#### COP 4710: Database Systems Spring 2004

#### -Day 18 – March 15, 2004 – Introduction to SQL – Part 2

Instructor :	Mark Llewellyn
	markl@cs.ucf.edu
	CC1 211, 823-2790
	http://www.cs.ucf.edu/courses/cop4710/spr2004

#### School of Electrical Engineering and Computer Science University of Central Florida



Page 1



#### An Example Database



COP 4710: Database Systems (Day 18)

Page 2

#### **Special Operators in SQL**

- ANSI standard SQL allows the use of special operators in conjunction with the WHERE clause. These special operators (see Day 17, page26) include:
  - BETWEEN Used to check whether an attribute value is within a range.
  - IS NULL Used to determine if an attribute value is null.
  - LIKE Used to match an attribute value to a string pattern. Many wildcard options are available.
  - **IN** Used to determine if an attribute value is within a list of values.
  - EXISTS Used to determine if a subquery returns an empty set or not.



### The BETWEEN Special Operator

Suppose that we want to see a listing for all products whose prices are between \$50 and \$100. The BETWEEN operator can be used for this query expression.

```
SELECT *
```

FROM PRODUCT

WHERE P\_PRICE BETWEEN 50.00 AND 100.00;

If your RDBMS does not support BETWEEN you would need to express this query as:

```
SELECT *
FROM PRODUCT
WHERE P_PRICE > 50.00 AND P_PRICE < 100.00;
```

COP 4710: Database Systems (Day 18)

Page 4

## The IS NULL Special Operator

Suppose that we want to see a listing for all products that do not currently have a vendor assigned, i.e., V\_CODE = null.
 The null entries could be found with the following query expression.

SELECT P\_CODE, P\_DESCRIPT, V\_CODE FROM PRODUCT WHERE V\_CODE IS NULL;

• NOTE: SQL uses a special operator for testing for nulls. You cannot use a condition such as V\_CODE = NULL. The reason is that NULL is technically not a "value", but a special property of an attribute that represents precisely the absence of any value at all.





### The LIKE Special Operator

- The LIKE special operator is used in conjunction with wildcards to find patterns within string attributes.
- Standard SQL allows you to use the percent sign (%) and underscore (\_) wildcard characters to make matches when the entire string is not known.

% means any and all following characters are eligible.

'M%' includes Mark, Marci, M-234x, etc.

\_ means any one character may be substituted for the underscore.

'\_07-345-887\_' includes 407-345-8871, 007-345-8875

• Note: Access uses \* instead of % and ? instead of \_. Oracle searches are case-sensitive, Access searches are not.



#### The LIKE Special Operator (cont.)

• Suppose that we would like to find all the VENDOR rows for contacts whose last names begin with Smith.

SELECT V\_NAME, V\_CONTACT, V\_AREACODE, V\_PHONE FROM VENDOR

WHERE V\_CONTACT LIKE 'Smith%';

Microsoft Access	x								
<u>File Edit View Insert Query Tools Window H</u> elp		Micro	osoft A	ccess					
	>>	e <u>E</u>	dit <u>V</u> iev	v <u>I</u> nsert I	Format	<u>R</u> ecords <u>T</u> o	ools <u>W</u> indow <u>H</u> e	lp	
		<u>-</u>		i 🕽 🗟 🗳	× 1	🖻 🛍 🚿	🗠 🍓 👌	XI 79 7	a⊽ 🗚 类
This Communication					100		- 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10		
SELECT VENDOR V, NAME, VENDOR V, CONTACT, VENDOR V, AREACODE, VENDOR V, PHONE		Ľ	📰 Like	• Query : 9	ielect (	Juery			
FROM VENDOR				V_NAME	V	CONTACT	V_AREACODE	V_PHONE	
WHERE (((VENDOR,V_CONTACT) Like Smith*));	_		► Bry	son, Inc.	Sr	nithson	615	223-3234	
			Dor	ne Supply	Sr	nith	901	678-1419	
			B&	K, Inc.	Sr	nith	904	227-0093	
🛄 Outof Warranty Query 📰 qryFig6-15			*						
🖽 Forms 🔣 gryCreateTablePRODUCT 📰 gryFig6-16	-				_				
			Record	<u>;</u> <u>।</u> ∢ ∢	1	1 + +	IF* of 3		<b>-</b>
Ready	1							1	Þ
Access wildcard	Da	atash	ieet View	E.				NUM	

COP 4710: Database Systems (Day 18)

Page 7

#### The IN Special Operator

- Many queries that would seem to require the use of the logical OR operator can be more easily handled with the help of the special operator IN.
- For example the query:

SELECT \*

FROM PRODUCT

WHERE V\_CODE = 21344 OR V\_CODE = 24288;

can be handled more efficiently with:

SELECT \*

FROM PRODUCT

WHERE V\_CODE IN (21344, 24288);

COP 4710: Database Systems (Day 18)

Page 8

### The IN Special Operator (cont.)

- The IN operator is especially valuable when it is used in conjunction with subqueries.
- For example, suppose you want to list the V\_CODE and V\_NAME of only those vendors that actually provide products. In this case, you could use a subquery within the IN operator to automatically generate the value list. The query expression would be:

```
SELECT V_CODE, V_NAME
FROM VENDOR
WHERE V_CODE IN ( SELECT V_CODE
FROM PRODUCT);
```

• We'll look more closely at the IN operator later when we deal more in depth with subqueries.



Page 9

#### The EXISTS Special Operator

- The EXISTS operator can be sued whenever there is a requirement to execute a command based on the result of another query. That is, if a subquery returns any rows, then run the main query, otherwise, don't. We'll see this operator in more detail when we look at subqueries in more depth.
- For example, suppose we want a listing of vendors, but only if there are products to order. The following query will accomplish our task.

```
SELECT *

FROM VENDOR

WHERE EXISTS (SELECT *

FROM PRODUCT

WHERE P_ONHAND <= P_MIN);

COP 4710: Database Systems (Day 18) Page 10 Mark Llewellyn
```

#### Advanced DDL Commands in SQL

- In the notes for Day 17 we covered the basics of table creation in SQL, including inserting, deleting, and updating rows in tables.
- Now we'll look at the SQL commands to alter a table's structure, by changing attribute characteristics and by adding columns. We'll also see how to add data into new or modified columns. We'll also see how to copy tables or parts of tables as well as deleting entire tables.



### The ALTER Command

- All changes in the table structure are made using the ALTER command, followed by a keyword that produces the specific change you want to make.
- There are three options for the keyword: ADD, MODIFY, and DROP.
  - ADD enables you to add a column to a table.
  - MODIFY enables you to change a column's characteristics.
  - DROP allows you to delete a column from a table Most RDBMSs do not allow you to delete a column from a table, unless that column does not contain any values, because such an action may delete crucial data that are used by other tables.





### The ALTER Command (cont.)

The basic syntax of the ALTER command is:

ALTER TABLE tablename {ADD | MODIFY} ( columnname datatype [ {ADD | MODIFY} columnname datatype] );

The ALTER TABLE command can also be used to add table constraints. In that case the syntax would be as follows:

ALTER TABLE tablename

ADD constraint [ADD constraint];

COP 4710: Database Systems (Day 18)



### The ALTER Command (cont.)

• You can also use the ALTER TABLE command to remove a column or table constraint. The basic syntax of this form of the ALTER command is:

ALTER TABLE tablename DROP { PRIMARY KEY | COLUMN columnname / CONSTRAINT constraintname } ;

Notice that when removing a constraint, you need to specify the name given to the constraint. This is one reason why it is always advisable to name the constraints in the CREATE TABLE or ALTER TABLE statements.





# Changing a Column's Data Type

- You can also use the ALTER TABLE command to change the data type of a column.
- For example, suppose we wanted to change the data type of V\_CODE attribute in the PRODUCT table from integer to character. The following SQL command would accomplish this task:

#### ALTER TABLE PRODUCT

```
MODFIY (V_CODE CHAR(5));
```

Most RDBMSs doe not allow you to change the data type of an attribute unless the column to be changed is empty. For example, if we issued the SQL command above on the database we've been using and error message would be generated because the V\_CODE column already contains data. The reason for the error is simple. The V\_CODE attribute in PRODUCT references the V\_CODE attribute in VENDOR. If the data types don't match, there is a referential integrity violation, thus triggering the error message. If the V\_CODE column in PRODUCT was empty **and** the foreign key reference was not specified during the PRODUCT table's creation, then the SQL command above would execute properly.

#### Changing a Column's Data Characteristics

- If the column to be changes already contains data, you can make any changes in the column's characteristics if those changes do not alter the data type.
- For example, if we wanted to increase the width of the P\_PRICE column from 8 digits to 9 digits, we would need to issue the following command:

#### ALTER TABLE PRODUCT

MODFIY (P\_PRICE DECIMAL(9,2));

• Many RDBMSs place restrictions on the types of changes to column characteristics that can occur. For example, Oracle will allow you to widen a column but will not allow you to narrow a column.





#### Adding a Column to a Table

- You can alter an existing table by adding one or more columns.
- For example, suppose that we want to add a column to the PRODUCT table called P\_SALECODE that will allow us to decide if products that have been in inventory for a certain length of time should be placed on sale. Let's assume that the P\_SALECODE entries will be 1, 2, or 3, and we're not going to do arithmetic on the column so we'll make it a character.

ALTER TABLE PRODUCT

ADD (P\_SALECODE CHAR(1));

COP 4710: Database Systems (Day 18)



#### Adding a Column to a Table (cont.)

- When adding a column, be careful not to include the NOT NULL clause for the new column. Doing so will cause an error message because when adding a new column to a table that already has rows, the existing rows will default to a value of null for the new column. Therefore, it is not possible to add the NOT NULL clause for this new column.
- You can add the NOT NULL clause to the table structure after all the data for the new column has been entered and the column no longer contains nulls.



### Deleting a Column from a Table

- Occasionally, you may want to modify a table by deleting a column.
- Suppose that we want to delete the V\_ORDER attribute from the VENDOR table. To accomplish this task we would use the following SQL command:

ALTER TABLE VENDOR

DROP COLUMN V\_ORDER;

As before, some RDBMSs will impose restrictions on the deletion of an attribute. For example, most RDBMSs will not allow you to delete attributes that are involved in foreign key relationships, nor may you delete an attribute of a table that contains only that one attribute.





#### **Advanced Data Updates**

- To make data entries in an existing row's columns, SQL uses the UPDATE command. The UPDATE command only updates data in existing rows.
- For example, to enter the P\_SALECODE value of '2' in the fourth row of the PRODUCT\_2 table, we need to use the UPDATE command in conjunction with the primary key value for that row. The following command produces this effect with the before and after results shown on the next page.

```
UPDATE PRODUCT_2
```

```
SET P_SALECODE = '2'
```

```
WHERE P_CODE = '1546-QQ2';
```

COP 4710: Database Systems (Day 18)



P\_SALECODE value before update

							2	
			* 7 🖽	U 1994			₩, •	
PRODUC	T_2 : Table							-02
P_CODE	P_DESCRIPT	P_INDATE	P_ONHAND	P_MIN	P_PRICE	P_DISCOUNT	V_CODE	P_SALECODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-03	8	5	\$109.99	0.00	25595	
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-03	32	15	\$14.99	0.05	21344	
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-03	18	12	\$17.49	0.00	21344	
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-04	15	8	\$39.95	0.00	23119	•
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-04	23	5	\$43.99	0.00	23119	
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-03	8	5	\$109.92	0.05	24288	1
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-03	6	5	\$99.87	0.05	24288	1
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-04	12	5	\$38.95	0.05	25595	
23109-HB	Claw hammer	20-Jan-04	23	10	\$9.95	0.10	21225	
23114-AA	Sledge hammer, 12 lb.	02-Jan-04	8	5	\$14.40	0.05		
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-03	43	20	\$4.99	0.00	21344	
89-WRE-G	Hicut chain saw, 16 in.	07-Feb-04	11	5	\$256.99	0.05	24288	
PVC23DR1	PVC pipe, 3.5-in., 8-ft	20-Feb-04	188	75	\$5.87	0.00		
CM 19077	1.25 in metal screw, 25	01_Mer_04	172	75	99 38	0.00	21225	

COP 4710: Database Systems (Day 18)

Page 21

#### P\_SALECODE value after update

-		nisert Pomiat <u>R</u> ecords 100s . • R ∰2 X Ba R ≪ ∽		🗸 🏹	V .	► • • • • • • • • • • • • • • • • • • •	Fi ⁄ -	2	
		_2 : Table	(8) <b>2 V</b> A						
-	P_CODE	P_DESCRIPT	P_INDATE	P_ONHAND	P_MIN	P_PRICE	P_DISCOUNT	V_CODE	P_SALECODE
	11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-03	8	5	\$109.99	0.00	25595	
	13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-03	32	15	\$14.99	0.05	21344	
	14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-03	18	12	\$17.49	0.00	21344	
Ø	1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-04	15	8	\$39.95	0.00	23119	2 ┥
	1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-04	23	5	\$43.99	0.00	23119	
	2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-03	8	5	\$109.92	0.05	24288	1
	2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-03	6	5	\$99.87	0.05	24288	1
	2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-04	12	5	\$38.95	0.05	25595	
	23109-HB	Claw hammer	20-Jan-04	23	10	\$9.95	0.10	21225	
	23114-AA	Sledge hammer, 12 lb.	02-Jan-04	8	5	\$14.40	0.05		
	54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-03	43	20	\$4.99	0.00	21344	
	89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-04	11	5	\$256.99	0.05	24288	
	PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-04	188	75	\$5.87	0.00		
	SM-18277	1.25-in. metal screw, 25	01-Mar-04	172	75	\$6.99	0.00	21225	
_		12.22 2. 22							

COP 4710: Database Systems (Day 18)

Page 22

- Subsequent data can be entered in the same fashion, defining each entry location by its primary key (P\_CODE) and its column location (P\_SALECODE).
- The table on the previous page, also reflects the execution of the following SQL command:

**UPDATE PRODUCT\_2** 

SET P\_SALECODE = '1'

WHERE P\_CODE IN ('2232/QWE', '2232/QTY');

COP 4710: Database Systems (Day 18)

Page 23

- Although the UPDATE sequences we've just seen allow you to enter values into specified table cells, the process is very cumbersome. Fortunately, there may be better method.
- If a relationship can be established between entries and the existing columns, that relationship may be used to assign values to their appropriate slots.
- For example, suppose that we want to place sales codes based on the P\_INDATE into the table using the following rule:
  - If P\_INDATE is before December 25, 2003, then P\_SALECODE = 2
  - If P\_INDATE between Jan. 16 and Feb 10, 2004, then P\_SALECODE = 1

Given the rule from the previous page and using the PRODUCT table to effect the update, the following two command sequences will make all the appropriate updates to the P\_SALECODE in the PRODUCT table. The effect of these two updates on the PRODUCT table is shown on the next page.

```
UPDATE PRODUCT
```

```
SET P_SALECODE = '2'
```

```
WHERE P_INDATE < '25-Dec-2003';
```

```
UPDATE PRODUCT
```

```
SET P_SALECODE = '1'
```

```
WHERE P_INDATE >= '16-Jan-2004'
```

```
AND P_INDATE <= '10-Feb-2004';
```

COP 4710: Database Systems (Day 18)

Page 25

		G 11	71 7 -					8
Carrier and	ਡ 🔍 ✔   & 🖽 🖽 ≫  *	ି   🖁   🛱 🖡	A+ 19 E		<b>#1</b>   •*	※   世 福 ·	· 🛛 🖓 🗸	
								52 
💼 Ch06	_SaleCo : Database					<u>×</u> ا		
DDODUCI				_				
PRODUCI	_3 : Table							
P_CODE	P_UESCRIPT	P_INUATE	P_ONHAND	P_MIN	P_PRICE	P_DISCOUNT	V_CODE	P_SALECODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-03	8	5	\$109.99	0.00	25595	2
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-03	32	15	\$14.99	0.05	21344	2
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-03	18	12	\$17.49	0.00	21344	2
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-04	15	8	\$39.95	0.00	23119	
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-04	23	5	\$43.99	0.00	23119	
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-03	8	5	\$109.92	0.05	24288	
2232/QVVE	B&D jigsavv, 8-in. blade	24-Dec-03	6	5	\$99.87	0.05	24288	2
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-04	12	5	\$38.95	0.05	25595	1
23109-HB	Claw hammer	20-Jan-04	23	10	\$9.95	0.10	21225	1
23114-AA	Sledge hammer, 12 lb.	02-Jan-04	8	5	\$14.40	0.05		
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-03	43	20	\$4.99	0.00	21344	2
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-04	11	5	\$256.99	0.05	24288	1
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-04	188	75	\$5.87	0.00		
SM-18277	1.25-in. metal screw, 25	01-Mar-04	172	75	\$6.99	0.00	21225	
SW-23116	2.5-in. wd. screw, 50	24-Feb-04	237	100	\$8.45	0.00	21231	
WR3/TT3	Steel matting, 4'x8'x1 <i>1</i> 6", .5" mesh	17-Jan-04	18	5	\$119.95	0.10	25595	1
S		1	0	0	\$0.00	0.00	0	

COP 4710: Database Systems (Day 18)

*Page 26* 

- The arithmetic operators are particularly useful in data updates.
- For example, if the quantity on hand in the PRODUCT table has dropped below the minimum desirable value, you'll order more of the product. Suppose that we've ordered 20 more units of product 2232/QWE. When the 20 units arrive, we'll need to add them to the inventory in the following manner:

#### UPDATE PRODUCT

SET P\_ONHAND = P\_ONHAND + 20

WHERE P\_\_CODE = '2232/QWE/;

COP 4710: Database Systems (Day 18)

Page 27

• Suppose that you want to add 10 percent to the price of all products that have current prices below \$50. This can be accomplished with the following command.

UPDATE PRODUCT SET P\_PRICE = P\_PRICE \* 1.10 WHERE P PRICE < 50.00;

COP 4710: Database Systems (Day 18)

# **Copying Parts of Tables**

- Although the database should be properly designed before it is implemented, there are times when it may be necessary to break up a table structure into several component parts (smaller tables).
- SQL allows for the copying of the contents of selected table  $columns_{PRODUCT}$  the data in those columns need not be reentered manually into the newly created table(s).
- Supposer For Prexample 00; that we want to copy the P\_CODE, P\_DESCRIPT, and P\_PRICE columns from the PRODUCT table to a new table named PART.
- First, we'll need to create the PART table as shown on the next page.



CREATE TABLE PART (	
PART_CODE	CHAR(8) NOT NULL UNIQUE,
PART_DESCRIPT	CHAR(35),
PART_PRICE	DECIMAL(8,2),
PRIMARY KEY (PART_CO	DDE) );

- Note that the PART column names need not be identical to those in the original table, and that the new table need not have the same number of columns as the original table.
  - In this case, the first column in the PART table is PART\_CODE, rather than the original P\_CODE found in the PRODUCT table. And the PART table contains only three columns, rather than the seven columns found in the PRODUCT table.
  - However, column characteristics must match: you cannot copy a character-based attribute into a numeric structure, and vice versa.

COP 4710: Database Systems (Day 18)

Page 30

- Next, we need to add the rows to the new PART table using the PRODUCT table rows. To do this we'll need to use the INSERT command.
- Recall, the basic syntax of this command:

INSERT INTO target\_tablename [(target\_columnlist)]

SELECT *source\_columnlist* 

FROM source-tablename;

- The target-columnlist is requires if the source-columnlist doesn't match all of the attribute names and characteristics f the target table (including the order of the columns!). Otherwise, you do not need to specify the targetcolumnlist.
  - In our example, we will need to specify the target-columnlist since we are changing the names of one or more of the columns.





• To effect the desired update, we need to issue the following INSERT command:

INSERT INTO PART (PART\_CODE, PART\_DESCRIPT, PART\_PRICE) SELECT P\_CODE, P\_DESCRIPT, P\_PRICE FROM PRODUCT;

The contents of the updated PARTS table can be viewed with the following query:

```
SELECT *
FROM PART;
```

• The results are shown on the next page.



\_ 🗆 ×

#### Microsoft Access

File Edit View Insert Format Records Tools Window Help

PART_CODE	PART_DESCRIPT	PART_PRICE
11QER/31	Power painter, 15 psi., 3-nozzle	\$109.99
13-Q2/P2	7.25-in. pwr. saw blade	\$14.99
14-Q1/L3	9.00-in. pwr. saw blade	\$17.49
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	\$39.95
1558-QW1	Hrd. cloth, 1/2-in., 3x50	\$43.99
2232/QTY	B&D jigsaw, 12-in. blade	\$109.92
2232/QWE	B&D jigsaw, 8-in. blade	\$99.87
2238/QPD	B&D cordless drill, 1/2-in.	\$38.95
23109-HB	Claw hammer	\$9.95
23114-AA	Sledge hammer, 12 lb.	\$14.40
54778-2T	Rat-tail file, 1/8-in. fine	\$4.99
89-WRE-Q	Hicut chain saw, 16 in.	\$256.99
PVC23DRT	PVC pipe, 3.5-in., 8-ft	\$5.87
SM-18277	1.25-in. metal screw, 25	\$6.99
SW-23116	2.5-in. wd. screw, 50	\$8.45
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	\$119.95
ecord: 💶 🔳	1 ▶ ▶ ▶ of 16	

Results of copying parts of a table with renaming of the columns.

COP 4710: Database Systems (Day 18)

Page 33

#### Adding Primary and Foreign Key Designations

- Although we were able to create a new table based on an existing table in the previous example, the process is not without its problems. Primarily, the new PART table was created without the inclusion of the integrity rules of the table on which it was based. In particular, there is no primary key designated for the table shown on the previous page.
- To define the primary key for this table we need to use the ALTER command as shown below:

ALTER TABLE PRODUCT

ADD PRIMARY KEY (P\_CODE);

COP 4710: Database Systems (Day 18)

Page 34

# Adding Primary and Foreign Key Designations (cont.)

- Quite aside from the fact that the integrity rules are not automatically transferred to a new table that derives its data from one or more other tables, there are several other scenarios that would leave you without entity and referential integrity enforcement.
- For example, you might have simply forgotten to define the primary and foreign keys when you created the tables.
- The integrity rules can be reestablished via the ALTER command as shown below:

```
ALTER TABLE PRODUCT
ADD PRIMARY KEY(P_CODE)
ADD FOREIGN KEY(V_CODE) REFERENCES VENDOR;
```

COP 4710: Database Systems (Day 18)

#### Deleting a Table From the Database

A table can be deleted from the database through the DROP command as shown below:

DROP TABLE PRODUCT

A table can only be dropped from a database if it is not participating as the "1" side of any relationships. If you attempt to delete such a table, the RDMS will issue an error message to indicate that a foreign key integrity violation has occurred.

