

SQL Questions and Answers

SQL Questions and Answers

Part 1/14: Basic Select Operations

emp_no	first_name	last_name	salary	birth_date	gender	joining_date	department
1001	James	Bond	45300	1998-01-19	M	2020-11-29	Marketing
1002	Robert	Lee	69410	1998-09-02	M	2019-12-26	IT
1003	Mary	Wood	39518	1997-11-07	F	2019-09-12	Business Development
1004	John	Cole	36264	1999-02-01	M	2021-04-14	HR
1005	Jennifer	Gill	48941	1996-06-05	F	2019-06-28	Marketing
1006	Michael	Watts	23800	1999-04-16	M	2022-06-09	HR
1007	David	Mills	39400	1998-04-26	M	2021-10-17	IT
1008	William	Hill	26392	1999-10-21	M	2019-09-30	Business Development
1009	Linda	Potter	55231	1998-01-24	F	2021-08-07	Finance
1010	Richard	Martin	25600	1996-08-07	M	2021-03-23	IT
1011	Susan	Hall	54152	1999-12-27	F	2020-08-07	Finance
1012	Joseph	Turner	60556	1997-05-13	M	2019-09-02	Marketing
1013	Mary	Gill	58246	1999-01-21	F	2021-04-30	Business Development
1014	Thomas	Ryan	23676	1996-10-12	M	2022-03-02	IT
1015	Charles	Williams	57226	2000-01-29	M	2019-11-05	

Q1. Write a query to display unique salary of employee table.

A:

```
SELECT DISTINCT salary FROM employees;
```

Q1. Write a query to display unique salary of employee table.

A:

```
SELECT DISTINCT salary FROM employees;
```

Q2. Write a query to display emp_no and the annual salary with the column name CTC from the employee table.

A:

```
SELECT emp_no, salary*12 AS CTC FROM employees;
```

Q1. Write a query to display unique salary of employee table.

A:

```
SELECT DISTINCT salary FROM employees;
```

Q2. Write a query to display emp_no and the annual salary with the column name CTC from the employee table.

A:

```
SELECT emp_no, salary*12 AS CTC FROM employees;
```

Q3. Write a query to display unique combination of salary and department of the employee table

A:

```
SELECT DISTINCT salary, department FROM employees;
```

SQL Questions and Answers

Part 2/14: **WHERE** clause

Q1. Write a query to display all the records of the employee whose joining_date is after 20 March 2022.

A:

```
SELECT * FROM employees WHERE joining_date > '2022-03-20';
```

Q2. Write a query to display emp_no, first_name, last_name, department of the employee whose department is 'Finance'.

A:

```
SELECT emp_no,first_name,last_name,department FROM employees WHERE  
department = 'Finance';
```

Q3. Write a query to display emp_no, salary, department of the employee whose salary is below 35000.

A:

```
SELECT emp_no, salary, department FROM employees WHERE salary < 35000;
```


Q4. Write a query to display all the records of the employee whose birth_date is after or equal to 24 January 1998.

A:

```
SELECT * FROM employees WHERE birth_date >= '1998-01-24';
```

Q5. Write a query to display emp_no, first_name, department of the employee whose department is other than 'IT'.

A:

```
SELECT emp_no, first_name, department FROM employees  
WHERE department != 'IT';
```

Q6. Write a query to display all the records of the employee whose emp_no is 1010 or below 1010.

A:

```
SELECT * FROM employees WHERE emp_no <= 1010;
```

Q7. Write a query to display unique first_name of the employee whose gender is Male.

A:

```
SELECT DISTINCT first_name FROM employees WHERE gender='M';
```

Q8. Write a query to display emp_no, first_name , salary and add 10000 to the salary of the employee whose salary is below 25000 and display final salary as column name 'final_salary'.

A:

```
SELECT emp_no,first_name,salary, salary+10000 AS final_salary FROM  
employees WHERE salary < 25000;
```

SQL Questions and Answers

Part 3/14: AND, OR and NOT Operators

Q1. Write a query to display all the records of the employee who either belong to 'HR' department or 'Marketing' department.

A:

```
SELECT * FROM employees WHERE department = 'HR' OR department =  
'Marketing';
```

Q2. Write a query to get the emp_no, first_name and salary of the employee whose emp_no is more than 1010 and salary is also more than 40000.

A:

```
SELECT emp_no, first_name, salary FROM employees WHERE emp_no > 1010  
AND salary > 40000;
```

Q3. Write a query to display all the records of the employee whose gender is Female and they don't belong to 'IT' department.

A:

```
SELECT * FROM employees WHERE gender = 'F' AND NOT department = 'IT';
```

Q4. Write a query to get emp_no, department, salary and joining_date of the employee who either joined the company after 1 April 2020 or who has salary below 33000.

A:

```
SELECT emp_no, department, salary, joining_date FROM employees WHERE  
joining_date > '2020-04-01' OR salary < 33000;
```

Q5. Write a query to get all the records of the employee whose gender is Male and their department should either be 'Business Development' or 'Marketing' or 'Finance'.

A:

```
SELECT * FROM employees WHERE gender = 'M' AND (department ='Business  
Development' OR department ='Marketing' OR department='Finance');
```

Q6. Write a query to get emp_no, first_name and birth_date of the employee whose emp_no is below 1020 and birth_date should not be more than 9 March 1999.

A:

```
SELECT emp_no, first_name, birth_date FROM employees WHERE emp_no < 1020  
AND birth_date < '1999-03-09';
```

SQL Questions and Answers

Part 4/14: BETWEEN and IN Operators

Q1. Write a query to get all the details of the employee who joined the company between 1st March 2019 to 1st April 2021.

A:

```
SELECT * FROM employees WHERE joining_date BETWEEN '2019-03- 01' AND '2021-04-01';
```

Q2. Write a query to get first_name, last_name and salary of the employee whose last_name is Gill, Hill, Mills or Hall.

A:

```
SELECT first_name, last_name, salary FROM employees WHERE last_name IN ('Gill','Hill','Mills','Hall');
```

Q3. Write a query to display all the details of the employee whose salary is not between 40000 to 50000.

A:

```
SELECT * FROM employees WHERE salary NOT BETWEEN 40000 AND 50000;
```

Q4. Write a query to get emp_no, first_name, salary and department of the employee whose department is not Marketing, HR and Finance. And the column name for department will be 'other_department'.

A:

```
SELECT emp_no, first_name, salary, department AS other_department FROM  
employees WHERE department NOT IN ('Marketing','HR','Finance');
```

Q5. Write a query to get all the details of the employee whose emp_no is more than 1020 and the first name of the employee is David, Mark, Joe.

A:

```
SELECT * FROM employees WHERE emp_no > 1020 AND first_name IN  
('David','Mark','Joe');
```

Q6. Write a query to get all the details of the employee whose gender is female and birth_date is between 9 July 1997 to 9 July 1998.

A:

```
SELECT * FROM employees WHERE gender = 'F' AND birth_date BETWEEN  
'1997-07-09' AND '1998-07-09';
```


SQL Questions and Answers

Part 5/14: LIKE & Wildcards

Q1. Select all employees whose first name ends with 'y'.

A:

```
SELECT * FROM employees WHERE first_name LIKE '%y';
```

Q2. Select all employees whose date of birth is in April.

A:

```
SELECT * FROM employees WHERE birth_date LIKE '%-04-%';
```

Q3. Select emp_no , first name and department of the employees whose department does not contains the word 'ing'

A:

```
SELECT emp_no, first_name, department FROM employees WHERE  
department NOT LIKE '%ing% ';
```

SQL Questions and Answers

Part 6/14: IS NULL- IS NOT NULL

Q1. Display all the details of the employees whose emp_no is more than 1020 and salary is not available.

A:

```
SELECT * FROM employees WHERE emp_no > 1020 AND salary IS NULL;
```

Q2. Retrieve those employees whose last_name is null.

A:

```
SELECT * FROM employees WHERE last_name IS NULL;
```

Q3. Retrieve emp_no, first_name, salary and department of those employees who works in any department.

A:

```
SELECT emp_no, first_name, salary, department FROM employees WHERE  
department IS NOT NULL;
```

SQL Questions and Answers

Part 7/14: ORDER BY Clause

Q1. Write a SQL query to retrieve all the records with joining_date sorted in descending order.

A:

```
SELECT * FROM employees ORDER BY joining_date DESC;
```

Q2. Write an SQL query to retrieve all records from employee table and sort them by last name in ascending order.

A:

```
SELECT * FROM employees ORDER BY last_name ASC;
```

Q3. Retrieve emp_no, first_name, gender and salary from the employee table whose salary is not NULL and sort them by gender in ascending order, then by first name in ascending order.

A:

```
SELECT emp_no, first_name, gender, salary FROM employees WHERE salary IS NOT NULL ORDER BY gender , first_name;
```

SQL Questions and Answers

Part 8/14: LIMIT and OFFSET

Q1. Return all rows from the "employee" table, but only show the first 10 rows.

A:

```
SELECT * FROM employees LIMIT 10;
```

Q2. Return the next 10 rows from the " employee " table (i.e. rows with emp_no 1011-1020):

A:

```
SELECT * FROM employees LIMIT 10 OFFSET 10
```

Q3. Fetch emp_no, first name ,last name , joining date of 6 employees who recently joined the company

A:

```
SELECT emp_no, first_name, last_name, joining_date  
FROM employees  
ORDER BY joining_date DESC  
LIMIT 6;
```


Q4. Retrieve all the details of the 2 youngest employees from the “employee” table

A:

```
SELECT * FROM employees  
ORDER BY birth_date DESC  
LIMIT 2;
```

Q5. Retrieve all the details of the 3 eldest male employees from the “employee” table

A:

```
SELECT * FROM employees  
WHERE gender='M'  
ORDER BY birth_date  
LIMIT 3;
```

SQL Questions and Answers

Part 9/14: Aggregate-Functions

Q1. Find the number of male employee as total_male from "employee" table.

A:

```
SELECT COUNT(*) AS total_male FROM employees WHERE gender='M';
```

Q2. Write a SQL query to find the number of employees whose first name starts with the letter 'J' from "employee" table.

A:

```
SELECT COUNT(*) FROM employees WHERE first_name LIKE 'j%';
```

Q3. Find the total salary of employees whose employee numbers (emp_no) are between 1010 and 1020 from "employee" table.

A:

```
SELECT SUM(salary) FROM employees  
WHERE emp_no  
BETWEEN 1010 AND 1020;
```

Q4. Write a SQL query to calculate the total annual salary of all employees as annual_salary in the HR department from "employee" table.

A:

```
SELECT SUM(salary*12) AS annual_salary FROM employees  
WHERE department = 'HR';
```

Q5. Find the average salary of all employees as avg_salary from "employee" table.

A:

```
SELECT AVG(salary) AS avg_salary FROM employees ;
```

Q6. Find the average salary of employees working in the IT department from "employee" table.

A:

```
SELECT AVG(salary) FROM employees WHERE department='IT';
```

Q7. Write a SQL query to find the earliest joining date as min_joining_date among all employees in the "employee" table.

A:

```
SELECT MIN(joining_date) AS min_joining_date FROM employees ;
```

Q8. Find the minimum salary of employees working in the Finance department as min_salary from "employee" table.

A:

```
SELECT MIN(salary) AS min_salary FROM employees WHERE  
department='Finance';
```

Q9. Find the maximum birth date among all employees from "employee" table as max_dob.

A:

```
SELECT MAX(birth_date) AS max_dob FROM employees;
```

SQL Questions and Answers

Part 10/14: GROUP BY CLAUSE

Q1. Find the total salary expenditure categorized by gender from employee table.

A:

```
SELECT gender, SUM(salary) FROM employees GROUP BY gender;
```

Q2. Retrieve the average salary for each department as avg_salary from the employee table and display the results ascending order of the average salary.

A:

```
SELECT department, AVG(salary) AS avg_salary FROM employees  
GROUP BY department  
ORDER BY avg_salary;
```

Q3. Find the minimum salary in each department as min_salary whose department is not null from the employee table?

A:

```
SELECT department, MIN(salary) AS min_salary FROM employees  
WHERE department IS NOT NULL  
GROUP BY department;
```

Q4. Find the average salary as average_salary and maximum salary as max_salary for each department from the employee table.

A:

```
SELECT department, AVG(salary) AS average_salary, MAX(salary) AS max_salary  
FROM employees  
GROUP BY department;
```


Q5. Determine the number of employees as total_employees in each department, sorted in descending order of the count from the employee table.

A:

```
SELECT department, COUNT(*) AS total_employees FROM employees  
GROUP BY department  
ORDER BY total_employees DESC;
```

SQL Questions and Answers

Part 11/14: HAVING CLAUSE

Q1. Write a SQL query to count the number of duplicate first name as duplicate_count in the employee table.

A:

```
SELECT first_name,COUNT(*) AS duplicate_count  
FROM employees  
GROUP BY first_name  
HAVING COUNT(*) > 1;
```

Q2. Find the departments with more than 5 employees as total_employees from the employee table.

A:

```
SELECT department,COUNT(*) AS total_employees  
FROM employees  
GROUP BY department  
HAVING COUNT(*) > 5;
```

Q3. Determine the department with an average salary greater than 50000 as average_salary from the employee table.

A:

```
SELECT department, AVG(salary) AS average_salary  
FROM employees  
GROUP BY department  
HAVING AVG(salary) > 50000;
```

Q4. Determine the department with at least 3 employees earning a salary of 30000 or more total_employees from as the employee table.

A:

```
SELECT department, COUNT(*) AS total_employees  
FROM employees  
WHERE salary >= 30000  
GROUP BY department  
HAVING COUNT(*) >= 3;
```

Q5. Find the departments with a maximum salary greater than 60000 as max_salary from the employee table.

A:

```
SELECT department, MAX(salary) AS max_salary  
FROM employee  
GROUP BY department  
HAVING MAX(salary) > 60000;
```

SQL Questions and Answers

Part 12/14: SET Operators

product_jan

prod_id	prod_name	price	quantity	expiry_date
1	Chocolate	10	15	2024-05-10
2	Biscuits	20	10	2025-10-30
3	Noodles	60	5	2025-11-01
4	Peanuts	15	20	2024-06-06
5	Lays	25	30	2025-05-17
6	Oats	40	25	2025-02-14
7	Almonds	150	6	2026-01-18
8	Pickle	80	3	2024-12-25

product_feb

prod_id	prod_name	price	quantity	expiry_date
1	Cold Drink	90	10	2024-08-21
2	Milk	30	50	2023-06-20
3	Noodles	60	5	2025-11-01
4	Oil	65	18	2024-02-28
5	Lays	25	30	2025-05-17
6	Oats	40	25	2025-02-14

Q1. Write an SQL query to retrieve all the products from both tables where the quantity sold is greater than 5 in January and February.

A:

```
SELECT * FROM product_jan WHERE quantity > 5
```

UNION

```
SELECT * FROM product_feb WHERE quantity > 5;
```

Q2. Write an SQL query to retrieve the products that were sold in both January and February.

A:

```
SELECT * FROM product_jan  
INTERSECT  
SELECT * FROM product_feb;
```

Q4. Write an SQL query to find the common product names that were sold in both January and February, along with their respective quantities, prices, and total sales.

A:

```
SELECT prod_name, quantity, price, quantity*price AS total FROM product_jan  
INTERSECT  
SELECT prod_name,quantity,price, quantity*price AS total FROM product_feb;
```

Q4. Write an SQL query to retrieve the products that were sold in January for price less than 30 but were not sold in February.

A:

```
SELECT * FROM product_jan WHERE price<30
```

EXCEPT

```
SELECT * FROM product_feb;
```

Q5. Write an SQL query to retrieve the details of products sold in either January or February, with a price between 10 and 50.

A:

SELECT * FROM product_jan WHERE price BETWEEN 10 AND 50

UNION

SELECT * FROM product_feb WHERE price BETWEEN 10 AND 50;

SQL Questions and Answers

Part 13/14: JOINS

vendors

vend_id	vend_name	vend_address	vend_city	vend_state	vend_zip	vend_country
BRS01	Bears R Us	123 Main Street	Bear Town	MI	44444	USA
BRE02	Bear Emporium	500 Park Street	Anytown	OH	44333	USA
DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
FRB01	Furball Inc.	1000 5th Avenue	New York	NY	11111	USA
FNG01	Fun and Games	42 Galaxy Road	London		N16 6PS	England
JTS01	Jouets et ours	1 Rue Amusement	Paris		45678	France

products

prod_id	vend_id	prod_name	prod_price	prod_desc
BR01	BRS01	8 inch teddy bear	5.99	8 inch teddy bear, comes with cap and jacket
BR02		12 inch teddy bear	8.99	12 inch teddy bear, comes with cap and jacket
BR03	BRS01	18 inch teddy bear	11.99	18 inch teddy bear, comes with cap and jacket
BNBG01	DLL01	Fish bean bag toy	3.49	Fish bean bag toy, complete with bean bag worms with which to feed it
BNBG02	DLL01	Bird bean bag toy	3.49	Bird bean bag toy, eggs are not included
BNBG03		Rabbit bean bag toy	3.49	Rabbit bean bag toy, comes with bean bag carrots
RGAN01	DLL01	Raggedy Ann	4.99	18 inch Raggedy Ann doll
RYL01	FNG01	King doll	9.49	12 inch king doll with royal garments and crown
RYL02	FNG01	Queen doll	9.49	12 inch queen doll with royal garments and crown

Q1. Write SQL query to retrieve the product ID, product name, and vendor name from the "Products" table and "Vendors" table, sorted in ascending order by product ID?

A:

```
SELECT prod_id,prod_name,vend_name  
FROM Products INNER JOIN vendors  
ON  
products.vend_id = vendors.vend_id  
ORDER BY prod_id;
```

Q2. Write SQL query to retrieve the product ID, product name, vendor name, and vendor country for products sold by vendors in the USA, sorted by product ID?

A:

```
SELECT prod_id,prod_name,vend_name ,vend_country  
FROM Products INNER JOIN vendors  
ON  
products.vend_id = vendors.vend_id  
WHERE vend_country='USA'  
ORDER BY prod_id;
```

Q3. Write SQL query to retrieve the product ID, product name, vendor name, and product price from the "Products" table, joined with the "Vendors" table, where the product price is less than 5, and the results are sorted by the product ID? (Use Natural Join)

A:

```
SELECT prod_id,prod_name,vend_name ,prod_price  
FROM Products NATURAL JOIN vendors  
WHERE prod_price < 5  
ORDER BY prod_id;
```

Q4. Write SQL query to retrieve the product ID, vendor ID, and vendor city for all products, sorted by the vendor ID?

A:

```
SELECT prod_id,vendors.vend_id ,vend_city  
FROM Products INNER JOIN vendors  
ON  
products.vend_id = vendors.vend_id  
ORDER BY vendors.vend_id;
```

Q5. Write SQL query to retrieve vendor name and calculates the total amount of products sold by each vendor as “total_amount”. Sorts the results in ascending order based on the total amount?

A:

```
SELECT vend_name,SUM(prod_price) AS total_amount  
FROM Products INNER JOIN vendors  
ON  
products.vend_id = vendors.vend_id  
GROUP BY vend_name  
ORDER BY total_amount;
```

Q6. Show all employees whose joining date is same?

A:

```
SELECT s1.name, s1.DateTime FROM  
employees s1 INNER JOIN employees s2  
ON  
s1.DOJ = s2.DOJ  
AND  
s1.empld <> s2.empld;
```

SQL Questions and Answers

Part 14/14: SUB QUERIES

Q1. Write a query to display highest salary from all employee.

select max(salary) from employee;

Q2. Write a query to display employee name who is taking highest salary.

select name from employee where salary = (select max(salary) from employee);

Q3. Write a query to display 2nd highest salary

select max(salary) from employees where salary not in (select max(salary) from employees);

OR

select max(salary) from employee where salary <>(select max(salary) from employee);

Q4. Write a query to display employee name who is taking 2nd highest salary.

A:

select name from employee where salary = (select max(salary) from employee where salary <>(select max(salary) from employee));

Q5. Write a query to display highest salary in each department

A:

select department,max(salary) from employee group by department;

Q6. Write a query to display name of employee who has highest salary in each department

A:

select name from employee where salary in (select max(salary) from employee group by department);

Q7. Write a query to to display all department names where number of employees are less than 3

A:

```
select department from employee group by department having count(*)  
<3;
```

Q8. Write a query to to display all employee names whose working in a department in which number of employees are less than 3

A:

```
select name from employee where department in (select department  
from employee group by department having count(*) <3);
```

Q9. How to find Nth highest salary from a table?

4th highest salary:

A:

```
SELECT name, salary FROM Employee1 e1 WHERE 4-1 =  
(  
    SELECT COUNT(DISTINCT salary) FROM Employee1 e2  
    WHERE e2.salary > e1.salary  
);
```

Explanation :

The distinct keyword is there to deal with duplicate salaries in the table. In order to find the Nth highest salary, we are only considering unique salaries. The highest salary means no salary is higher than it, the Second highest means only one salary is higher than it, 3rd highest means two salaries are higher than it, similarly Nth highest salary means N-1 salaries are higher than it.

Q
A: