



Structured
Query
Language

Top 30 Câu Hỏi Phỏng Vấn SQL

(Bản Thực Hành)

Học nhanh — Nhớ lâu — Phỏng vấn tự tin

Trang 1

Luyện SQL mỗi ngày,
Giải thật nhiều bài,
Tư duy như một
Database !

Q1. Tìm tất cả nhân viên có lương lớn hơn 50.000

Bảng: Employees

Lời giải SQL:

EmpId	Name	Salary	HireDate
1	Hung	65000	2023-07-15
2	Lan	42000	2024-02-20
3	Minh	75000	2022-11-05
4	Nga	52000	2021-06-18

```
SELECT *  
FROM Employees  
WHERE Salary > 50000;
```

✅ Kết quả: Trả về các nhân viên có mức lương lớn hơn 50.000.

Mẹo phỏng vấn 💡

Mệnh đề WHERE lọc từng dòng trước
khi gom nhóm và tổng hợp dữ liệu.

Q2. Tìm nhân viên có tên bắt đầu bằng chữ 'A'

Lời giải SQL:

Ký tự đại diện với LIKE

Ví dụ

```
SELECT *  
FROM Employees  
WHERE Name LIKE 'A%';
```

Mẫu	Ý nghĩa
A%	Bắt đầu bằng A
%A	Kết thúc bằng A
%A%	Chứa A ở bất kỳ đâu
_	Đúng một ký tự

Name	Khớp 'A%' ?
An	✓ Có
Anh	✓ Có
Am	✓ Có
Khoa	✗ Không
Nga	✗ Không

★ Mẹo phỏng vấn

LIKE là câu hỏi rất hay gặp ở vòng sàng lọc SQL.

Q3. Tìm nhân viên được tuyển dụng trong năm 2024

Lời giải SQL (Cách 1)

Cách 2 [Thân thiện với index]

```
SELECT *  
FROM Employees  
WHERE YEAR(HireDate) = 2024;
```

```
SELECT *  
FROM Employees  
WHERE HireDate >= '2024-01-01'  
AND HireDate < '2025-01-01';
```

Mẹo thực chiến 🚀

Tránh bọc cột ngày đã đánh index vào
trong hàm (YEAR(), MONTH()) vì sẽ làm
mất tác dụng của index. Lọc theo khoảng
ngày sẽ nhanh hơn nhiều.

Thứ tự thực thi SQL

- ① FROM
- ↓
- ② WHERE
- ↓
- ③ SELECT

✗ Lỗi thường gặp

- Dùng '=' thay cho LIKE.
- Dùng YEAR(HireDate) trên bảng lớn đã đánh index.
- Quên dấu % khi so khớp mẫu.

Ôn nhanh 🎯

- ✓ WHERE lọc theo từng dòng.
- ✓ LIKE dùng để so khớp mẫu chuỗi.
- ✓ Lọc theo khoảng ngày giúp tận dụng index.



Q4. Tính lương trung bình theo từng phòng ban

Bảng: Employees

EmpId	Name	DeptId	Salary
1	Hung	10	65000
2	Lan	10	42000
3	Minh	20	75000
4	Nga	20	52000
5	Tuan	30	61000
6	Mai	30	48000

Lời giải SQL:

```
SELECT DeptId,  
       AVG(Salary) AS AvgSalary  
FROM   Employees  
GROUP BY DeptId  
ORDER BY DeptId;
```

Kết quả:

DeptId	AvgSalary
10	53500.00
20	63500.00
30	54500.00

★ Mẹo phỏng vấn

Dùng GROUP BY để gom dữ liệu và tính giá trị tổng hợp cho từng nhóm.

Khái niệm cốt lõi

- ✓ AVG() bỏ qua giá trị NULL
- ✓ GROUP BY đứng sau mệnh đề WHERE
- ✓ Dùng ORDER BY để sắp xếp kết quả

Q5. Đếm số nhân viên trong mỗi phòng ban

Lời giải SQL:

```
SELECT DeptId,  
       COUNT(*) AS EmpCount  
FROM   Employees  
GROUP BY DeptId  
ORDER BY EmpCount DESC;
```

Kết quả:

DeptId	EmpCount
10	2
20	2
30	2

Mẹo phỏng vấn 💡

COUNT(*) đếm tất cả các dòng, kể cả dòng có NULL.
COUNT(cột) chỉ đếm các giá trị khác NULL.

Giải thích:

- COUNT(*) đếm số dòng trong mỗi nhóm (mỗi phòng ban).
- Kết quả được sắp xếp theo EmpCount giảm dần.

Biến thể thường gặp

- Đếm nhân viên có lương > 50000
- Đếm số nhân viên khác nhau trong mỗi phòng
- Đếm phòng ban có nhiều hơn 3 nhân viên

Q6. Tìm mức lương cao nhất theo từng phòng ban

Lời giải SQL:

```
SELECT DeptId,  
       MAX(Salary) AS MaxSalary  
FROM   Employees  
GROUP BY DeptId  
ORDER BY MaxSalary DESC;
```

Kết quả:

DeptId	MaxSalary
20	75000
10	65000
30	61000

★ Mẹo phỏng vấn

MAX() trả về giá trị lớn nhất trong mỗi nhóm.
Dùng MIN() để lấy giá trị nhỏ nhất.

Tra nhanh hàm tổng hợp

COUNT() → Đếm số dòng
SUM() → Tổng các giá trị
AVG() → Giá trị trung bình
MAX() → Giá trị lớn nhất

✗ Lỗi thường gặp

- Quên GROUP BY khi dùng hàm tổng hợp.
- Dùng hàm tổng hợp trong WHERE (phải dùng HAVING).
- Chọn cột không nằm trong GROUP BY.

Ghi nhớ nhanh 📝

Thứ tự thực thi khi tổng hợp

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY



Q7. Lấy tên nhân viên kèm tên phòng ban (INNER JOIN)

Các bảng:

Employees

EmpId	Name	DeptId
1	Hung	10
2	Lan	20
3	Minh	10
4	Nga	30

Departments

DeptId	DeptName
10	HR
20	Finance
30	IT

Lời giải SQL:

```
SELECT E.EmpId, E.Name, D.DeptName
FROM Employees E
INNER JOIN Departments D
ON E.DeptId = D.DeptId
ORDER BY E.EmpId;
```

Kết quả:

EmpId	Name	DeptName
1	Hung	HR
2	Lan	Finance
3	Minh	HR
4	Nga	IT

★ **Mẹo phỏng vấn:** INNER JOIN chỉ trả về những dòng khớp nhau ở cả hai bảng.

Q8. Tìm khách hàng chưa từng đặt đơn hàng nào (LEFT JOIN)

Các bảng:

Customers

CustomerId	Name
1	An
2	Trang
3	Khoa
4	Mai

Orders

OrderId	CustomerId	Amount
101	1	2500
102	1	3200
103	3	1500

Lời giải SQL:

```
SELECT C.CustomerId, C.Name
FROM Customers C
LEFT JOIN Orders O
ON C.CustomerId = O.CustomerId
WHERE O.OrderId IS NULL;
```

Kết quả:

CustomerId	Name
2	Trang
4	Mai

Khái niệm cốt lõi:

LEFT JOIN giữ lại toàn bộ dòng của bảng bên trái (Customers) dù không có dòng khớp bên phải.

💡 **Mẹo phỏng vấn:** Dùng LEFT JOIN + WHERE cột_bảng_phải IS NULL để tìm các bản ghi "thiếu" hoặc "không khớp".

Q9. Tìm đơn hàng không có khách hàng hợp lệ (RIGHT JOIN)

Customers

CustomerId	Name
1	An
2	Trang
3	Khoa

Orders

OrderId	CustomerId	Amount
101	1	2500
102	5	1800
103	2	3200
104	6	700

Lời giải SQL:

```
SELECT O.OrderId, O.CustomerId, O.Amount
FROM Customers C
RIGHT JOIN Orders O
ON C.CustomerId = O.CustomerId
WHERE C.CustomerId IS NULL;
```

Kết quả:

OrderId	CustomerId	Amount
102	5	1800
104	6	700

💡 **Mẹo phỏng vấn:** RIGHT JOIN giữ toàn bộ dòng của bảng bên phải (Orders) dù không có dòng khớp bên trái.

Khi nào dùng loại JOIN nào?

- LEFT JOIN → tìm dữ liệu thiếu ở bảng bên phải.
- RIGHT JOIN → tìm dữ liệu thiếu ở bảng bên trái.
- INNER JOIN → chỉ lấy dữ liệu khớp nhau.

Sơ đồ trực quan các loại JOIN

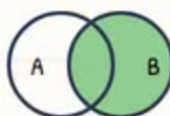
INNER JOIN



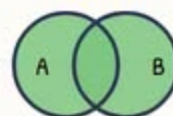
LEFT JOIN



RIGHT JOIN



FULL JOIN



Điểm cần nhớ

- ✓ Hiểu rõ quan hệ giữa các bảng trước.
- ✓ Chọn đúng loại JOIN theo yêu cầu.
- ✓ IS NULL là chìa khóa tìm dữ liệu thiếu.



Q10. Phòng ban có nhiều hơn 2 nhân viên (GROUP BY + HAVING)

Bảng: Employees

EmpId	Name	DeptId
1	Hung	10
2	Lan	10
3	Minh	10
4	Nga	20
5	Tuan	20
6	Mai	30
7	Khoa	30
8	Trang	30

Lời giải SQL:

```
SELECT DeptId, COUNT(*) AS EmpCount
FROM Employees
GROUP BY DeptId
HAVING COUNT(*) > 2
ORDER BY EmpCount DESC;
```

Kết quả:

DeptId	EmpCount
10	3
30	3

★ Mẹo phỏng vấn

HAVING đi cùng GROUP BY để lọc các nhóm dựa trên giá trị tổng hợp.

Điểm mấu chốt

- ✓ COUNT(*) đếm mọi dòng.
- ✓ GROUP BY gom các dòng lại.
- ✓ HAVING lọc theo nhóm.
- ✓ ORDER BY sắp xếp kết quả cuối.

Q11. Sản phẩm bán được hơn 100 lượt

Các bảng:

Products

ProductId	ProductName
1	Laptop
2	Điện thoại
3	Máy tính bảng

OrderItems

OrderItemId	ProductId	Quantity
1	1	50
2	1	70
3	2	30
4	3	120
5	2	50

Lời giải SQL:

```
SELECT P.ProductId, P.ProductName,
       SUM(O.Quantity) AS TotalSold
FROM Products P
JOIN OrderItems O
  ON P.ProductId = O.ProductId
GROUP BY P.ProductId, P.ProductName
HAVING SUM(O.Quantity) > 100
ORDER BY TotalSold DESC;
```

Kết quả:

ProductId	ProductName	TotalSold
1	Laptop	120
3	Máy tính bảng	120

Mẹo phỏng vấn 💡

Trong HAVING luôn phải dùng hàm tổng hợp (SUM, COUNT, AVG...).



Ta dùng SUM để tính tổng số lượng bán ra của mỗi sản phẩm, rồi dùng HAVING để lọc những sản phẩm có tổng bán > 100.

Q12. Khách hàng có tổng giá trị mua hàng lớn hơn 10.000

Customers

CustomerId	Name
1	An
2	Trang
3	Khoa

Orders

OrderId	CustomerId	Amount
101	1	6000
102	1	5000
103	2	3000
104	2	3000
105	3	15000

Lời giải SQL:

```
SELECT C.CustomerId, C.Name,
       SUM(O.Amount) AS TotalSpent
FROM Customers C
JOIN Orders O
  ON C.CustomerId = O.CustomerId
GROUP BY C.CustomerId, C.Name
HAVING SUM(O.Amount) > 10000
ORDER BY TotalSpent DESC;
```

Kết quả:

CustomerId	Name	TotalSpent
1	An	11000
3	Khoa	15000

Khái niệm cốt lõi

- ✓ JOIN kết nối dữ liệu.
- ✓ GROUP BY gom theo khách hàng.
- ✓ HAVING lọc theo tổng chi tiêu.

WHERE và HAVING

WHERE	HAVING
Lọc dòng trước khi gom nhóm	Lọc nhóm sau khi gom nhóm
Không dùng được hàm tổng hợp	Bắt buộc dùng hàm tổng hợp

Hàm tổng hợp

COUNT(*) → Tổng số dòng
COUNT(col) → Số dòng khác NULL
SUM(col) → Tổng giá trị
AVG(col) → Trung bình

Lỗi thường gặp

- ✗ Dùng WHERE thay vì HAVING.
- ✗ Quên mệnh đề GROUP BY.
- ✗ Chọn cột không có trong GROUP BY.



Q13. Tìm mức lương cao thứ 2 trong bảng Employees

Bảng: Employees

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	70000
4	Nga	50000
5	Tuan	40000

Lời giải SQL (Cách 1 — dùng truy vấn con)

```
SELECT MAX(Salary) AS SecondHighestSalary
FROM Employees
WHERE Salary < (SELECT MAX(Salary)
FROM Employees);
```

Lời giải SQL (Cách 2 — dùng ORDER BY + LIMIT)

```
SELECT DISTINCT Salary
FROM Employees
ORDER BY Salary DESC
LIMIT 1 OFFSET 1;
```

Kết quả:

SecondHighestSalary
60000

Mẹo phỏng vấn 💡

DISTINCT rất quan trọng vì nhiều nhân viên có thể cùng mức lương cao nhất.

Khái niệm cốt lõi

- ✓ Truy vấn con lấy giá trị lớn nhất.
- ✓ Sau đó lọc các giá trị nhỏ hơn nó.
- ✓ Cuối cùng lấy MAX của phần còn lại.

Q14. Nhân viên có lương cao hơn mức lương trung bình

Bảng: Employees

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	70000
4	Nga	50000
5	Tuan	40000

Lời giải SQL:

```
SELECT *
FROM Employees
WHERE Salary > (SELECT AVG(Salary)
FROM Employees);
```

★ Mẹo phỏng vấn

Truy vấn con trả về một giá trị duy nhất (lương trung bình) và được dùng ngay trong WHERE.

Kết quả:

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	70000

Khái niệm cốt lõi

- ✓ Truy vấn con trả về một giá trị vô hướng.
- ✓ Được dùng bên trong mệnh đề WHERE.
- ✓ Hữu ích khi so sánh với giá trị tổng hợp.

Q15. Tìm sản phẩm chưa từng được đặt mua

Các bảng:

Products

ProductId	ProductName
1	Laptop
2	Điện thoại
3	Máy tính bảng
4	Tai nghe

Orders

OrderId	ProductId
101	1
102	3
103	1

Lời giải SQL (dùng LEFT JOIN)

```
SELECT P.ProductId, P.ProductName
FROM Products P
LEFT JOIN Orders O
ON P.ProductId = O.ProductId
WHERE O.OrderId IS NULL;
```

💡 Mẹo phỏng vấn

LEFT JOIN + IS NULL là mẫu câu cực kỳ quan trọng để tìm dữ liệu bị thiếu.

Kết quả:

ProductId	ProductName
2	Điện thoại
4	Tai nghe

Khái niệm cốt lõi

- ✓ LEFT JOIN giữ mọi dòng của bảng trái.
- ✓ Dòng không khớp sẽ có NULL ở cột bảng phải.
- ✓ Ta lọc chúng bằng IS NULL.

Các loại truy vấn con

1. **Vô hướng**: trả về 1 giá trị.
2. **Nhiều dòng**: trả về nhiều dòng.
3. **Nhiều cột**: trả về nhiều cột.

Toán tử so sánh

- = Bằng
- <> Khác
- > < Lớn hơn / nhỏ hơn
- >= <= Lớn hơn / nhỏ hơn hoặc bằng

Hàm tổng hợp hay dùng

- COUNT(*) → Số dòng
- SUM() → Tổng
- AVG() → Trung bình
- MIN() → Nhỏ nhất

Lỗi thường gặp

- ✗ Quên DISTINCT trong truy vấn con.
- ✗ Dùng WHERE với hàm tổng hợp thay cho HAVING.
- ✗ Xử lý giá trị NULL chưa đúng.



Q16. Xếp hạng nhân viên theo lương (cao đến thấp)

Bảng: Employees

EmpId	Name	DeptId	Salary
1	Hung	10	70000
2	Lan	10	60000
3	Minh	10	60000
4	Nga	20	50000
5	Tuan	20	40000
6	Mai	30	40000

Lời giải SQL (dùng RANK)

```
SELECT EmpId, Name, DeptId, Salary,  
       RANK() OVER (ORDER BY Salary DESC) AS SalaryRank  
FROM Employees;
```

Kết quả:

EmpId	Name	Salary	SalaryRank
1	Hung	70000	1
2	Lan	60000	2
3	Minh	60000	2
4	Nga	50000	4
5	Tuan	40000	5
6	Mai	40000	5

Mẹo phỏng vấn

Window function rất hay được hỏi ở các vị trí từ SDE-2 trở lên.

Khái niệm cốt lõi

- ✓ RANK() cho cùng hạng với giá trị bằng nhau và bỏ qua hạng kế tiếp.
- ✓ Dùng ORDER BY bên trong OVER() để định nghĩa thứ tự.

Q17. Xếp hạng liên tiếp (DENSE_RANK) theo lương

Lời giải SQL (dùng DENSE_RANK)

```
SELECT EmpId, Name, Salary,  
       DENSE_RANK() OVER (ORDER BY Salary DESC)  
       AS DenseRank  
FROM Employees  
ORDER BY Salary DESC;
```

Kết quả:

EmpId	Name	Salary	DenseRank
1	Hung	70000	1
2	Lan	60000	2
3	Minh	60000	2
4	Nga	50000	3
5	Tuan	40000	4
6	Mai	40000	4

Khác biệt: RANK và DENSE_RANK

RANK()	DENSE_RANK()
1, 2, 2, 4, 5, 5	1, 2, 2, 3, 4, 4
(Bỏ qua hạng kế tiếp)	(Không có khoảng trống)

★ Khi nào dùng

Dùng DENSE_RANK khi bạn không muốn thứ hạng bị ngắt quãng.

Q18. Đánh số thứ tự nhân viên trong từng phòng ban (theo lương giảm dần)

Lời giải SQL (dùng ROW_NUMBER)

```
SELECT EmpId, Name, DeptId, Salary,  
       ROW_NUMBER() OVER (PARTITION BY DeptId  
                           ORDER BY Salary DESC) AS RowNum  
FROM Employees  
ORDER BY DeptId, RowNum;
```

Kết quả:

EmpId	Name	DeptId	Salary	RowNum
1	Hung	10	70000	1
2	Lan	10	60000	2
3	Minh	10	60000	3
4	Nga	20	50000	1
5	Tuan	20	40000	2
6	Mai	30	40000	1

Khái niệm cốt lõi

- ✓ PARTITION BY chia tập kết quả thành các nhóm.
- ✓ ROW_NUMBER đánh số duy nhất cho từng dòng trong mỗi nhóm.
- ✓ Rất hữu ích cho phân trang và bài toán top-N.

Mẹo phỏng vấn

ROW_NUMBER được dùng rất nhiều cho bảng xếp hạng, phân trang, khử trùng lặp và truy vấn top N.

Dùng hàm nào?



ROW_NUMBER() → số duy nhất cho mọi dòng.

RANK() → xếp hạng có khoảng trống.

DENSE_RANK() → xếp hạng liên tiếp.

Ví dụ nhanh (Lương)

Salary	RANK()	DENSE_RANK()	ROW_NUMBER()
70000	1	1	1
60000	2	2	2
60000	2	2	3

Tình huống hay gặp

- Lấy top N của mỗi nhóm (ROW_NUMBER).
- Bảng xếp hạng (RANK / DENSE_RANK).
- Khử bản ghi trùng lặp.

✗ Lỗi thường gặp

- Quên ORDER BY bên trong OVER().
- Không dùng PARTITION BY khi cần chia nhóm.
- Nhầm lẫn RANK với ROW_NUMBER.



Q19. Tìm các bản ghi bị trùng lặp trong bảng

Bảng: Employees

EmpId	Name	Email
1	Hung	hung@gmail.com
2	Lan	lan@gmail.com
3	Minh	minh@gmail.com
4	Hung	hung@gmail.com
5	Nga	nga@gmail.com
6	Lan	lan@gmail.com

Lời giải SQL:

```
SELECT Name, Email, COUNT(*) AS Cnt
FROM Employees
GROUP BY Name, Email
HAVING COUNT(*) > 1;
```

Kết quả:

Name	Email	Cnt
Hung	hung@gmail.com	2
Lan	lan@gmail.com	2

Mẹo phỏng vấn 💡

Dùng GROUP BY + HAVING để phát hiện trùng lặp. COUNT(*) > 1 nghĩa là bản ghi xuất hiện nhiều hơn một lần.

Khái niệm cốt lõi

- ✓ GROUP BY gom các giá trị giống nhau.
- ✓ HAVING lọc theo nhóm.

Q20. Xóa bản ghi trùng lặp (giữ lại một bản ghi)

Bảng: Employees

EmpId	Name	Email
1	Hung	hung@gmail.com
2	Lan	lan@gmail.com
3	Minh	minh@gmail.com
4	Hung	hung@gmail.com
5	Nga	nga@gmail.com
6	Lan	lan@gmail.com

Lời giải SQL (dùng CTE + ROW_NUMBER):

```
WITH CTE AS (
  SELECT EmpId, Name, Email,
         ROW_NUMBER() OVER (PARTITION BY Name, Email
                             ORDER BY EmpId) AS RowNum
  FROM Employees
)
DELETE FROM CTE
WHERE RowNum > 1;
```

Mẹo phỏng vấn 💡

ROW_NUMBER() gán số duy nhất cho các dòng trùng nhau. Ta xóa những dòng có RowNum > 1 và giữ lại dòng đầu tiên.

Giải thích:

- PARTITION BY Name, Email để xác định các dòng trùng.
- ORDER BY EmpId để giữ lại EmpId nhỏ nhất.
- Xóa các dòng có RowNum > 1 (các bản dư thừa).

Hãy cẩn thận ⚠

- Luôn thử nghiệm trên bản sao của bảng trước.
- Kiểm tra kỹ các cột trong PARTITION BY.

Q21. Tìm các nhân viên vào làm trong cùng một tháng

Bảng: Employees

EmpId	Name	HireDate
1	Hung	2024-01-15
2	Lan	2024-01-25
3	Minh	2024-02-05
4	Nga	2024-02-20
5	Tuan	2024-02-28

Lời giải SQL:

```
SELECT YEAR(HireDate) AS HireYear,
       MONTH(HireDate) AS HireMonth,
       COUNT(*) AS EmpCount
FROM Employees
GROUP BY YEAR(HireDate), MONTH(HireDate)
HAVING COUNT(*) > 1
ORDER BY HireYear, HireMonth;
```

Kết quả:

HireYear	HireMonth	EmpCount
2024	1	2
2024	2	3

Mẹo phỏng vấn 💡

Tách YEAR và MONTH rồi gom nhóm theo cả hai để tìm các tháng trùng nhau.

Tra nhanh Window Function

Hàm	Công dụng
ROW_NUMBER()	Số thứ tự duy nhất cho mỗi dòng
RANK()	Xếp hạng, có khoảng trống

Hàm tổng hợp

COUNT(*) → Số dòng
COUNT(col) → Số dòng khác NULL
SUM(col) → Tổng giá trị

GROUP BY và HAVING

- WHERE lọc dòng trước khi gom nhóm.
- HAVING lọc sau khi đã gom nhóm.

✗ Lỗi thường gặp

- Dùng WHERE với hàm tổng hợp.
- Quên mệnh đề GROUP BY.
- Xóa dữ liệu mà chưa sao lưu.



Q22. Tính thâm niên của nhân viên theo năm và tháng

Bảng: Employees

Lời giải SQL:

EmpId	Name	JoinDate
1	Hung	2022-01-15
2	Lan	2021-05-10
3	Minh	2023-02-20
4	Nga	2020-11-05
5	Tuan	2019-07-01

```
SELECT EmpId, Name, JoinDate,  
DATEDIFF(YEAR, JoinDate, GETDATE()) AS ExpYears,  
DATEDIFF(MONTH, JoinDate, GETDATE()) % 12 AS ExpMonths  
FROM Employees;
```

Kết quả:

EmpId	Name	JoinDate	ExpYears	ExpMonths
1	Hung	2022-01-15	2	5
2	Lan	2021-05-10	3	1
3	Minh	2023-02-20	1	3
4	Nga	2020-11-05	3	7
5	Tuan	2019-07-01	4	11

Mẹo phỏng vấn 💡

DATEDIFF trả về khoảng cách giữa hai mốc thời gian theo đơn vị bạn chỉ định. Dùng % 12 để lấy số tháng lẻ.

Khái niệm cốt lõi

- ✓ DATEDIFF(YEAR, ...) cho tổng số năm.
- ✓ DATEDIFF(MONTH, ...) % 12 cho số tháng còn lại.
- ✓ GETDATE() trả về ngày giờ hiện tại.

Q23. Tách tên miền từ địa chỉ email

Bảng: Users

Lời giải SQL (dùng SUBSTRING + CHARINDEX):

UserId	Email
1	hung@gmail.com
2	lan@outlook.com
3	minhfpt.com.vn
4	nga@company.org
5	tuan@yahoo.com

```
SELECT UserId, Email,  
SUBSTRING(Email,  
CHARINDEX('@', Email) + 1,  
LEN(Email) - CHARINDEX('@', Email)) AS Domain  
FROM Users;
```

Kết quả:

UserId	Email	Domain
1	hung@gmail.com	gmail.com
2	lan@outlook.com	outlook.com
3	minhfpt.com.vn	fpt.com.vn
4	nga@company.org	company.org
5	tuan@yahoo.com	yahoo.com

Cách khác (ANSI SQL)

```
SELECT UserId, Email,  
SUBSTRING(Email FROM POSITION('@'  
IN Email) + 1) AS Domain  
FROM Users;  
(Chạy được trên PostgreSQL / MySQL)
```

Mẹo phỏng vấn 💡

CHARINDEX('@', Email) trả về vị trí của ký tự @. Ta lấy toàn bộ phần phía sau @ để có tên miền.

Khái niệm cốt lõi

- ✓ CHARINDEX tìm vị trí chuỗi con.
- ✓ SUBSTRING cắt một phần chuỗi.
- ✓ LEN trả về độ dài chuỗi.

Q24. Tìm đơn hàng đặt trong 30 ngày gần nhất

Bảng: Orders

Lời giải SQL:

OrderId	CustomerId	OrderDate	Amount
101	1	2024-05-10	2500
102	2	2024-05-20	3200
103	3	2024-06-01	1500
104	1	2024-06-05	4100
105	2	2024-06-10	2800
106	3	2024-06-18	1900

```
SELECT *  
FROM Orders  
WHERE OrderDate >= DATEADD(DAY, -30, GETDATE())  
ORDER BY OrderDate DESC;
```

Kết quả (giả sử hôm nay là 2024-06-19):

OrderId	CustomerId	OrderDate	Amount
106	3	2024-06-18	1900
105	2	2024-06-10	2800
104	1	2024-06-05	4100
103	3	2024-06-01	1500

Mẹo phỏng vấn 💡

DATEADD(DAY, -30, GETDATE()) cho ra mốc ngày cách hôm nay 30 ngày để so sánh với OrderDate.

Khái niệm cốt lõi

- ✓ DATEADD cộng / trừ khoảng thời gian.
- ✓ GETDATE() cho ngày hiện tại.
- ✓ Rất hữu ích cho báo cáo và dashboard.

Tra nhanh hàm ngày tháng

GETDATE() → Ngày giờ hiện tại
DATEADD() → Cộng / trừ thời gian
DATEDIFF() → Khoảng cách 2 mốc

Tra nhanh hàm chuỗi

LEN() → Độ dài chuỗi
SUBSTRING() → Cắt chuỗi con
CHARINDEX() → Vị trí chuỗi con

Thực hành tốt

- ✓ Lọc theo khoảng ngày thay vì bọc cột trong hàm.
- ✓ Chú ý múi giờ khi so sánh thời gian.

✗ Lỗi thường gặp

- Dùng hàm lên cột đã đánh index.
- Quên định dạng ngày của hệ quản trị.



Q25. Tìm mức lương cao thứ 2 trong bảng Employees

Bảng: Employees

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	60000
4	Nga	50000
5	Tuan	40000
6	Mai	30000

Lời giải SQL (Cách 1 — DISTINCT + LIMIT/OFFSET)

```
SELECT Salary
FROM (
  SELECT DISTINCT Salary
  FROM Employees
  ORDER BY Salary DESC
  LIMIT 1 OFFSET 1 -- bỏ qua mức cao nhất
) AS t;
```

Kết quả:

SecondHighestSalary
60000

Mẹo phỏng vấn

Câu này kiểm tra cách bạn xử lý giá trị trùng và tư duy về xếp hạng.

Khái niệm cốt lõi

- ✓ DISTINCT loại bỏ giá trị trùng.
- ✓ ORDER BY DESC sắp xếp giảm dần.
- ✓ OFFSET 1 bỏ qua bản ghi đầu tiên.

Lời giải SQL (Cách 2 — dùng truy vấn con)

```
SELECT MAX(Salary) AS SecondHighestSalary
FROM Employees
WHERE Salary < (SELECT MAX(Salary) FROM Employees);
```

Q26. Tìm 3 nhân viên có lương cao nhất

Bảng: Employees

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	60000
4	Nga	50000
5	Tuan	40000
6	Mai	30000

Lời giải SQL (dùng LIMIT)

```
SELECT EmpId, Name, Salary
FROM Employees
ORDER BY Salary DESC
LIMIT 3;
```

Kết quả:

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	60000

Khái niệm cốt lõi

- ✓ ORDER BY luôn đứng trước LIMIT.
- ✓ LIMIT N lấy N dòng đầu tiên.
- ✓ Rất hợp cho báo cáo top N.

Cách khác (ANSI SQL — FETCH FIRST)

```
SELECT EmpId, Name, Salary
FROM Employees
ORDER BY Salary DESC
FETCH FIRST 3 ROWS ONLY;
(Chạy trên PostgreSQL, DB2, Oracle 12c+ ...)
```

Q27. Xoay dữ liệu (Pivot): tổng doanh thu mỗi sản phẩm theo tháng

Bảng: Sales

SaleId	Product	SaleDate	Amount
1	Laptop	2024-01-10	1000
2	Laptop	2024-02-12	1500
3	Điện thoại	2024-01-15	800
4	Điện thoại	2024-02-20	900
5	Máy tính bảng	2024-01-18	600
6	Máy tính bảng	2024-02-25	700

Lời giải SQL (dùng tổng hợp có điều kiện)

```
SELECT
  Product,
  SUM(CASE WHEN MONTH(SaleDate) = 1 THEN Amount ELSE 0 END) AS T1_Sales,
  SUM(CASE WHEN MONTH(SaleDate) = 2 THEN Amount ELSE 0 END) AS T2_Sales
FROM Sales
GROUP BY Product
ORDER BY Product;
```

Kết quả:

Product	T1_Sales	T2_Sales
Laptop	1000	1500
Điện thoại	800	900
Máy tính bảng	600	700

Mẹo phỏng vấn

Câu hỏi pivot cho thấy bạn có biết biến dòng thành cột để phục vụ phân tích hay không.

Giải thích

- MONTH() dùng để lấy số tháng.
- SUM(CASE WHEN ...) tính tổng cho từng tháng.
- Đây là kỹ thuật PIVOT cơ bản, chạy được trên hầu hết hệ quản trị.

Hàm tổng hợp hay dùng

COUNT(*) → Số dòng
SUM() → Tổng giá trị
AVG() → Trung bình

Hàm ngày / giờ

NOW() / GETDATE() → Hiện tại
MONTH() → Lấy tháng
YEAR() → Lấy năm

Hàm xử lý chuỗi

LENGTH() → Độ dài
UPPER() → Chữ hoa
TRIM() → Cắt khoảng trắng

✗ Lỗi thường gặp

- Quên ORDER BY bên trong OVER().
- Dùng LIMIT mà không sắp xếp trước.
- Quên ELSE 0 trong CASE WHEN.



Q28. Tìm khách hàng đã đặt nhiều hơn 2 đơn hàng

Bảng: Orders

OrderId	CustomerId	OrderDate
101	1	2024-05-10
102	2	2024-05-12
103	2	2024-05-11
104	1	2024-06-01
105	1	2024-06-05

Lời giải SQL (GROUP BY + HAVING)

```
SELECT CustomerId, COUNT(*) AS OrderCount
FROM Orders
GROUP BY CustomerId
HAVING COUNT(*) > 2;
```

Khái niệm cốt lõi

- ✓ GROUP BY gom các đơn hàng theo CustomerId.
- ✓ HAVING lọc các nhóm có số đơn > 2.

Kết quả:

CustomerId	OrderCount
1	3

Cách khác (truy vấn con)

```
SELECT CustomerId
FROM Orders
GROUP BY CustomerId
HAVING COUNT(*) > 2;
```

Mẹo phỏng vấn

HAVING đi cùng GROUP BY để lọc kết quả đã tổng hợp. WHERE không dùng được với hàm tổng hợp.

Q29. Lấy mức lương cao thứ 2 bằng LIMIT + OFFSET

Bảng: Employees

EmpId	Name	Salary
1	Hung	70000
2	Lan	60000
3	Minh	60000
4	Nga	50000
5	Tuan	40000

Lời giải SQL (LIMIT + OFFSET)

```
SELECT DISTINCT Salary
FROM Employees
ORDER BY Salary DESC
LIMIT 1 OFFSET 1; -- bỏ qua mức cao nhất
```

Khái niệm cốt lõi

- ✓ DISTINCT loại bỏ các mức lương trùng nhau.
- ✓ OFFSET 1 bỏ qua mức lương cao nhất.

Kết quả:

SecondHighestSalary
60000

Cách khác (truy vấn con)

```
SELECT MAX(Salary)
FROM Employees
WHERE Salary < (SELECT MAX(Salary)
FROM Employees);
```

Mẹo phỏng vấn

DISTINCT là bắt buộc vì nhiều nhân viên có thể cùng mức lương. LIMIT + OFFSET rất gọn nếu hệ quản trị hỗ trợ.

Q30. Hiển thị lương của từng nhân viên kèm lương trung bình phòng ban

Bảng: Employees

EmpId	Name	DeptId	Salary
1	Hung	10	70000
2	Lan	10	60000
3	Minh	20	60000
4	Nga	20	50000
5	Tuan	30	40000

Lời giải SQL (window function AVG() OVER)

```
SELECT EmpId, Name, DeptId, Salary,
AVG(Salary) OVER (PARTITION BY DeptId)
AS AvgDeptSalary
FROM Employees;
```

Khái niệm cốt lõi

- ✓ Window function tính trung bình theo từng phòng ban mà không gộp mất các dòng.
- ✓ PARTITION BY xác định nhóm (phòng ban).

Kết quả (ví dụ):

EmpId	Name	DeptId	Salary	AvgDeptSalary
1	Hung	10	70000	65000.00
2	Lan	10	60000	65000.00
3	Minh	20	60000	55000.00
4	Nga	20	50000	55000.00
5	Tuan	30	40000	40000.00

Mẹo phỏng vấn

Window function rất mạnh và thường được hỏi ở vị trí SDE-2 trở lên: chúng cho phép tính toán trên một tập dòng liên quan mà vẫn giữ nguyên chi tiết từng dòng.

Thứ tự thực thi mệnh đề

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY
7. LIMIT / OFFSET

Điểm quan trọng cần nhớ

- WHERE lọc dòng trước khi gom nhóm.
- HAVING lọc nhóm sau khi gom nhóm.
- Đánh index cho cột dùng trong WHERE, JOIN, GROUP BY, ORDER BY.
- Tránh SELECT * trong môi trường production.
- Luôn kiểm tra trường hợp biên (NULL, trùng lặp, bảng rỗng).

Hàm tổng hợp hay dùng

COUNT(*) → Số dòng
COUNT(col) → Số dòng khác NULL
SUM(col) → Tổng giá trị
AVG(col) → Trung bình
MIN(col) → Nhỏ nhất
MAX(col) → Lớn nhất

Luyện tập thêm

→ LeetCode SQL 50
→ HackerRank SQL
→ Mode Analytics SQL
→ StrataScratch 😊