Chapter 8: Advanced SQL

Modern Database Management 6th Edition Jeffrey A. Hoffer, Mary B. Prescott, Fred R. McFadden

Processing Multiple Tables – Joins

Join - a relational operation that causes two or more tables with a common domain to be combined into a single table or view

Equi-join - a join in which the joining condition is based on equality between values in the common columns; common columns appear redundantly in the result table

Natural join — an equi-join in which one of the duplicate columns is eliminated in the result table

Outer join - a join in which rows that do not have matching values in common columns are nonetheless included in the result table (as opposed to *inner* join, in which rows must have matching values in order to appear in the result table)

Union join – includes all columns from each table in the join, and an instance for each row of each table

The common columns in joined tables are usually the primary key of the dominant table and the foreign key of the dependent table in 1:M relationships.

Figure 7-3 revisited: Sample Pine Valley Furniture data

oft Access - [CUS	STOMER_t : Table]												
<u>E</u> dit ⊻iew <u>I</u> nsert f	Format Records Tools W	jindow <u>H</u> elp				<u> - 6 ×</u>							
a 🕹 🖪 🖤 🛛	X 🖻 🖻 🚿 🕫 🤮		M +* 1X E	口 / 2	Q -								
stomer_ID	Customer_Name	Customer_Address	City	State P	ostal_Code								
1 Conte	emporary Casuals	1355 S Hines Blvd	Gainesville	FL 32	2601-								
2 Value	e Furniture	15145 S.W. 17th St.	Plano	TX 75	5094-								
3 Home	ie Furnishings	1900 Allard Ave.	Albany	NY 12	2209-			<					
4 Easte	ern Furniture	1925 Beltline Rd.	Carteret	NJ 07	7008-				lal				
5 Impre	essions	5585 Westcott Ct.	Sacramento	CA 94	4206-		Solt Access - [u	INDEN_C. Tab					
6 Furnit	iture Gallery	325 Flatiron Dr.	Boulder	CO 80	0514-	j 🛄 Ele	Edit View Inser	rt Format <u>R</u> eco	ords <u>T</u> ools <u>W</u> ind	dow Help			
7 Perio	od Furniture	394 Rainbow Dr.	Seattle	WA 97	7954-	₩ -		* •	🚿 n 🍓	₹↓ ₹↓ ¥ 面	7 🗰 H* MK 🛅	〕 徇 • ② •	
8 Calfor	ornia Classics	816 Peach Rd.	Santa Clara	CA 98	5915-		Order Id	Order Date	Customer	ID			_
9 M & H	H Casual Furniture	3			4620-	• •	1001	10/21/2000	D 1				
10 Semi	inole Interiors	2			4646-	+	1002	10/21/2000	0 8				
11 Amer	rican Euro Lifestyles	2 CIST	nme	rs	7508-	•	1003	10/22/2000	D 15				
12 Battle	le Creek Furniture	3			9015-	•	1004	10/22/2000	0 5	;			
13 Herita	age Furnishings	6			/013-	•	1005	10/24/2000	D 3				
14 Kane	eohe Homes	112 Kiowai St.	Kaneohe	HI 98	5/44-	•	1006	10/24/2000	0 2		order		
15 Moun	ntain Scenes	4132 Main Street	Ugden	UT 84	4403-	•	1007	10/27/2000	D 11				
itoNumber)						•	1008	10/30/2000	0 12	2			
1.0	a late later					+	1009	11/5/2000	D 4				
	E FEIRE OF 15					•	1010	11/5/2000	D 1				
						the second se							
mber to identfy custo	tomer					*	0		U				
Mer to identfy custo	tomer					*	0		U				
Microsoft Access	tomer					Record:		1 1 1 1 +] of 10				
Microsoft Access	tomer s - [Order_line_t : Table] [Insert Format Records]	[ools <u>W</u> indow <u>H</u> elp				Record: Datashe	0 II III eet View	1 • • • • •	u of 10				
Microsoft Access	tomer s - [Order_line_t: Table] Insert Format Records] \$ ** \$ B @ #\$ \$	[ools Window Help ∽ @ Q↓ X↓ Vy P	5 7 A ►	× © 2	á · (?) ·	Record: Datashe	0 M M	1 • • •] of 10				
Microsoft Access	Insert Format Records 1 *** Order_line_t: Table) Insert Format Records 1 *** & @ @ @ %	[ools Window Help 19 영화 출부 조부 정 전 19 Quan	1 7 4 4 ••	× © 4	1 · 2 .	Record: Datashe	0 eet View	1 1 1 1] of 10				
Microsoft Access Microsoft Access ■ Elle Edit View Source_Id 1001 ~1 1001 ~1 1000 ~	Insert Format Records I Product Id Order	[ools <u>Wi</u> ndow Help 의 🛞 입니 조니 🏹 원 ed_Quan 2	j7 (A) ▶•	× © 2	a • (2) •	Record: Datashe	0	1 + 11 ++] of 10				
	s - [Order_line_t: Table] Insert Format Records 1 Product_Id Order 1 2	[ools Window Help 이 (1) 1 조나 지 또 된 ed_Quan 2 2	j7 (4) ▶•	× @ 2	5 • Q •	Record: Datashe	0 Inter View	1 + 11 ++] of 10				
I mber to identify cust Microsoft Access Ele Edit View ✓ Order_Id 1001 1001	Insert Format Records] Product_Id Order 1 2 4 4	[ools Window Help 이 에 순부 조부 장 전 ed_Quan 2 1	∃ 7 (4) ▶•	× © 2	1 - 1 - 1	Record: Datashe	o net View	1 1 1 1 1 1	of 10				
▲ 1 mber to identfy custom Microsoft Access Bile Edit View ▲ Order_Id 1001 1001 1002	s - [Order_line_t: Table] Insert Format Records I	[ools Window Help 전	7 4 ••	× D 2	9 • 12 •	Record: Datashe	0	1 F F F] of 10				
Import to identify customer to identify and identify to identify and identify	s - [Order_line_t: Table] Insert Format Records 1 Product_Id Order 1 2 4 3 3	[cols ₩indow Help 2	5 7 4 ••	× © 2	1 - C) -	Record: Datashe	0 It I I I I I I I I I I I I I I I I I I	1 b b b] of 10				- 0
▲ 1 mber to identfy custo Microsoft Access ■ Ele ■ 1001 1001 1002 1003 1004	s-[Order_line_t: Table] Insert Format Records 1 Product_ld Order 1 2 4 3 3 6	[ools ₩indow Help 2 ed_Quan 2 1 5 3 2	57 A >*	× © 2	±	Record: Datashe	0 et View PRODUCT. PRODUCT.	1 DI	Jof 10 Window ⊞elp				
▲ 1 mber to identfy custo Microsoft Access ■ Ele Edit View ▲ ▲ ■ Ele Edit View ▲ ■ Order Id ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1004	Insert Format Records 1 Product_Id Coder Product_Id Order 1 2 4 3 6 8	[ools Window Help ▷ ④ ▲ ▲ ▲ ▼ ♥ ed_Quan 2 1 5 3 2 2 2	5 7 (41) •	× @ 4	a • • • •	Microsoft Accord	0 ■ () et View • sse - [PRODUCT] • Insert Format Q ♥ % Fib	1 DI	Undow Help	V To V M	·· × □ 盘·		
▲ 1 mber to identfy cust Microsoft Access ■ Ele Edit View ▲ ■ Order_Id 1001 1001 1001 1001 1001 1003 1004 1005	Insert Format Records I Product_Id Order 1 2 4 3 4 3 6 8 4 4	[ools Window Help ▷ @ 24 & V Y ced_Quan 2 1 5 3 2 2 - -	9 7 44 > •	× 🗁 2	5 • C •	Microsoft Acoc Detashe	0 ■ I I I I I set View ■ Set - [PRODUCT] ■ Insert Fgrmat Q ♥ & B P D [Product De	I I I I I I I I I I I I I I I I I I I	U ∫of 10 Window Help @ 2↓ Z↓ oduct Finish	Ÿ ⅔ ♥ å	▶* ₩ □ @ -		
▲ 1 mber to identfy cust Microsoft Access ■ Eile Edit Yiew L Eile Edit Yiew L ■ ■ ■ ■ Order_Id ■ ■ ■ ■ ■	s - [Order_line_t: Table] Insert Format Records I	[ools Window Help		× © #	a · 2 .	Microsoft Accord	0 1 () tet View ss - (PRODUCT) insert Format Comparison () () () () () () () () () ()	I DIF	Undow Help € 2↓ X↓ oduct_Finish erv	V To V A Standard_Price \$175.00	▶• 咪 应 @ - I Product Line I 0 10001	2 .	
▲ 1 mber to identfy custom Microsoft Access Bile Edit View 2 Order_Id 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1002 1003 1004 1005 1006	s - [Order_line_t: Table] Insert Format Records I Product_Id Order 1 2 4 3 3 6 8 4 4 4 5	[ools Window Help 2 ed_Quan 2 2 1 5 3 2 2 0 Cord	ava	× © #	a · ℚ . S	Microsoft Acco Datashe	0 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	T Die Table Records Icols Records Icols Scription Pro Che Nat	Window Help	▼ ▼ ▲4 Standard_Price \$175.00 \$175.00 \$200.00	▶• ▶× ☐ 20 - 1 • Tooluct_Line_1 0 10001 0 20001	2 . 10	_ 0 _ 0
▲ 1 mber to identfy custom Microsoft Access ■ Ele ■ 1001 1001 1001 1002 1003 1004 1005 1006 1006	s - [Order_line_t: Table] Insert Format Records 1 Product_Id Order 1 2 4 3 6 8 4 4 5 7	[ools Window Help ed_Quan 2 1 5 3 2 2 1 5 3 2 2 2 2 0 COCC	a⊽ a	× © 2	a · ₽ . S	Microsoft Acce Microsoft Acce Edt yew Product	0 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	I I I I I I I I I I I I I I I I I I I	Window Help Window Help Wind		▶• ★ 臣 泡 - 1 Product_Line_ 0 10001 0 20001 0 20001	2 .	_ 8 _ 8
▲ ▲ 1 mber to identfy custo Microsoft Access ■ Ele Edit View ▲ ▼ ■ ● ● ● ■ ● ● ● ● ● ■ ● </td <td>Insert Format Records I Product_Id Order 1 2 4 3 4 3 6 8 4 4 5 7 1</td> <td>[ools Window Help ▷ ④ ② 爻 ¾ Ў Ў ed_Quan 2 1 5 3 2 0 Cord</td> <td>∎⊽ M →</td> <td>× © 2</td> <td>s</td> <td>Microsoft Accord Detashe Microsoft Accord Ele Edt View Product + + +</td> <td>0 ■ I I I I et View ■ I I I I I I I I I I I I I I I I End Table 2 Coffee Table 3 Computer DA 4 Entertainmet</td> <td>1 PI PI C: Table) Records Iools Records Iools R</td> <td>Window Help Window Help Markov Lep Markov Lep Mark</td> <td></td> <td>▶ ★ ★ 日 @ @ ~ Product_Line_I 0 10001 0 20001 0 30001</td> <td></td> <td>_ 8 _ 8</td>	Insert Format Records I Product_Id Order 1 2 4 3 4 3 6 8 4 4 5 7 1	[ools Window Help ▷ ④ ② 爻 ¾ Ў Ў ed_Quan 2 1 5 3 2 0 Cord	∎⊽ M →	× © 2	s	Microsoft Accord Detashe Microsoft Accord Ele Edt View Product + + +	0 ■ I I I I et View ■ I I I I I I I I I I I I I I I I End Table 2 Coffee Table 3 Computer DA 4 Entertainmet	1 PI PI C: Table) Records Iools Records Iools R	Window Help Window Help Markov Lep Markov Lep Mark		▶ ★ ★ 日 @ @ ~ Product_Line_I 0 10001 0 20001 0 30001		_ 8 _ 8
▲ ▲ 1 mber to identfy cust Microsoft Access ■ Ele Edit Yiew L ■ ■ ● ● Order_Id ■ ● ● ● ■ ■ ● <td< td=""><td>Insert Format Records I Product_Id Order 1 2 4 3 4 3 6 8 4 4 5 7 1 2 4 3 6 8 4 4 5 7 1 2</td><td>[ools Window Help</td><td>er li</td><td>× © ŧ</td><td>s</td><td>Microsoft Acoc Detashe</td><td>0 ■ • • • • • • • • • • • • • • • • • • •</td><td>I I I I I I I I I I I I I I I I I I I</td><td>Window Help Window Help Windo</td><td>∑ Ta ♥ A Standard_Price \$175.00 \$200.00 \$375.00 \$650.00 \$650.00 \$650.00</td><td>▶ ★ X ② ② - 1 Product_Line_1 0 10001 0 20001 0 20001 0 30001 0 10001</td><td></td><td>_ () _ () _ ()</td></td<>	Insert Format Records I Product_Id Order 1 2 4 3 4 3 6 8 4 4 5 7 1 2 4 3 6 8 4 4 5 7 1 2	[ools Window Help	er li	× © ŧ	s	Microsoft Acoc Detashe	0 ■ • • • • • • • • • • • • • • • • • • •	I I I I I I I I I I I I I I I I I I I	Window Help Window Help Windo	∑ Ta ♥ A Standard_Price \$175.00 \$200.00 \$375.00 \$650.00 \$650.00 \$650.00	▶ ★ X ② ② - 1 Product_Line_1 0 10001 0 20001 0 20001 0 30001 0 10001		_ () _ () _ ()
▲ ▲ 1 mber to identfy cust	s - [Order_line_t: Table] Insert Format Records I Troduct_Id Order 1 2 4 3 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 8 4 4 5 7 1 2 3 6 8 8 8 4 4 5 7 1 1 2 3 6 8 8 8 8 8 8 8 8 8 8 8 8 8	[ools Window Help	er l	× © ₹	a - ② . S	Microsoft Accc Datashe	0 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	T Die PI Eccords Icols Eccords Ico	Window Help	♥ 10 ♥ 44 Standard_Price \$175.00 \$200.00 \$375.00 \$325.00 \$325.00 \$325.00			
▲ 1 mber to identfy cust Microsoft Access ■ File ■ 1001 ■ 1002 ■ 1003 ■ 1004 ■ 1005 ■ 1006 ■ 1006 ■ 1007 ■ 1008 ■ 1008	s - [Order_line_t: Table] Insert Format Records I Product_Id Order 1 2 4 4 3 3 6 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 8 8 8 8 8 8 8 8 8 8 8 8	[ools Window Help ed_Quan 2 1 5 3 2 2 1 5 3 2 2 1 5 3 2 3 3 2 3 3 3 3	a⊽ a	× © 2	a - ② . S		0 iet View iet View ist - [PRODUCT] insert Format Computer Du 2 Coffee Table 3 Computer Du 4 Entertainmer 5 Witter's Des 6 8-Drawer Draf 6 B-Drawer Draf 7 Dining Table	I I I I I I I I I I I I I I I I I I I	Window Help Window Help Wind	▼ ★ ★ ↓ Standard_Price \$175.00 \$200.00 \$375.00 \$650.00 \$225.00 \$750.00 \$800.00 \$800.00			_ 0 _ 0
▲ 1 mber to identfy custo Microsoft Access Bile Edit Bile Edit Yew Edit	Insert Format Records 1 Insert Format Records 1 Insert Format Records 1 Product_Id Order 1 2 4 3 6 8 4 4 5 7 1 2 3 8 4 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 4 5 7 1 2 4 5 8 4 4 5 7 1 2 4 5 8 8 4 4 5 7 7 1 2 4 5 8 8 8 8 8 8 8 8 8 8 8 8 8	[ools Window Help	er l	× © 2	s	Microsoft Acce Detashe Microsoft Acce Detashe Product +	0 ■ I I I I I I I I I I I I I I I I I I I	I I I I I I I I I I I I I I I I I I I	Window Help Window Help Wind	✓ Y a ♥ a Standard_Price \$175.00 \$200.00 \$375.00 \$650.00 \$325.00 \$750.00 \$800.00 \$250.00 \$250.00			_ 0 _ 0
▲ ▲ 1 mber to identfy cust ■ Microsoft Access ■ ■ Ele Edit View 业 ■ ■ ● ● ● ■ ■ ● ● ● ● ■ ■ ● <td>Insert Format Records I Product Id Order 1 1 2 4 3 6 8 4 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 4 5 7 1 6 8 8 4 5 7 7 1 7 7 7 7 7 7 7 7 7 7 7 7 7</td> <td>Cools Window Help</td> <td>er l</td> <td>× © 2</td> <td>s.</td> <td>Hicrosoft Accord: Detashe Microsoft Accord Detashe Product +</td> <td>0 ■ () et View • Insert Format • Insert Format • Insert Format • O • Product_De 2 Coffee Table 2 Coffee Table 2 Coffee Table 2 Coffee Table 3 Computer Dr 4 Entertainme 5 Write's Des 6 8-Drawer Dre 7 Dining Table 8 Computer Do 8 Computer Dre 9 Comput</td> <td>I I I I I I I I I I I I I I I I I I I</td> <td>Window Help Window Help Windo</td> <td>▼ Tan Atan Standard_Price \$175.00 \$200.00 \$375.00 \$255.00 \$750.00 \$800.00 \$220.00 \$800.00 \$220.00 \$200.00 \$250.00 \$200.00 \$200.00</td> <td>▶ ★ X @ 泡 - 1 Product_Line_1 0 10001 0 20001 0 20001 0 30001 0 10001 0 20001 0 20001 0 20001</td> <td>2 .</td> <td>_ () _ () _ ()</td>	Insert Format Records I Product Id Order 1 1 2 4 3 6 8 4 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 4 4 5 7 1 2 3 6 8 4 5 7 1 6 8 8 4 5 7 7 1 7 7 7 7 7 7 7 7 7 7 7 7 7	Cools Window Help	er l	× © 2	s.	Hicrosoft Accord: Detashe Microsoft Accord Detashe Product +	0 ■ () et View • Insert Format • Insert Format • Insert Format • O • Product_De 2 Coffee Table 2 Coffee Table 2 Coffee Table 2 Coffee Table 3 Computer Dr 4 Entertainme 5 Write's Des 6 8-Drawer Dre 7 Dining Table 8 Computer Do 8 Computer Dre 9 Comput	I I I I I I I I I I I I I I I I I I I	Window Help Window Help Windo	▼ Tan Atan Standard_Price \$175.00 \$200.00 \$375.00 \$255.00 \$750.00 \$800.00 \$220.00 \$800.00 \$220.00 \$200.00 \$250.00 \$200.00 \$200.00	▶ ★ X @ 泡 - 1 Product_Line_1 0 10001 0 20001 0 20001 0 30001 0 10001 0 20001 0 20001 0 20001	2 .	_ () _ () _ ()
▲ ▲ 1 mber to identfy cust ■ Microsoft Access ■ Bile Edit View ▲ 2 ■ 0 001 = 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1001 ■ 1002 1003 1004 1005 1006 1006 1007 1007 1008 1009 1009 1009	Insert Format Records I The set Format	[ools Window Help 2 2 2 1 5 3 2 2 3 2 3 2 3 3 2 3 10	er l	× © ŧ	s ∎- Ձ,	Microsoft Accc Datashe	0 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	T Dolo Records Tools Records T	Window Help window Help wind	▼ 10 ▼ 44 Standard_Price \$175.00 \$200.00 \$375.00 \$650.00 \$250.00 \$250.00 \$250.00 \$250.00			
▲ ▲ 1 mber to identfy customer to identify custom	Insert Format Records I S - [Order_line_t: Table] Insert Format Records I Product_Id Order 1 1 2 4 3 3 6 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 4 4 5 7 1 2 3 8 8 4 4 5 7 7 1 8 8 8 4 4 5 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8	[ools Window Help 2 2 3 2 2 1 5 3 2 2 2 1 5 3 2 3 2 3 3 3 3 2 3 3 10 0 0	s a	× © 2	s 		0 1 I I I I I I I I I I I I I I I I I I I	T Table Records Tools Records Tools Records Tools Records Nat Resk Nat Nat Resser Wh Nat Resser Wh Nat Resser Wh Nat Resser Wh	Window Help Window Help Window Help Window Help 24 X↓ 24 X↓	▼ To To At a standard Price \$175.00 \$200.00 \$375.00 \$325.00 \$325.00 \$325.00 \$325.00 \$325.00 \$325.00 \$325.00 \$325.00 \$325.00 \$325.00 \$300.00 \$250.00 \$300.00 \$250.00 \$300.00 \$300.00 \$300.0			_ Ø, _ Ø,

Natural Join Example

For each customer who placed an order, what is the customer's name and order number?

Join involves multiple tables in FROM clause

SELECT CUSTOMER_T.CUSTOMER_ID, CUSTOMER_NAME, ORDER_ID FROM CUSTOMER_T, ORDER_T WHERE CUSTOMER_T.CUSTOMER_ID = ORDER_T.CUSTOMER_ID

> WHERE clause performs the equality check for common columns of the two tables

Chapter 8

Outer Join Example

List the customer name, ID number, and order number for all customers. Include customer information even for customers that do have an order

SELECT CUSTOMER_T.CUSTOMER_ID, CUSTOMER_NAME, ORDER_ID FROM CUSTOMER_T, LEFT OUTER JOIN ORDER_T WHERE CUSTOMER_T.CUSTOMER_ID = ORDER_T.CUSTOMER_ID

> LEFT OUTER JOIN syntax will cause customer data to appear even if there is no corresponding order data

Chapter 8

Multiple Table Join Example

Assemble all information necessary to create an invoice for order number 1006 Four tables involved in this

join

SELECT CUSTOMER_T.CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ADDRESS, CITY, SATE, POSTAL_CODE, ORDER_T.ORDER_ID, ORDER_DATE, QUANTITY, PRODUCT_NAME, UNIT_PRICE, (QUANTITY * UNIT_PRICE)

FROM CUSTOMER_T, ORDER_T, ORDER_LINE_T, PRODUCT_T

WHERE CUSTOMER_T.CUSTOMER_ID = ORDER_LINE.CUSTOMER_ID AND ORDER_T.ORDER_ID = ORDER_LINE_T.ORDER_ID AND ORDER LINE T.PROEUCT ID = PRODUCT PRODUCT ID

AND ORDER_T.ORDER_ID = 1006;

Each pair of tables requires an equality-check condition in the WHERE clause, matching primary keys against foreign keys

Chapter 8

Figure 8-1 – Results from a four-table join

From CUSTOMER_T table

CUSTOMER	_ID (CUSTON	IER_NAME	CUSTOMER_ADDRESS	CITY	ST	POSTAL_CODE	
	2 V 2 V	Value Fun Value Fun Value Fun	niture niture niture	15145 S.W. 17th St. 15145 S.W. 17th St. 15145 S.W. 17th St.	Plano Plano Plano	TX TX TX	75094 75094 75094	
ORDER_ID	ORDE	R-DAT	QUANTITY	PRODUCT_NAME	UNIT_PRK	CE	(QUANTITY*UNIT	PRICE)
1006 1006 1006	24-OC 24-OC 24-OC	T-00 T-00 T-00	1 2 2	Entertainment Center Writer's Desk Dining Table	6: 3: 8:	50 25 20		650 650 1600

From ORDER_T table

From PRODUCT_T table

Processing Multiple Tables --Subqueries

Subquery = placing an inner query (SELECT statement) inside an outer query

Options:

- In a condition of the WHERE clause
- As a "table" of the FROM clause
- Within the HAVING clause
- Subqueries can be:
- Non correlated execute once for the entire outer query
- Correlated execute once for each row returned by the outer query

Subquery Example

Show all customers who have placed an order

The IN operator will test to see if the CUSTOMER_ID value of a row is included in the list returned from the subquery

SELECT CUSTOMER_NAME FROM CUSTOMER_T WHERE CUSTOMER_ID IN

(SELECT DISTINCT CUSTOMER_ID FROM ORDER_T);

Subquery is embedded in parentheses. In this case it returns a list that will be used in the WHERE clause of the outer query

Chapter 8

Correlated vs. Noncorrelated Subqueries

Non-correlated subqueries:

- Do not depend on data from the outer query
- Executes once for the entire outer query
- Correlated subqueries:
 - Does make use of data from the outer query
 - Executes once for each row of the outer query
 - Can make use of the EXISTS operator

Figure 8-2(a) – Processing a noncorrelated subquery

SELECT CUSTOMER_NAME FROM CUSTOMER_T WHERE CUSTOMER_ID IN

(SELECT DISTINCT CUSTOMER_ID FROM ORDER_T);

 The subquery (shown in the box) is processed first and an intermediate results table created:

CUSTOMER_ID

8

15 5

3 2 11

12

No reference to data
in outer query, so
subquery executes
once only

9 rows selected.

The outer query returns the requested customer information for each customer included in the intermediate results table:

CUSTOMER_NAME

Contemporary Casuals Value Furniture Home Furnishings Eastern Furniture Impressions California Classics American Euro Lifestyles Battle Creek Furniture Mountain Scenes 9 rows selected.

Correlated Subquery Example

Show all orders that include furniture finished in natural ash

The EXISTS operator will return a TRUE value if the subquery resulted in a non-empty set, otherwise it returns a FALSE

SELECT DISTINCT ORDER_ID FROM ORDER_LINE_T

(SELECT * FROM PRODUCT_T

WHERE PRODUCT_ID = ORDER_LINE_T.PRODUCT_ID

AND PRODUCT_FINISH = 'Natural ash');

The subquery is testing for a value that comes from the outer query

WHERE EXISTS

Figure 8-2(b) – Processing a correlated subquery





		Product_ID	Product Description	Product_Finish	Standard_Price	Product_Line_Id
Þ	۲	1	End Table	Cherry	\$175.00	10001
	۲	2→2	Coffee Table 🛛 🤇	Natural Ash	\$200.00	20001
	۲	4→ 3	Computer Desk	Natural Ash	\$375.00	20001
	۲	4	Entertainment Center	Natural Maple	\$650.00	30001
	۲	5	Writer's Desk	Cherry	\$325.00	10001
	۲	6	8-Drawer Dresser	White Ash	\$750.00	20001
	۲	7	Dining Table 🔇	Natural Ash	\$800.00	20001
	۲	8	Computer Desk	Walnut	\$250.00	30001
*		(AutoNumber)			\$0.00	

- 1. The first order ID is selected from ORDER _LINE _T: ORDER _ID =1001.
- The subquery is evaluated to see if any product in that order has a natural ash finish. Product 2 does, and is part of the order. EXISTS is valued as *true* and the order ID is added to the result table.
- The next order ID is selected from ORDER _LINE _T: ORDER _ID =1002.
- The subquery is evaluated to see if the product ordered has a natural ash finish. It does. EXISTS is valued as true and the order ID is added to the result table.
- Processing continues through each order ID. Orders 1004, 1005, and 1010 are not included in the result table because they do not include any furniture with a natural ash finish. The final result table is shown in the text on page 303.

Show all orders that include furniture finished in natural ash

Subquery forms the derived table used in the FROM clause of the outer query One column of the subquery is an aggregate function that has an alias name. That alias can then be referred to in the outer query

SELECT PRODUCT_DESCRIPTION, STANDARD_PRICE, AVGPRICE

(SELECT AVG(STANDARD_PRICE) AVGPRICE FROM PRODUCT_T), PRODUCT_T WHERE STANDARD_PRICE > AVG_PRICE;

The WHERE clause normally cannot include aggregate functions, but because the aggregate is performed in the subquery its result can be used in the outer query's WHERE clause

Ensuring Transaction Integrity

Transaction = A discrete unit of work that must be completely processed or not processed at all

- May involve multiple updates
- If any update fails, then all other updates must be cancelled
- SQL commands for transactions
 - BEGIN TRANSACTION/END TRANSACTION
 - Marks boundaries of a transaction
 - COMMIT
 - Makes all updates permanent
 - ROLLBACK

• Cancels updates since the last COMMIT

Figure 8-4: An SQL Transaction sequence (in pseudocode)

```
BEGIN transaction
  INSERT Order ID, Order date, Customer ID into Order t;
  INSERT Order ID, Product ID, Quantity into Order line t;
  INSERT Order ID, Product ID, Quantity into Order line t;
  INSERT Order ID, Product ID, Quantity into Order line t;
END transaction
                                           Invalid Product ID entered
   Valid information inserted.
   COMMIT work
                                    Transaction will be ABORTED.
                                    ROLLBACK all changes made to Order_t
   All changes to data
                                    All changes made to Order_t
   are made permanent.
                                    and Order_line_t are removed.
                                    Database state is just as it was
```

Chapter 8

before the transaction began.

Data Dictionary Facilities

System tables that store metadata Users usually can view some of these tables Users are restricted from updating them Examples in Oracle8i

- DBA_TABLES descriptions of tables
- DBA_CONSTRAINTS description of constraints
- DBA_USERS information about the users of the system
- DBA_TAB_PRIVS descriptions of grants on objects in the database

SQL-99 Enhancements/Extensions User-defined data types (UDT) - Subclasses of standard types or an object type Analytical functions (for OLAP) Persistent Stored Modules (SQL/PSM) Capability to create and drop code modules – New statements: • CASE, IF, LOOP, FOR, WHILE, etc. • Makes SQL into a procedural language SQL-99 Standard not widely adopted yet Oracle has propriety version called PL/SQL

Routines and Triggers

$ightarrow \mathbf{Routines}$

- Program modules that execute on demand
- Functions routines that return values and take input parameters
- Procedures routines that do not return values and can take input or output parameters

Origgers

 Routines that execute in response to a database event (INSERT, UPDATE, or DELETE)

Figure 8-5: Triggers contrasted with stored procedures



Source: adapted from Mullins, 1995.

Figure 8-6: Oracle PL/SQL trigger syntax

CREATE [OR REPLACE] TRIGGER trigger_name {BEFORE AFTER} {INSERT | DELETE | UPDATE} ON table_name [FOR EACH ROW [WHEN (trigger_condition)]] trigger_body_here;

Figure 8-7: SQL-99 Create routine syntax

{CREATE PROCEDURE | CREATE FUNCTION} routine_name ([parameter [{,parameter} . . .]]) [RETURNS data_type result_cast] /* for functions only */ [LANGUAGE {ADA | C | COBOL | FORTRAN | MUMPS | PASCAL | PLI | SQL}] [PARAMETER STYLE {SQL | GENERAL}] [SPECIFIC specific_name] [DETERMINISTIC | NOT DETERMINISTIC] [NO SQL | CONTAINS SQL | READS SQL DATA | MODIFIES SQL DATA] [RETURN NULL ON NULL INPUT | CALL ON NULL INPUT] [DYNAMIC RESULT SETS unsigned_integer] /* for procedures only */ [STATIC DISPATCH] /* for functions only */ routine_body

Embedded and Dynamic SQL

Embedded SQL

- Including hard-coded SQL statements in a program written in another language such as C or Java
- Dynamic SQL
 - Ability for an application program to generate SQL code on the fly, as the application is running