

Questions on SQL – Part 1

- By [admin](#)
- September 16, 2015
- [Database, Some Solved Questions on DBMS, SQL](#)

Questions on SQL

Question 1 : Consider the following schema:

Suppliers(sid: integer, sname: string, address: string)

Parts(pid: integer, pname: string, color: string)

Catalog(sid: integer, pid: integer, cost: real)

The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in SQL:

1. Find the pnames of parts for which there is some supplier.

Solution :

```
SELECT p.pname FROM Parts AS p  
  
WHERE EXISTS (  
  
SELECT * FROM Catalog AS c  
  
WHERE c.pid = p.pid)
```

2. Find the snames of suppliers who supply every red part.

Solution : SELECT s.sid, s.sname

```
FROM Suppliers AS s, Catalog AS c, Parts AS p  
  
WHERE s.sid = c.sid  
  
AND p.pid = c.pid AND p.color = 'red'
```

```
GROUP BY s.sid, s.sname
```

```
HAVING COUNT(*) = (SELECT COUNT(*) FROM Parts AS p1
```

```
WHERE p1.color='red')
```

3. Find the snames of suppliers who supply every part.

Solution : SELECT S.sname FROM Suppliers S

```
WHERE NOT EXISTS (( SELECT P.pid FROM Parts P )
```

```
EXCEPT
```

```
( SELECT C.pid FROM Catalog C
```

```
WHERE C.sid = S.sid ))
```

4. Find the pnames of parts supplied by Acme Widget Suppliers and no one else.

Solution : SELECT p.pname FROM Parts AS p

```
WHERE p.pid IN (
```

```
(SELECT c1.pid AS pid FROM Catalog AS c1, Suppliers s1
```

```
WHERE c1.pid = s1.sid
```

```
AND s1.sname = 'Acme Widget Suppliers')
```

```
EXCEPT
```

```
(SELECT DISTINCT c2.pid AS pid
```

```
FROM Catalog AS c2, Suppliers s2
```

```
WHERE c2.pid = s2.sid
```

```
AND s2.sname != 'Acme Widget Suppliers'))
```

5. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part.

Solution : SELECT DISTINCT c.sid FROM Catalog AS c,

(SELECT c1.pid AS avg pid, AVG(c1.cost) AS avg cost

FROM Catalog AS c1 GROUP BY c1.pid) AS TEMP

WHERE c.pid = TEMP.avg pid AND c.cost > TEMP.avg cost

6. Find the sids of suppliers who supply a red part and a green part.

Solution : (SELECT DISTINCT c1.sid AS SID

FROM Parts AS p1, Catalog AS c1

WHERE p1.pid = c1.pid AND p1.color = 'green')

INTERSECT (

SELECT DISTINCT c2.sid AS SID

FROM Parts AS p2, Catalog AS c2

WHERE p2.pid = c2.pid AND p2.color = 'red')

or

SELECT DISTINCT c.sid

FROM Catalog c, Parts p

WHERE c.pid = p.pid and p.color = 'red'

INTERSECT

SELECT DISTINCT c.sid

FROM Catalog c, Parts p

```
WHERE c.pid = p.pid and p.color = 'green'
```

7. For every supplier who supplies at least 1 red part, print the name of the supplier and the total number of red parts that he or she supplies.

Solution : SELECT S.sname, COUNT(*) as PartCount

```
FROM Suppliers S, Catalog C, Parts P
```

```
WHERE C.sid = S.sid and P.pid = C.pid and P.color = 'Red'
```

```
GROUP BY S.sname, S.sid
```

8. For each part, find the sname of the supplier who charges the most for that part.

Solution : SELECT P.pid, S.sname

```
FROM Parts P, Suppliers S, Catalog C
```

```
WHERE C.pid = P.pid
```

```
AND C.sid = S.sid
```

```
AND C.cost = (SELECT MAX (C1.cost)
```

```
FROM Catalog C1
```

```
WHERE C1.pid = P.pid)
```

9. Find the sids of suppliers who supply only red parts.

Solution : SELECT DISTINCT C.sid

```
FROM Catalog C
```

```
WHERE NOT EXISTS ( SELECT *
```

```
FROM Parts P
```

```
WHERE P.pid = C.pid AND P.color <> 'Red' )
```

10. Find the sids of suppliers who supply a red part or a green part.

Solution : SELECT DISTINCT C.sid

FROM Catalog C, Parts P

WHERE C.pid = P.pid AND P.color = 'Red'

UNION

SELECT DISTINCT C1.sid

FROM Catalog C1, Parts P1

WHERE C1.pid = P1.pid AND P1.color = 'Green'

11. For every supplier that supplies a green part and a red part, print the name and price of the most expensive part that she supplies.

Solution : SELECT S.sname, MAX(C.cost) as MaxCost

FROM Suppliers S, Parts P, Catalog C

WHERE P.pid = C.pid AND C.sid = S.sid

GROUP BY S.sname, S.sid

HAVING ANY (P.color='green') AND ANY (P.color = 'red')

12. Find the distinct pname of all parts.

Solution : SELECT DISTINCT pname

FROM Parts;

13. Find the distinct pname of parts for which there is some supplier.

Solution : SELECT DISTINCT P.pname

FROM Parts P, Catalog C

WHERE P.pid = C.pid;

14. Find the distinct pnames of all parts sold (by some supplier) for less than 5.50

Solution : SELECT DISTINCT P.pname

FROM Parts P, Catalog C

WHERE P.pid = C.pid

AND C.cost<5.50;

15. Find the price of the least expensive red part.

Solution : SELECT MIN(C.cost)

FROM Catalog C, Parts P

WHERE C.pid = P.pid

AND P.color = 'Red';

16. For every supplier, print the name of the supplier and the total number of parts that he or she supplies.

Solution : SELECT S.sname, COUNT(*) as PartCount

FROM Suppliers S, Catalog C

WHERE C.sid = S.sid

GROUP BY S.sname, S.sid

17. Find the pids of the most expensive parts supplied by suppliers named Yosemite Sham.

Solution : SELECT C.pid

FROM Catalog C, Suppliers S

WHERE S.sname = 'Yosemite Sham' AND

C.sid = S.sid AND

```
C.cost ≥ ALL (SELECT C2.cost  
  
FROM Catalog C2, Suppliers S2  
  
WHERE S2.sname = 'Yosemite Sham' AND  
  
C2.sid = S2.sid)
```

18. For every supplier, print the name and price of the most expensive part that she supplies.

Solution : SELECT S.sname, MAX(C.cost) as MaxCost

```
FROM Suppliers S, Parts P, Catalog C  
  
WHERE P.pid = C.pid AND C.sid = S.sid  
  
GROUP BY S.sname, S.sid
```

Questions Based on SQL – Part 6

- By [admin](#) |
- September 17, 2015 |
- [Database, Question Bank of DBMS, Some Solved Questions on DBMS, SQL](#)

Questions Based on SQL – Part 6

Question 1 : Consider the following relations:

Student(snum: integer, sname: string, major: string,
level: string, age: integer)

Class(name: string, meets at: string, room: string,
fid: integer)

Enrolled(snum: integer, cname: string)

Faculty(fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.

Write the following queries in SQL. No duplicates should be printed in any of the answers.

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by I. Teach.

Solution : `SELECT DISTINCT S.Sname`

`FROM Student S, Class C, Enrolled E, Faculty F`

`WHERE S.snum = E.snum`

`AND E.cname = C.name`

`AND C.fid = F.fid AND`

`F.fname = 'I.Teach' AND S.level = 'JR'`

2. Find the age of the oldest student who is either a History major or enrolled in a course taught by I. Teach.

Solution : `SELECT MAX(S.age)`

`FROM Student S`

`WHERE (S.major = 'History')`

`OR S.snum IN (SELECT E.snum`

`FROM Class C, Enrolled E, Faculty F`

`WHERE E.cname = C.name AND C.fid = F.fid`

```
AND F.fname = 'I.Teach' )
```

3. Find the names of all classes that either meet in room R128 or have five or more students enrolled.

Solution : SELECT C.name

```
FROM Class C
```

```
WHERE C.room = 'R128'
```

```
OR C.name IN (SELECT E.cname
```

```
FROM Enrolled E
```

```
GROUP BY E.cname
```

```
HAVING COUNT (*) >= 5)
```

4. Find the names of all students who are enrolled in two classes that meet at the same time.

Solution : SELECT DISTINCT S.sname

```
FROM Student S
```

```
WHERE S.snum IN (SELECT E1.snum
```

```
FROM Enrolled E1, Enrolled E2, Class C1, Class C2
```

```
WHERE E1.snum = E2.snum AND E1.cname <> E2.cname
```

```
AND E1.cname = C1.name
```

```
AND E2.cname = C2.name AND C1.meets at = C2.meets at)
```

5. Find the names of faculty members who teach in every room in which some class is taught.

Solution : SELECT DISTINCT F.fname

```
FROM Faculty F
```

```
WHERE NOT EXISTS (( SELECT *  
  
FROM Class C )  
  
EXCEPT  
  
(SELECT C1.room  
  
FROM Class C1  
  
WHERE C1.fid = F.fid ))
```

6. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

Solution : SELECT DISTINCT F.fname

```
FROM Faculty F  
  
WHERE 5 > (SELECT COUNT (E.snum)  
  
FROM Class C, Enrolled E  
  
WHERE C.name = E.cname  
  
AND C.fid = F.fid)
```

7. For each level, print the level and the average age of students for that level.

Solution : SELECT S.level, AVG(S.age)

```
FROM Student S  
  
GROUP BY S.level
```

8. For all levels except JR, print the level and the average age of students for that level.

Solution : SELECT S.level, AVG(S.age)

```
FROM Student S
```

```
WHERE S.level <> 'JR'
```

```
GROUP BY S.level
```

9. For each faculty member that has taught classes only in room R128, print the faculty member's name and the total number of classes she or he has taught.

Solution : SELECT F.fname, COUNT(*) AS CourseCount

```
FROM Faculty F, Class C
```

```
WHERE F.fid = C.fid
```

```
GROUP BY F.fid, F.fname
```

```
HAVING EVERY ( C.room = 'R128' )
```

10. Find the names of students enrolled in the maximum number of classes.

Solution : SELECT DISTINCT S.sname

```
FROM Student S
```

```
WHERE S.snum IN (SELECT E.snum
```

```
FROM Enrolled E
```

```
GROUP BY E.snum
```

```
HAVING COUNT (*) >= ALL (SELECT COUNT (*)
```

```
FROM Enrolled E2
```

```
GROUP BY E2.snum ))
```

11. Find the names of students not enrolled in any class.

Solution : SELECT DISTINCT S.sname

```
FROM Student S
```

```
WHERE S.snum NOT IN (SELECT E.snum
```

```
FROM Enrolled E )
```

12. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

Solution : SELECT S.age, S.level

```
FROM Student S
```

```
GROUP BY S.age, S.level,
```

```
HAVING S.level IN (SELECT S1.level
```

```
FROM Student S1
```

```
WHERE S1.age = S.age
```

```
GROUP BY S1.level, S1.age
```

```
HAVING COUNT (*) >= ALL (SELECT COUNT (*)
```

```
FROM Student S2
```

```
WHERE s1.age = S2.age
```

```
GROUP BY S2.level, S2.age))
```