

Job Exam SQL Practice Guide: SELECT Queries (Beginner to Advanced)

Section 1: Sample Tables and Data

Table: employees

id	name	salary	department_id	hire_date	manager_id
1	Ali	50000	1	2020-01-01	NULL
2	Rafi	60000	1	2021-02-15	1
3	Mina	45000	2	2019-06-20	1
4	Sumi	70000	3	2022-03-10	2

Table: departments

id	name	location
1	HR	Dhaka
2	IT	Chittagong
3	Sales	Dhaka

Table: students

id	name	age	course_id
1	Arafat	20	1
2	Kabir	22	2
3	Nira	18	NULL

Table: courses

id	name
1	Math
2	Science

Section 2: 50 SQL SELECT Queries

1. `SELECT * FROM employees;`
2. `SELECT name, age FROM students;`
3. `SELECT DISTINCT department_id FROM employees;`
4. `SELECT name FROM students WHERE age > 20;`

Job Exam SQL Practice Guide: SELECT Queries (Beginner to Advanced)

5. `SELECT name FROM students WHERE age BETWEEN 18 AND 25;`
6. `SELECT * FROM employees WHERE department_id = 3;`
7. `SELECT name FROM employees WHERE name LIKE 'A%';`
8. `SELECT name FROM employees WHERE name LIKE '%an%';`
9. `SELECT name, salary FROM employees ORDER BY salary DESC;`
10. `SELECT name FROM employees LIMIT 5;`
11. `SELECT COUNT(*) FROM employees;`
12. `SELECT AVG(salary) FROM employees;`
13. `SELECT MIN(salary) FROM employees;`
14. `SELECT MAX(salary) FROM employees;`
15. `SELECT department_id, COUNT(*) FROM employees GROUP BY department_id;`
16. `SELECT department_id, AVG(salary) FROM employees GROUP BY department_id;`
17. `SELECT department_id, COUNT(*) FROM employees GROUP BY department_id HAVING COUNT(*) > 1;`
18. `SELECT * FROM students WHERE age IN (18, 21, 25);`
19. `SELECT name FROM students WHERE NOT age = 20;`
20. `SELECT * FROM employees WHERE hire_date > '2020-01-01';`
21. `SELECT e.name, d.name FROM employees e JOIN departments d ON e.department_id = d.id;`
22. `SELECT s.name, c.name FROM students s LEFT JOIN courses c ON s.course_id = c.id;`
23. `SELECT orders.id, customers.name FROM orders RIGHT JOIN customers ON orders.customer_id = customers.id;`
24. `SELECT name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);`
25. `SELECT name FROM employees WHERE department_id IN (SELECT id FROM departments WHERE location = 'Dhaka');`
26. `SELECT name FROM employees WHERE EXISTS (SELECT 1 FROM departments WHERE employees.department_id = departments.id AND departments.name = 'IT');`
27. `SELECT name FROM employees WHERE salary IS NULL;`

Job Exam SQL Practice Guide: SELECT Queries (Beginner to Advanced)

28. `SELECT name FROM employees WHERE salary IS NOT NULL;`
29. `SELECT UPPER(name) FROM employees;`
30. `SELECT LENGTH(name) FROM students;`
31. `SELECT e.name, e.salary, d.name AS department FROM employees e INNER JOIN departments d ON e.department_id = d.id WHERE e.salary = (SELECT MAX(salary) FROM employees WHERE department_id = d.id);`
32. `SELECT product_id, SUM(quantity) FROM order_items GROUP BY product_id ORDER BY SUM(quantity) DESC LIMIT 5;`
33. `SELECT department_id, COUNT(*) FROM employees GROUP BY department_id HAVING COUNT(*) > 10;`
34. `SELECT s.name, COUNT(e.id) FROM students s LEFT JOIN enrollments e ON s.id = e.student_id GROUP BY s.name;`
35. `SELECT c.name, AVG(g.grade) FROM courses c JOIN grades g ON c.id = g.course_id GROUP BY c.name;`
36. `SELECT name FROM students WHERE id NOT IN (SELECT student_id FROM enrollments);`
37. `SELECT name, salary, RANK() OVER (ORDER BY salary DESC) AS salary_rank FROM employees;`
38. `SELECT department_id, salary, DENSE_RANK() OVER (PARTITION BY department_id ORDER BY salary DESC) AS dept_rank FROM employees;`
39. `SELECT customer_id, SUM(amount) FROM payments GROUP BY customer_id ORDER BY SUM(amount) DESC LIMIT 3;`
40. `SELECT e.name, d.name AS department, l.city FROM employees e JOIN departments d ON e.department_id = d.id JOIN locations l ON d.location_id = l.id;`
41. `SELECT name, hire_date FROM employees WHERE hire_date BETWEEN '2020-01-01' AND '2023-01-01';`
42. `SELECT customer_id FROM orders GROUP BY customer_id HAVING COUNT(*) = (SELECT MAX(order_count) FROM (SELECT customer_id, COUNT(*) AS order_count FROM orders GROUP BY customer_id) AS sub);`
43. `SELECT department_id, name, salary FROM (SELECT *, ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY salary DESC) AS rn FROM employees) AS ranked WHERE rn = 1;`
44. `SELECT name, salary FROM employees WHERE salary > ALL (SELECT salary FROM employees WHERE department_id = 2);`
45. `SELECT name FROM employees WHERE salary > (SELECT salary FROM employees WHERE name =`

Job Exam SQL Practice Guide: SELECT Queries (Beginner to Advanced)

```
'Sabbir');
```

```
46. SELECT e1.name, e2.name AS manager_name FROM employees e1 JOIN employees e2 ON  
e1.manager_id = e2.id;
```

```
47. SELECT name FROM employees WHERE name REGEXP '^[A-Z][a-z]+$';
```

```
48. SELECT DATE(hire_date) AS hire_day, COUNT(*) FROM employees GROUP BY  
DATE(hire_date);
```

```
49. SELECT MONTH(hire_date) AS hire_month, COUNT(*) FROM employees GROUP BY  
MONTH(hire_date);
```

```
50. SELECT * FROM (SELECT *, NTILE(4) OVER (ORDER BY salary DESC) AS quartile FROM  
employees) AS salary_ranked WHERE quartile = 1;
```