Module 14: Analyzing Queries

Overview

- **Queries That Use**
  - the AND Operator
  - the OR Operator
  - Join Operations

Queries That Use the AND Operator

- **Processing the AND Operator**
  - Returns rows that meet all conditions for every criterion specified in the WHERE clause
  - Progressively limits the number of rows returned with each additional search condition
  - Can use an index for each search condition of the WHERE clause
- **Indexing Guidelines and Performance Considerations**
  - Define an index on one highly selective search criterion
  - Evaluate performance between creating multiple, single-column indexes and creating a composite index

Queries That Use the OR Operator

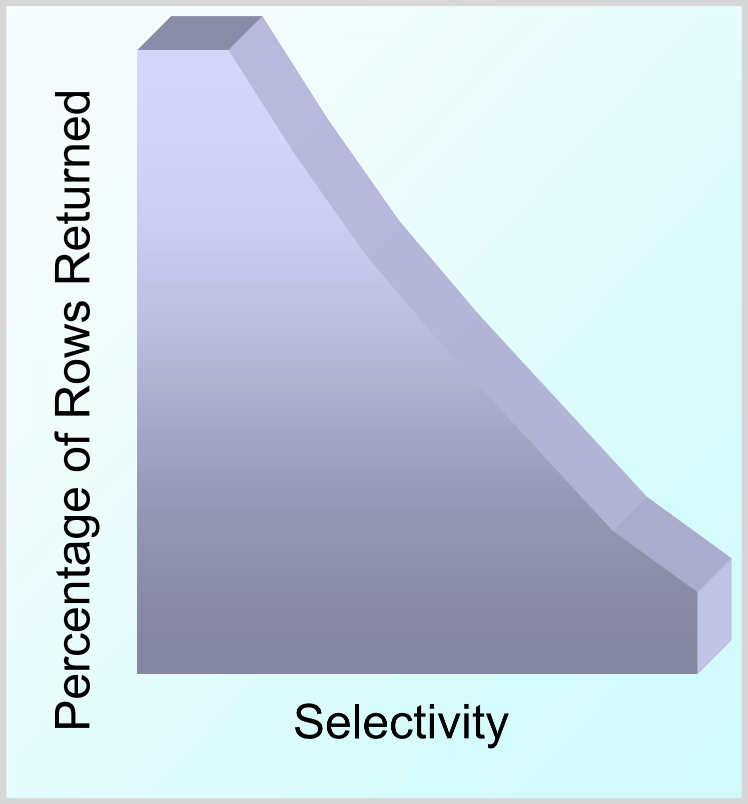
- **Processing the AND Operator**
  - Returns Rows That Meet Any of the Conditions for Every Criterion Specified in the WHERE Clause
  - Progressively Increases the Number of Rows Returned with Each Additional Search Condition
  - Can Use One Index or Different Indexes for Each Part of the OR Operator
- **Always Performs a Table Scan or Clustered Index Scan If One Column Referenced in the OR Operator Does Not Have an Index or If the Index Is Not Useful**
- **Can Use Multiple Indexes**

◆Queries That Use Join Operations

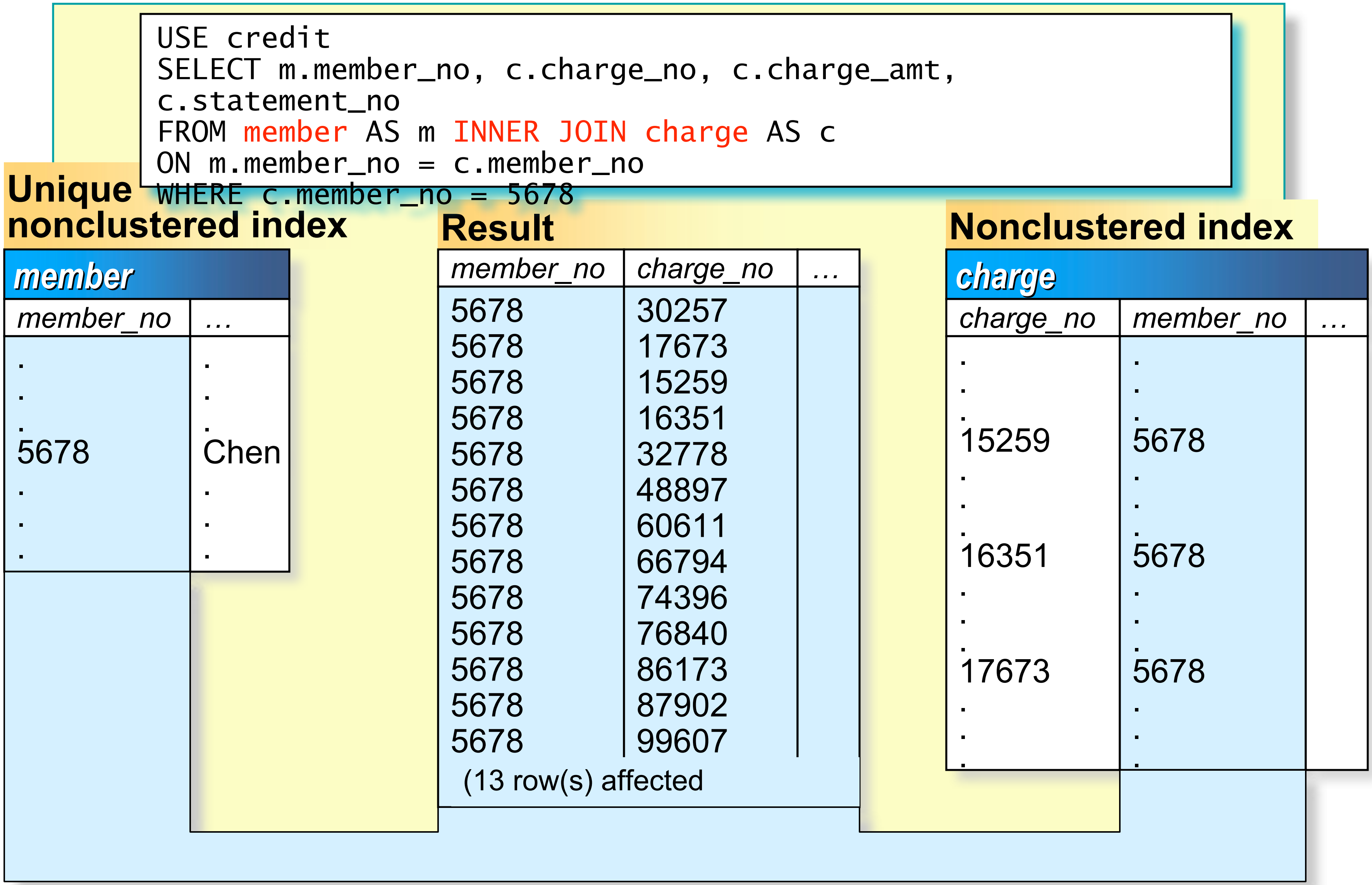
- Selectivity and Density of a JOIN Clause
- How Joins Are Processed
- How Nested Loop Joins Are Processed
- Multimedia: How Joins Are Processed
- Considerations When Merge Joins Are Used
- How Hash Joins Are Processed

Selectivity and Density of a JOIN Clause

- **Selectivity of a JOIN Clause**
  - Based on index density, if statistics are available
  - Based on a number of considerations, if statistics are unavailable
- **Density of a JOIN Clause**
  - An index with large number of duplicates has high join density
  - A unique index has low join density



How Joins Are Processed



```
USE credit
SELECT m.member_no, c.charge_no, c.charge_amt,
c.statement_no
FROM member AS m INNER JOIN charge AS c
ON m.member_no = c.member_no
WHERE c.member_no = 5678
```

Unique nonclustered index

member

member_no	…
.	.
.	.
.	.
5678	Chen
.	.
.	.
.	.

Result

member_no	charge_no	…
5678	30257	
5678	17673	
5678	15259	
5678	16351	
5678	32778	
5678	48897	
5678	60611	
5678	66794	
5678	74396	
5678	76840	
5678	86173	
5678	87902	
5678	99607	

(13 row(s) affected)

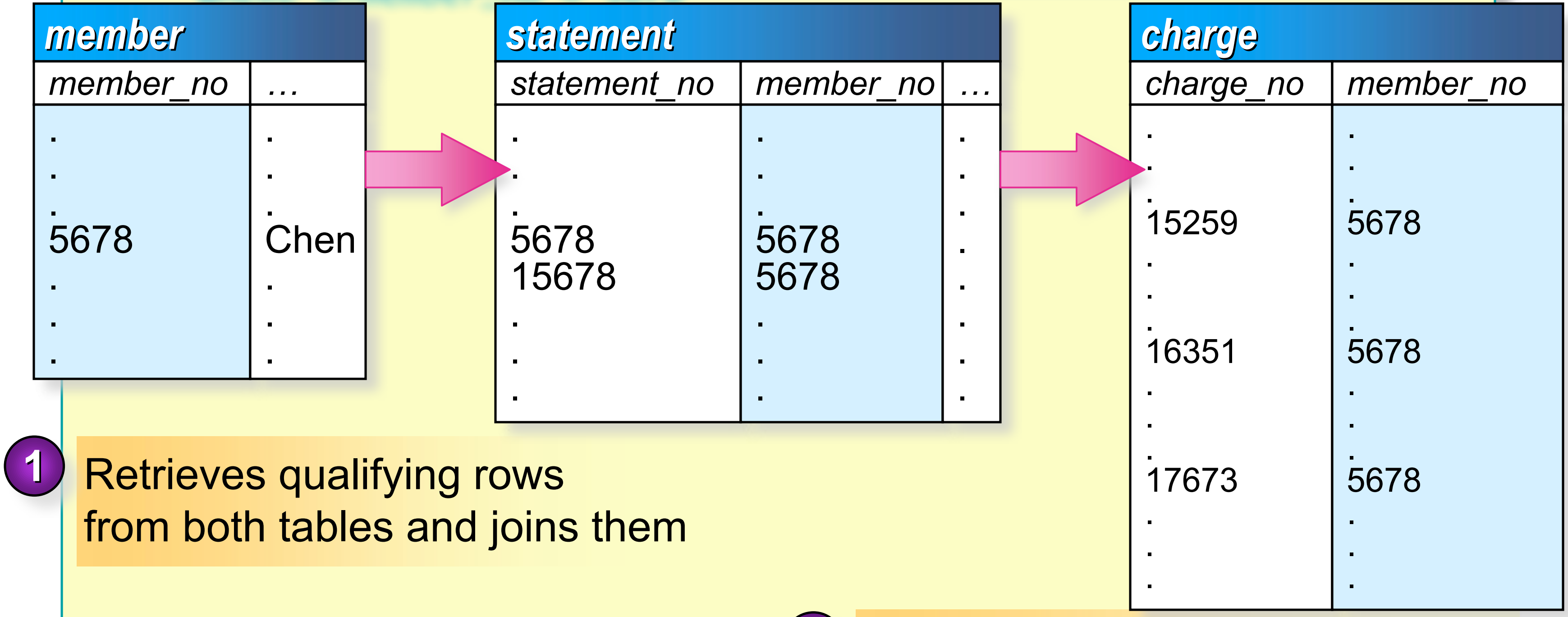
Nonclustered index

charge

charge_no	member_no	…
.	.	
.	.	
.	.	
15259	5678	
.	.	
.	.	
.	.	
16351	5678	
.	.	
.	.	
.	.	
17673	5678	
.	.	
.	.	
.	.	

How Nested Loop Joins Are Processed

```
USE credit
SELECT m.member_no, c.charge_no, c.charge_amt,
s.statement_no
FROM member AS m INNER JOIN charge AS c
ON m.member_no = c.member_no
INNER JOIN statement AS s
ON c.member_no = s.member_no
WHERE m.member_no = 5678
```



member

member_no	…
.	.
.	.
.	.
5678	Chen
.	.
.	.
.	.

statement

statement_no	member_no	…
.	.	.
.	.	.
.	.	.
5678	5678	.
15678	5678	.
.	.	.
.	.	.
.	.	.

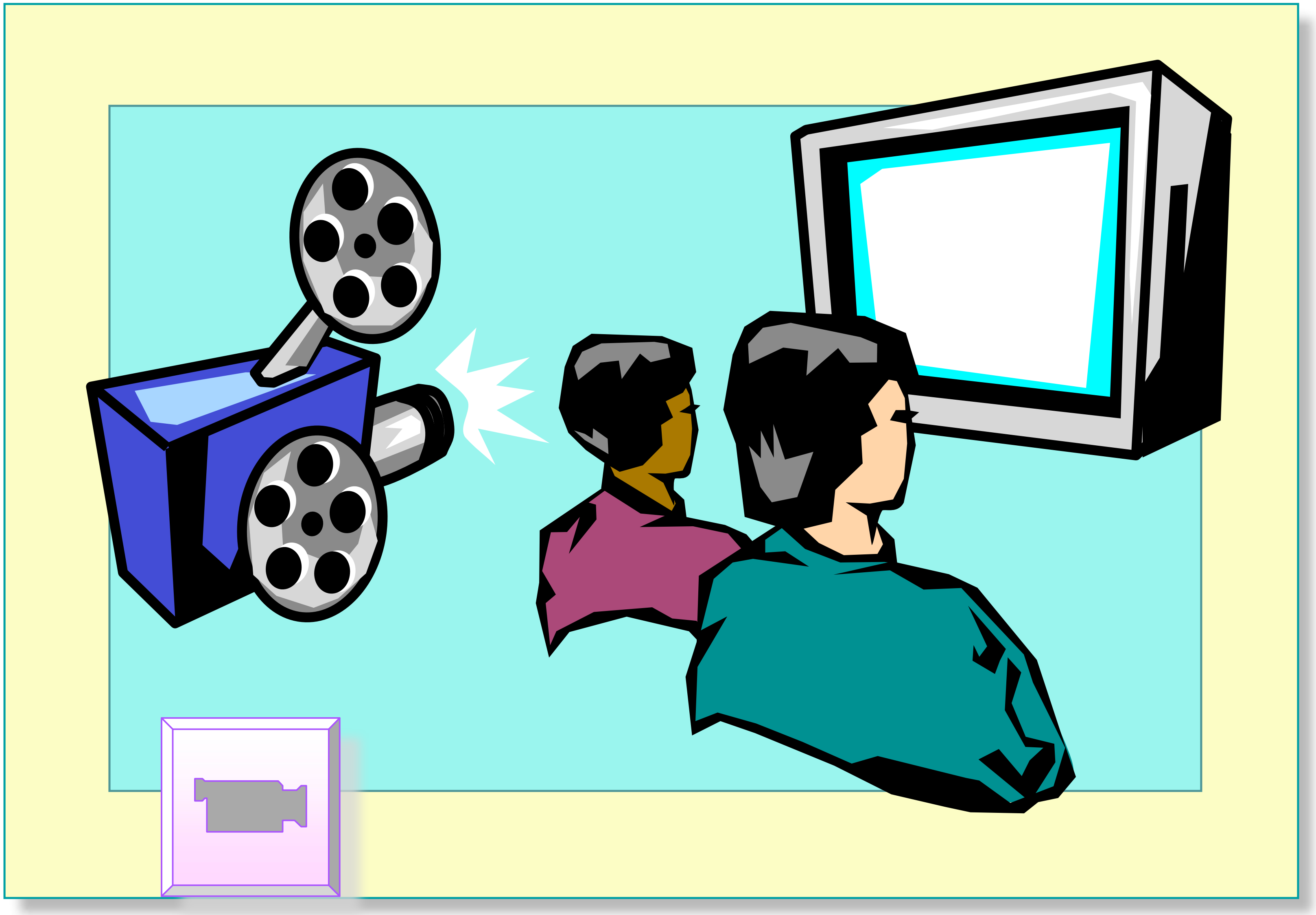
charge

charge_no	member_no
.	.
.	.
.	.
15259	5678
.	.
.	.
.	.
16351	5678
.	.
.	.
.	.
17673	5678
.	.
.	.
.	.

1. Retrieves qualifying rows from both tables and joins them

2. Joins the results with the qualifying rows of the charge table

Multimedia: How Merge Joins Are Processed



Considerations When Merge Joins Are Used

- **Requires That Joined Columns Are Sorted**
- **Evaluates Sorted Values**
  - Uses an existing index tree
  - Leverages sort operations
  - Performs its own sort operation
- **Performance Considerations**
  - Usually default

```
USE credit
SELECT m.lastname, p.payment_amt
 FROM member AS m INNER JOIN payment AS p
 ON m.member_no = p.member_no
 WHERE p.payment_amt < 7000 AND m.firstname < 'Jak'
```

Recommended Practices

☑ Define an Index on a Highly Selective Column

☑ Ensure That Useful Indexes Exist for All Columns Referenced in the OR Operator

☑ Minimize the Use of Hash Joins

Review

- **Queries That Use**
  - the AND Operator
  - the OR Operator
  - Join Operations