

## More SQL Queries

-- Sample Solutions --

1. Select the names of the customers who in one day bought at least three trousers of different prices.

```
SELECT C.name
FROM   sell S, trouser T, customer C
WHERE  S.tcode = T.tcode AND
       S.cid = C.cid
GROUP BY S.day, C.cid, C.name
HAVING COUNT(DISTINCT T.price) >= 3
```

2. Select the trousers that were sold with the highest discount.

```
SELECT S.tcode
FROM   sell S
WHERE  S.discount >= ALL
      (SELECT discount
       FROM   sell)
```

3. Select the customers who bought in one week at least three trousers of different price categories.

I.e., there exists a week during which such a customer bought at least three such trousers.

```
SELECT C.name
FROM   sell S, trouser T,
       datetable D, customer C
WHERE  S.tcode = T.tcode AND
       S.day = D.day
GROUP BY S.cid, C.name, D.week
HAVING count(DISTINCT T.category) >= 3
```

4. Select the trousers that were sold the highest number of times.

```
CREATE VIEW salesCount(tcode,sCount) AS
SELECT S.tcode, count(*)
FROM   sell S
GROUP BY S.tcode

SELECT tcode
FROM   salesCount
WHERE  sCOUNT = (SELECT max(sCount)
                  FROM   salescount)
```

5. Select, for each price, the customer who bought the highest number

of trousers for the price.

```
CREATE VIEW customerPurchasesPrice(cid,price,sCount) AS
SELECT cid, price, count(*)
FROM   sell S, trouser T
WHERE  S.tcode = T.tcode
GROUP BY cid, price
```

```
SELECT P.price, P.cid
FROM   customerPurchasesPrice P
WHERE  P.sCount = (SELECT max(P0.sCount)
                  FROM   customerPurchasesPrice P0
                  WHERE  P.price = P0.price)
```

6. Select, for each category, the customer who bought the lowest number of different trousers in that category.

```
CREATE VIEW customerPurchasesByCat(cid,category,dCount) AS
SELECT cid, category, count(DISTINCT S.tcode)
FROM   sell S, trouser T
WHERE  S.tcode = T.tcode
GROUP BY cid, category
```

```
SELECT P.cid, P.category
FROM   customerPurchasesByCat P
WHERE  P.dCount = (SELECT min(P0.dCount)
                  FROM   customerPurchasesByCat P0
                  WHERE  P.category = P0.category)
```

7. Select the luxury trousers that have been sold at least two times every week.

Idea: Select luxury trousers such that there does not exist a week during which there were sales of that trouser that numbered less than 2, or during which there were no sales at all.

```
SELECT T.tcode
FROM   trouser T
WHERE  T.category = 'luxury' AND
      NOT EXISTS
      (SELECT *
       FROM   datetable DW
       WHERE  2 >
            (SELECT count(*)
             FROM   sell S, datetable DS
             WHERE  DW.week = DS.week AND
                   S.day = DS.day AND
                   S.tcode = T.tcode)
       OR NOT EXISTS
       (SELECT *
        FROM   sell S, datetable DS
        WHERE  DW.week = DS.week AND
              S.day = DS.day AND
              S.tcode = T.tcode))
```

An alternative with outer join:

```
SELECT T.tcode
FROM   trouser T
WHERE  T.category = 'luxury' AND
```

```

2 <= ALL
(SELECT count(*)
FROM (SELECT DISTINCT week
      FROM datetable DW) weeks
LEFT JOIN
(SELECT week
FROM datetable, sell
WHERE datetable.day = sell.day AND
      sell.tcode = T.tcode) sellPerWeek
ON weeks.week = sellPerWeek.week
GROUP BY weeks.week)

```

Note: This query also uses two inline views in the outer join, one of which is correlated with the outer query.

Interestingly, this query runs correctly under PostgreSQL, but not on Oracle, which does not accept the correlated inline view.

8. Select the trousers on offer that have been sold during the week after Christmas (week 52) a number of times below the average of any trouser at any week.

I.e., for each trouser and each week, compute the number of times it has been sold during that week. Then, form the average of weekly sales for each trouser. Then take the minimum of these averages. Finally, look for the trousers where the number of sales in week 52 is below that average.

```

CREATE VIEW trouserSalesPerWeek(tcode, week, scount) AS
SELECT S.tcode, D.week, COUNT(*)
FROM sell S, dateTable D
WHERE S.day = D.day
GROUP BY S.tcode, D.week

```

```

CREATE VIEW avgOfWeeklySales(tcode, wavg) AS
SELECT tspw.tcode, AVG(scount)
FROM trouserSalesPerWeek tspw
GROUP BY tspw.tcode

```

```

SELECT tsp.tcode
FROM trouserSalesPerWeek tsp
WHERE tsp.week = 52 AND
      tsp.scount < (SELECT MIN(wavg)
                   FROM avgOfWeeklySales)

```

One could have written the query using only inline view, but this would have made the query much less readable.

