About the Presentations

- The presentations cover the objectives found in the opening of each chapter.
- All chapter objectives are listed in the beginning of each presentation.
- You may customize the presentations to fit your class needs.
- Some figures from the chapters are included. A complete set of images from the book can be found on the Instructor Resources disc.

COURSE TECHNOLOGY
CENGAGE Learning™

Oracle 12*c*: SQL

Chapter 4
Constraints

Objectives

- Explain the purpose of constraints in a table
- Distinguish among PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK, and NOT NULL constraints and the appropriate use for each constraint
- Understand how constraints can be created when creating a table or modifying an existing table
- Distinguish between creating constraints at the column level and table level

Objectives (continued)

- Create PRIMARY KEY constraints for a single column and a composite primary key
- Create a FOREIGN KEY constraint
- Create a UNIQUE constraint
- Create a CHECK constraint

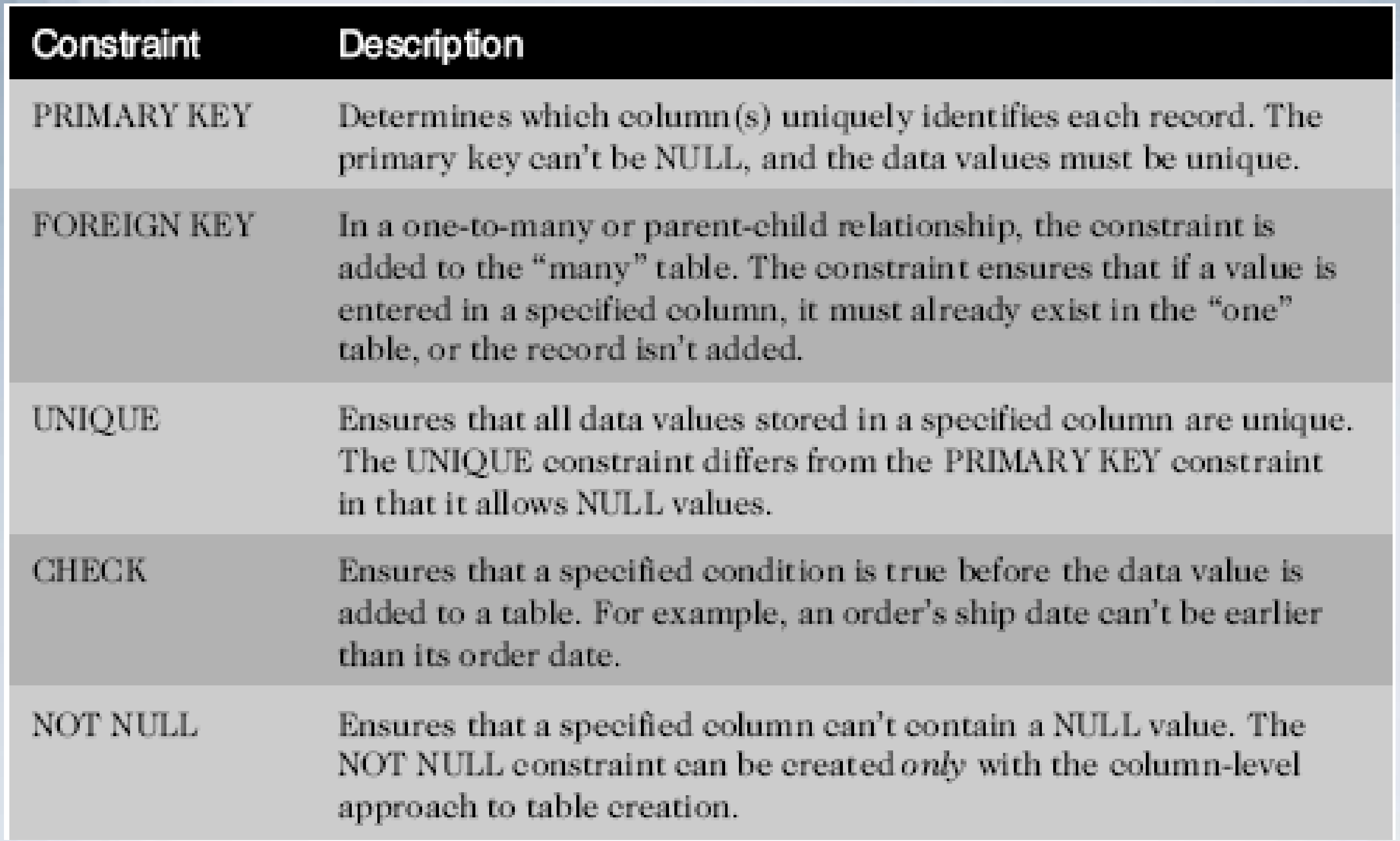
Objectives (continued)

- Create a NOT NULL constraint using the ALTER TABLE…MODIFY command
- Include constraints during table creation
- Use DISABLE and ENABLE commands
- Use the DROP command

Constraints

- Rules used to enforce business rules, practices, and policies
- Rules used to ensure accuracy and integrity of data

Constraint Types



Constraint	Description
PRIMARY KEY	Determines which column(s) uniquely identifies each record. The primary key can't be NULL, and the data values must be unique.
FOREIGN KEY	In a one-to-many or parent-child relationship, the constraint is added to the "many" table. The constraint ensures that if a value is entered in a specified column, it must already exist in the "one" table, or the record isn't added.
UNIQUE	Ensures that all data values stored in a specified column are unique. The UNIQUE constraint differs from the PRIMARY KEY constraint in that it allows NULL values.
CHECK	Ensures that a specified condition is true before the data value is added to a table. For example, an order's ship date can't be earlier than its order date.
NOT NULL	Ensures that a specified column can't contain a NULL value. The NOT NULL constraint can be created *only* with the column-level approach to table creation.

Creating Constraints

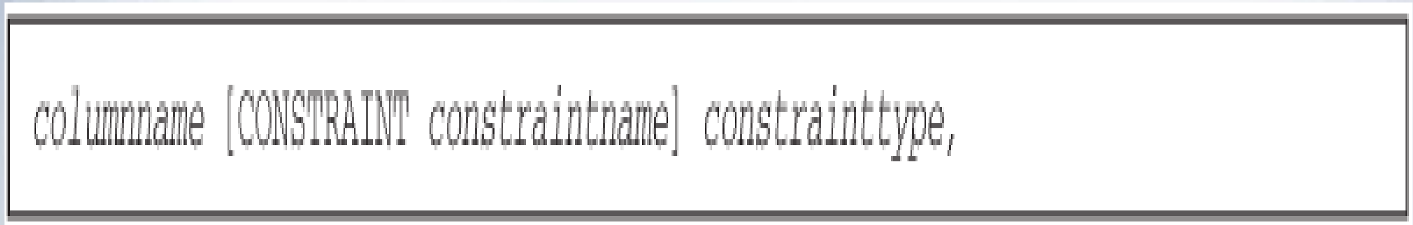
- Use the optional CONSTRAINT keyword during creation to assign a name
- Let the server name the constraint using the default format SYS_C*n*
- Informative names can assist in debugging

Creating Constraints (continued)

- When
  - During table creation
  - After table creation, by modifying the existing table
- How
  - Column level approach
  - Table level approach

Creating Constraints at the Column Level

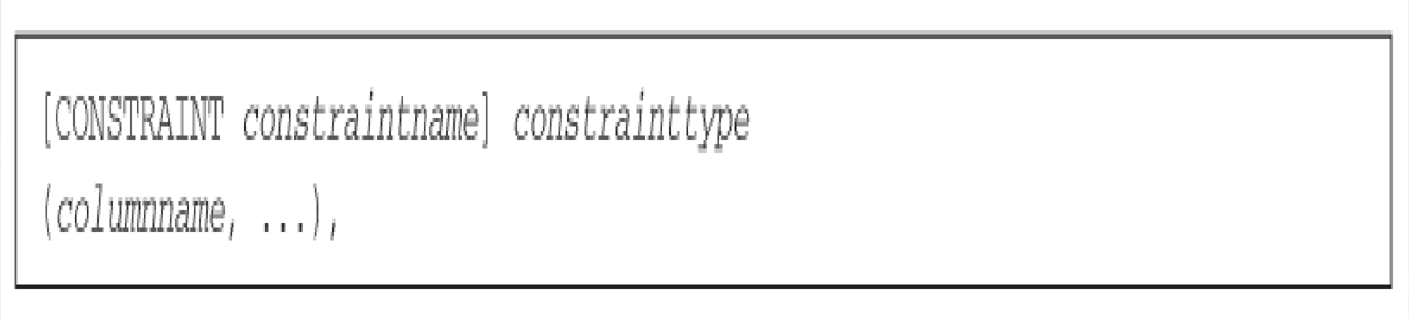
- If a constraint is being created at the column level, the constraint applies to the column specified



```
columnname [CONSTRAINT constraintname] constrainttype,
```

Creating Constraints at the Table Level

- Approach can be used to create any constraint type except NOT NULL
- Required if constraint is based on multiple columns



```
[CONSTRAINT constraintname] constrainttype
(columnname, ...),
```

Enforcement of Constraints

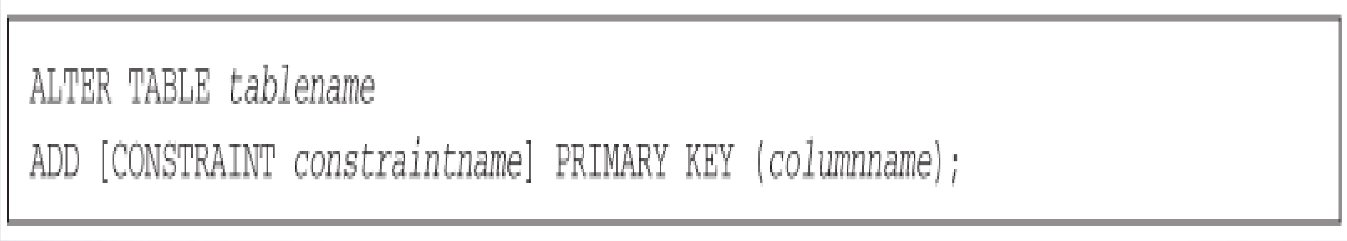
- All constraints are enforced at the table level
- If a data value violates a constraint, the entire row is rejected

Adding Constraints to Existing Tables

- Constraints are added to an existing table with the ALTER TABLE command
- Add a NOT NULL constraint using MODIFY clause
- All other constraints are added using ADD clause

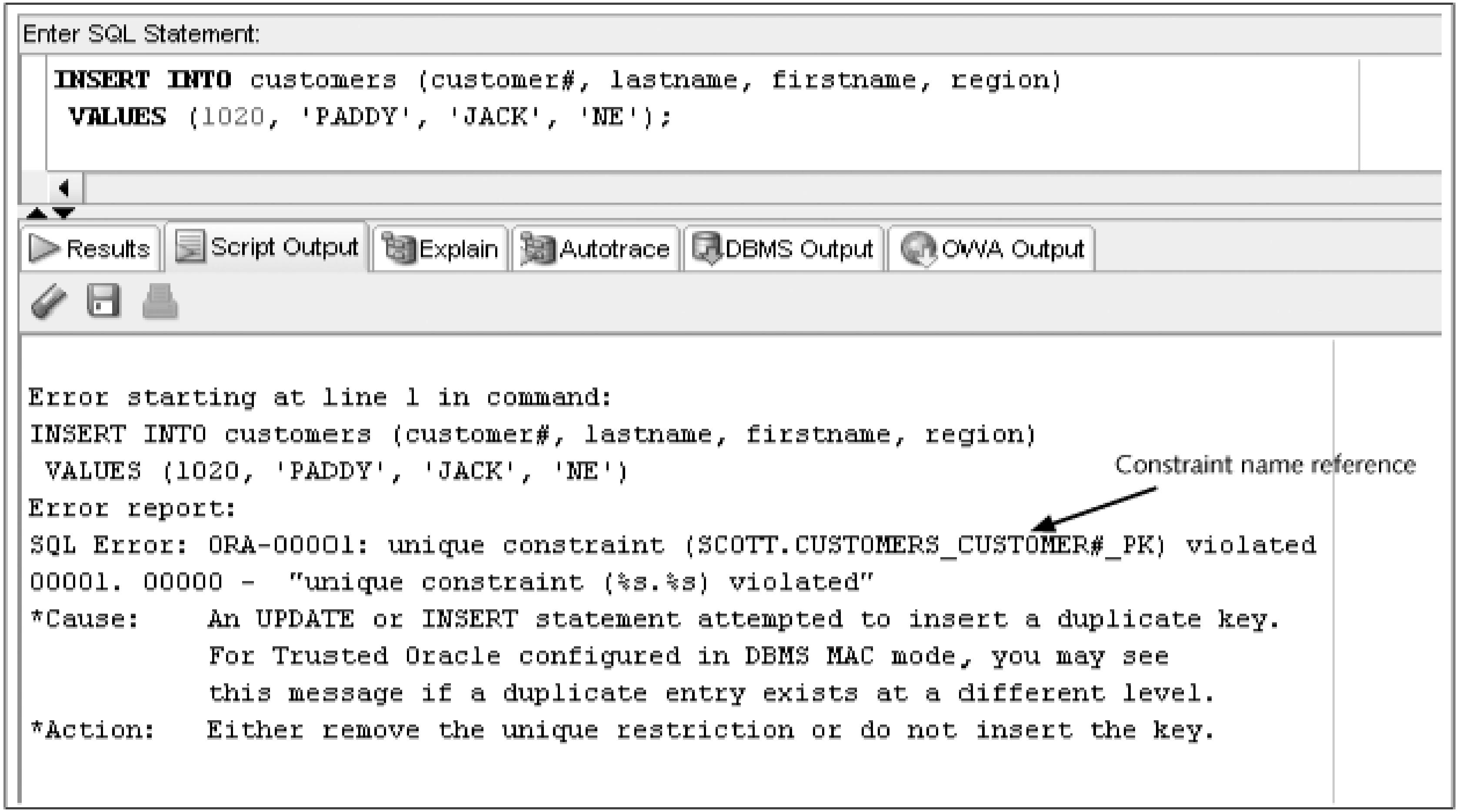
Using the PRIMARY KEY Constraint

- Ensures that columns do not contain duplicate or NULL values
- Only one per table is allowed



```
ALTER TABLE tablename
ADD [CONSTRAINT constraintname] PRIMARY KEY (columnname);
```

Constraint Checked with Data Input

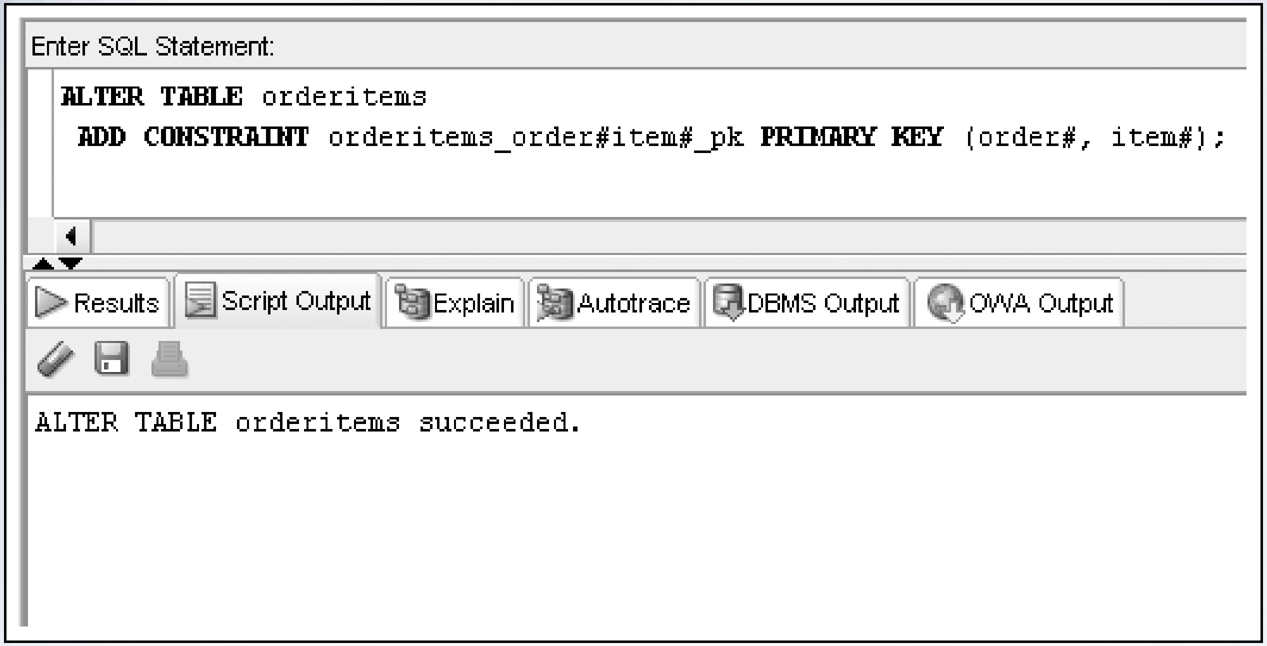


```
INSERT INTO customers (customer#, lastname, firstname, region)
 VALUES (1020, 'PADDY', 'JACK', 'NE');
```

```
Error starting at line 1 in command:
INSERT INTO customers (customer#, lastname, firstname, region)
 VALUES (1020, 'PADDY', 'JACK', 'NE')
Error report:
SQL Error: ORA-00001: unique constraint (SCOTT.CUSTOMERS_CUSTOMER#_PK) violated
00001. 00000 -  "unique constraint (%s.%s) violated"
*Cause:    An UPDATE or INSERT statement attempted to insert a duplicate key.
           For Trusted Oracle configured in DBMS MAC mode, you may see
           this message if a duplicate entry exists at a different level.
*Action:   Either remove the unique restriction or do not insert the key.
```

PRIMARY KEY Constraint for Composite Key

- List column names within parentheses separated by commas



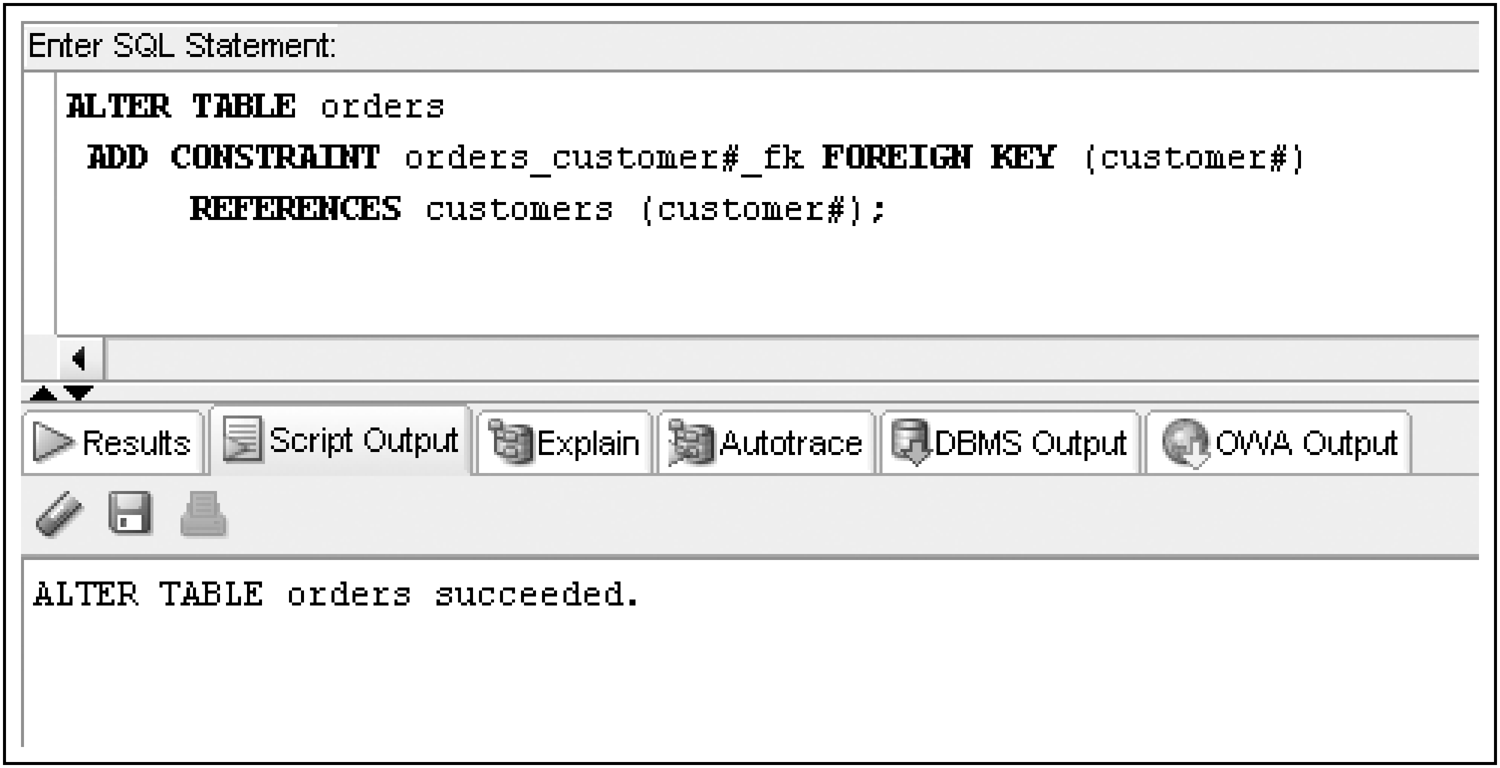
```
ALTER TABLE orderitems
 ADD CONSTRAINT orderitems_order#item#_pk PRIMARY KEY (order#, item#);
```

```
ALTER TABLE orderitems succeeded.
```

Using the FOREIGN KEY Constraint

- Requires a value to exist in the referenced column of another table
- NULL values are allowed
- Enforces referential integrity
- Maps to the PRIMARY KEY in parent table

FOREIGN KEY Constraint Example



```
ALTER TABLE orders
 ADD CONSTRAINT orders_customer#_fk FOREIGN KEY (customer#)
      REFERENCES customers (customer#);
```

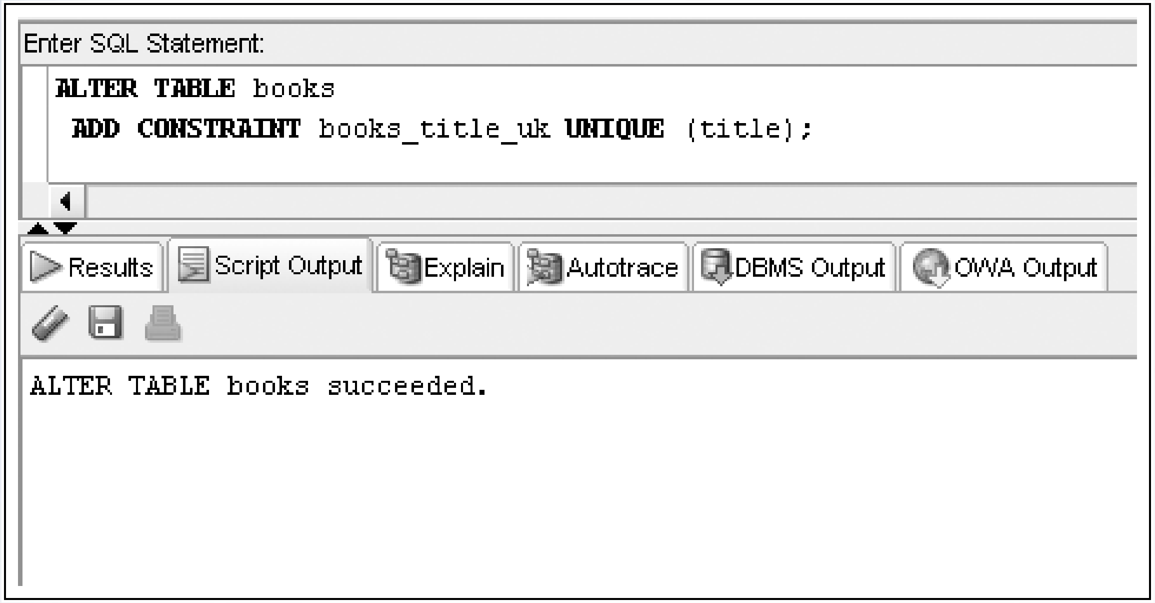
```
ALTER TABLE orders succeeded.
```

Deletion of Foreign Key Values

- You cannot delete a value in a parent table referenced by a row in a child table
- Use ON DELETE CASCADE keywords when creating FOREIGN KEY constraint – it automatically deletes a parent row when the row in a child table is deleted

Using the UNIQUE Constraint

- No duplicates are allowed in the referenced column
- NULL values are permitted

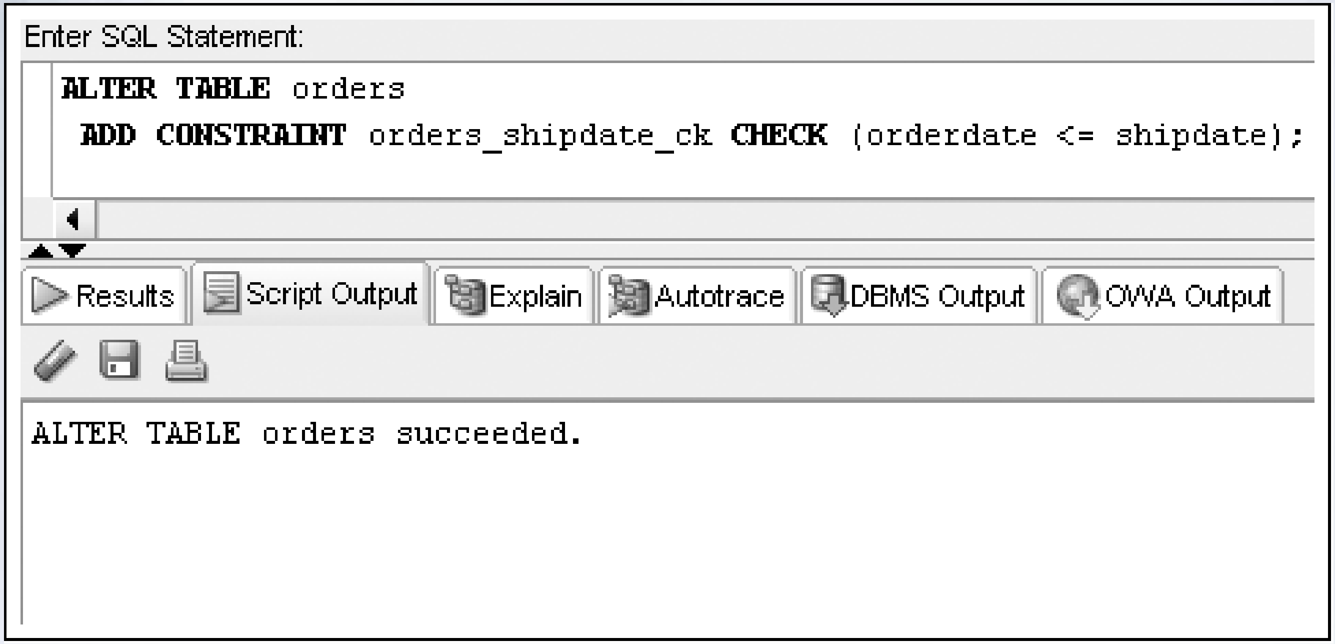


```
ALTER TABLE books
 ADD CONSTRAINT books_title_uk UNIQUE (title);
```

```
ALTER TABLE books succeeded.
```

Using the CHECK Constraint

- Updates and additions must meet specified condition



Enter SQL Statement:

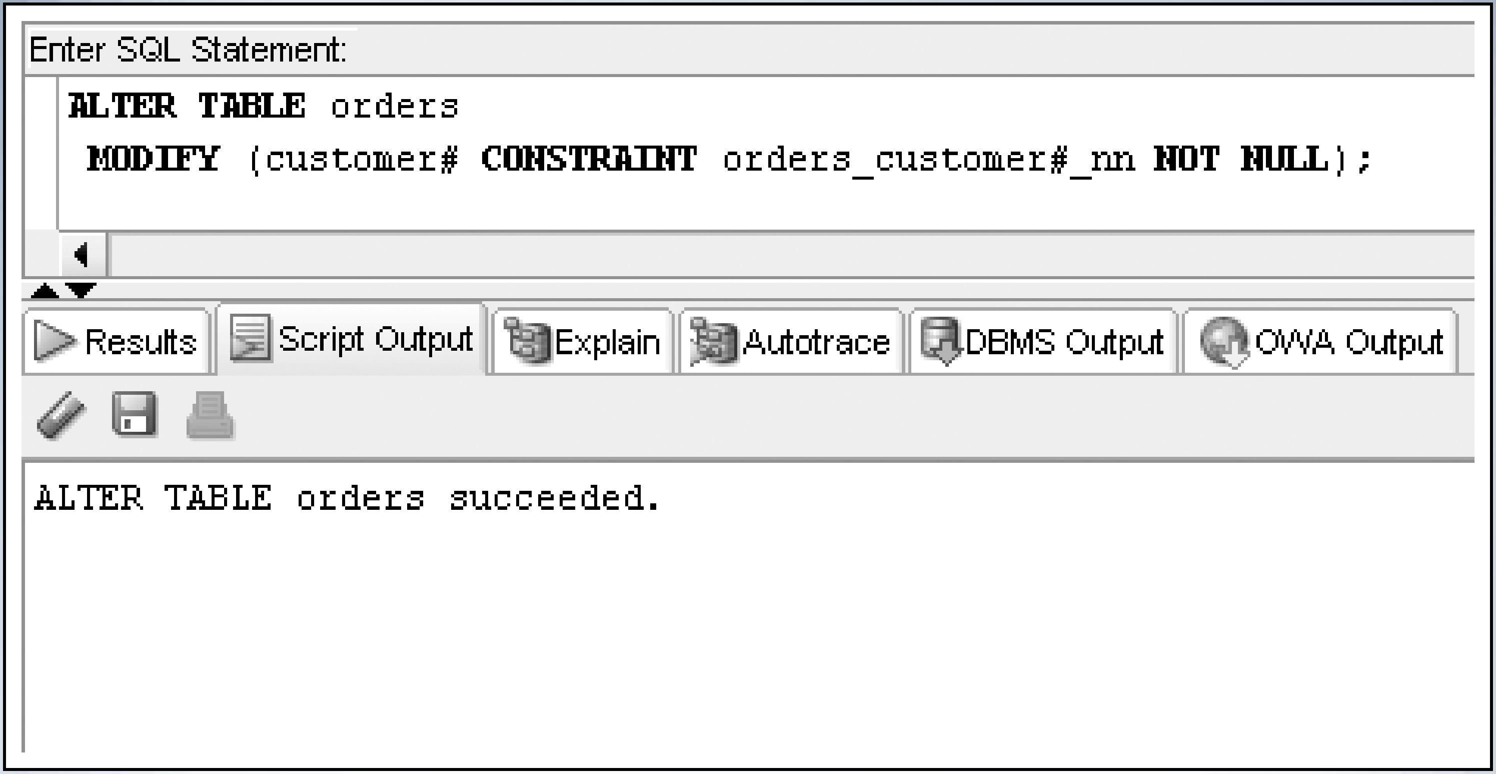
```
ALTER TABLE orders
 ADD CONSTRAINT orders_shipdate_ck CHECK (orderdate <= shipdate);
```

```
ALTER TABLE orders succeeded.
```

Using the NOT NULL Constraint

- The NOT NULL constraint is a special CHECK constraint with IS NOT NULL condition
- Can only be created at column level
- Included in output of DESCRIBE command
- Can only be added to an existing table using ALTER TABLE…MODIFY command

NOT NULL Constraint Example

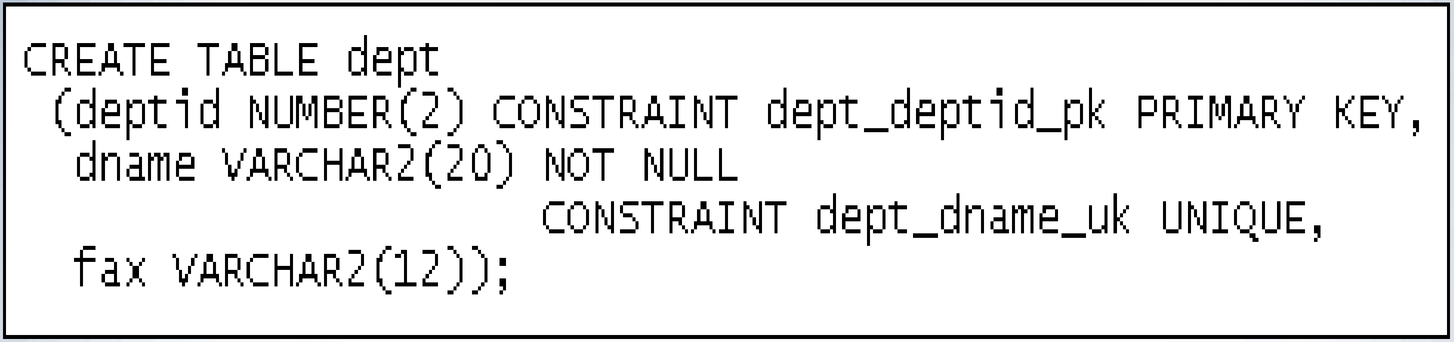


```
ALTER TABLE orders
 MODIFY (customer# CONSTRAINT orders_customer#_nn NOT NULL);
```

```
ALTER TABLE orders succeeded.
```

Including Constraints during Table Creation – Column Level

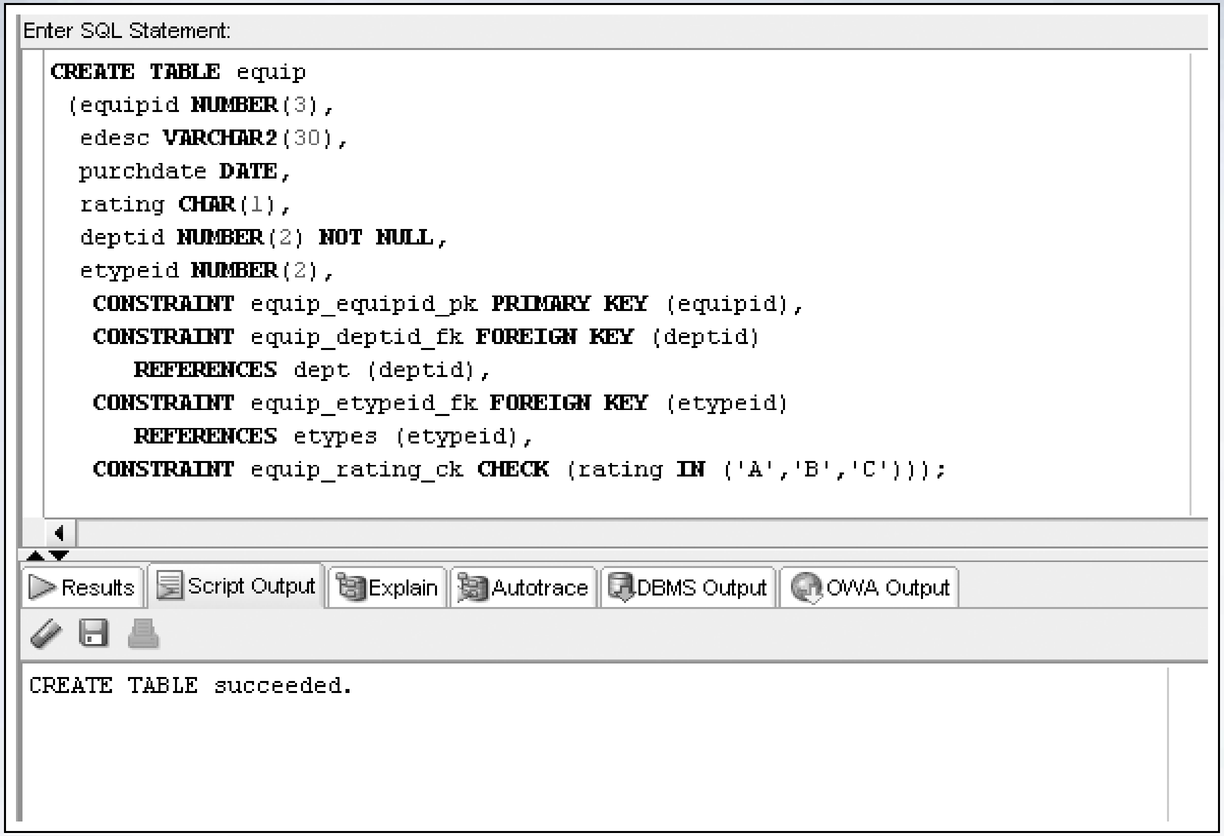
- Include in column definition



```
CREATE TABLE dept
 (deptid NUMBER(2) CONSTRAINT dept_deptid_pk PRIMARY KEY,
  dname VARCHAR2(20) NOT NULL
                     CONSTRAINT dept_dname_uk UNIQUE,
  fax VARCHAR2(12));
```

Including Constraints during Table Creation – Table Level

- Include at end of column list

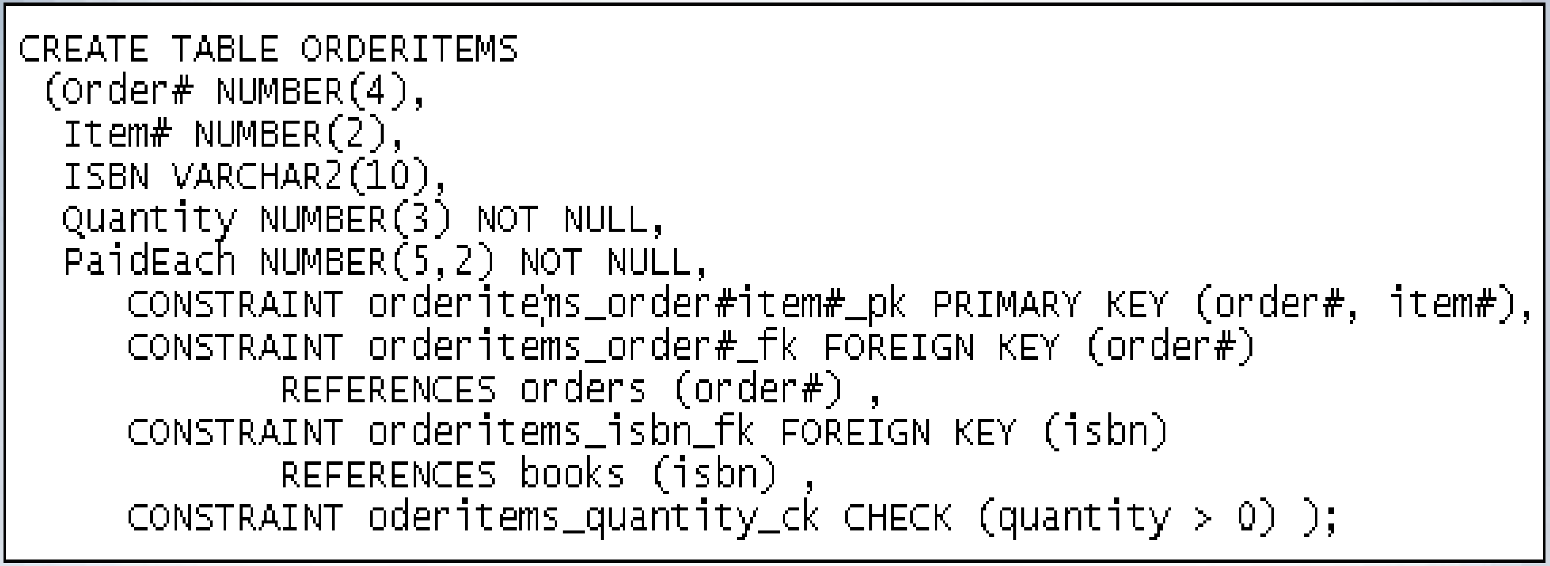


```
CREATE TABLE equip
 (equipid NUMBER(3),
  edesc VARCHAR2(30),
  purchdate DATE,
  rating CHAR(1),
  deptid NUMBER(2) NOT NULL,
  etypeid NUMBER(2),
   CONSTRAINT equip_equipid_pk PRIMARY KEY (equipid),
   CONSTRAINT equip_deptid_fk FOREIGN KEY (deptid)
      REFERENCES dept (deptid),
   CONSTRAINT equip_etypeid_fk FOREIGN KEY (etypeid)
      REFERENCES etypes (etypeid),
   CONSTRAINT equip_rating_ck CHECK (rating IN ('A','B','C')));
```

```
CREATE TABLE succeeded.
```

Multiple Constraints on a Single Column

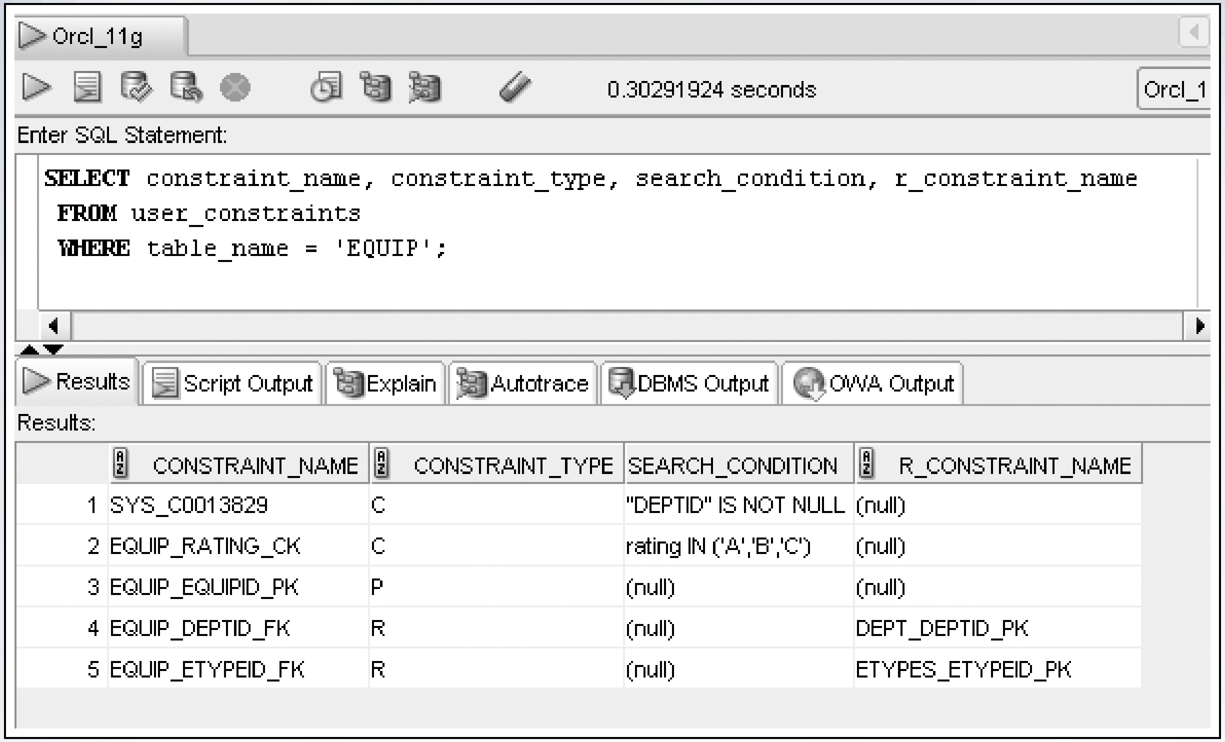
- A column may be included in multiple constraints
- The order# column is included in a primary key and a foreign key constraint



```
CREATE TABLE ORDERITEMS
 (Order# NUMBER(4),
  Item# NUMBER(2),
  ISBN VARCHAR2(10),
  Quantity NUMBER(3) NOT NULL,
  PaidEach NUMBER(5,2) NOT NULL,
     CONSTRAINT orderitems_order#item#_pk PRIMARY KEY (order#, item#),
     CONSTRAINT orderitems_order#_fk FOREIGN KEY (order#)
            REFERENCES orders (order#) ,
     CONSTRAINT orderitems_isbn_fk FOREIGN KEY (isbn)
            REFERENCES books (isbn) ,
     CONSTRAINT oderitems_quantity_ck CHECK (quantity > 0) );
```

Viewing Constraints – USER_CONSTRAINTS

- Display constraint listing for a specific table

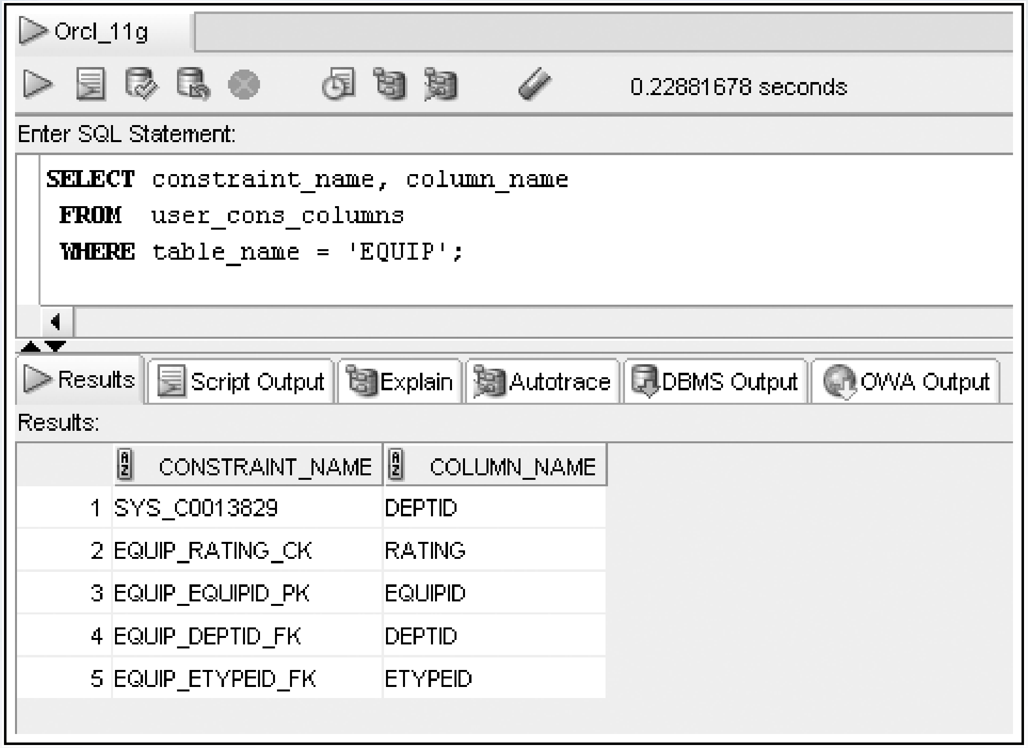


```
SELECT constraint_name, constraint_type, search_condition, r_constraint_name
 FROM user_constraints
 WHERE table_name = 'EQUIP';
```

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_CONSTRAINT_NAME
1	SYS_C0013829	C	"DEPTID" IS NOT NULL	(null)
2	EQUIP_RATING_CK	C	rating IN ('A','B','C')	(null)
3	EQUIP_EQUIPID_PK	P	(null)	(null)
4	EQUIP_DEPTID_FK	R	(null)	DEPT_DEPTID_PK
5	EQUIP_ETYPEID_FK	R	(null)	ETYPES_ETYPEID_PK

Viewing Constraints – USER_CONS_COLUMNS

- Display constraint listing by column

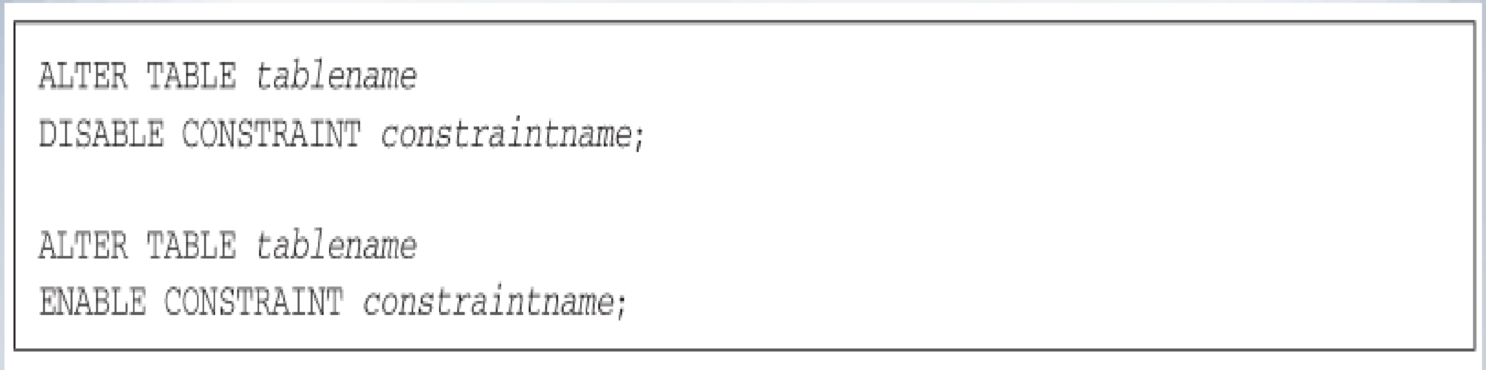


```
SELECT constraint_name, column_name
 FROM  user_cons_columns
 WHERE table_name = 'EQUIP';
```

	CONSTRAINT_NAME	COLUMN_NAME
1	SYS_C0013829	DEPTID
2	EQUIP_RATING_CK	RATING
3	EQUIP_EQUIPID_PK	EQUIPID
4	EQUIP_DEPTID_FK	DEPTID
5	EQUIP_ETYPEID_FK	ETYPEID

Using DISABLE/ENABLE

- Use DISABLE or ENABLE clause of ALTER TABLE command



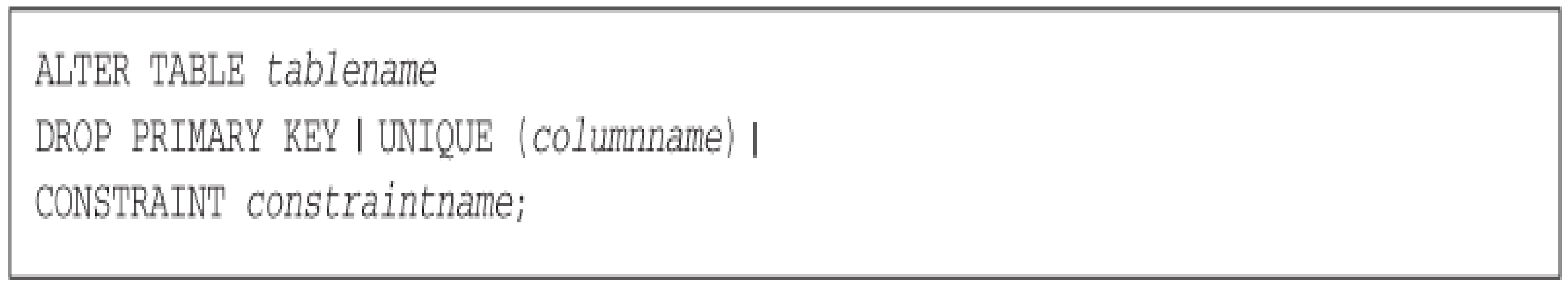
```
ALTER TABLE tablename
DISABLE CONSTRAINT constraintname;

ALTER TABLE tablename
ENABLE CONSTRAINT constraintname;
```

Dropping Constraints

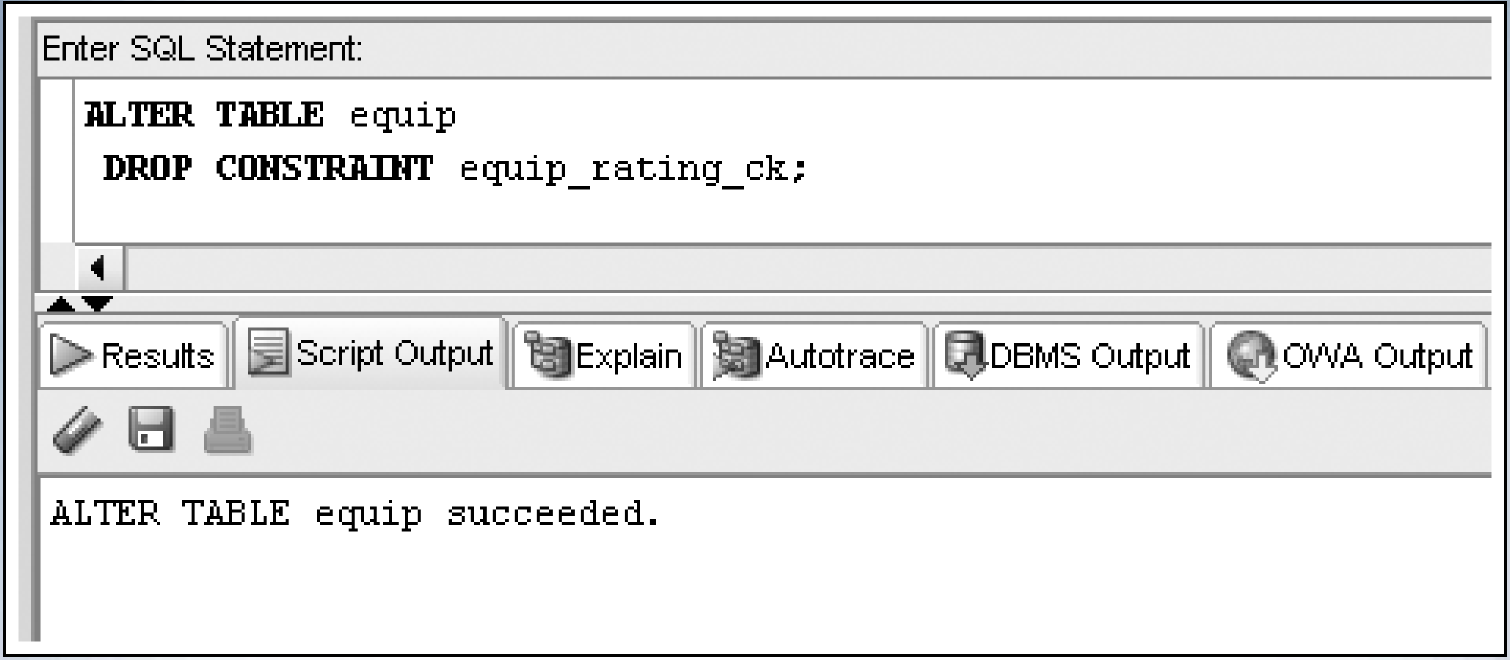
- Constraints cannot be modified; they must be dropped and recreated
- Actual syntax depends on type of constraint
  - PRIMARY KEY – just list type of constraint
  - UNIQUE – include column name
  - All others – reference constraint name

ALTER TABLE…DROP Syntax



```
ALTER TABLE tablename
DROP PRIMARY KEY | UNIQUE (columnname) |
CONSTRAINT constraintname;
```

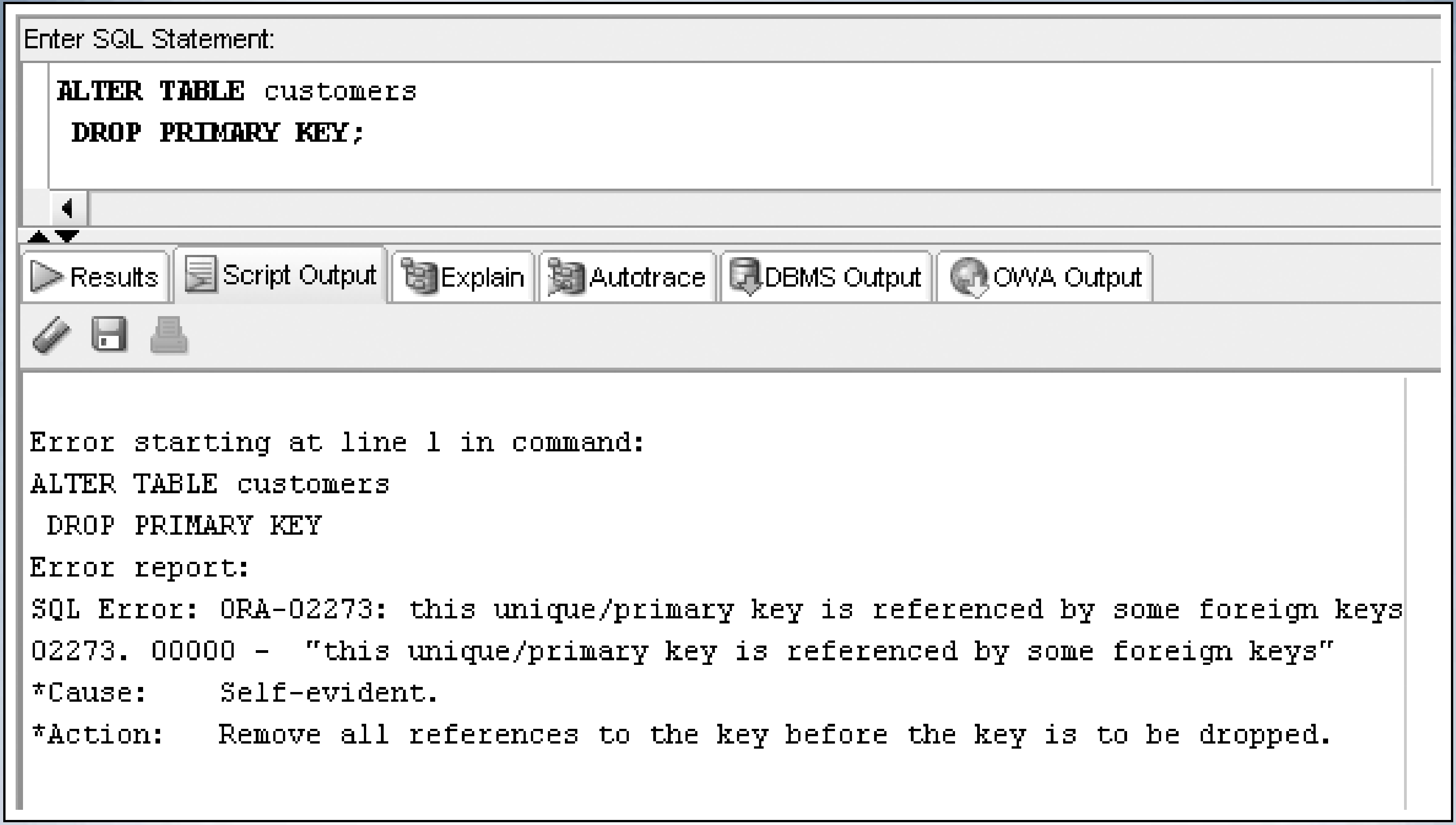
Drop Constraint Example



```
ALTER TABLE equip
 DROP CONSTRAINT equip_rating_ck;
```

```
ALTER TABLE equip succeeded.
```

Drop Constraint Example – Error



```
ALTER TABLE customers
 DROP PRIMARY KEY;
```

```
Error starting at line 1 in command:
ALTER TABLE customers
 DROP PRIMARY KEY
Error report:
SQL Error: ORA-02273: this unique/primary key is referenced by some foreign keys
02273. 00000 -  "this unique/primary key is referenced by some foreign keys"
*Cause:    Self-evident.
*Action:   Remove all references to the key before the key is to be dropped.
```

Summary

- A constraint is a rule that is applied to data being added to a table
  - The constraint represents business rules, policies, and/or procedures
  - Data violating the constraint is not added to the table
- A constraint can be included during table creation as part of the CREATE TABLE command or added to an existing table using the ALTER TABLE command

Summary (continued)

- A PRIMARY KEY constraint does not allow duplicate or NULL values in the designated column
- Only one PRIMARY KEY constraint is allowed in a table
- A FOREIGN KEY constraint requires that the column entry match a referenced column entry in the referenced table or be NULL
- A UNIQUE constraint is similar to a PRIMARY KEY constraint except it allows NULL values to be stored in the specified column
- A CHECK constraint ensures a value meets a specified condition

Summary (continued)

- A NOT NULL constraint ensures a value is provided for a column
- A constraint can be disabled or enabled using the ALTER TABLE command and the DISABLE and ENABLE keywords
- A constraint cannot be modified
  - To change a constraint, the constraint must first be dropped with the DROP command and then re-created
- USER_CONSTRAINTS and USER_CONS_COLUMNS data dictionary views provide information regarding constraints