1.264 Lecture 13

SQL: Joins, views
(Views may be covered in lecture 14)

Please start SQL Server Management Studio
Next class: Murach chapter 12. Exercises due after class

Joins

- Relational model permits you to bring data from separate tables into new and unanticipated relationships.
- Relationships become explicit when data is manipulated: when you query the database, not when you create it.
  - This is critical; it allows extensibility in databases. The FAA or EPA never thought its data would be used in 1.264 along with DOT carrier data, and some new order tables.
  - You can join on any columns in tables, as long as data types match and the operation makes sense. They don’t need to be keys, though they usually are.
- Good joins
  - Join columns must have compatible data types
  - Join column is usually key column:
    - Either primary key or foreign key
  - Nulls will never join

Joins

- List all orders, showing order number and amount, and name and credit limit of customer
  - Orders has order number and amount, but no customer names or credit limits
  - Customers has customer names and credit limit, but no order info
- SELECT OrderNbr, Amt, Company, CreditLimit FROM Customers, Orders WHERE Cust = CustNbr; (Implicit syntax)
- SELECT OrderNbr, Amt, Company, CreditLimit FROM Customers INNER JOIN Orders ON Customers.CustNbr = Orders.Cust; (SQL-92)

OrderNbr	Cust	Prod	Qty	Amt	Disc
1	211	Bulldozer	7	$31,000.00	0.2
2	522	Riveter	2	$4,000.00	0.3
3	522	Crane	1	$500,000.00	0.4

Join

CustNbr	Company	CustRep	CreditLimit
211	Connor Co	89	$50,000.00
522	Amaratunga Enterprises	89	$40,000.00
890	Feni Fabricators	53	$1,000,000.00

Join with 3 tables

- List orders over $25,000, including the name of the salesperson who took the order and the name of the customer who placed it.
  - SELECT OrderNbr, Amt, Company, Name FROM Orders, Customers, SalesReps WHERE Cust = CustNbr AND CustRep = RepNbr AND Amt >= 25000; (Implicit syntax)

OrderNbr	Cust	Prod	Qty	Amt	Disc
1	211	Bulldozer	7	$31,000.00	0.2
2	522	Riveter	2	$4,000.00	0.3
3	522	Crane	1	$500,000.00	0.4

CustNbr	Company	CustRep	CreditLimit
211	Connor Co	89	$50,000.00
522	Amaratunga Enterprises	89	$40,000.00
890	Feni Fabricators	53	$1,000,000.00

RepNbr	Name	RepOffice	Quota	Sales
53	Bill Smith	1	$100,000.00	$0.00
89	Jen Jones	2	$50,000.00	$130,000.00

Result:

OrderNbr	Amt	Company	Name
1	$31,000.00	Connor Co	Jen Jones
3	$500,000.00	AmaratungaEnterprise	Jen Jones

Join notes

- SQL-92 syntax for previous example:
  - SELECT OrderNbr, Amt, Company, Name FROM SalesReps

    INNER JOIN Customers ON SalesReps.RepNbr = Customers.CustRep

    INNER JOIN Orders ON Customers.CustNbr = Orders.Cust

    WHERE Amt >= 25000;
- Use * carefully in joins
  - It gives all columns from all tables being joined
- If a field has the same name in the tables being joined, qualify the field name:
  - Use table1.fieldname, table2.fieldname
  - Customers.CustNbr, Orders.Amt, etc.

Exercises 1

- Write the SQL queries in the previous slides
- List, for each customer:
  - Customer name, credit limit, rep name serving the customer and the rep sales (of that rep)
- List, for each customer:
  - Customer name, their rep’s name, their rep’s office name

Solutions 1

- List, for each customer: customer name, credit limit,
  rep name serving the customer and the rep sales (of that rep)

  -
    ```
    SELECT Company, CreditLimit, Name, Sales
      FROM Customers, SalesReps
      WHERE CustRep= RepNbr
    ```

- List, for each customer: customer name, their rep name, their rep’s office

  -
    ```
    SELECT Company, Name, City
      FROM Customers, SalesReps, Offices
      WHERE CustRep= RepNbr AND RepOffice= OfficeNbr
    ```

Self joins (recursive relationship)

EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	

- We want to list employees and their manager's name
  - Manager could be foreign key into manager table, but it has to be a 'foreign' key into the employee table itself in this case
- SQL essentially lets us do this by giving aliases:
  - SELECT Emp.Name, Manager.Name FROM Employees Emp, Employees Manager WHERE Emp.Mgr = Manager.EmpNbr (Implicit syntax)
  - SELECT Emp.Name, Manager.Name FROM Employees AS Emp INNER JOIN Employees AS Manager ON Emp.Mgr = Manager.EmpNbr (SQL-92)
  - We actually only need to use 1 alias (Manager)—see solution file
- Self joins used in bills of materials, or any hierarchy

Self joins

Emp (alias/virtual)

EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	

Manager (alias/virtual)

EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	

Employees (physical table)

EmpNbr	Name	Title	Mgr
105	Mary Smith	Analyst	104
109	Jill Jones	Sr Analyst	107
104	Sally Silver	Manager	111
107	Pat Brown	Manager	111
111	Eileen Howe	President	

Exercises 2

- For practice, write the SQL statements from the previous slide

- Then, create a BillOfMaterials table (New Table, in design view).
    - Table has 3 columns: PartID, PartName, PartParent
- Fill it: Edit top 200 rows…
    - PartID is A, B, C, D, E
    - PartName is AA, BB, CC, DD, EE
    - PartParent: part of the exercise
        - Part A consists of parts B and C
        - Part C consists of parts D and E
- Query the “parent” part of each part in the table
    - Name the alias tables BOMPart and BOMParent
        - You may use only one alias if you wish

Solutions 2

1. BillOfMaterials table:

PartID	PartName	Parent
A	AA	A
B	BB	A
C	CC	A
D	DD	C
E	EE	C

2. SQL:

```
SELECT BOMPart.PartName, BOMParent.PartName AS Parent
FROM BillOfMaterials AS BOMPart
INNER JOIN BillOfMaterials AS BOMParent
ON BOMPart.PartParent = BOMParent.PartID
```

JOIN types

- INNER join: returns just rows with matching keys (join column values)
- RIGHT join: returns all rows from right (second) table, whether they match a row in the first table or not
- LEFT join: returns all rows from left (first) table, whether they match a row in the second table or not
- OUTER join: Returns all rows from both tables, whether they match or not
- (We'll do an exercise on these)

Left Join Example and Exercise

- Display all customers with orders > $50,000 or credit limits > $50,000.
  - Use a RIGHT or LEFT JOIN since you want all the customers, whether they have an order or not, to be the ‘raw material’ for the WHERE clause
  - SELECT DISTINCT CustNbr

    FROM Customers LEFT JOIN Orders ON CustNbr = Cust

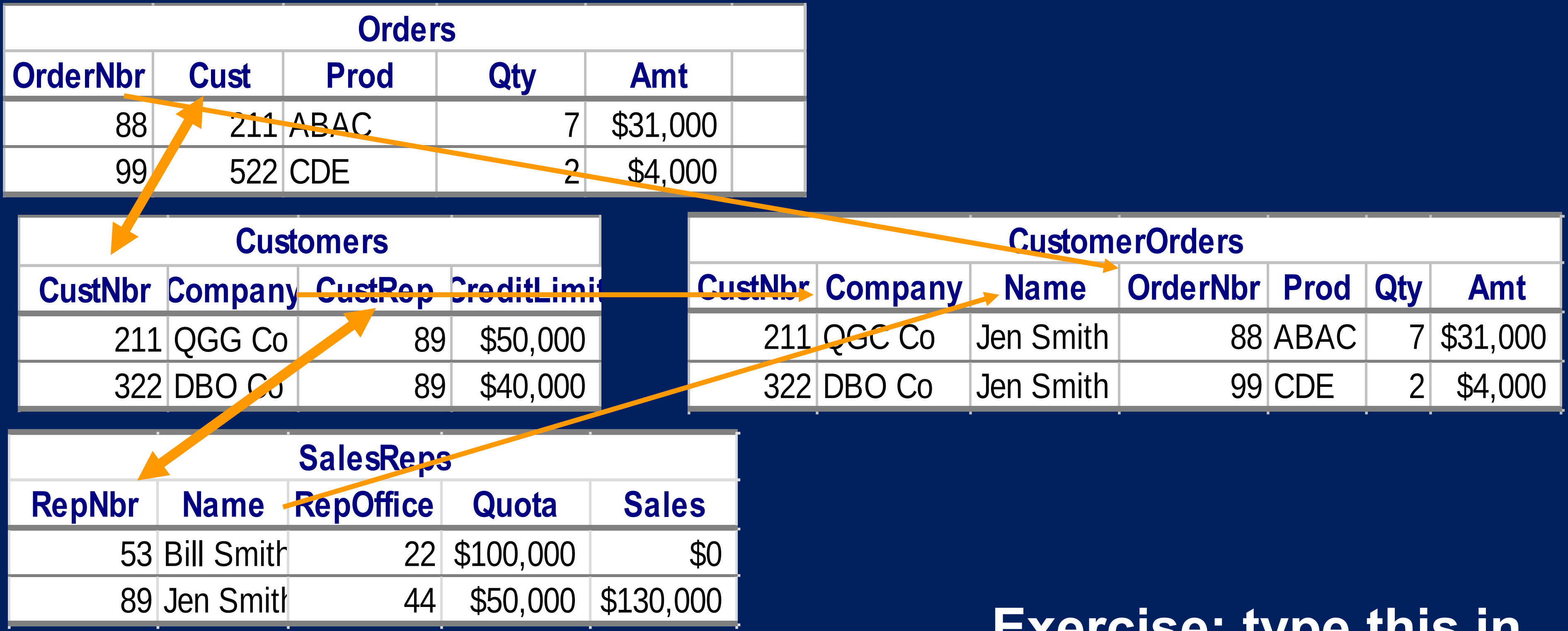
    WHERE (CreditLimit > 50000 OR Amt > 50000)
  - Exercises: (Tables should have original values-slide 4)
    - Type this SQL statement in.
    - Reverse the order of the tables, use RIGHT keyword
    - See what happens when you omit the WHERE clause; this gives you all the rows in the join.
    - Type this in without the LEFT keyword and see what happens.
    - Remove DISTINCT and see what happens.

Views

- Virtual tables that present data in denormalized form to users
- They are NOT separate copies of the data; they reference the data in the underlying tables
- Database stores definition of view; the data is updated when the underlying tables are updated
- Advantages:
  - Designed to meet specific needs of specific users
  - Much simpler queries for users on views constructed for them
  - Security: give access only for data in views
  - Independence: layers user or program away from change in underlying tables

Views (and exercise)

- CREATE VIEW CustomerOrders AS SELECT CustNbr, Company, Name, OrderNbr, Prod, Qty, Amt FROM Customers, SalesReps, Orders WHERE CustRep = RepNbr AND CustNbr = Cust
  (Implicit syntax)



Orders					
OrderNbr	Cust	Prod	Qty	Amt	
88	211	ABAC	7	$31,000	
99	522	CDE	2	$4,000	

Customers			
CustNbr	Company	CustRep	CreditLimit
211	QGG Co	89	$50,000
322	DBO Co	89	$40,000

CustomerOrders						
CustNbr	Company	Name	OrderNbr	Prod	Qty	Amt
211	QGC Co	Jen Smith	88	ABAC	7	$31,000
322	DBO Co	Jen Smith	99	CDE	2	$4,000

SalesReps				
RepNbr	Name	RepOffice	Quota	Sales
53	Bill Smith	22	$100,000	$0
89	Jen Smith	44	$50,000	$130,000

Exercise: type this in

Writing views in SQL Server

- SQL Server client helps you create views (and write queries) through query-by-example (QBE) interface
  - Views, right click, ‘New View’
  - Add tables in query (control click)
  - Check boxes for columns in result
  - Write WHERE clauses in grid
- Not all views can be updated. View is read-only if:
  - DISTINCT is in the SELECT statement
  - Statement contains expressions (averages, totals, etc.)
  - Statement references views that are not updatable
  - Statement has GROUP BY or HAVING clauses

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies
Fall 2013