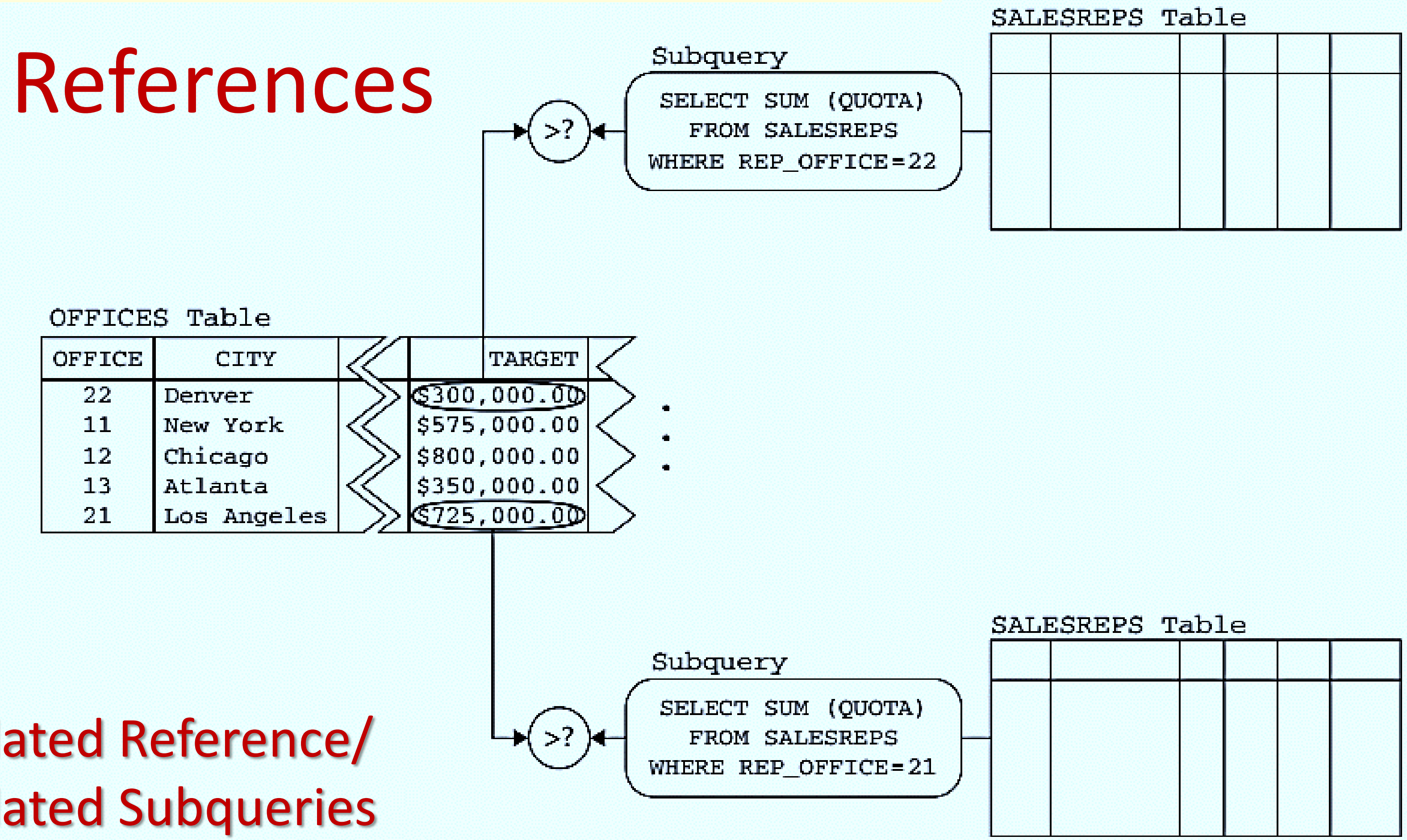
```
SELECT CITY
  FROM OFFICES
 WHERE TARGET > (SELECT SUM(QUOTA)
                   FROM SALESREPS
                  WHERE REP_OFFICE = OFFICE);
```

In this case, the subquery cannot be calculated once for the entire query.

Subqueries

```
SELECT CITY
  FROM OFFICES
 WHERE TARGET > (SELECT SUM(QUOTA)
                   FROM SALESREPS
                  WHERE REP_OFFICE = OFFICE);
```

Outer References



OFFICES Table

OFFICE	CITY	TARGET
22	Denver	$300,000.00
11	New York	$575,000.00
12	Chicago	$800,000.00
13	Atlanta	$350,000.00
21	Los Angeles	$725,000.00

Subquery

```
SELECT SUM (QUOTA)
FROM SALESREPS
WHERE REP_OFFICE=22
```

Subquery

```
SELECT SUM (QUOTA)
FROM SALESREPS
WHERE REP_OFFICE=21
```

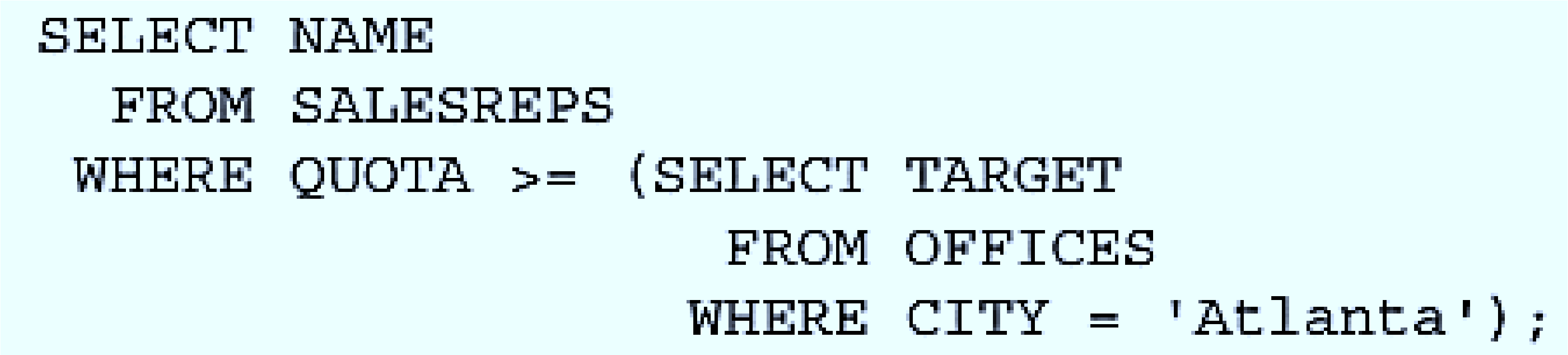
- Correlated Reference/ Correlated Subqueries

Subquery Search Conditions

- Subquery comparison test
  - Compares the value of an expression with a single value produced by a subquery (resembles the simple comparison test)
- Subquery set membership test
  - Checks whether the value of an expression matches one of the set of values produced by a subquery (resembles the simple set membership test)
- Existence test
  - Tests whether a subquery produces any rows of query results.
- Quantified comparison test
  - Compares the value of an expression with each of the sets of values produced by a subquery.

Subquery Comparison Test

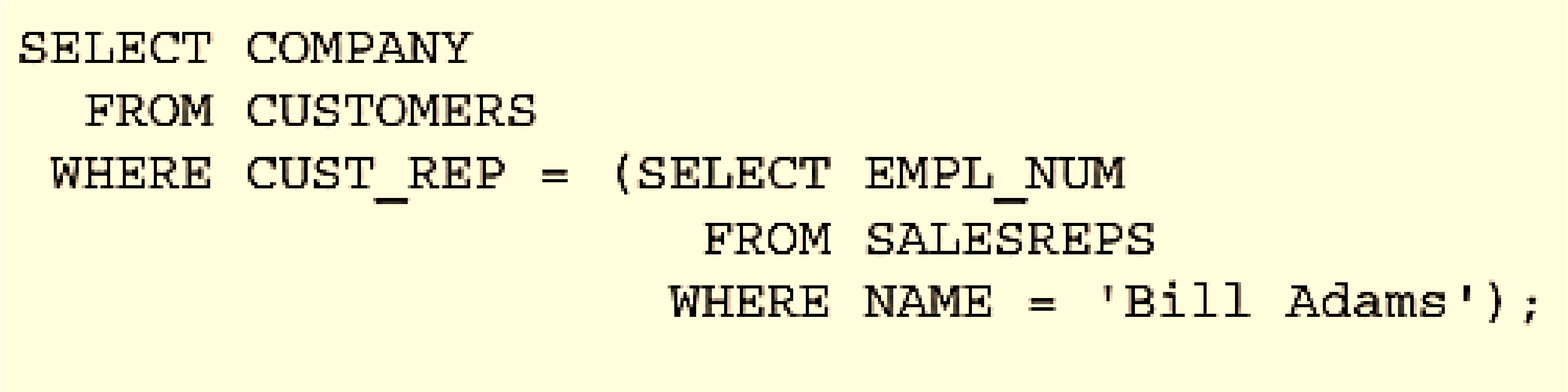
- List the salespeople (name) whose quotas are equal to or higher than the target of the Atlanta sales office.



```
SELECT NAME
  FROM SALESREPS
 WHERE QUOTA >= (SELECT TARGET
                   FROM OFFICES
                  WHERE CITY = 'Atlanta');
```

Subquery Comparison Test

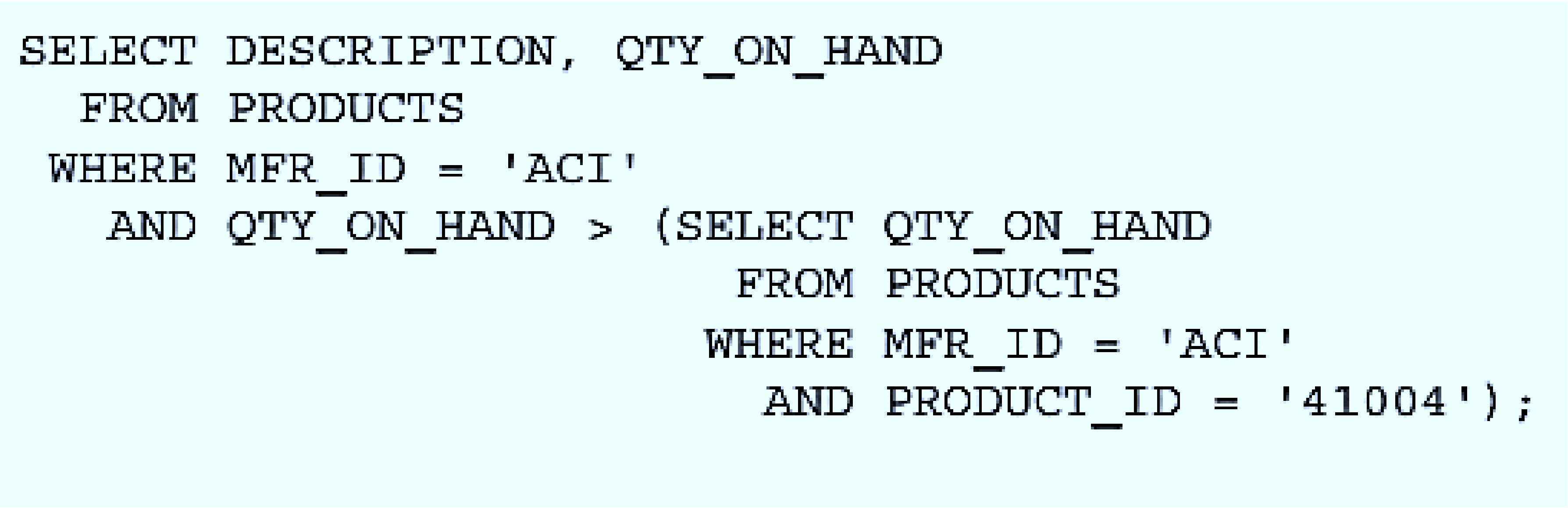
- List all customers served by Bill Adams.



```
SELECT COMPANY
  FROM CUSTOMERS
 WHERE CUST_REP = (SELECT EMPL_NUM
                     FROM SALESREPS
                    WHERE NAME = 'Bill Adams');
```

Subquery Comparison Test

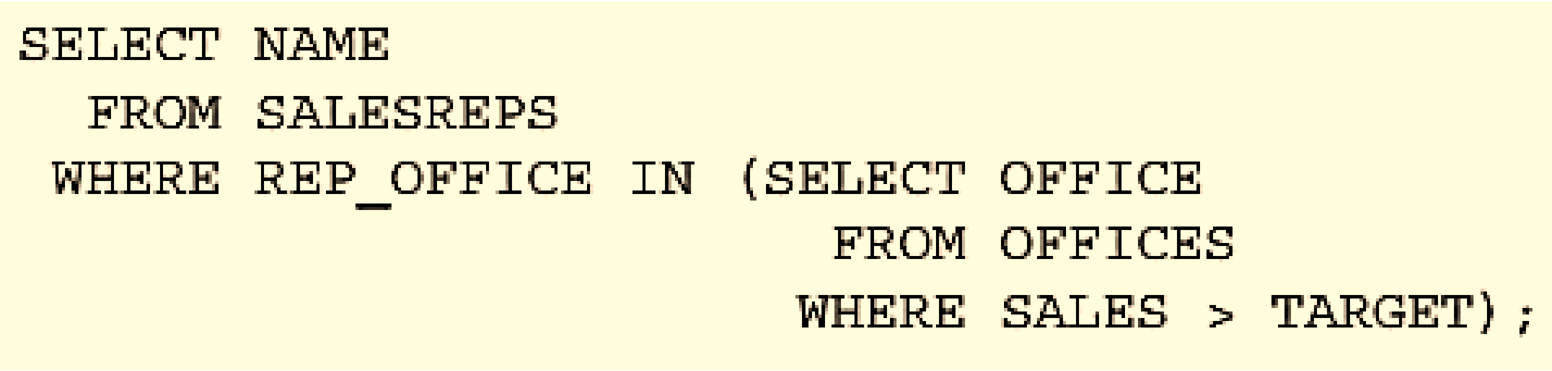
- List all products from manufacturer ACI where the quantity on hand is above the quantity on hand of product ACI-41004.



```
SELECT DESCRIPTION, QTY_ON_HAND
  FROM PRODUCTS
 WHERE MFR_ID = 'ACI'
   AND QTY_ON_HAND > (SELECT QTY_ON_HAND
                        FROM PRODUCTS
                       WHERE MFR_ID = 'ACI'
                         AND PRODUCT_ID = '41004');
```

The Set Membership Test

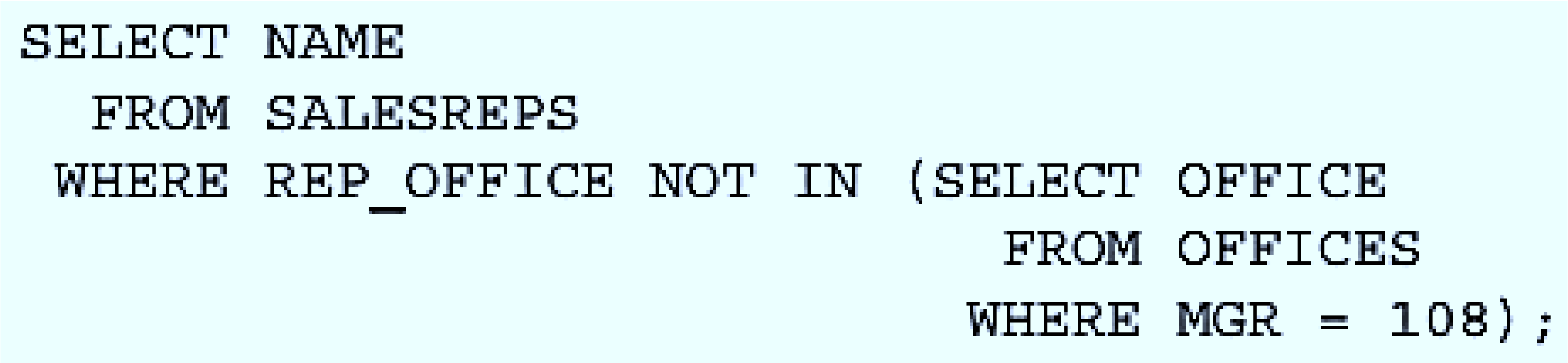
- List the salespeople who work in offices that are overtarget.



```
SELECT NAME
  FROM SALESREPS
 WHERE REP_OFFICE IN (SELECT OFFICE
                        FROM OFFICES
                       WHERE SALES > TARGET);
```

The Set Membership Test

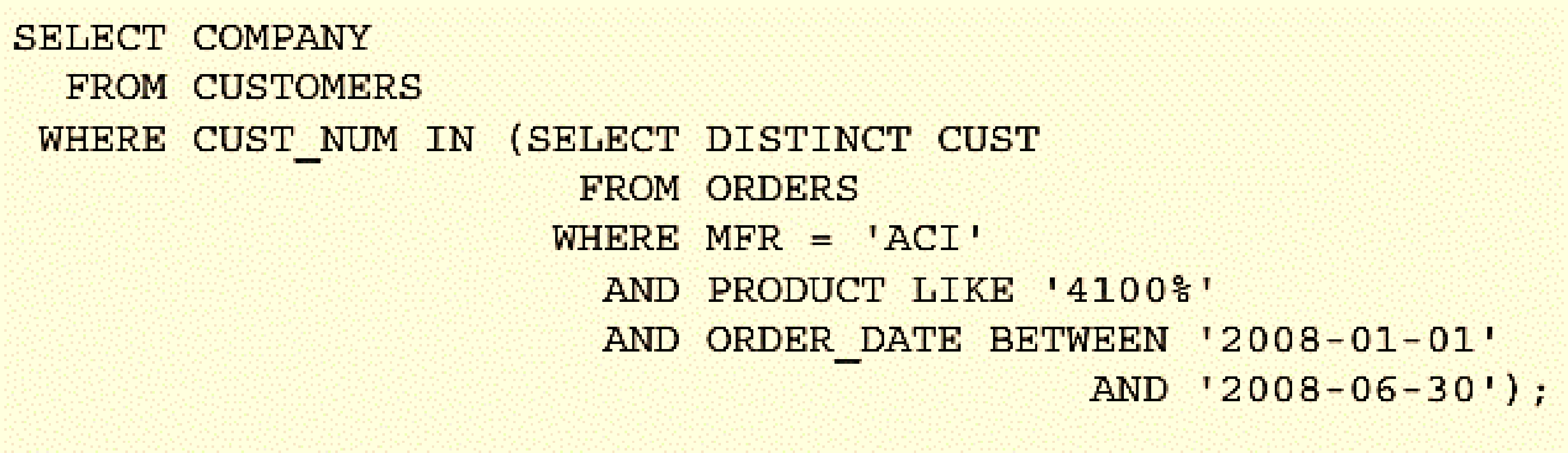
- List the salespeople who do not work in offices managed by Larry Fitch (emp. No. 108).



```
SELECT NAME
  FROM SALESREPS
 WHERE REP_OFFICE NOT IN (SELECT OFFICE
                            FROM OFFICES
                           WHERE MGR = 108);
```

The Set Membership Test

- List all of the customers who have placed orders for ACI Widgets (manufacturer ACI, product numbers starting with 4100) between January and June 2008.

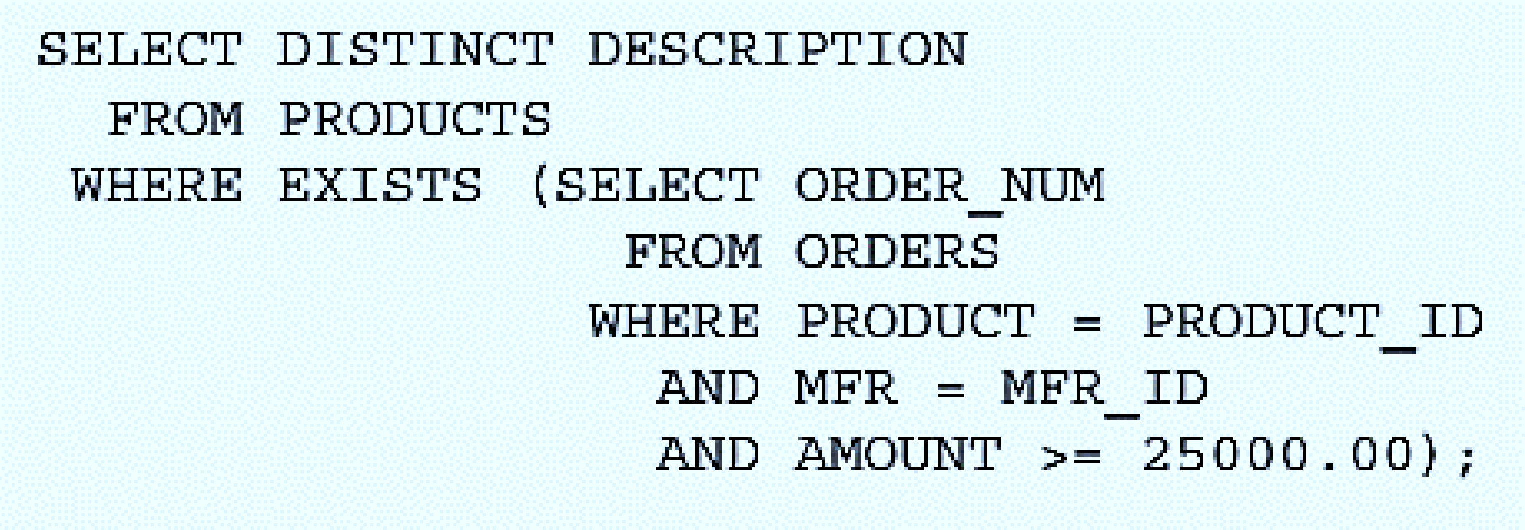


```
SELECT COMPANY
  FROM CUSTOMERS
 WHERE CUST_NUM IN (SELECT DISTINCT CUST
                      FROM ORDERS
                     WHERE MFR = 'ACI'
                       AND PRODUCT LIKE '4100%'
                       AND ORDER_DATE BETWEEN '2008-01-01'
                                          AND '2008-06-30');
```

The Existence Test

- List the products for which an order of Rs. 25,000 or more has been received.

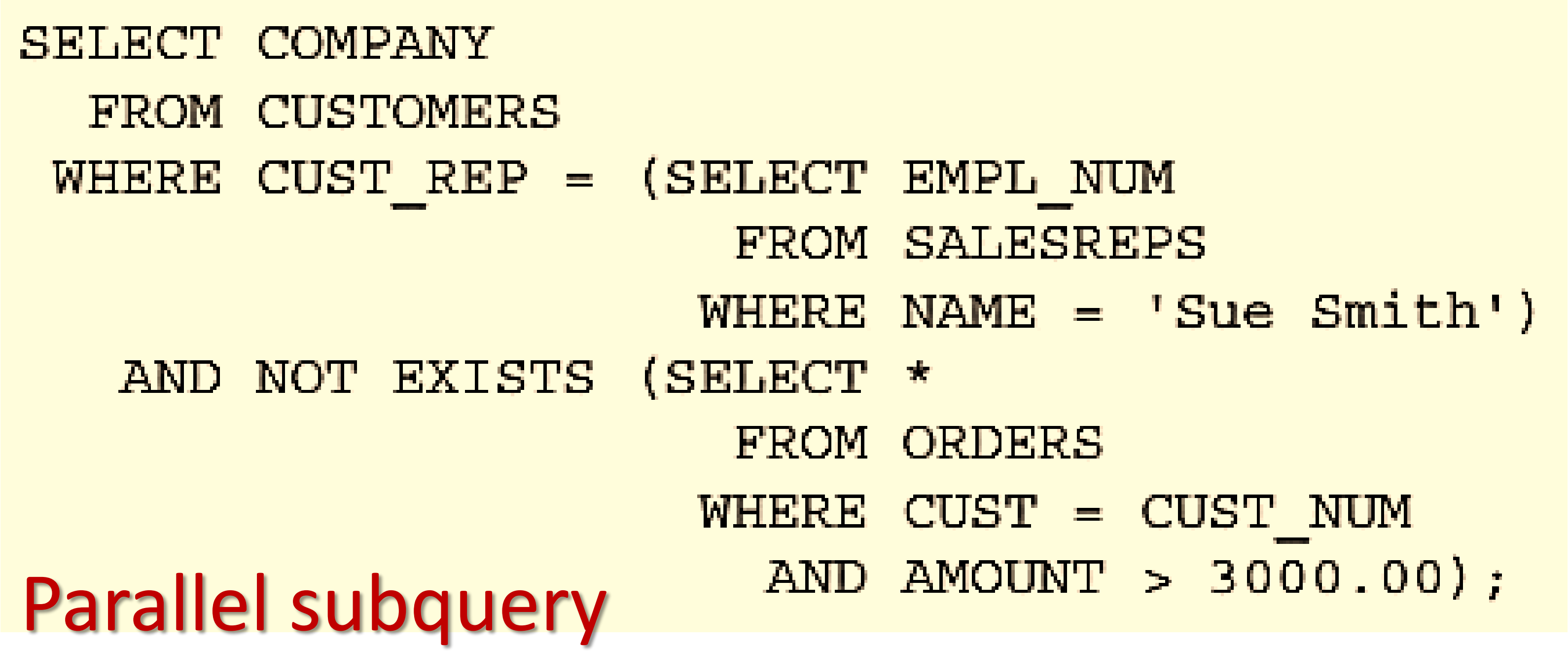
List the products for which there exists at least one order in the ORDERS table (a) that is for the product in question and (b) that has an amount of at least Rs. 25,000.



```
SELECT DISTINCT DESCRIPTION
  FROM PRODUCTS
 WHERE EXISTS (SELECT ORDER_NUM
                 FROM ORDERS
                WHERE PRODUCT = PRODUCT_ID
                  AND MFR = MFR_ID
                  AND AMOUNT >= 25000.00);
```

The Existence Test

- List any customers assigned to Sue Smith who have not placed an order for over Rs. 3000.

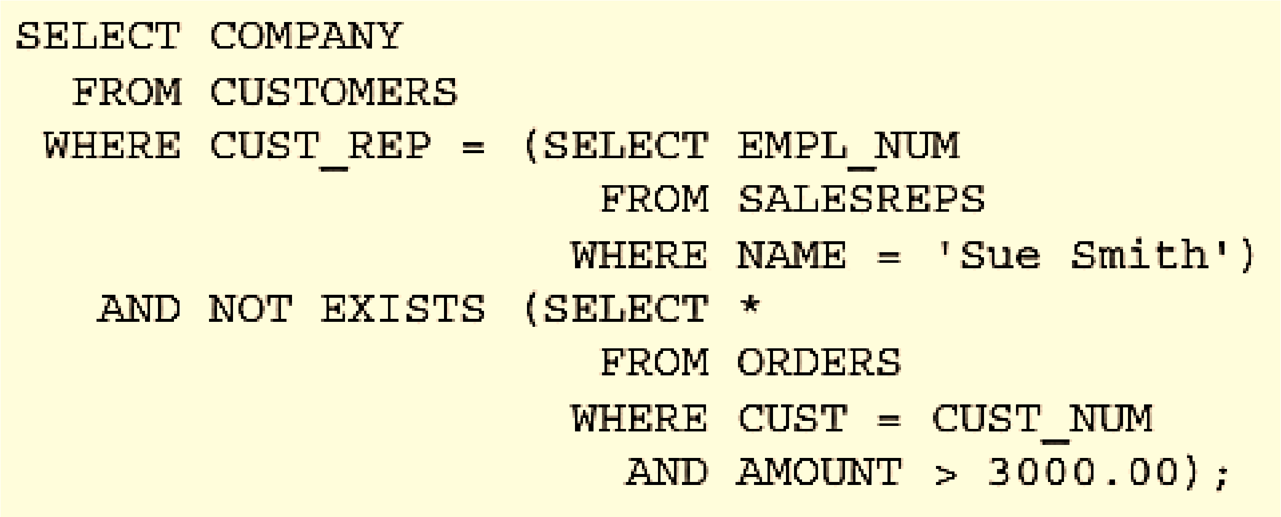


```
SELECT COMPANY
  FROM CUSTOMERS
 WHERE CUST_REP = (SELECT EMPL_NUM
                     FROM SALESREPS
                    WHERE NAME = 'Sue Smith')
   AND NOT EXISTS (SELECT *
                     FROM ORDERS
                    WHERE CUST = CUST_NUM
                      AND AMOUNT > 3000.00);
```

Parallel subquery

The Existence Test-

- List any customers assigned to Sue Smith who have not placed an order for over Rs. 3000.



```
SELECT COMPANY
  FROM CUSTOMERS
 WHERE CUST_REP = (SELECT EMPL_NUM
                     FROM SALESREPS
                    WHERE NAME = 'Sue Smith')
   AND NOT EXISTS (SELECT *
                     FROM ORDERS
                    WHERE CUST = CUST_NUM
                      AND AMOUNT > 3000.00);
```

Customer 1101

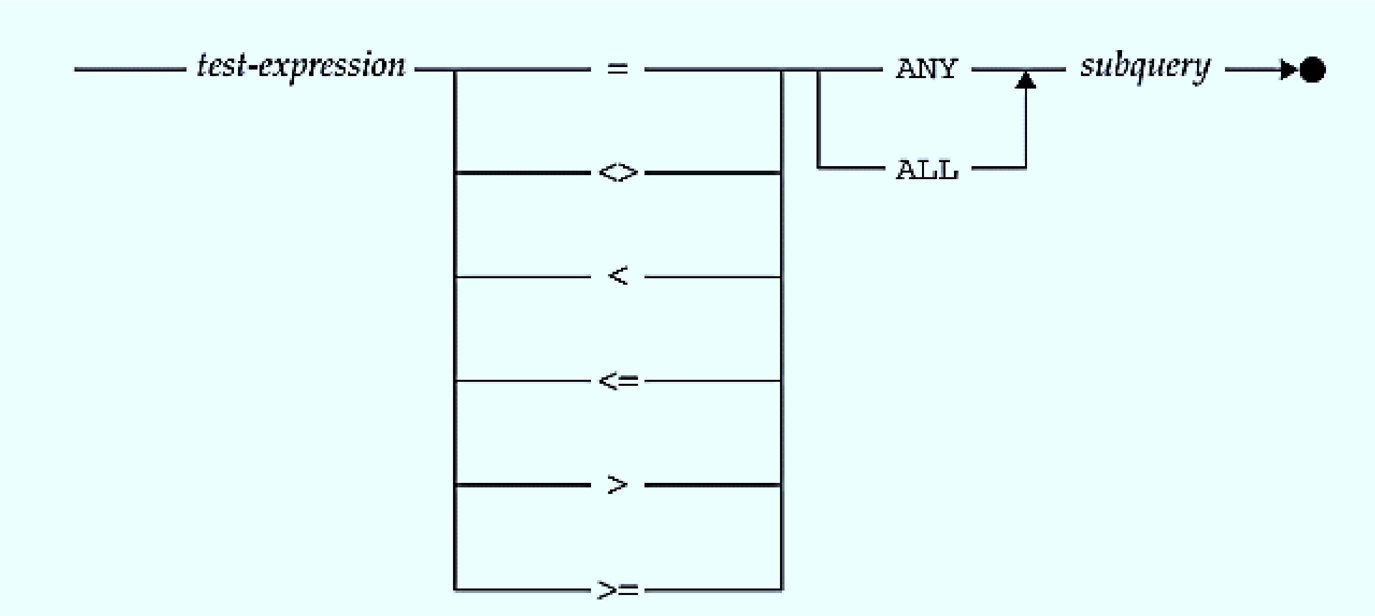
Amount
1000
2000
3000
6000
8000

What about using ‘EXISTS’ and ‘<=3000’ condition?

Customer 1101 should not be selected. But if we use ‘EXISTS’ and ‘<=3000’ conditions, then customer 1101 will be selected.

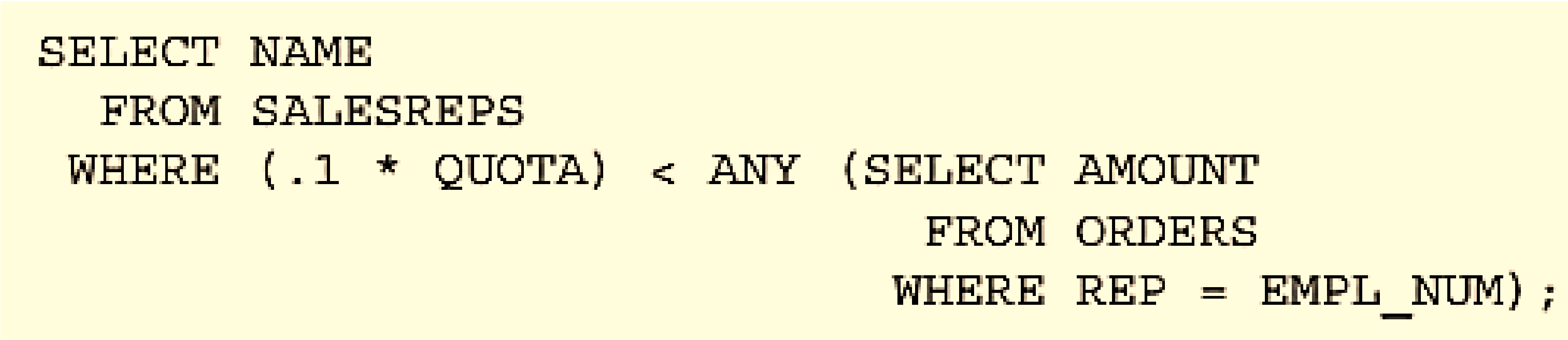
Quantified Comparison Tests

- Any and All



The ANY Test

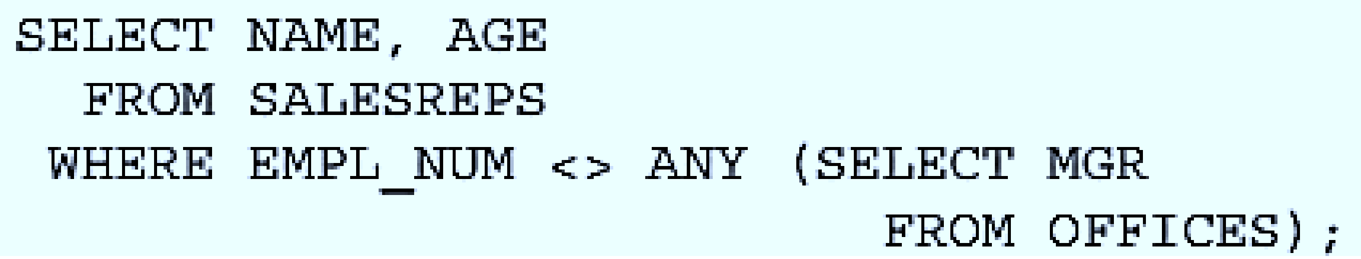
- List the salespeople who have taken an order that represents more than 10% of their quota.



```
SELECT NAME
  FROM SALESREPS
 WHERE (.1 * QUOTA) < ANY (SELECT AMOUNT
                             FROM ORDERS
                            WHERE REP = EMPL_NUM);
```

Quantified Comparison Tests

- List the name and ages of all the people in the sales force who do not manage an office.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE EMPL_NUM <> ANY (SELECT MGR
                          FROM OFFICES);
```

SALESREPS

EMPL_NUM	…
101	
102	
103	
104	
105	
106	
107	

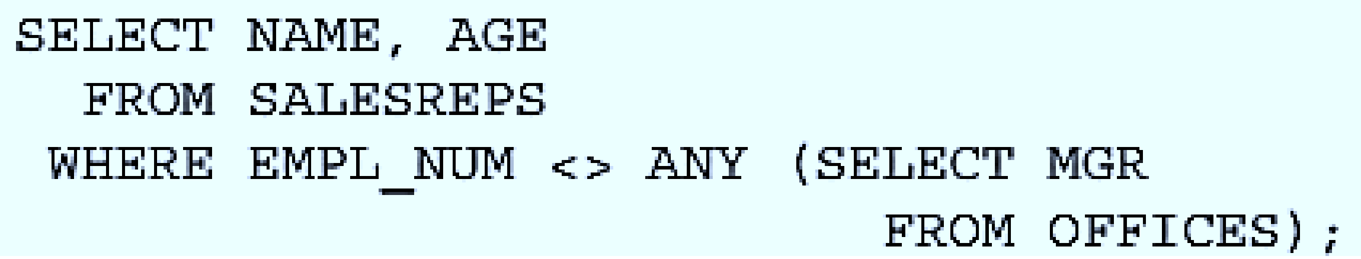
OFFICE	…	MGR
11		102
12		107
13		105

OFFICES

The result will include all salespersons.

Quantified Comparison Tests

- List the name and ages of all the people in the sales force who do not manage an office.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE EMPL_NUM <> ANY (SELECT MGR
                          FROM OFFICES);
```

SALESREPS

EMPL_NUM	…
101	
102	
103	
104	
105	
106	
107	

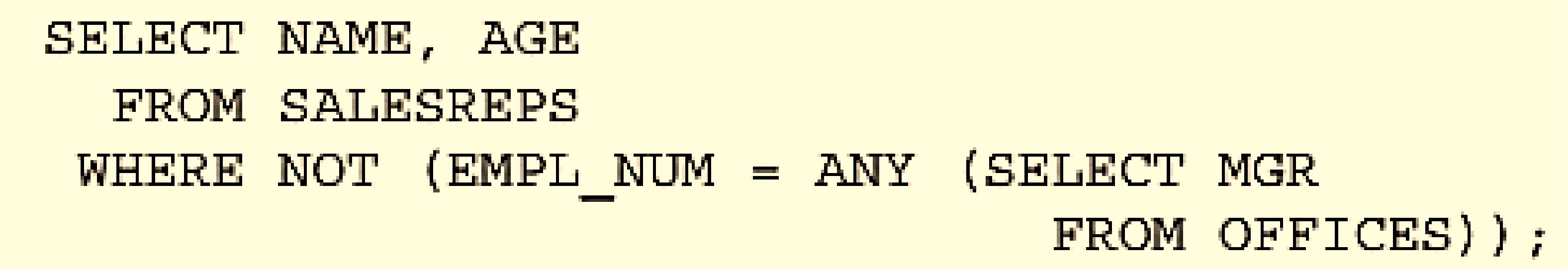
OFFICES

OFFICE	…	MGR
11		102
12		107
13		105

If any of the individual comparison yields a true result, the ‘ANY’ test returns a ‘TRUE’ result

Quantified Comparison Tests

- List the name and ages of all the people in the sales force who do not manage an office.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE NOT (EMPL_NUM = ANY (SELECT MGR
                               FROM OFFICES));
```

SALESREPS

EMPL_NUM	…
101	
102	
103	
104	
105	
106	
107	

OFFICE	…	MGR
11		102
12		107
13		105

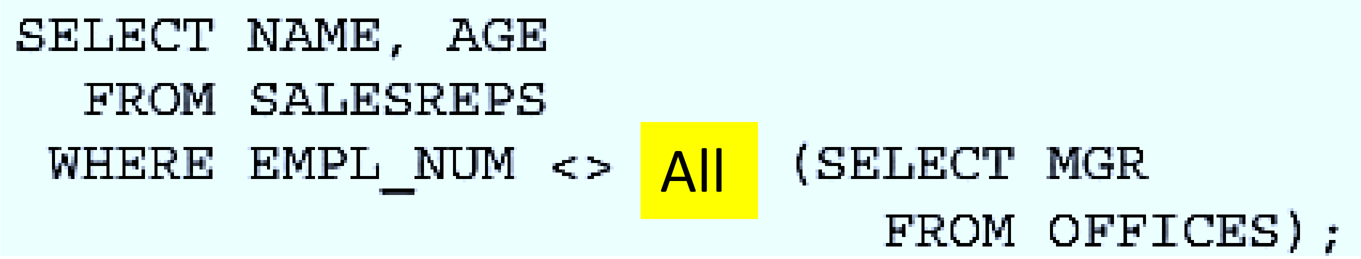
OFFICES

101 → NOT (FALSE) – TRUE (included)

102 → NOT (TRUE) – FALSE (excluded)

Quantified Comparison Tests

- List the name and ages of all the people in the sales force who do not manage an office.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE EMPL_NUM <> All  (SELECT MGR
                           FROM OFFICES);
```

SALESREPS

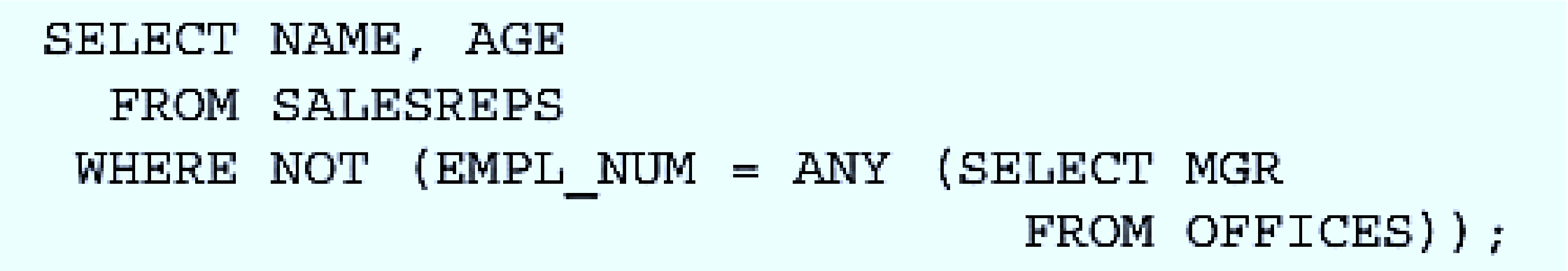
EMPL_NUM	…
101	
102	
103	
104	
105	
106	
107	

OFFICES

OFFICE	…	MGR
11		102
12		107
13		105

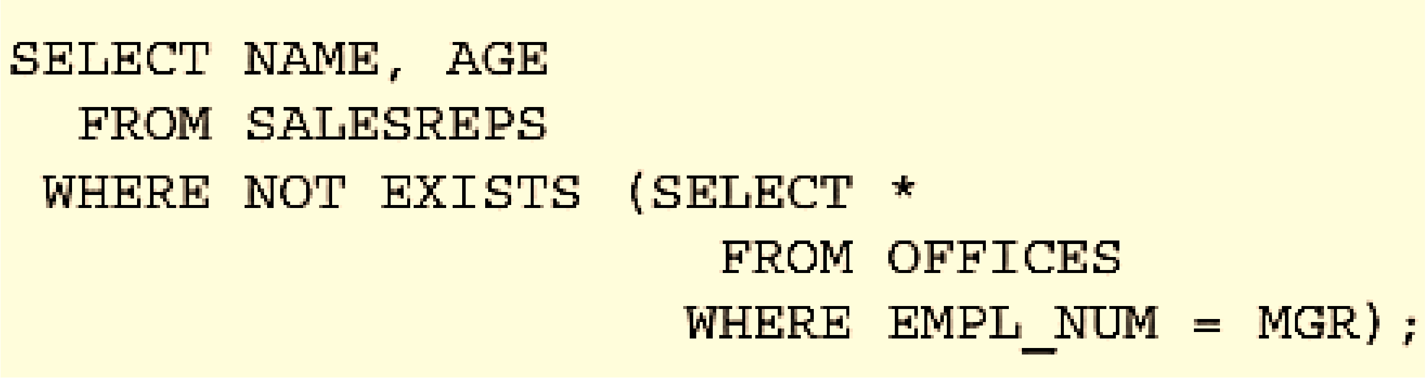
Quantified Comparison Tests

- List the name and ages of all the people in the sales force who do not manage an office.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE NOT (EMPL_NUM = ANY (SELECT MGR
                              FROM OFFICES));
```

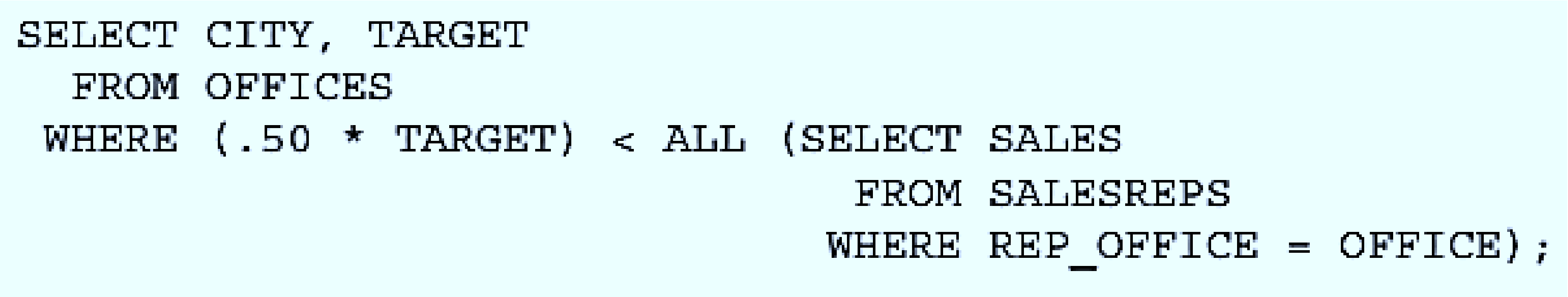
You can always turn a query with an ANY test into a query with an EXISTS test by moving the comparison inside the search condition of the subquery.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE NOT EXISTS (SELECT *
                     FROM OFFICES
                    WHERE EMPL_NUM = MGR);
```

The ALL Test

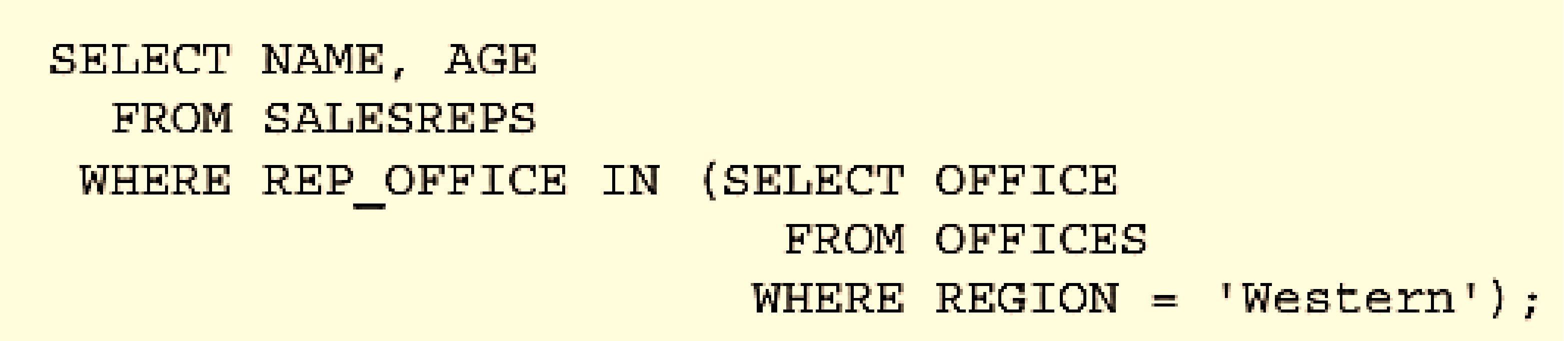
- List the offices (city) and their targets where all of the salespeople have sales that exceeds 50% of the office’s target.



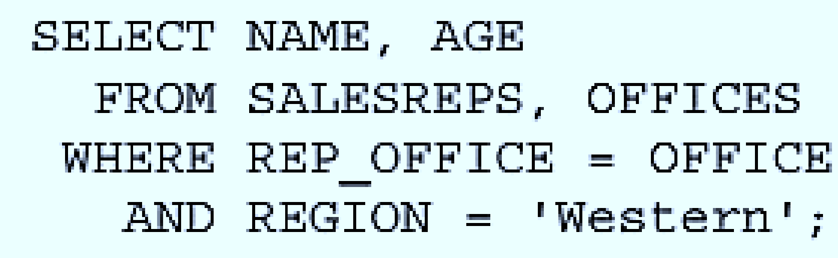
```
SELECT CITY, TARGET
  FROM OFFICES
 WHERE (.50 * TARGET) < ALL (SELECT SALES
                               FROM SALESREPS
                              WHERE REP_OFFICE = OFFICE);
```

Subqueries and Joins

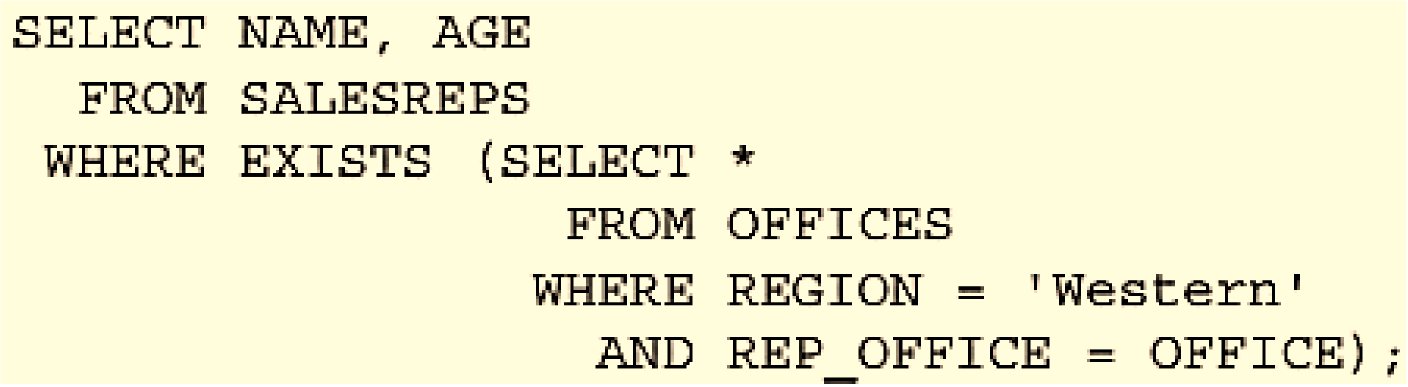
- List the names and ages of salespeople who work in offices in the western region.



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE REP_OFFICE IN (SELECT OFFICE
                        FROM OFFICES
                       WHERE REGION = 'Western');
```



```
SELECT NAME, AGE
  FROM SALESREPS, OFFICES
 WHERE REP_OFFICE = OFFICE
   AND REGION = 'Western';
```



```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE EXISTS (SELECT *
                 FROM OFFICES
                WHERE REGION = 'Western'
                  AND REP_OFFICE = OFFICE);
```

Subqueries and Joins

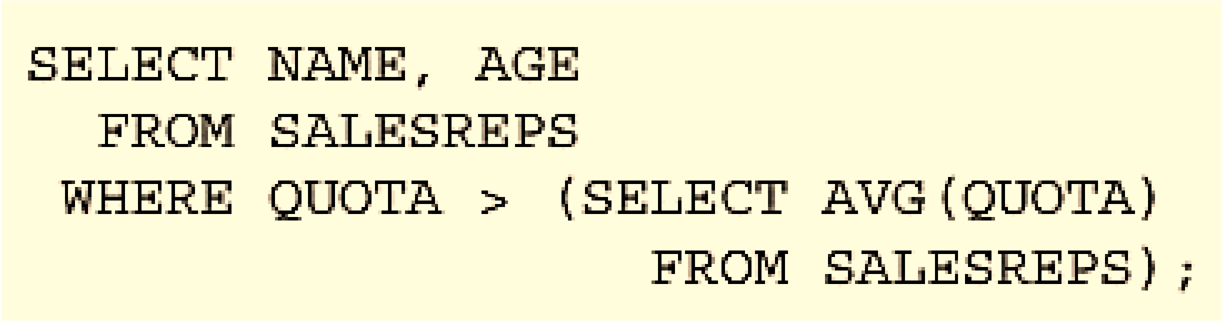
- List the names and ages of salespeople who work in offices in the western region.
- What about the following query
- List the names, ages, and office (city) of the salespeople who work in offices in the western region.

Subquery or Join ?

Ans: Join Why??

Subqueries and Joins

- List the names and ages of salespeople who have above average quotas.

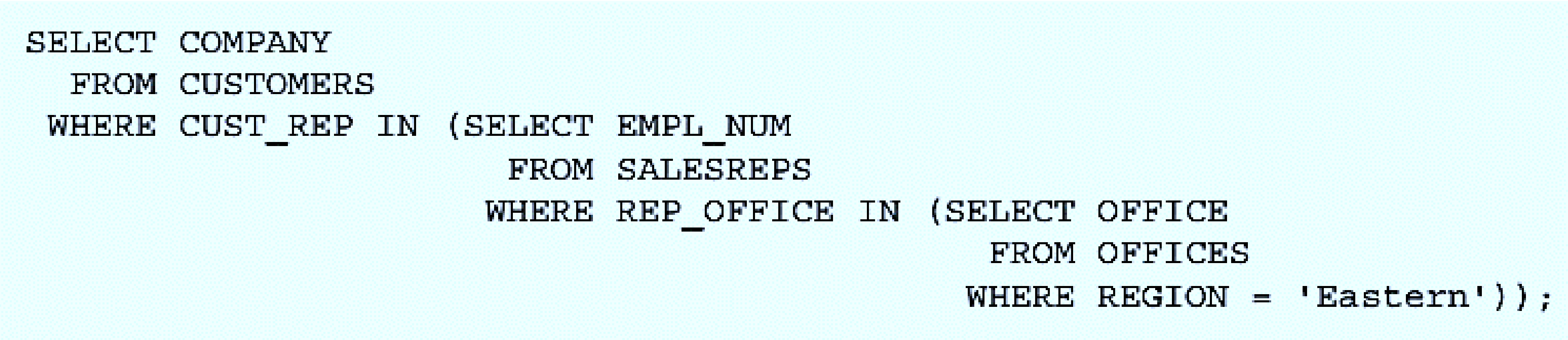


```
SELECT NAME, AGE
  FROM SALESREPS
 WHERE QUOTA > (SELECT AVG(QUOTA)
                  FROM SALESREPS);
```

Many queries with subqueries cannot be translated into an equivalent join.

Nested Subqueries

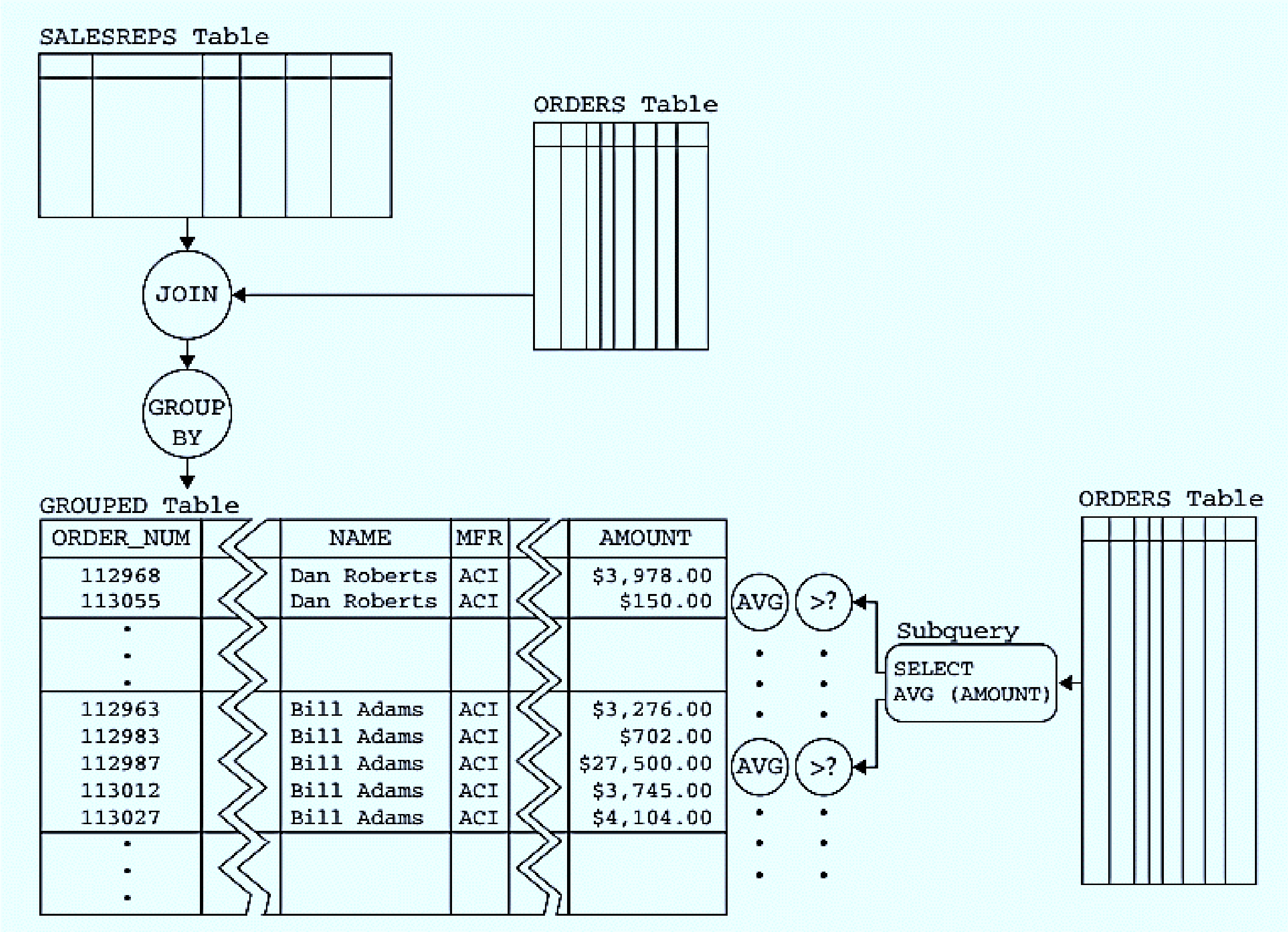
- List the customers whose salespeople are assigned to offices in the eastern sales region.



```
SELECT COMPANY
  FROM CUSTOMERS
 WHERE CUST_REP IN (SELECT EMPL_NUM
                      FROM SALESREPS
                     WHERE REP_OFFICE IN (SELECT OFFICE
                                            FROM OFFICES
                                           WHERE REGION = 'Eastern'));
```

Subqueries in the HAVING Clause

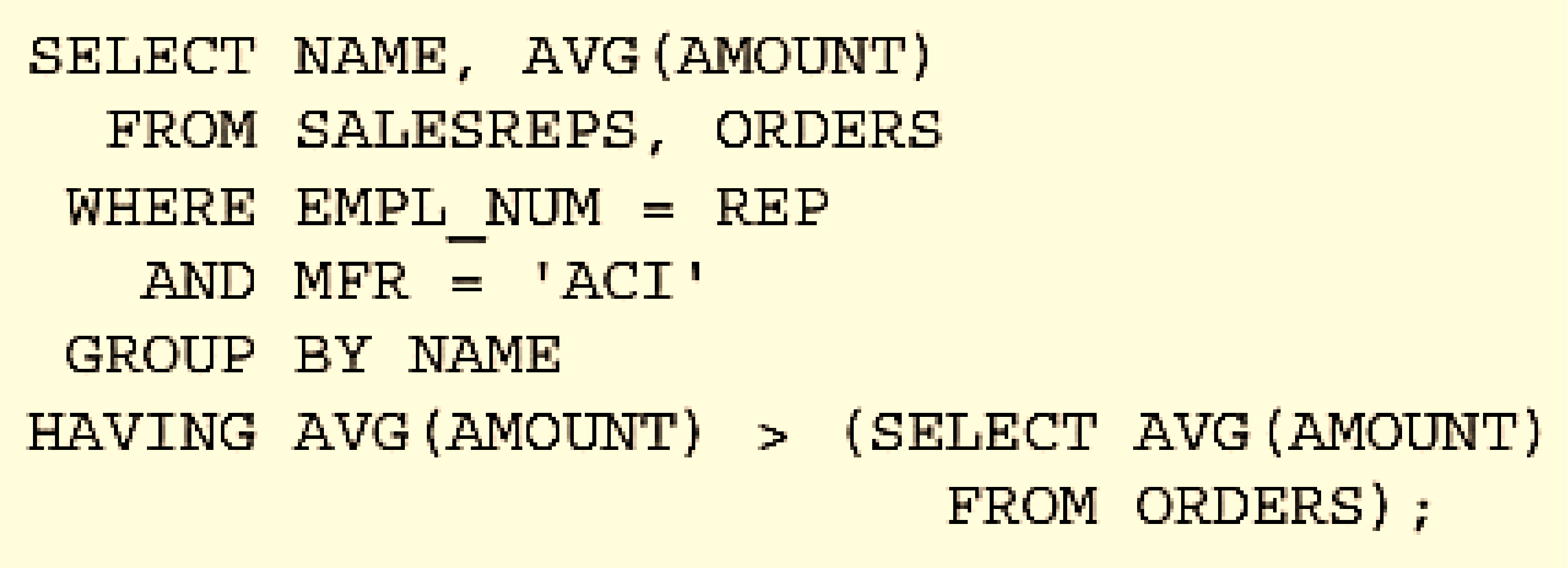
- List the salespeople whose average order size for products manufactured by ACI is higher than the overall average order size.



ORDER_NUM	NAME	MFR	AMOUNT
112968 113055	Dan Roberts Dan Roberts	ACI ACI	$3,978.00 $150.00
. . .			
112963 112983 112987 113012 113027	Bill Adams Bill Adams Bill Adams Bill Adams Bill Adams	ACI ACI ACI ACI ACI	$3,276.00 $702.00 $27,500.00 $3,745.00 $4,104.00
. . .			

Subqueries in the HAVING Clause

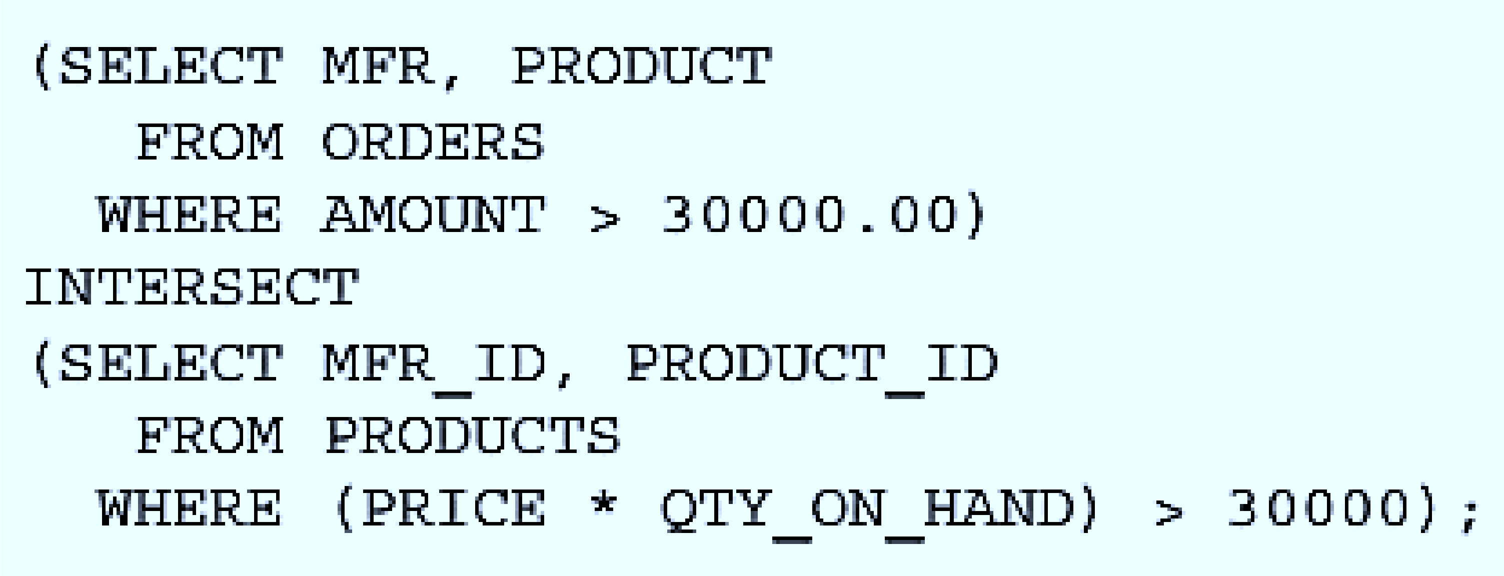
- List the salespeople whose average order size for products manufactured by ACI is higher than the overall average order size.



```
SELECT NAME, AVG(AMOUNT)
  FROM SALESREPS, ORDERS
 WHERE EMPL_NUM = REP
   AND MFR = 'ACI'
 GROUP BY NAME
HAVING AVG(AMOUNT) > (SELECT AVG(AMOUNT)
                        FROM ORDERS);
```

INTERSECT Operation

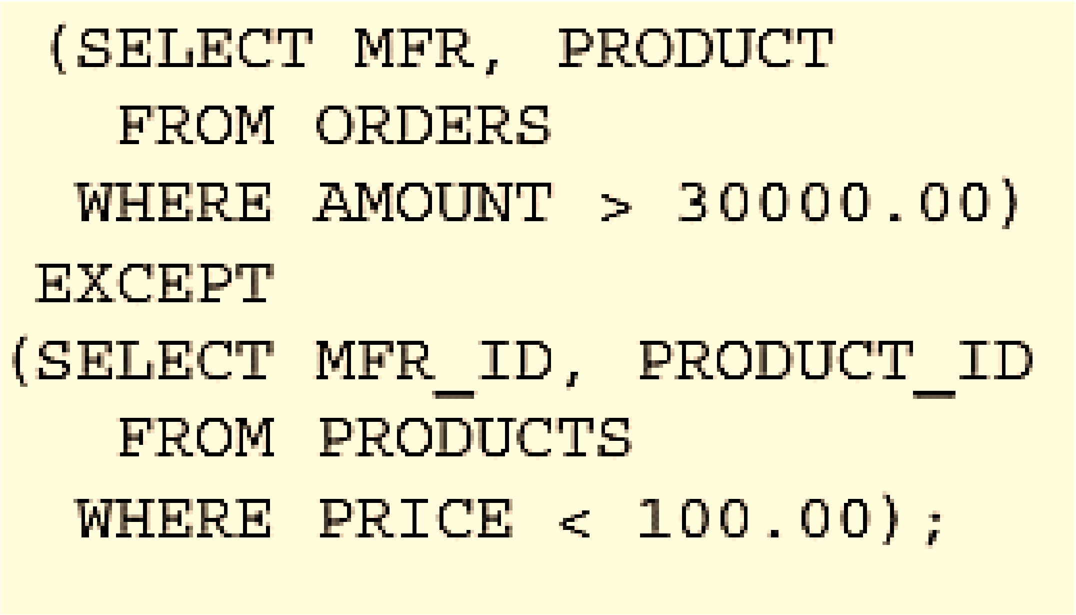
- Show all products for which there is an order over $30,000 and more than $30,000 worth of inventory on hand.



```
(SELECT MFR, PRODUCT
    FROM ORDERS
   WHERE AMOUNT > 30000.00)
INTERSECT
(SELECT MFR_ID, PRODUCT_ID
    FROM PRODUCTS
   WHERE (PRICE * QTY_ON_HAND) > 30000);
```

EXCEPT Operation

- Show all products for which there is an order over $30,000 except for those products that sell for under $100.



```
 (SELECT MFR, PRODUCT
   FROM ORDERS
  WHERE AMOUNT > 30000.00)
 EXCEPT
(SELECT MFR_ID, PRODUCT_ID
   FROM PRODUCTS
  WHERE PRICE < 100.00);
```