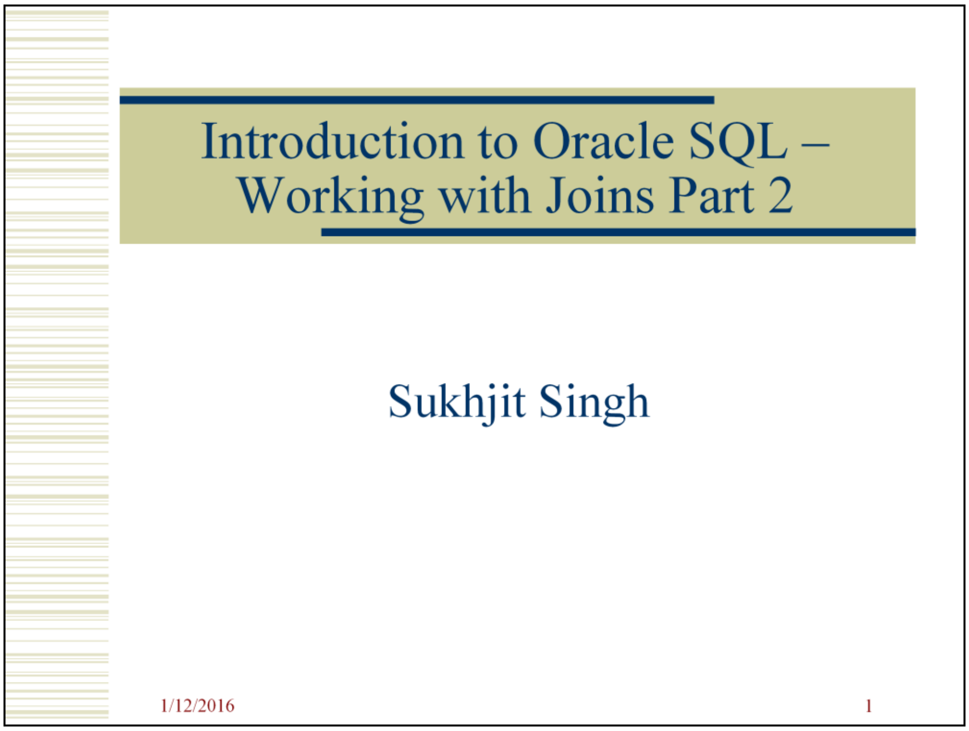
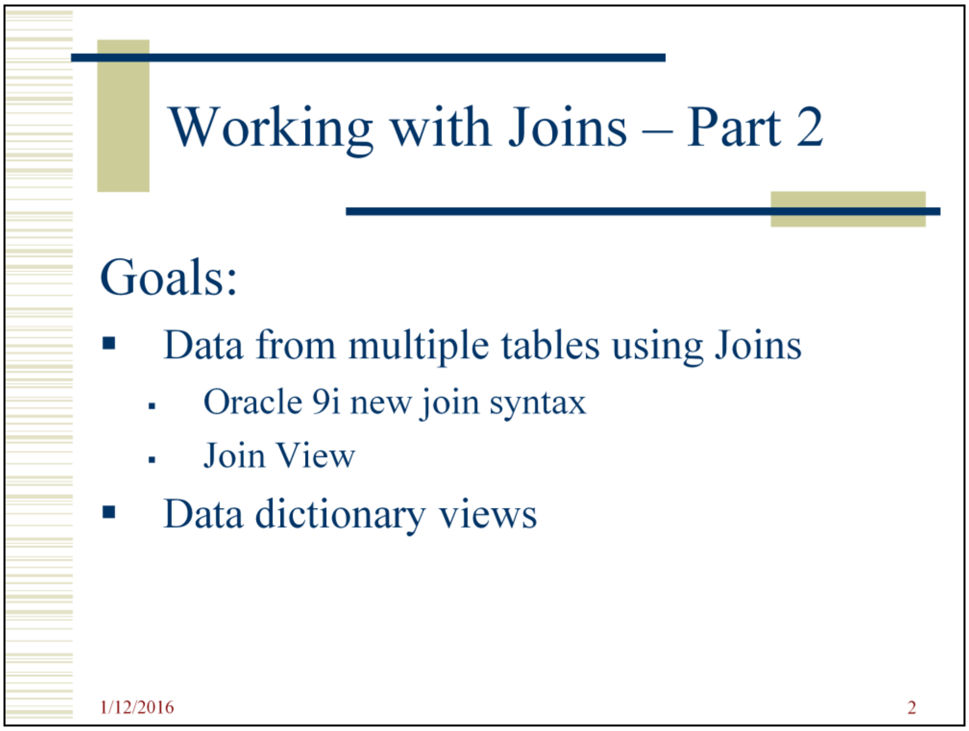


Introduction to Oracle SQL – Working with Joins Part 2

Sukhjit Singh

1/12/2016

1

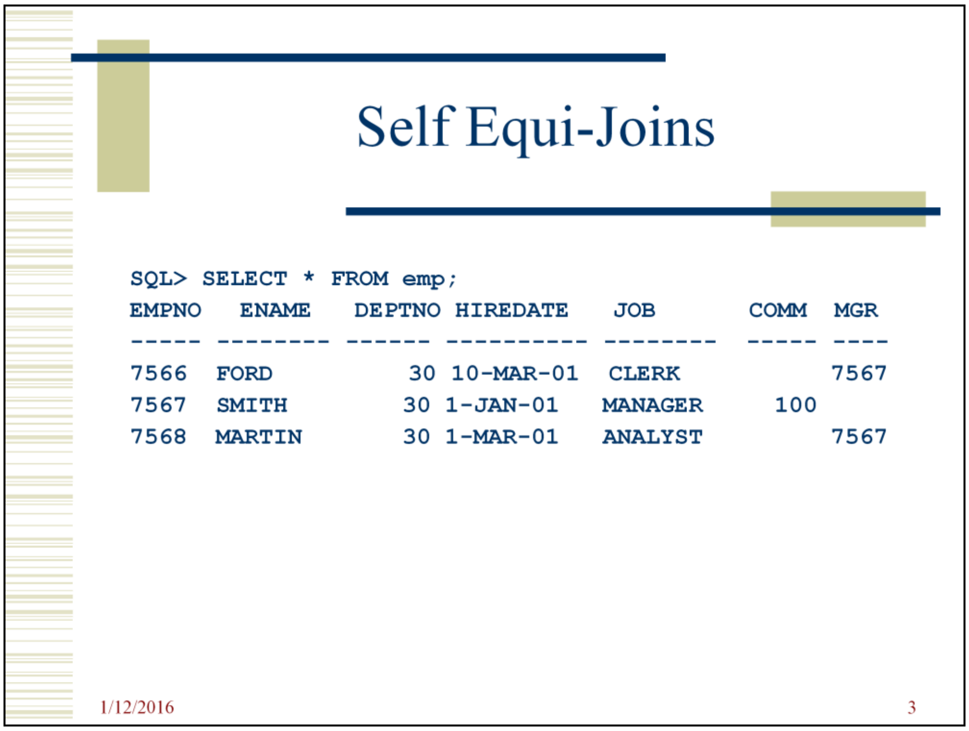


Working with Joins – Part 2

Goals:

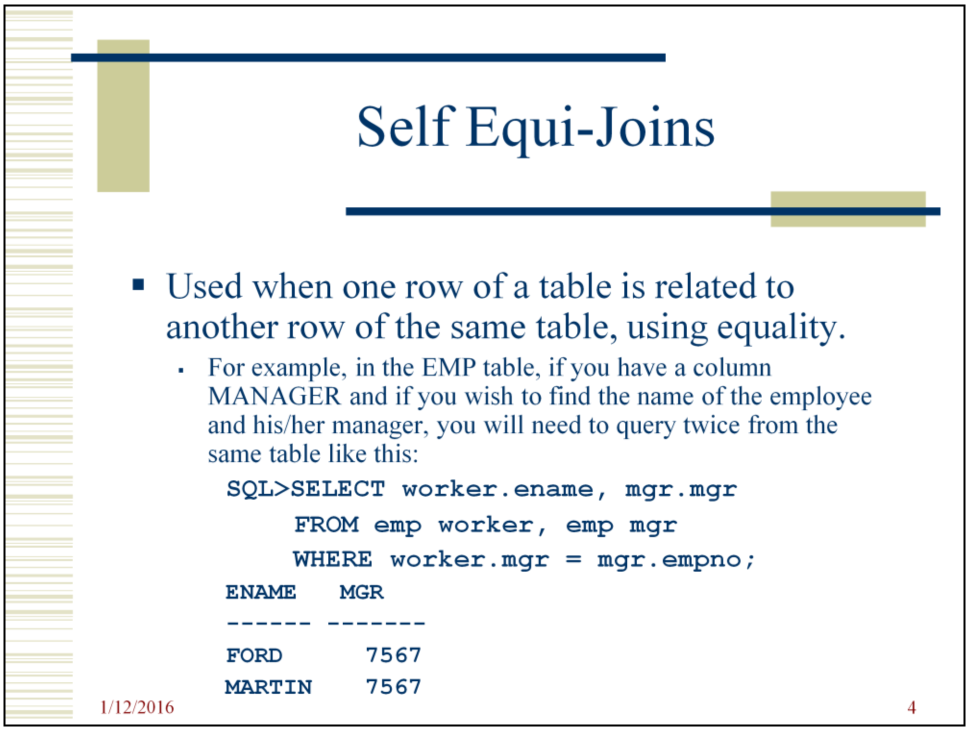
- Data from multiple tables using Joins
  - Oracle 9i new join syntax
  - Join View
- Data dictionary views

1/12/2016 2



Self Equi-Joins

```
SQL> SELECT * FROM emp;
EMPNO   ENAME    DEPTNO HIREDATE    JOB        COMM  MGR
----- -------- ------ ---------- --------   ----- ----
7566  FORD         30 10-MAR-01  CLERK             7567
7567  SMITH        30 1-JAN-01   MANAGER     100
7568  MARTIN       30 1-MAR-01   ANALYST           7567
```



Self Equi-Joins

- Used when one row of a table is related to another row of the same table, using equality.
  - For example, in the EMP table, if you have a column MANAGER and if you wish to find the name of the employee and his/her manager, you will need to query twice from the same table like this:

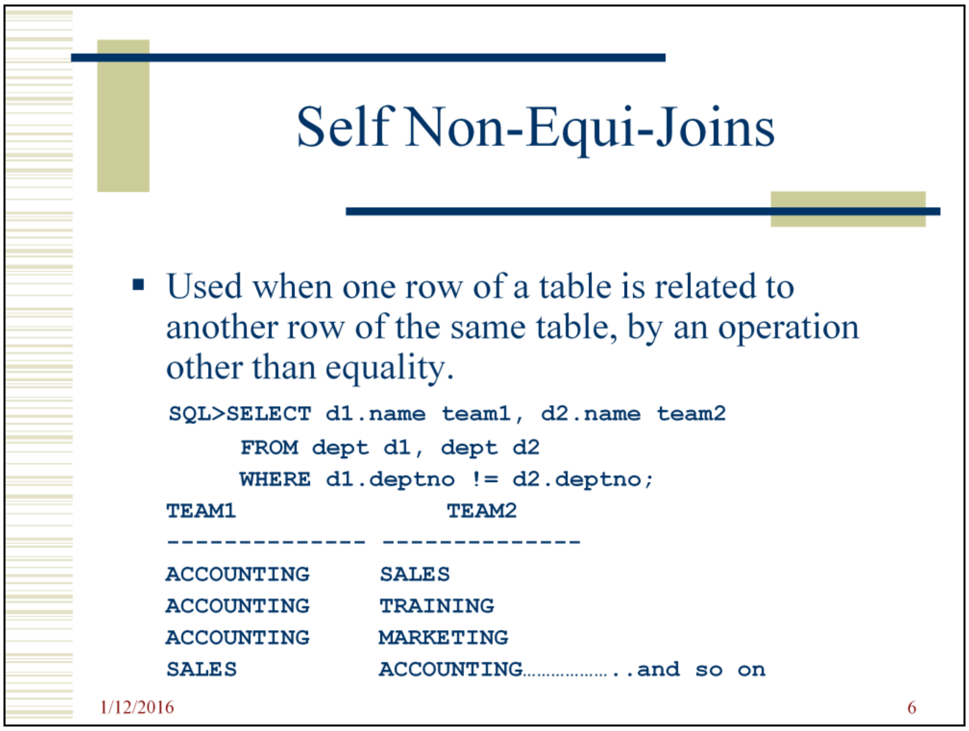
```
SQL>SELECT worker.ename, mgr.mgr
    FROM emp worker, emp mgr
    WHERE worker.mgr = mgr.empno;
ENAME    MGR
------ -------
FORD      7567
MARTIN    7567
```

1/12/2016 4

Self Outer Equi-Joins

- In the above example, if there are employees without any manager id, they would not be retrieved in the previous query. For that, self equi-outer join is required.

```
SQL>SELECT worker.ename, mgr.mgr
      FROM emp worker, emp mgr
     WHERE worker.mgr =  mgr.empno(+);
ENAME    MGR
------ -------
FORD    7567
MARTIN 7567
KING
```



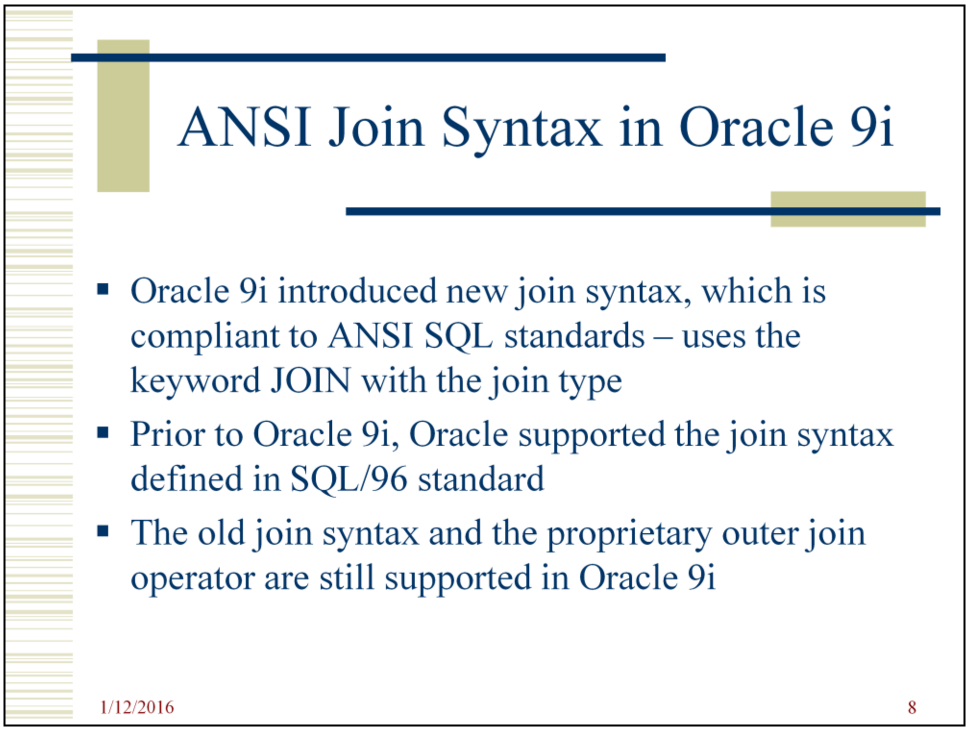
Self Non-Equi-Joins

- Used when one row of a table is related to another row of the same table, by an operation other than equality.

```
SQL>SELECT d1.name team1, d2.name team2
     FROM dept d1, dept d2
     WHERE d1.deptno != d2.deptno;
TEAM1                TEAM2
-------------- --------------
ACCOUNTING     SALES
ACCOUNTING     TRAINING
ACCOUNTING     MARKETING
SALES          ACCOUNTING..................and so on
```

Joining more than two tables

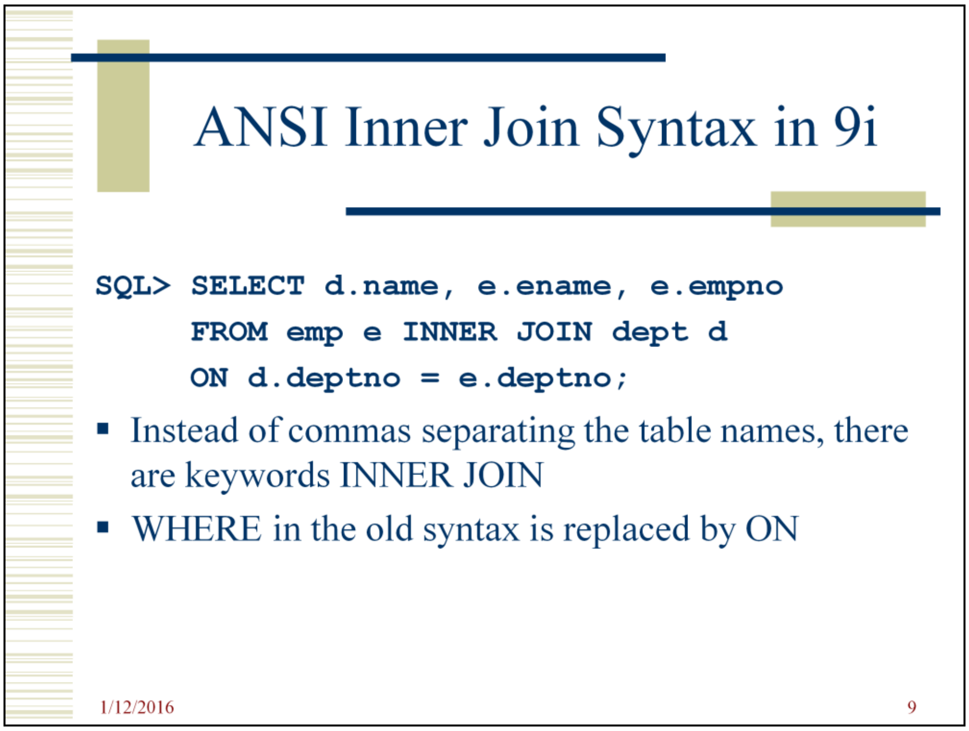
```
SQL> SELECT e.ename, d.deptno, e.sal,
     s.grade
     FROM emp e, dept d, salgrade s
     WHERE e.sal BETWEEN s.losal AND
     s.highsal
     AND e.deptno(+) = d.deptno
     AND E.NAME = ‘MARTIN’;
```



ANSI Join Syntax in Oracle 9i

- Oracle 9i introduced new join syntax, which is compliant to ANSI SQL standards – uses the keyword JOIN with the join type
- Prior to Oracle 9i, Oracle supported the join syntax defined in SQL/96 standard
- The old join syntax and the proprietary outer join operator are still supported in Oracle 9i

1/12/2016 8

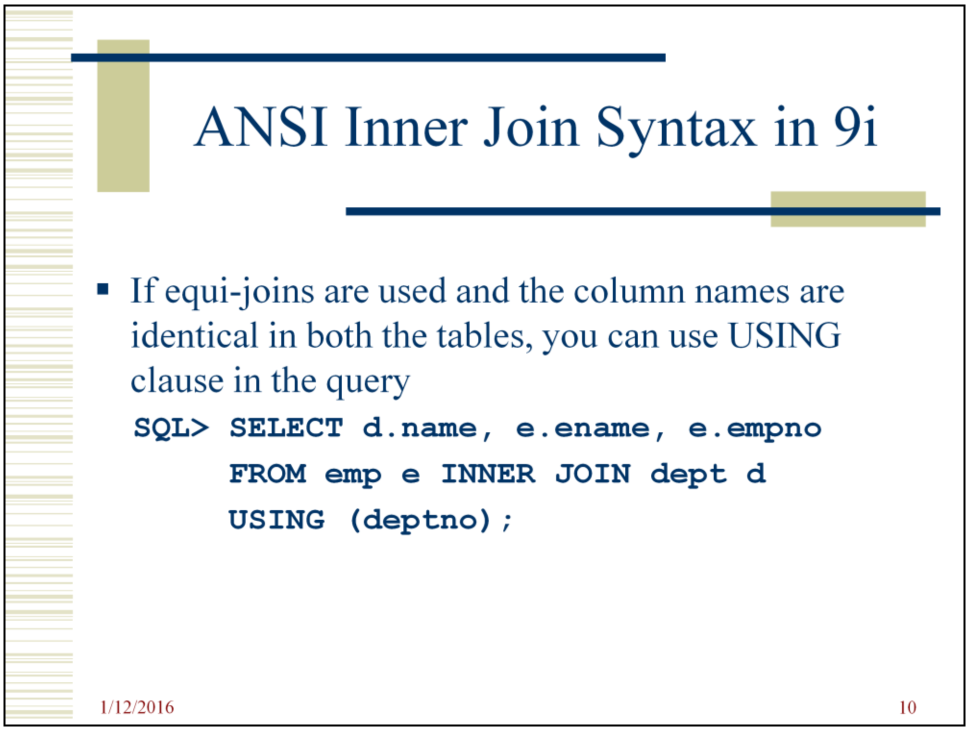


ANSI Inner Join Syntax in 9i

```
SQL> SELECT d.name, e.ename, e.empno
     FROM emp e INNER JOIN dept d
     ON d.deptno = e.deptno;
```

- Instead of commas separating the table names, there are keywords INNER JOIN
- WHERE in the old syntax is replaced by ON

1/12/2016 9

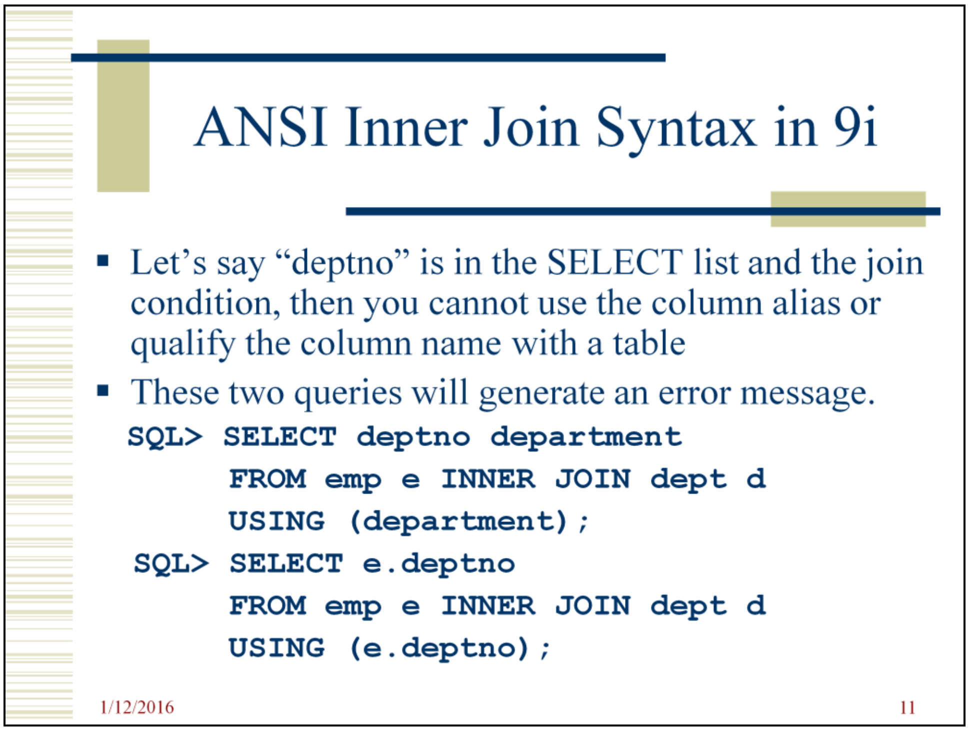


ANSI Inner Join Syntax in 9i

- If equi-joins are used and the column names are identical in both the tables, you can use USING clause in the query

```
SQL> SELECT d.name, e.ename, e.empno
     FROM emp e INNER JOIN dept d
     USING (deptno);
```

1/12/2016 10

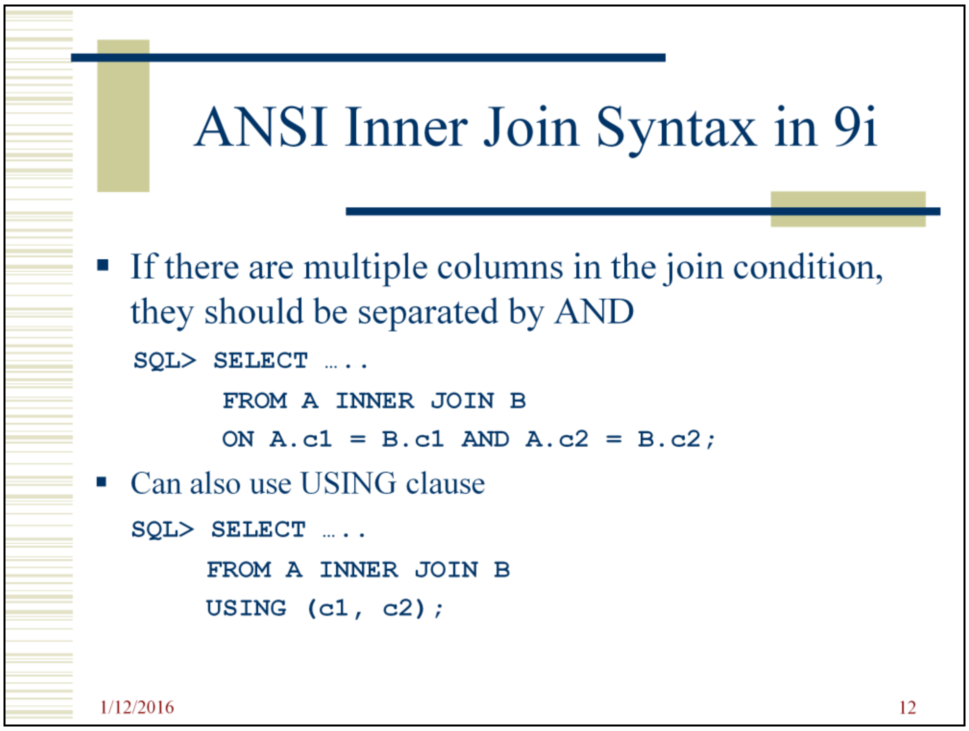


ANSI Inner Join Syntax in 9i

- Let's say "deptno" is in the SELECT list and the join condition, then you cannot use the column alias or qualify the column name with a table
- These two queries will generate an error message.

```
SQL> SELECT deptno department
        FROM emp e INNER JOIN dept d
        USING (department);
SQL> SELECT e.deptno
        FROM emp e INNER JOIN dept d
        USING (e.deptno);
```

1/12/2016 11



ANSI Inner Join Syntax in 9i

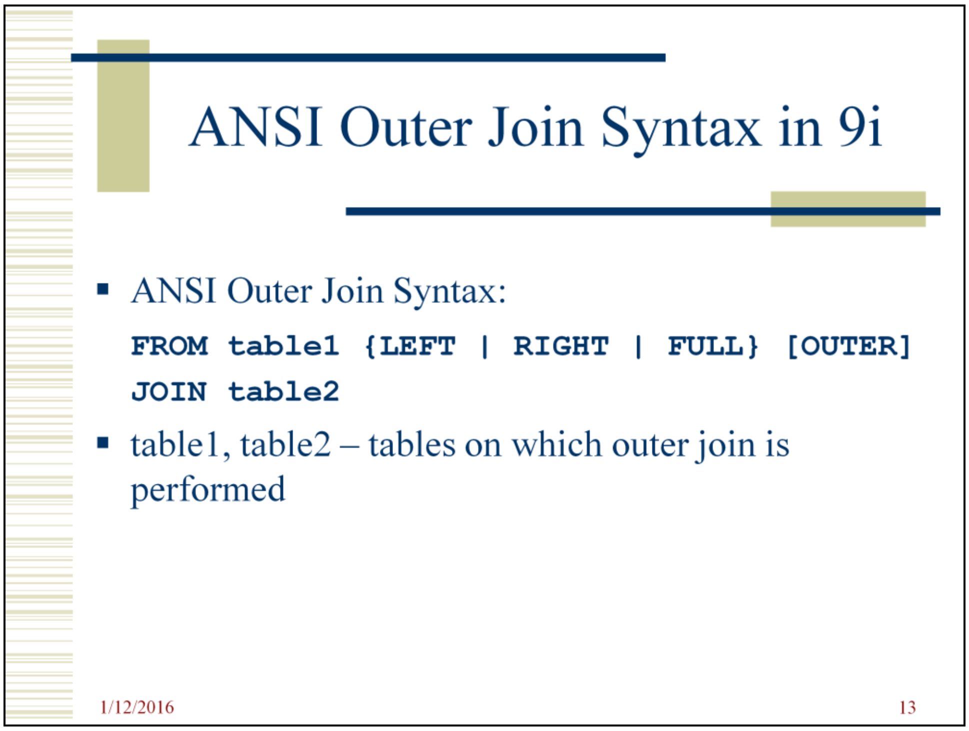
- If there are multiple columns in the join condition, they should be separated by AND

```
SQL> SELECT …..
        FROM A INNER JOIN B
        ON A.c1 = B.c1 AND A.c2 = B.c2;
```

- Can also use USING clause

```
SQL> SELECT …..
       FROM A INNER JOIN B
       USING (c1, c2);
```

1/12/2016 12

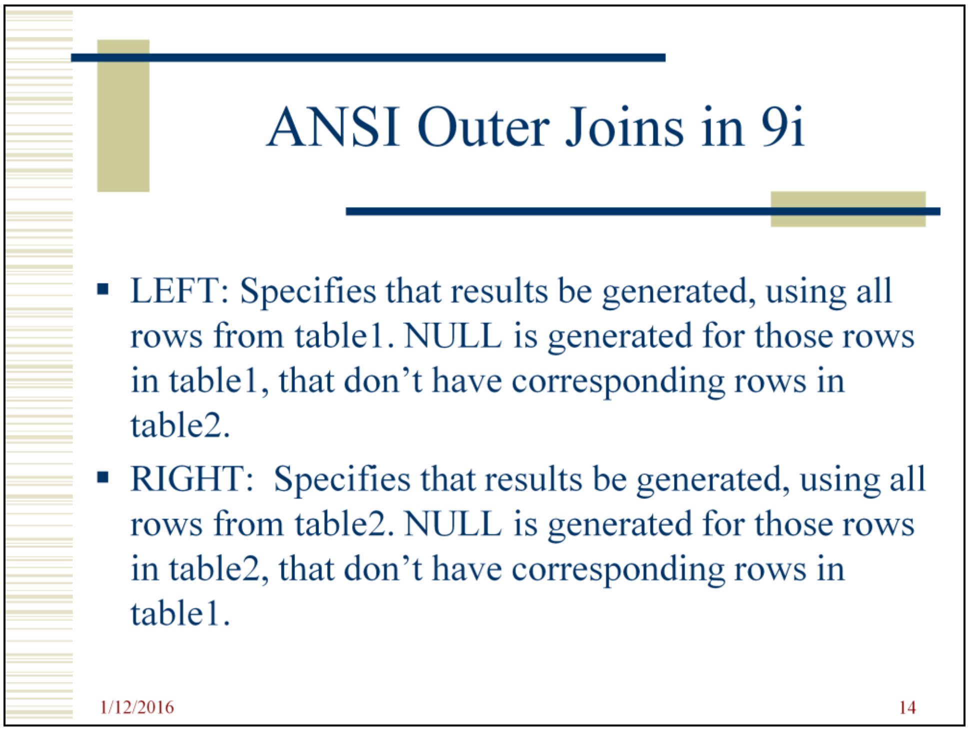

ANSI Outer Join Syntax in 9i

- ANSI Outer Join Syntax:

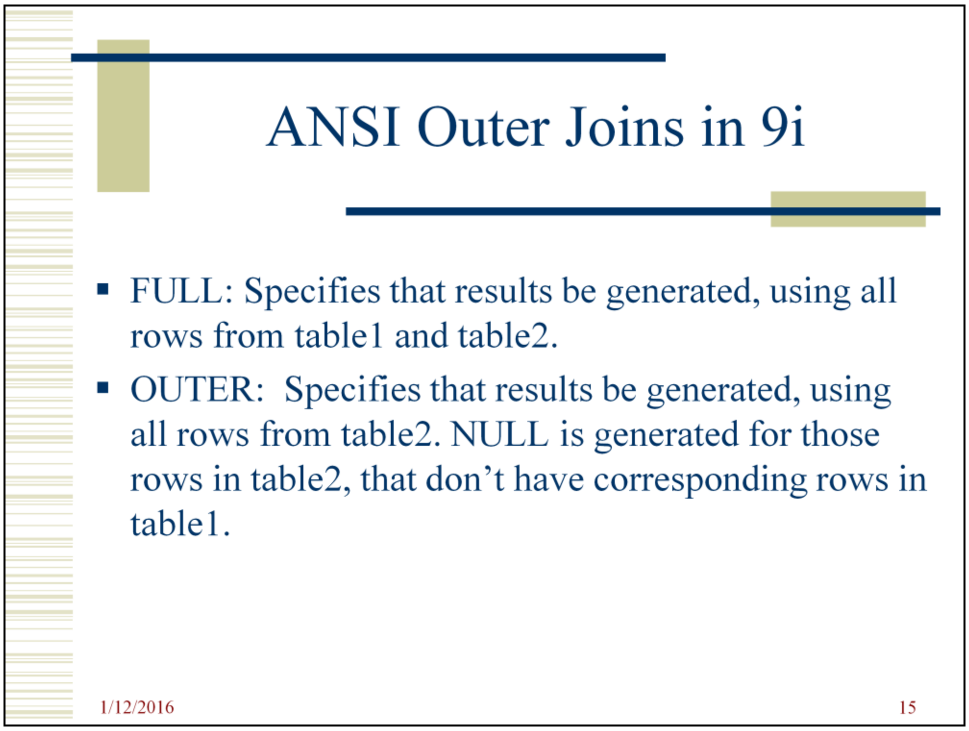
```
FROM table1 {LEFT | RIGHT | FULL} [OUTER]
JOIN table2
```

- table1, table2 – tables on which outer join is performed

1/12/2016 13


ANSI Outer Joins in 9i

- LEFT: Specifies that results be generated, using all rows from table1. NULL is generated for those rows in table1, that don't have corresponding rows in table2.
- RIGHT: Specifies that results be generated, using all rows from table2. NULL is generated for those rows in table2, that don't have corresponding rows in table1.

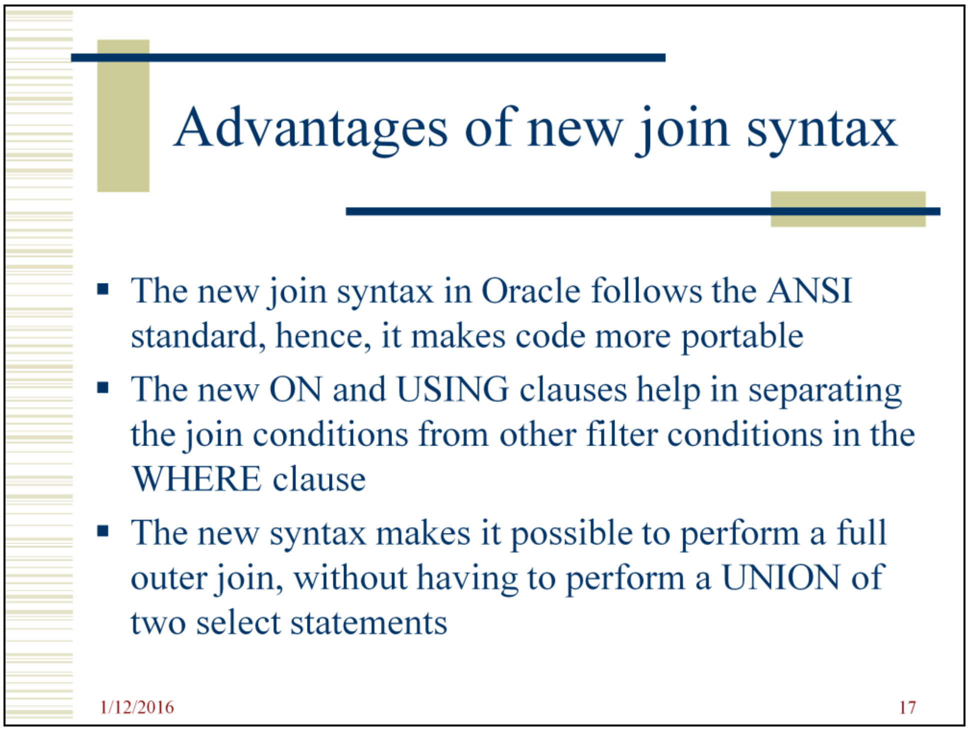


ANSI Outer Joins in 9i

- FULL: Specifies that results be generated, using all rows from table1 and table2.
- OUTER: Specifies that results be generated, using all rows from table2. NULL is generated for those rows in table2, that don't have corresponding rows in table1.

ANSI Outer Joins in 9i

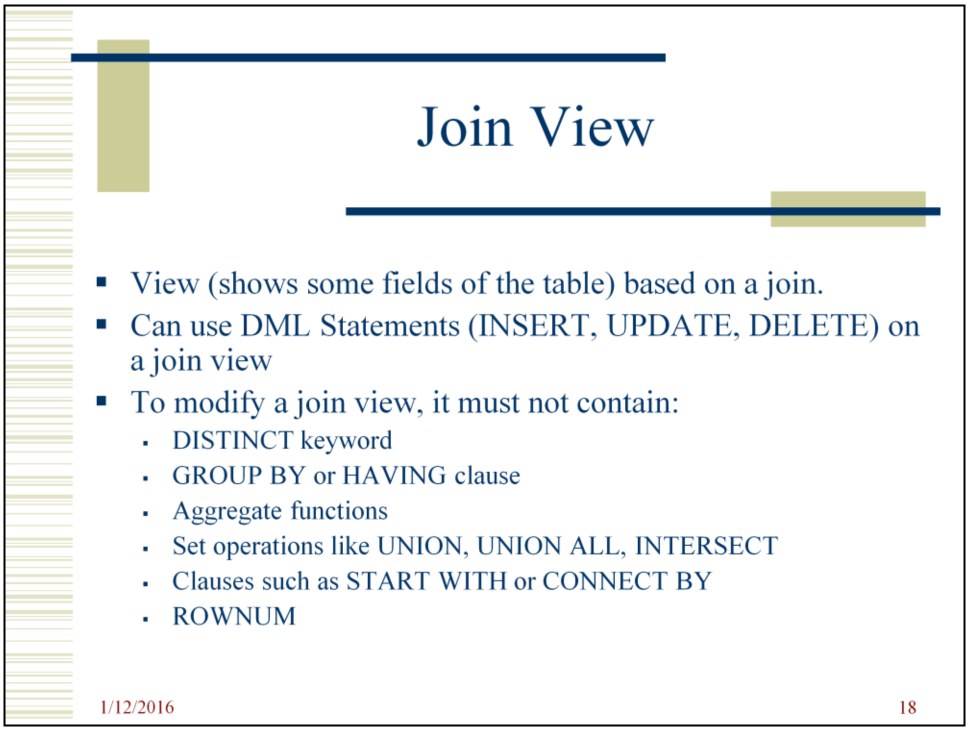
- FULL: Specifies that results be generated, using all rows from table1 and table2.
- OUTER: Specifies that results be generated, using all rows from table2. NULL is generated for those rows in table2, that don’t have corresponding rows in table1.



Advantages of new join syntax

- The new join syntax in Oracle follows the ANSI standard, hence, it makes code more portable
- The new ON and USING clauses help in separating the join conditions from other filter conditions in the WHERE clause
- The new syntax makes it possible to perform a full outer join, without having to perform a UNION of two select statements

1/12/2016 17



Join View

- View (shows some fields of the table) based on a join.
- Can use DML Statements (INSERT, UPDATE, DELETE) on a join view
- To modify a join view, it must not contain:
  - DISTINCT keyword
  - GROUP BY or HAVING clause
  - Aggregate functions
  - Set operations like UNION, UNION ALL, INTERSECT
  - Clauses such as START WITH or CONNECT BY
  - ROWNUM

1/12/2016 18

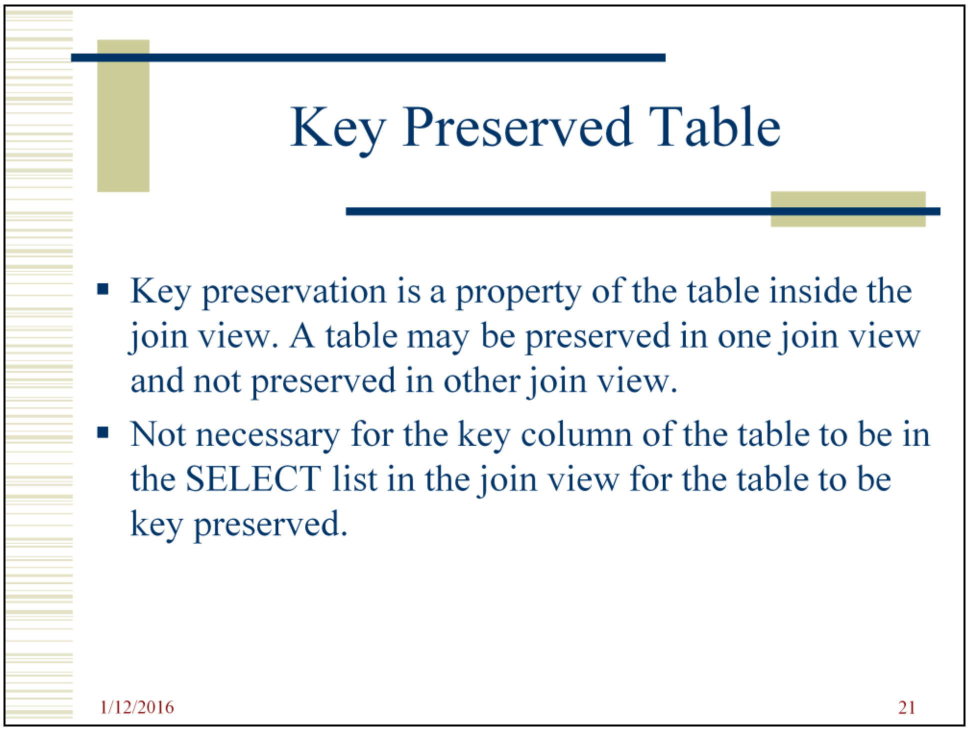
Creating a Join View

```
SQL> CREATE VIEW v_emp_dept AS
     SELECT e.empno, e.ename, d.name,
     e.hiredate
     FROM emp e, dept d
     WHERE e.deptno = d.deptno;
```

- In this query, every row has a unique empno. Therefore, emp table is a key preserved table in this view because its keys are preserved through the join.

Creating a Join View

```
SQL> SELECT * FROM v_emp_dept;
EMPNO  ENAME    NAME       HIREDATE
----- --------- ---------- ----------
7566   FORD     SALES      10-MAR-01
7567   SMITH    TRAINING   1-JAN-01
7568   MARTIN   ACCOUNTING 1-MAR-01
```



Key Preserved Table

- Key preservation is a property of the table inside the join view. A table may be preserved in one join view and not preserved in other join view.
- Not necessary for the key column of the table to be in the SELECT list in the join view for the table to be key preserved.

1/12/2016 21

INSERT Statement - Join View

- Can insert values into the key preserved table of the join view

```
SQL> INSERT into v_emp_dept (empno, ename,
hiredate) VALUES (7598, 'SMITH', '02-JAN-02');
```

- Not allowed to insert values into non-key preserved table of join view

```
SQL> INSERT into v_emp_dept (empno, ename,
hiredate, name) VALUES (7598, 'SMITH', '02-JAN-
02', 20); -- error
```

INSERT Statement - Join View

- Cannot insert values into the join view, if the join view is created using “WITH CHECK OPTION”

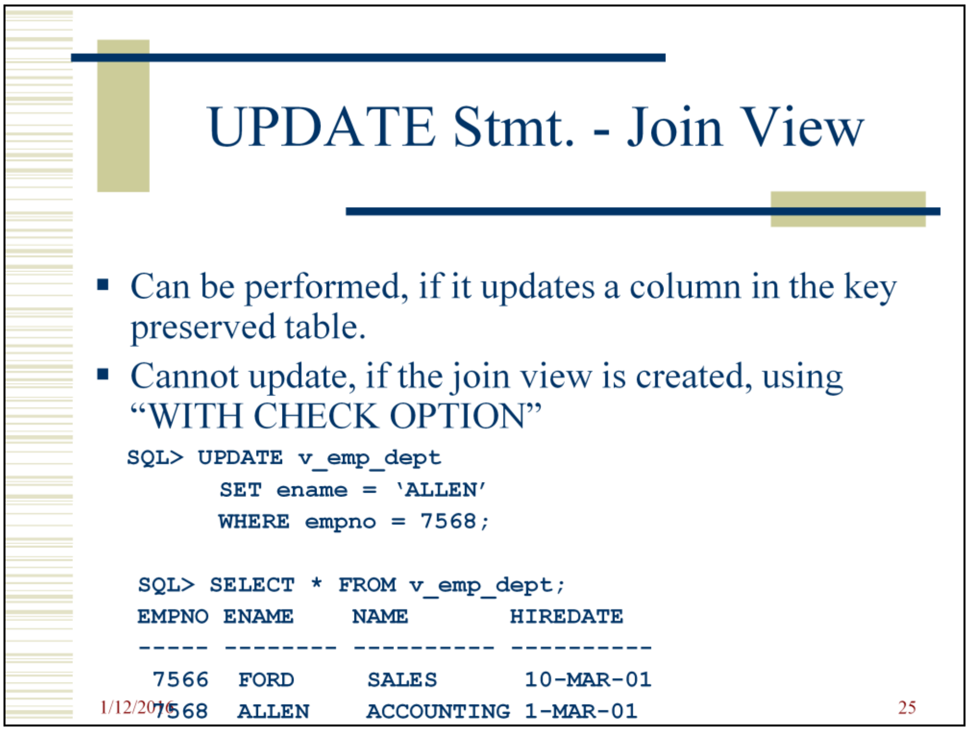
```
SQL> CREATE VIEW v_emp_dept AS
     SELECT e.empno, e.ename, d.name,
     e.hiredate FROM emp e, dept d
     WHERE e.deptno = d.deptno
     WITH CHECK OPTION;
```

DELETE Stmt. - Join View

▪Can be performed, if join view has one and only one key preserved table

```
SQL> DELETE FROM v_emp_dept
           WHERE empno = 7567;

SQL> SELECT * FROM v_emp_dept;
EMPNO ENAME     NAME       HIREDATE
----- --------- ---------- ----------
7566  FORD      SALES      10-MAR-01
7568  MARTIN    ACCOUNTING 1-MAR-01
```

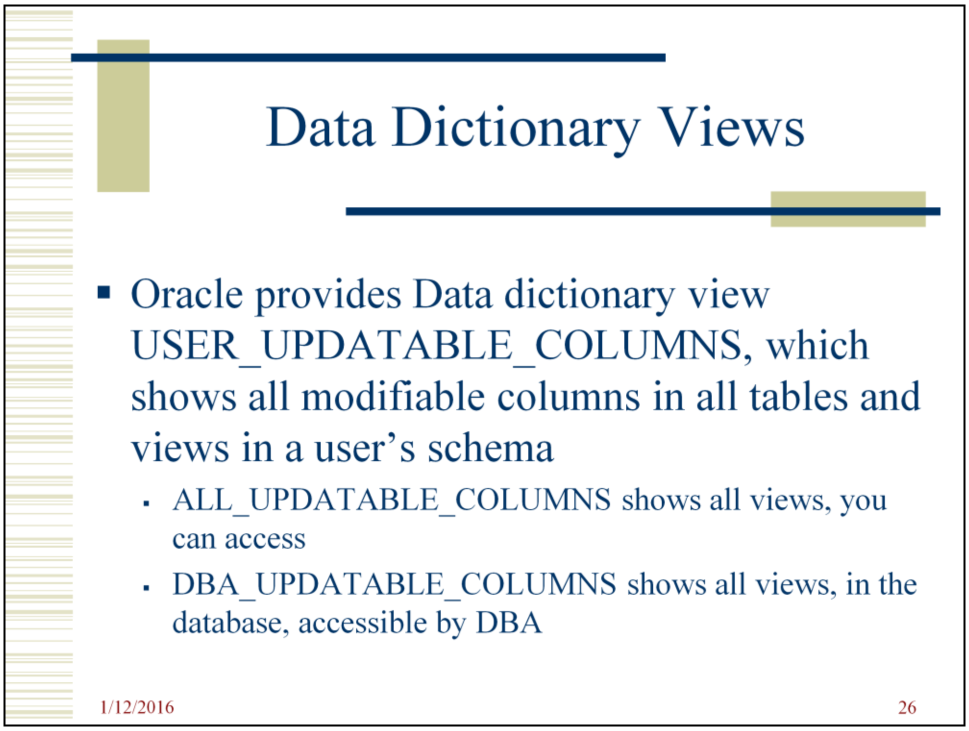


UPDATE Stmt. - Join View

- Can be performed, if it updates a column in the key preserved table.
- Cannot update, if the join view is created, using “WITH CHECK OPTION”

```
SQL> UPDATE v_emp_dept
        SET ename = 'ALLEN'
        WHERE empno = 7568;

SQL> SELECT * FROM v_emp_dept;
EMPNO ENAME     NAME       HIREDATE
----- --------- ---------- ----------
 7566  FORD      SALES      10-MAR-01
 7568  ALLEN     ACCOUNTING 1-MAR-01
```



Data Dictionary Views

- Oracle provides Data dictionary view USER_UPDATABLE_COLUMNS, which shows all modifiable columns in all tables and views in a user's schema
  - ALL_UPDATABLE_COLUMNS shows all views, you can access
  - DBA_UPDATABLE_COLUMNS shows all views, in the database, accessible by DBA

1/12/2016 26

Data Dictionary Views

- DESC USER_UPDATABLE_COLUMNS

```
Name                         Null?       Type
-------------------------    ----------  -----------
OWNER                        NOT NULL   VARCHAR2(30)
TABLE_NAME                   NOT_NULL   VARCHAR2(30)
COLUMN_NAME                  NOT_NULL   VARCHAR2(30)
UPDATABLE                               VARCHAR2(30)
INSERTABLE                              VARCHAR2(30)
DELETABLE                               VARCHAR2(30)
```