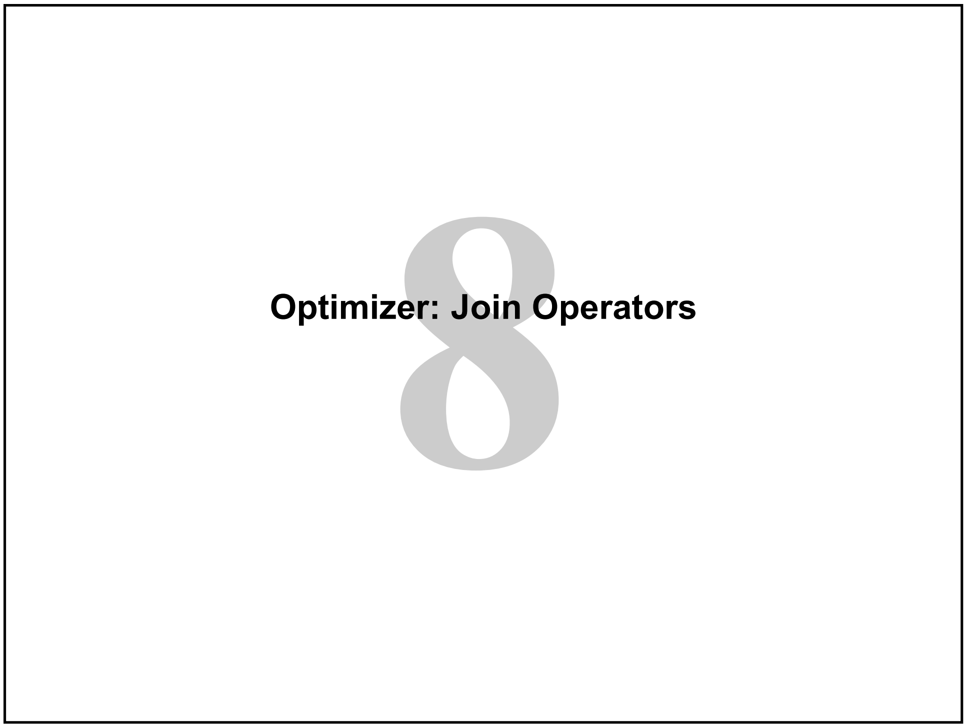
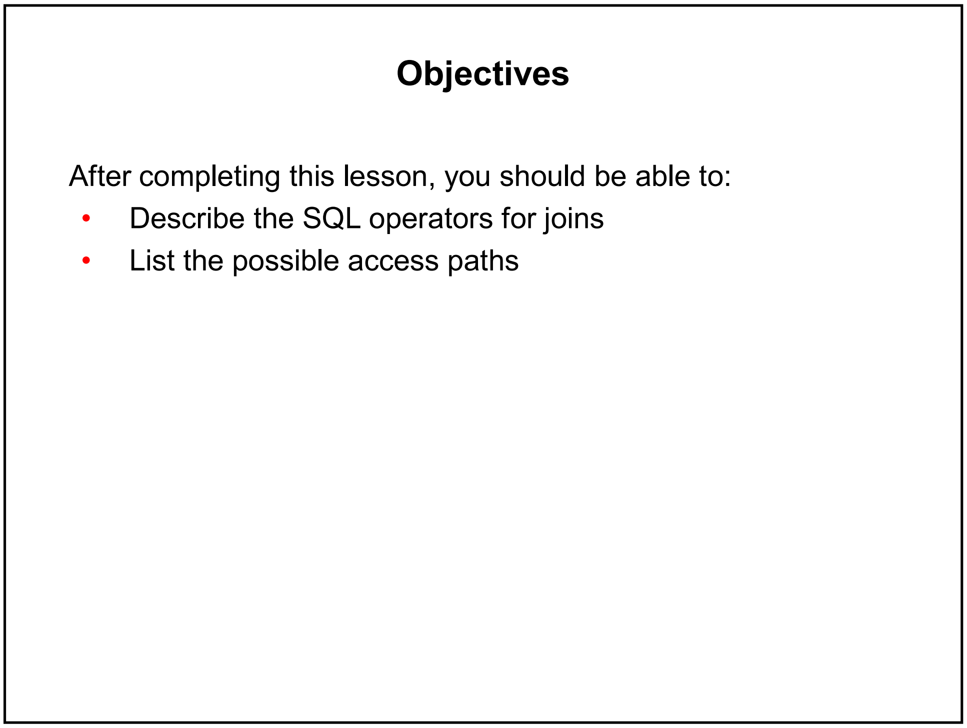


8

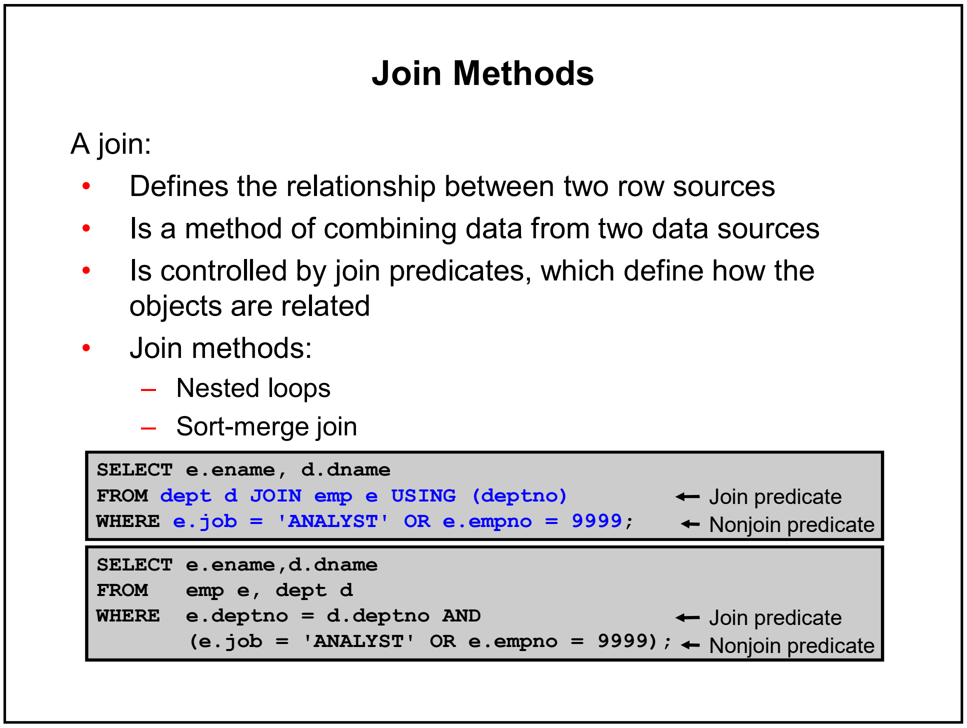
Optimizer: Join Operators



Objectives

After completing this lesson, you should be able to:

- Describe the SQL operators for joins
- List the possible access paths



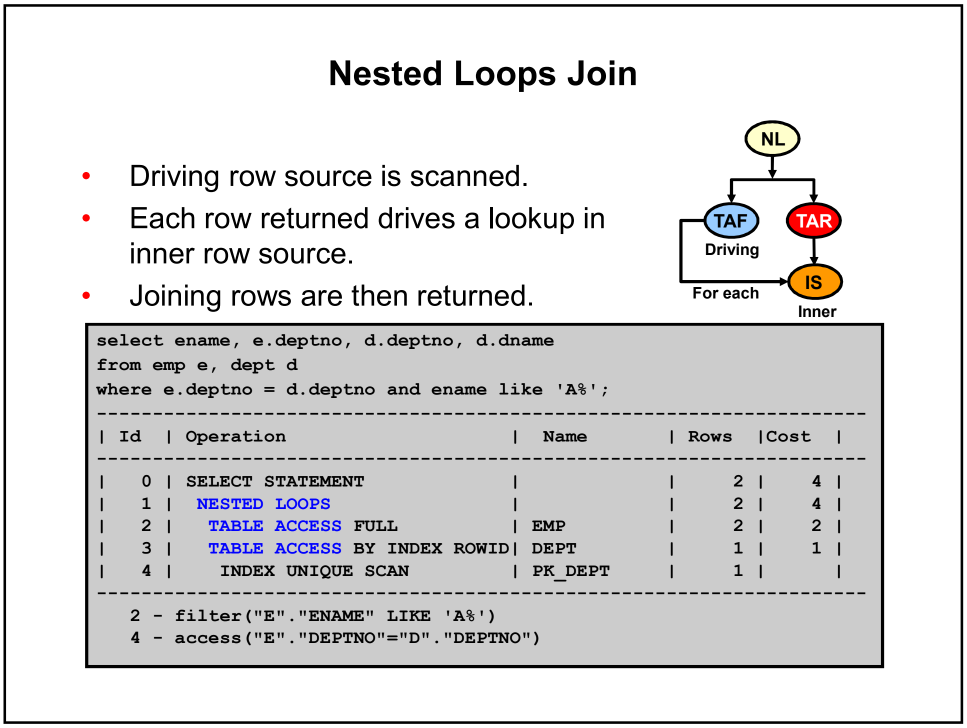
Join Methods

A join:

- Defines the relationship between two row sources
- Is a method of combining data from two data sources
- Is controlled by join predicates, which define how the objects are related
- Join methods:
  - Nested loops
  - Sort-merge join

```
SELECT e.ename, d.dname
FROM dept d JOIN emp e USING (deptno)                  ← Join predicate
WHERE e.job = 'ANALYST' OR e.empno = 9999;             ← Nonjoin predicate
```

```
SELECT e.ename,d.dname
FROM   emp e, dept d
WHERE  e.deptno = d.deptno AND                         ← Join predicate
       (e.job = 'ANALYST' OR e.empno = 9999);          ← Nonjoin predicate
```



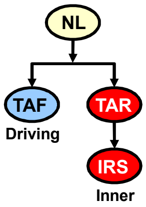
Nested Loops Join

- Driving row source is scanned.
- Each row returned drives a lookup in inner row source.
- Joining rows are then returned.

NL
TAF
TAR
Driving
For each
IS
Inner

```
select ename, e.deptno, d.deptno, d.dname
from emp e, dept d
where e.deptno = d.deptno and ename like 'A%';
---------------------------------------------------------------------
| Id  | Operation                    |  Name        | Rows  |Cost  |
---------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |     2 |     4 |
|   1 |  NESTED LOOPS                |              |     2 |     4 |
|   2 |   TABLE ACCESS FULL          | EMP          |     2 |     2 |
|   3 |   TABLE ACCESS BY INDEX ROWID| DEPT         |     1 |     1 |
|   4 |    INDEX UNIQUE SCAN         | PK_DEPT      |     1 |       |
---------------------------------------------------------------------
   2 - filter("E"."ENAME" LIKE 'A%')
   4 - access("E"."DEPTNO"="D"."DEPTNO")
```

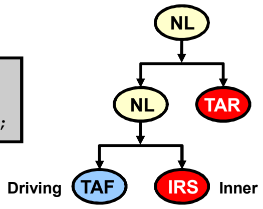
Nested Loops Join: Prefetching

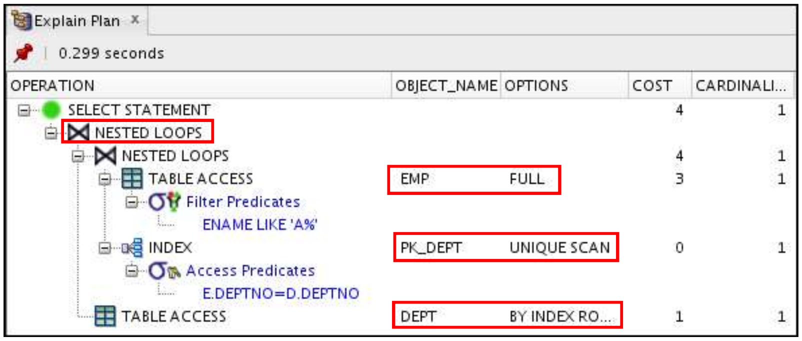


```
select ename, e.deptno, d.deptno, d.dname
from emp e, dept d
where e.deptno = d.deptno and ename like 'A%';
----------------------------------------------------------------------
|   0 | SELECT STATEMENT             |       |     2 |    84 |     5
|   1 |  TABLE ACCESS BY INDEX ROWID| DEPT  |     1 |    22 |     1
|   2 |   NESTED LOOPS               |       |     2 |    84 |     5
|*  3 |    TABLE ACCESS FULL         | EMP   |     2 |    40 |     3
|*  4 |    INDEX RANGE SCAN          | IDEPT |     1 |       |     0
----------------------------------------------------------------------
   3 - filter("E"."ENAME" LIKE 'A%')
   4 - access("E"."DEPTNO"="D"."DEPTNO")
```

Nested Loops Join: 11*g* Implementation

```
select ename, e.deptno, d.deptno, d.dname
from emp e, dept d
where e.deptno = d.deptno and ename like 'A%';
```

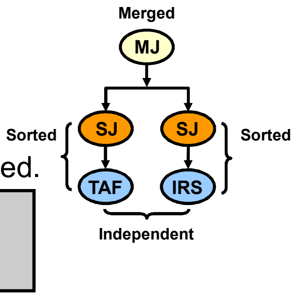




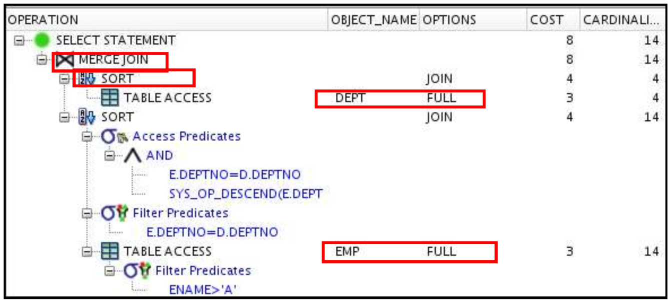
```
0.299 seconds

OPERATION                              OBJECT_NAME  OPTIONS         COST   CARDINALI...
SELECT STATEMENT                                                       4              1
  NESTED LOOPS
    NESTED LOOPS                                                       4              1
      TABLE ACCESS                     EMP          FULL               3              1
        Filter Predicates
          ENAME LIKE 'A%'
      INDEX                            PK_DEPT      UNIQUE SCAN        0              1
        Access Predicates
          E.DEPTNO=D.DEPTNO
    TABLE ACCESS                       DEPT         BY INDEX RO...     1              1
```

Sort-Merge Join

- First and second row sources are sorted by the same sort key.
- Sorted rows from both tables are merged.



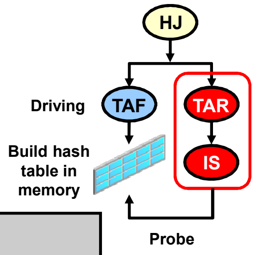
```
select /*+ USE_MERGE(d e) NO_INDEX(d) */
  ename, e.deptno, d.deptno, dname
  from emp e, dept d
  where e.deptno = d.deptno and ename > 'A'
```



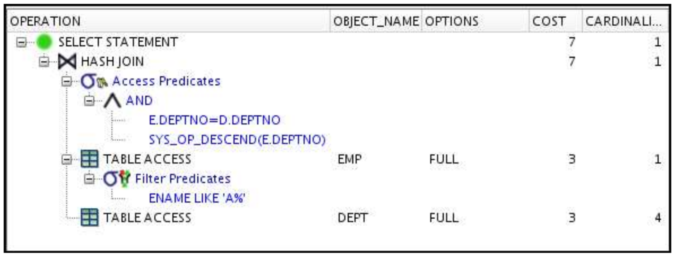
```
OPERATION                              OBJECT_NAME OPTIONS   COST  CARDINALI...
SELECT STATEMENT                                                8     14
  MERGE JOIN                                                    8     14
    SORT                                           JOIN         4      4
      TABLE ACCESS                     DEPT        FULL         3      4
    SORT                                           JOIN         4     14
      Access Predicates
        AND
          E.DEPTNO=D.DEPTNO
          SYS_OP_DESCEND(E.DEPT
      Filter Predicates
        E.DEPTNO=D.DEPTNO
      TABLE ACCESS                     EMP         FULL         3     14
        Filter Predicates
          ENAME>'A'
```

Hash Join

- The smallest row source is used to build a hash table.
- The second row source is hashed and checked against the hash table.

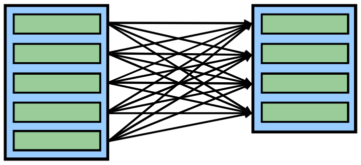


```
select /*+ USE_HASH(e d) */
  ename, e.deptno, d.deptno, dname
  from emp e, dept d
  where e.deptno = d.deptno and ename like 'A%’
```

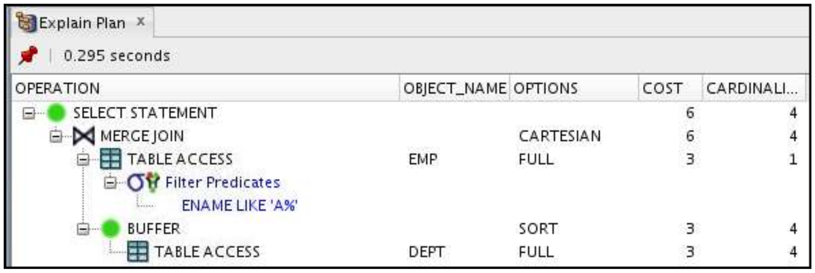


```
OPERATION                              OBJECT_NAME OPTIONS   COST  CARDINALI...
SELECT STATEMENT                                                7      1
  HASH JOIN                                                     7      1
    Access Predicates
      AND
        E.DEPTNO=D.DEPTNO
        SYS_OP_DESCEND(E.DEPTNO)
    TABLE ACCESS                       EMP         FULL         3      1
      Filter Predicates
        ENAME LIKE 'A%'
    TABLE ACCESS                       DEPT        FULL         3      4
```

Cartesian Join



```
select ename, e.deptno, d.deptno, dname
from emp e, dept d where ename like 'A%';
```



0.295 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALI...
SELECT STATEMENT			6	4
MERGE JOIN		CARTESIAN	6	4
TABLE ACCESS	EMP	FULL	3	1
Filter Predicates				
ENAME LIKE 'A%'				
BUFFER		SORT	3	4
TABLE ACCESS	DEPT	FULL	3	4

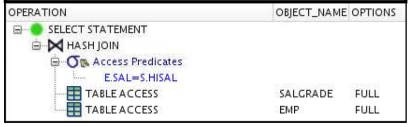
Join Types

- A join operation combines the output from two row sources and returns one resulting row source.
- Join operation types include the following :
  - Join (Equijoin/Natural – Nonequijoin)
  - Outer join (Full, Left, and Right)
  - Semi join: `EXISTS` subquery
  - Anti join: `NOT IN` subquery
  - Star join (Optimization)

Equijoins and Nonequijoins

Equijoin

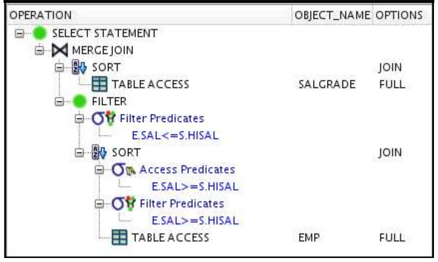
```
SELECT e.ename, e.sal, s.grade
FROM   emp e ,salgrade s
WHERE  e.sal = s.hisal;
```



```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  HASH JOIN
    Access Predicates
      E.SAL=S.HISAL
    TABLE ACCESS                         SALGRADE     FULL
    TABLE ACCESS                         EMP          FULL
```

Nonequijoin

```
SELECT e.ename, e.sal, s.grade
FROM   emp e ,salgrade s
WHERE  e.sal between s.hisal and s.hisal;
```

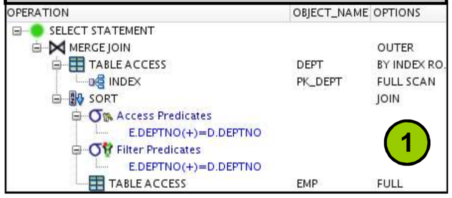


```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  MERGE JOIN
    SORT                                              JOIN
      TABLE ACCESS                       SALGRADE     FULL
    FILTER
      Filter Predicates
        E.SAL<=S.HISAL
      SORT                                            JOIN
        Access Predicates
          E.SAL>=S.HISAL
        Filter Predicates
          E.SAL>=S.HISAL
        TABLE ACCESS                     EMP          FULL
```

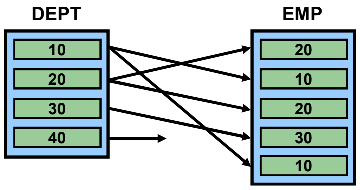
Outer Joins

An outer join returns a row even if no match is found.

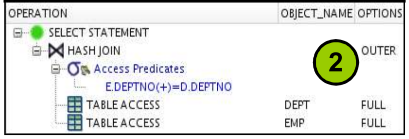
```
SELECT d.deptno,d.dname,e.empno,e.ename
FROM   emp e, dept d
WHERE  e.deptno(+)=d.deptno;
```



```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  MERGE JOIN                                          OUTER
    TABLE ACCESS                         DEPT         BY INDEX RO
      INDEX                              PK_DEPT      FULL SCAN
    SORT                                              JOIN
      Access Predicates
        E.DEPTNO(+)=D.DEPTNO
      Filter Predicates
        E.DEPTNO(+)=D.DEPTNO
      TABLE ACCESS                       EMP          FULL
```

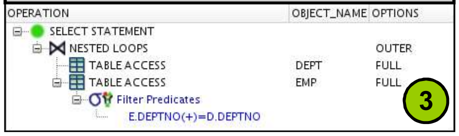


```
SELECT /*+ USE_HASH(e d) */
d.deptno,d.dname,e.empno,e.ename
FROM   emp e, dept d
WHERE  e.deptno(+)=d.deptno;
```



```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  HASH JOIN                                           OUTER
    Access Predicates
      E.DEPTNO(+)=D.DEPTNO
    TABLE ACCESS                         DEPT         FULL
    TABLE ACCESS                         EMP          FULL
```

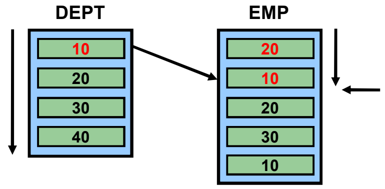
```
SELECT /*+ USE_NL(e d) */
d.deptno,d.dname,e.empno,e.ename
FROM   emp e, dept d
WHERE  e.deptno(+)=d.deptno;
```



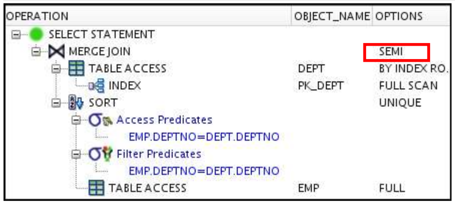
```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  NESTED LOOPS                                        OUTER
    TABLE ACCESS                         DEPT         FULL
    TABLE ACCESS                         EMP          FULL
      Filter Predicates
        E.DEPTNO(+)=D.DEPTNO
```

Semijoins

Semijoins look only for the first match.



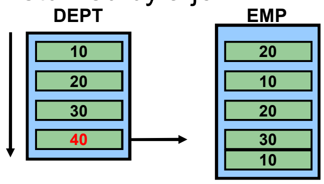
```
SELECT deptno, dname
FROM   dept
WHERE  EXISTS (SELECT 1 FROM emp WHERE emp.deptno=dept.deptno);
```



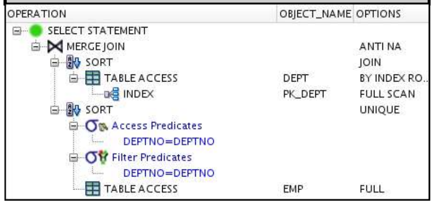
```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  MERGE JOIN                                          SEMI
    TABLE ACCESS                         DEPT         BY INDEX RO.
      INDEX                              PK_DEPT      FULL SCAN
    SORT                                              UNIQUE
      Access Predicates
        EMP.DEPTNO=DEPT.DEPTNO
      Filter Predicates
        EMP.DEPTNO=DEPT.DEPTNO
      TABLE ACCESS                       EMP          FULL
```

Antijoins

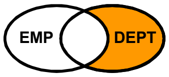
Reverse of what would have been returned by a join



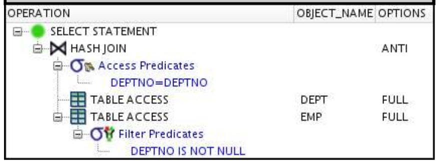
```
SELECT deptno, dname
FROM   dept
WHERE  deptno not in
      (SELECT deptno FROM emp);
```



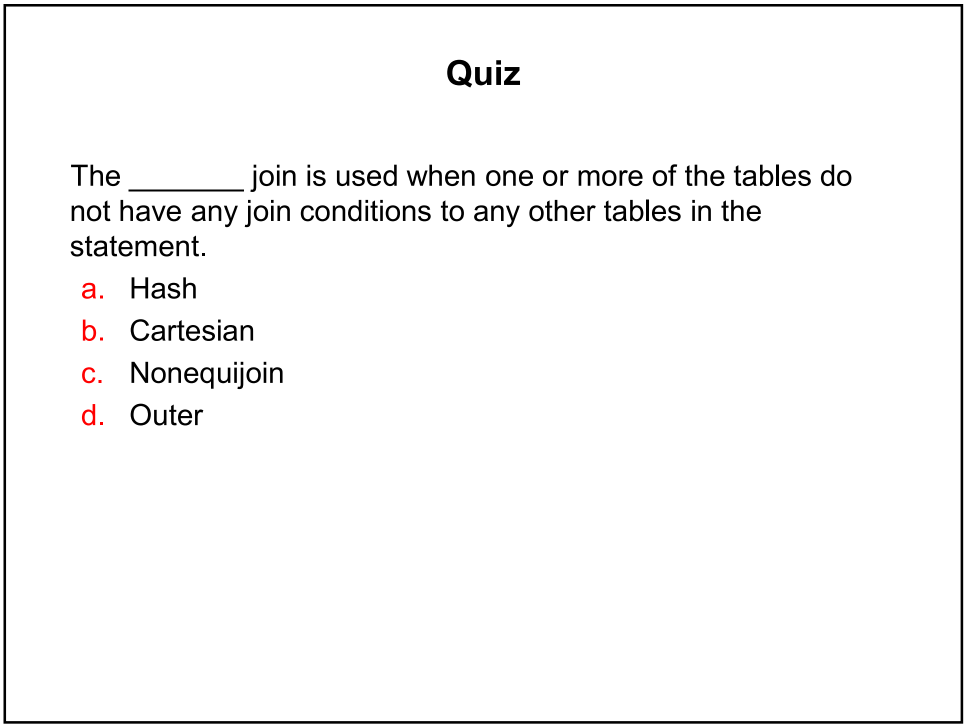
```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  MERGE JOIN                                          ANTI NA
    SORT                                              JOIN
      TABLE ACCESS                       DEPT         BY INDEX RO..
        INDEX                            PK_DEPT      FULL SCAN
    SORT                                              UNIQUE
      Access Predicates
        DEPTNO=DEPTNO
      Filter Predicates
        DEPTNO=DEPTNO
      TABLE ACCESS                       EMP          FULL
```



```
SELECT deptno, dname FROM dept
WHERE deptno IS NOT NULL AND
deptno NOT IN
 (SELECT /*+ HASH_AJ */ deptno FROM
emp WHERE deptno IS NOT NULL);
```



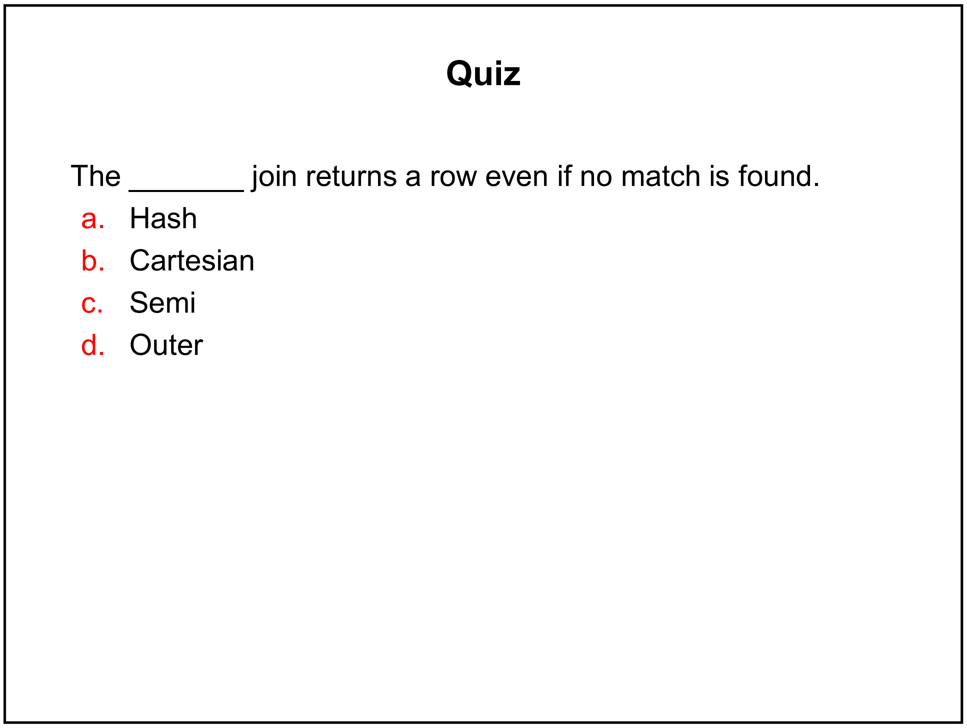
```
OPERATION                                OBJECT_NAME  OPTIONS
SELECT STATEMENT
  HASH JOIN                                           ANTI
    Access Predicates
      DEPTNO=DEPTNO
    TABLE ACCESS                         DEPT         FULL
    TABLE ACCESS                         EMP          FULL
      Filter Predicates
        DEPTNO IS NOT NULL
```



Quiz

The ________ join is used when one or more of the tables do not have any join conditions to any other tables in the statement.

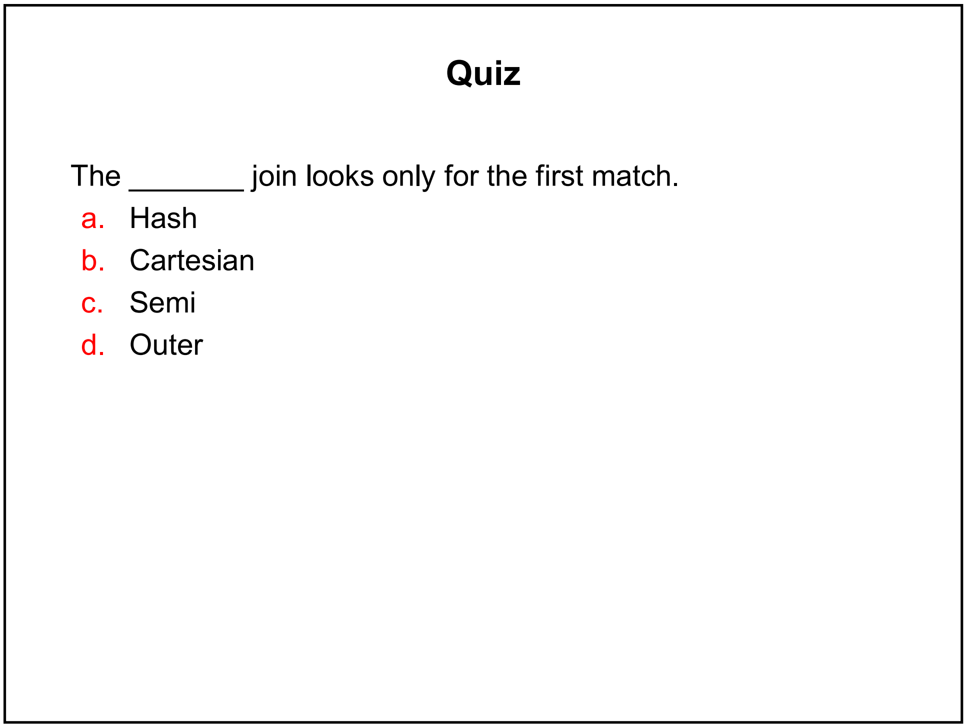
a. Hash
b. Cartesian
c. Nonequijoin
d. Outer



Quiz

The ________ join returns a row even if no match is found.

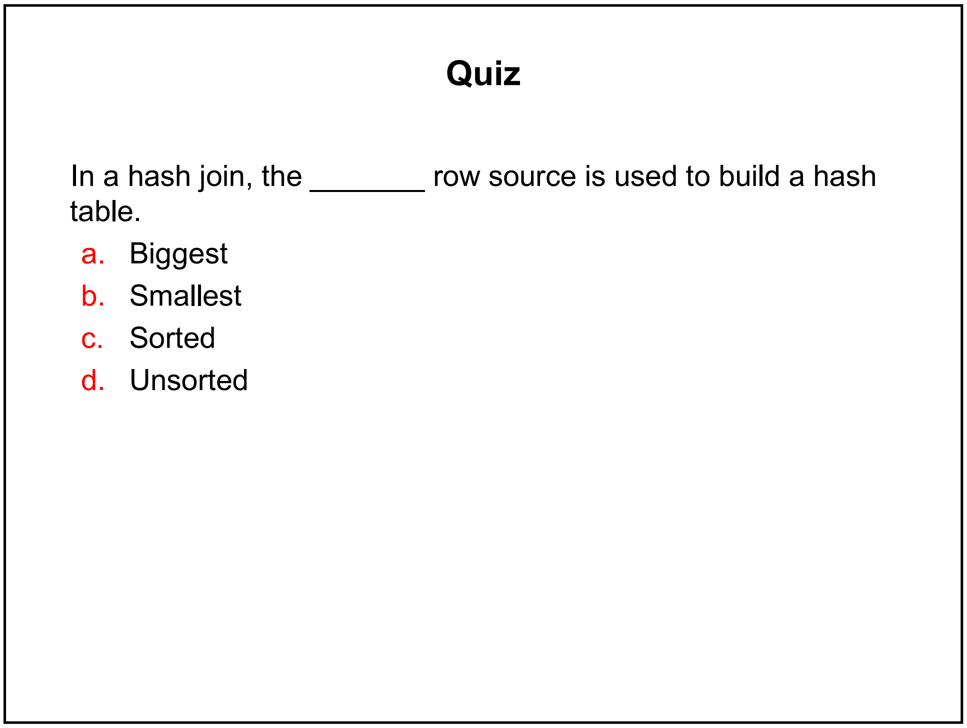
a. Hash
b. Cartesian
c. Semi
d. Outer



Quiz

The ________ join looks only for the first match.

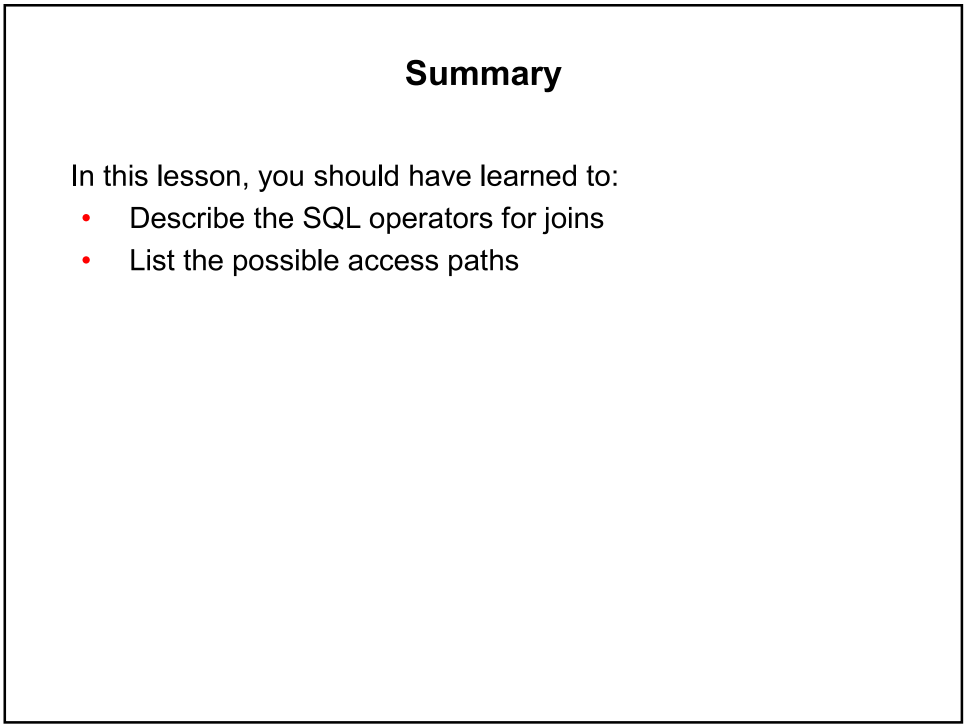
a. Hash
b. Cartesian
c. Semi
d. Outer



Quiz

In a hash join, the ________ row source is used to build a hash table.

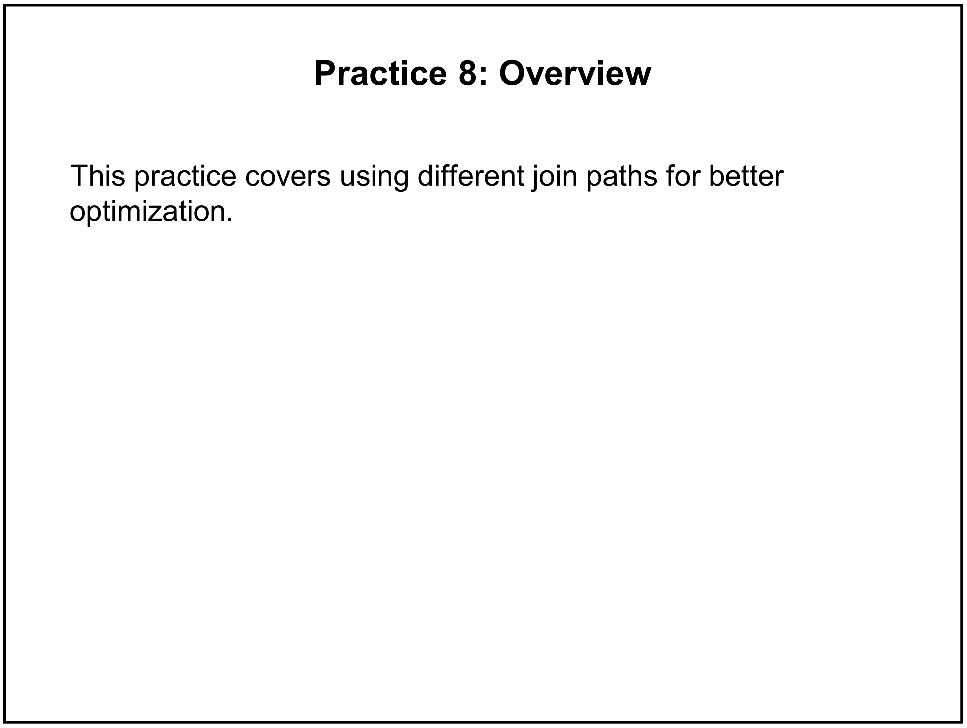
a. Biggest
b. Smallest
c. Sorted
d. Unsorted



Summary

In this lesson, you should have learned to:

- Describe the SQL operators for joins
- List the possible access paths



Practice 8: Overview

This practice covers using different join paths for better optimization.