

Problem Statement 1:

Write an SQL query to retrieve all records from a "Products" table where the price is greater than \$50. Additionally, include a condition to filter results by a specific category. Optimize the query for performance.

Solution:

```
SELECT *  
FROM Products  
WHERE price > 50  
AND category = "books"  
ORDER BY price DESC;
```

Optimizations:

1. Indexing: Ensure that price and category columns are indexed to improve query performance.
2. Filtering Early: The WHERE clause reduces the dataset before sorting.
3. Sorting Efficiently: Ordering by price DESC ensures that the most expensive products appear first.

Best Indexing Strategy

A **composite index** on (category, price) is ideal because:

1. Filtering Efficiency: The database first filters by category and then efficiently searches for price > 50 within that category.
2. Ordering Benefit: If queries often sort by price DESC, the index can further speed up sorting.

Creating the Index (indexName-> idx_category_price)

```
CREATE INDEX idx_category_price  
ON Products (category, price);
```