

What Is It?

An index that contains all columns a query needs, so SQL can answer the query from the index, without touching the main table.

How It Works

A normal index lookup has two steps:

1. find the row in the index
2. then go back to the table for any extra columns (called a key lookup or bookmark lookup).

A covering index eliminates that second step.

How to Create One

MySQL, Oracle:	<pre>CREATE INDEX idx ON orders (customer_id, order_date, status)</pre>
SQL Server, PostgreSQL:	<pre>CREATE INDEX idx ON orders (customer_id) INCLUDE (order_date, status)</pre>

The Column Order Rule

Put filter/join columns first, then sort columns, then SELECT columns.
The index can only be used from left to right.

When Does the Database Use a Covering Index?

It could be used when:

- All columns from the SELECT clause are in the index
- The WHERE clause uses the leading columns
- SELECT * is not used

In an execution plan, it appears as:

SQL Server: "Index Seek" with no Key Lookup

PostgreSQL: "Index Only Scan"

MySQL: "Using index" in EXPLAIN

Oracle: INDEX RANGE SCAN or INDEX FAST FULL SCAN

When To Use

- High-frequency read queries on large tables
- Queries returning a small subset of columns
- Reports hitting the same table repeatedly

When To Avoid

- Columns change often (index must update too)
- Table is small (full scan is fine)
- Too many indexes already slow down writes