Commonly Asked SQL Questions :-

by Akarshan Kapoor

1. Write a query to add multiple columns in a table.

```
INSERT INTO users VALUES
(Null, 'abc', 2323),
(Null, 'xyz' , 6969)
```

2. Write a query to add/ delete multiple columns in a table.

```
#adding multiple columns into table#
ALTER TABLE users
ADD email varchar(255) NOT NULL UNIQUE,
ADD password varchar(255) NOT NULL;


#deleting multiple columns from table#
ALTER TABLE users
DROP COLUMN email,
DROP COLUMN password;
```

3. Write a query to find out the second highest value in a table.

```
SELECT * FROM student
ORDER BY score DESC
LIMIT 1,1; #2nd highest;
```

4. Find the max value without using order by

```
SELECT name, budget FROM movies
WHERE budget = ( SELECT MAX(budget) FROM movies );
```

5. Find managers of all employees (**FINDING HIERARCHY)

```
SELECT e1.fname, e2.fname   #using_selfjoin
FROM employees e1
JOIN employees e2
ON e1.manager_id = e2.emp_id;
```

6. Use of Wild cards.

```
SELECT * FROM movies
WHERE name LIKE 'A_S_';

       or

SELECT * FROM movies
WHERE name LIKE '%man';  #batman,antman
```

7. If else based questions

```
SELECT name, family_size,
CASE
```

```
    WHEN family_size <=2 THEN 'small'
    WHEN family_size > AND family_size <=4 THEN 'medium'
    ELSE 'large'

END AS 'type'
FROM family;
```

8. Find category- wise top value. #correlated_query

```
SELECT genre, name, score
FROM movies m1
WHERE score = ( SELECT MAX(score) FROM movies m2 WHERE m2.genre = m1.genre );
```

9. Find / Delete all duplicate values.

```
DELETE FROM contacts
WHERE id NOT IN ( SELECT MIN(id) FROM contacts GROUP BY first_name,
last_name, email )
```

10. Joining Tables.

```
SELECT name, age, group_name
FROM users u
JOIN membership m
 ON u.user_id = m.user_id
JOIN groups g
 ON m.group_id = g.group_id;
```

11. Write a SQL query to fetch all the duplicate records in a table

```
SELECT user_id, user_name, email
(
SELECT *,
row_number() over(partition by user_name order by user id) as rn
FROM users
order by user_id )x
where x.rn>1;

           --OR--

DELETE FROM employees
WHERE id NOT IN (
    SELECT MIN(id)
    FROM employees
    GROUP BY first_name, last_name
);
```

12. Fetch the details of doctors who work in the same hospital but in different specialty.

```
SELECT d1.*
FROM doctors d1
JOIN doctors d2
       ON d1.id <> d2.id AND
          d1.hospital = d2.hospital AND
          d1.speciality <> d2.speciality;
```

13. Fetch users who logged in consecutively 3 or more times.

```
SELECT login_id, user_name, login_date
FROM (
SELECT *,
CASE WHEN user_name = lead(user_name) over(order by login_id)
          AND user_name =lead(user_name,2) over(order by login_id)
     THEN user_name
     else null
end as repeated_users
FROM users_login) x
WHERE x.repeated_users is not null;
```

Follow me for more .. [click here]