1.
```sql
-- Find the SIDs of suppliers who supply a red part and a green part.
SELECT DISTINCT SID
FROM CATALOG
WHERE PID IN (
    SELECT PID FROM PARTS WHERE COLOR = 'red'
)
INTERSECT
SELECT DISTINCT SID
FROM CATALOG
WHERE PID IN (
    SELECT PID FROM PARTS WHERE COLOR = 'green'
);
```

2.
```sql
-- Find the SIDs of suppliers who supply a red part or a green part.
SELECT DISTINCT SID
FROM CATALOG
WHERE PID IN (
    SELECT PID FROM PARTS WHERE COLOR = 'red'
)
UNION
SELECT DISTINCT SID
FROM CATALOG
WHERE PID IN (
    SELECT PID FROM PARTS WHERE COLOR = 'green'
);
```

3.
```sql
-- Find the SNAMEs of suppliers who supply every red part and every
   green part.
SELECT S.SNAME
FROM SUPPLIERS S
WHERE NOT EXISTS (
    SELECT PID
    FROM PARTS
    WHERE PARTS.COLOR = 'red'
    MINUS
    SELECT P.PID
    FROM PARTS P JOIN CATALOG C ON P.PID = C.PID
    WHERE C.SID = S.SID)
INTERSECT
SELECT S.SNAME
FROM SUPPLIERS S
WHERE NOT EXISTS (
    SELECT PID
    FROM PARTS
    WHERE PARTS.COLOR = 'green'
    MINUS
    SELECT P.PID
```

```
    FROM PARTS P JOIN CATALOG C ON P.PID = C.PID
    WHERE C.SID = S.SID
);
```

4.
```
-- Find the SNAMEs of suppliers who do not supply every red part.
SELECT S.SNAME
FROM SUPPLIERS S
WHERE EXISTS (
    SELECT PID
    FROM PARTS
    WHERE PARTS.COLOR = 'red'
    MINUS
    SELECT P.PID
    FROM PARTS P JOIN CATALOG C ON P.PID = C.PID
    WHERE C.SID = S.SID
);
```

5.
```
-- For every supplier that only supplies red parts, print the SID and
   the name of the supplier and the average cost of parts that she
   supplies.
SELECT S.SID, S.SNAME, AVG(C.COST) AS AVG_COST
FROM SUPPLIERS S
JOIN CATALOG C ON S.SID = C.SID
JOIN PARTS P ON C.PID = P.PID
WHERE NOT EXISTS (
    SELECT c.SID
    FROM CATALOG c
    JOIN PARTS p ON c.PID = p.PID
    WHERE S.SID = c.SID AND p.COLOR <> 'red'
)
GROUP BY S.SID, S.SNAME;

-- optimal solution
SELECT S.SID, S.SNAME, AVG(C.COST) AS AVG_COST
FROM SUPPLIERS S
JOIN CATALOG C ON S.SID = C.SID
JOIN PARTS P ON C.PID = P.PID
GROUP BY S.SID, S.SNAME
HAVING SUM(CASE WHEN P.COLOR <> 'red' THEN 1 ELSE 0 END) = 0;
```

6.
```
-- For each part, find the SNAMEs of the suppliers who do not charge
   the most for that part. The answer of this query should have two
   columns: PID and SNAME.
SELECT P.PID, S.SNAME
FROM PARTS P
```

```
JOIN CATALOG C ON P.PID = C.PID
JOIN SUPPLIERS S ON C.SID = S.SID
WHERE (P.PID, C.COST) NOT IN (
    SELECT PID, MAX(COST)
    FROM CATALOG
    GROUP BY PID
);
```

7.
```
-- For every part supplied by a supplier who is at the city of Newark,
   print the PID and the SID and the name of the suppliers who sell it
   at the highest price.
SELECT P.PID, S.SID, S.SNAME
FROM CATALOG C
JOIN PARTS P ON C.PID = P.PID
JOIN SUPPLIERS S ON C.SID = S.SID
WHERE S.CITY = 'Newark' AND C.COST = (
    SELECT MAX(COST)
    FROM CATALOG C
    WHERE C.PID = P.PID
);
```

8.
```
-- For every part which has at least two suppliers, find its PID, its
   PNAME and the total number of suppliers who sell it.
SELECT P.PID, P.PNAME, COUNT(DISTINCT C.SID) AS NUMBER_OF_SUPPLIERS
FROM PARTS P
JOIN CATALOG C ON P.PID = C.PID
GROUP BY P.PID, P.PNAME
HAVING COUNT(DISTINCT C.SID) >= 2;
```

9.
```
-- Find the PIDs of parts supplied by every supplier who is at the
   city of Newark or by every supplier who is at the city of Trenton.
SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS (
    (SELECT S.SID
     FROM SUPPLIERS S
     WHERE S.CITY = 'Newark')
    MINUS
    (SELECT C.SID
     FROM CATALOG C
     JOIN SUPPLIERS S ON C.SID = S.SID
     WHERE S.CITY = 'Newark' AND C.PID = P.PID)
)
UNION
SELECT P.PID
FROM PARTS P
```

```
WHERE NOT EXISTS (
    (SELECT S.SID
     FROM SUPPLIERS S
     WHERE S.CITY = 'Trenton')
    MINUS
    (SELECT C.SID
     FROM CATALOG C
     JOIN SUPPLIERS S ON C.SID = S.SID
     WHERE S.CITY = 'Trenton' AND C.PID = P.PID)
);
```

10.
```
-- Find the PIDs of parts supplied by every supplier who is at the
   city of Newark and by every supplier who is at the city of Trenton.
SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS (
    (SELECT S.SID
     FROM SUPPLIERS S
     WHERE S.CITY = 'Newark')
    MINUS
    (SELECT C.SID
     FROM CATALOG C
     JOIN SUPPLIERS S ON C.SID = S.SID
     WHERE S.CITY = 'Newark' AND C.PID = P.PID)
)
INTERSECT
SELECT P.PID
FROM PARTS P
WHERE NOT EXISTS (
    (SELECT S.SID
     FROM SUPPLIERS S
     WHERE S.CITY = 'Trenton')
    MINUS
    (SELECT C.SID
     FROM CATALOG C
     JOIN SUPPLIERS S ON C.SID = S.SID
     WHERE S.CITY = 'Trenton' AND C.PID = P.PID)
);
```

11.
```
-- Find the SIDs of suppliers who supply a red part but do not supply
   a blue part.
SELECT DISTINCT C.SID
FROM CATALOG C
WHERE EXISTS (
    SELECT 1
    FROM PARTS P
    JOIN CATALOG C2 ON P.PID = C2.PID
```

```
    WHERE C2.SID = C.SID AND P.COLOR = 'red')
    AND NOT EXISTS (
    SELECT 1
    FROM PARTS P
    JOIN CATALOG C2 ON P.PID = C2.PID
    WHERE C2.SID = C.SID AND P.COLOR = 'blue'
);

-- optimal solution on Oracle SQL Developer
SELECT DISTINCT SID
FROM CATALOG
WHERE PID IN (
    SELECT PID FROM PARTS WHERE COLOR = 'red'
)
MINUS
SELECT DISTINCT SID
FROM CATALOG
WHERE PID IN (
    SELECT PID FROM PARTS WHERE COLOR = 'blue'
);
```

12.
```
-- For every supplier who supplies at least 4 parts, find his SID,
   SNAME and the PID of the most expensive part(s) that he supplies.
SELECT S.SID, S.SNAME, C.PID
FROM SUPPLIERS S
JOIN CATALOG C ON S.SID = C.SID
WHERE C.SID IN (
    SELECT SID
    FROM CATALOG
    GROUP BY SID
    HAVING COUNT(PID) > 3)
AND C.COST = (
    SELECT MAX(COST)
    FROM CATALOG C1 WHERE C1.SID = C.SID GROUP BY SID HAVING
       COUNT(PID) > 3
);
```

13.
```
-- For every distinct color of the parts, find the total number of
   suppliers who supply a part of this color.
SELECT P.COLOR, COUNT(DISTINCT S.SID) AS NUMBER_OF_SUPPLIERS
FROM CATALOG C
JOIN PARTS P ON C.PID = P.PID
JOIN SUPPLIERS S ON S.SID = C.SID
GROUP BY P.COLOR;
```

14.
```
-- Find the SIDs of suppliers who supply at least two parts of
   different color.
```

```sql
SELECT C.SID
FROM CATALOG C
JOIN PARTS P ON C.PID = P.PID
JOIN SUPPLIERS S ON S.SID = C.SID
GROUP BY C.SID
HAVING COUNT(DISTINCT P.COLOR) >= 2;
```

15.
```sql
-- For every part which has a supplier, find its PID, PNAME, its
   average cost, maximum cost, and minimum cost.
SELECT P.PID, P.PNAME, AVG(C.COST) AS AVG_COST, MAX(C.COST) AS
   MAX_COST, MIN(C.COST) AS MIN_COST
FROM PARTS P
JOIN CATALOG C ON P.PID = C.PID
GROUP BY P.PID, P.PNAME;
```