

READY-TO-USE TEMPLATES

SQL Query Templates

Replace the bracketed placeholders with your own table and column names. Each template is followed by a filled-in example so you can see exactly how it works.

1. Basic Filter & Sort

Retrieve specific columns, filter to a condition, and sort the results.

```
SELECT column1, column2
FROM table_name
WHERE column1 = 'value'
ORDER BY column2 DESC;
```

EXAMPLE

```
SELECT Name, AdmissionDate
FROM Admissions
WHERE Department = 'Cardiology'
ORDER BY AdmissionDate DESC;
```

2. Range & Partial Text Filter

Filter to a numeric range and a partial text match.

```
SELECT *
FROM table_name
WHERE numeric_column BETWEEN low AND high
AND text_column LIKE '%keyword%';
```

EXAMPLE

```
SELECT *
FROM Patients
WHERE Age BETWEEN 18 AND 65
AND Diagnosis LIKE '%diabetes%';
```

3. Joining Two Tables

Combine matching rows from two related tables.

```
SELECT a.column1, b.column2
FROM table_a a
INNER JOIN table_b b
ON a.id_column = b.id_column;
```

EXAMPLE

```
SELECT p.Name, a.AdmissionDate
FROM Patients p
INNER JOIN Admissions a
ON p.PatientID = a.PatientID;
```

4. Grouped Summary

Count or summarize rows by category.

```
SELECT category_column, COUNT(*) AS total
FROM table_name
GROUP BY category_column;
```

EXAMPLE

```
SELECT Department, COUNT(*) AS total
FROM Admissions
GROUP BY Department;
```

5. Filtering a Grouped Result

Use HAVING (not WHERE) to filter after grouping/aggregation.

```
SELECT category_column, COUNT(*) AS total
FROM table_name
GROUP BY category_column
HAVING COUNT(*) > threshold;
```

EXAMPLE

```
SELECT Department, COUNT(*) AS total
FROM Admissions
GROUP BY Department
HAVING COUNT(*) > 50;
```

6. Subquery (a Question Inside a Question)

Filter using a value calculated by an inner query.

```
SELECT *
FROM table_name
WHERE numeric_column > (
    SELECT AVG(numeric_column) FROM table_name
);
```

EXAMPLE

```
SELECT *
FROM Admissions
WHERE LengthOfStay > (
    SELECT AVG(LengthOfStay) FROM Admissions
);
```

7. Left Join (Keep Everything on One Side)

Return all rows from the first table, even without a match in the second.

```
SELECT a.column1, b.column2
FROM table_a a
LEFT JOIN table_b b
    ON a.id_column = b.id_column;
```

EXAMPLE

```
SELECT p.Name, a.AdmissionDate
FROM Patients p
LEFT JOIN Admissions a
    ON p.PatientID = a.PatientID;
```