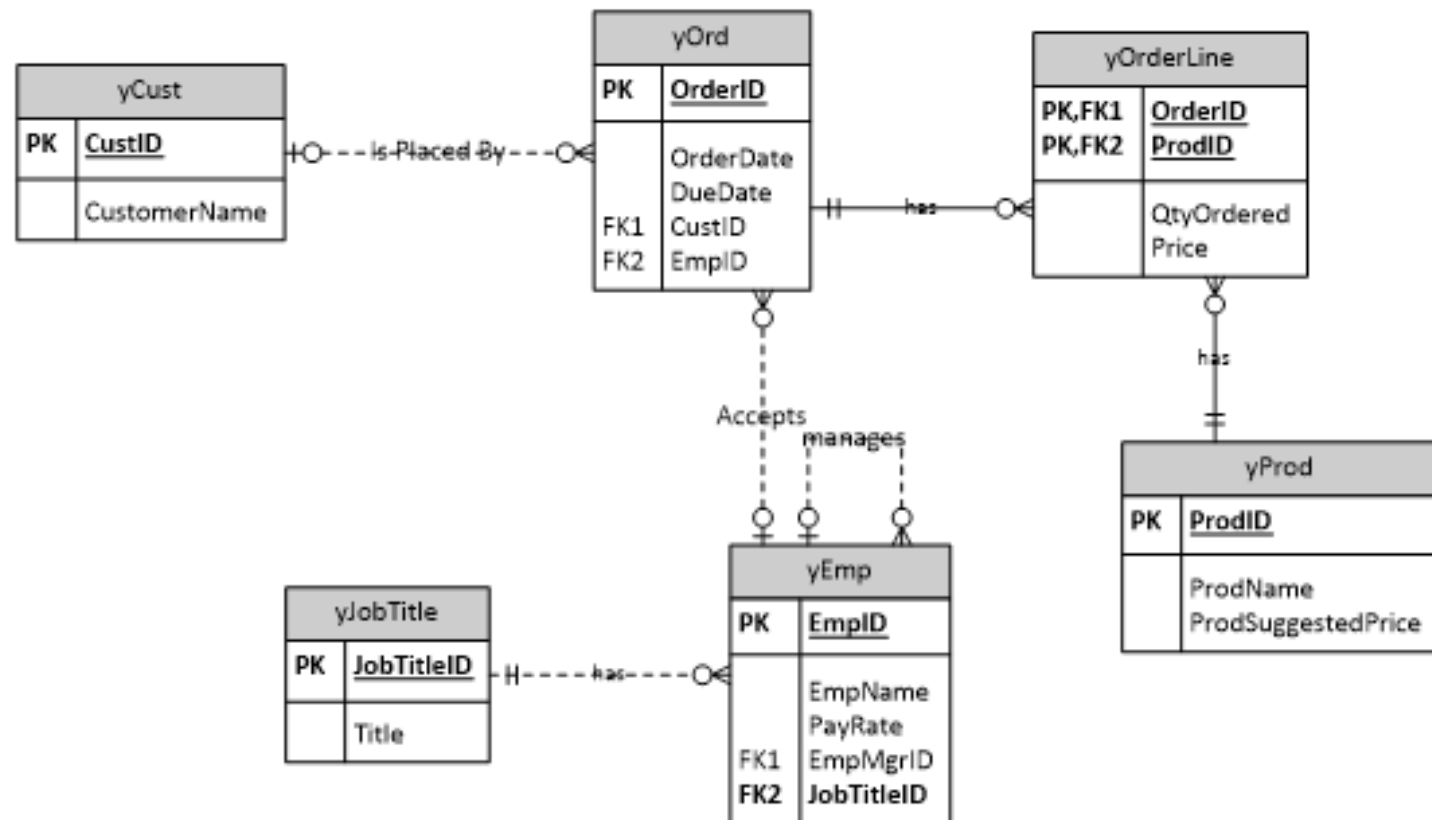


IS475/675 Agenda for 04/07/2025

- Answer questions
- Review writing non-correlated sub-queries
- Write sub-queries using the IN and NOT IN relational operators
- While waiting for class to start: Login to SQL Server Management Studio
- Create and populate six tables by opening and executing in SQL Server the file named:
K:\CoB\IS475\LabFiles\SQLLab7.sql



Create these tables by opening and executing the file: K:\CoB\IS475\LabFiles\SQLLab7.sql

Look at the content of the tables

- SELECT * FROM yEmp;
- SELECT * FROM yProd;
- SELECT * FROM yOrd;
- SELECT * FROM yOrderLine;
- SELECT * FROM yCust
- SELECT * FROM yJobTitle;

Count the rows in each table

- ```
SELECT COUNT(*) EmpTable
FROM yEmp;
```
- ```
SELECT COUNT(*) ProdTable  
FROM yProd;
```
- ```
SELECT COUNT(*) OrdTable
FROM yOrd;
```
- ```
SELECT COUNT(*) OrderLineTable  
FROM yOrderLine;
```
- ```
SELECT COUNT(*) CustTable
FROM yCust;
```
- ```
SELECT COUNT(*) JobTitle  
FROM yJobTitle;
```

What is a sub-query?

- A sub-query is a query embedded inside another query.
- The sub-query is executed in the normal operation of the query in which it is embedded.
- The sub-query will return an “answer” result table to the query in which it is embedded.
- A sub-query can be placed in the SELECT list, FROM statement, WHERE clause &/or HAVING clause.
- Two types of sub-queries: non-correlated and correlated.

Non-correlated vs. correlated

- A **non-correlated** sub-query means that the inner query executes independently of the outer query.
- A **correlated** sub-query means that the inner query relies on data passed to it from the outer query.
- The outer query always relies on data generated from the inner query.

Which employees have the lowest payrate?

| | EmpID | EmpName | PayRate |
|---|-------|---------|---------|
| 1 | 7 | Johnson | 23.50 |
| 2 | 15 | Anand | 23.50 |
| 3 | 16 | Smith | 23.50 |

```
SELECT      MIN(PayRate)
FROM        yemp;
```

```
SELECT      EmpID,
            EmpName,
            MIN(PayRate)
FROM        yEmp;
```

```
SELECT      EmpID,
            EmpName,
            MIN(PayRate)
FROM        yEmp
GROUP BY    empid, empname
```

Final query for this task

```
SELECT      EmpID,  
            EmpName,  
            Payrate  
FROM        yemp  
WHERE       PayRate =  
            (SELECT MIN(Payrate)  
             FROM yEmp)
```

Outer Query

Inner Query

Which employees have a higher payrate than the average payrate for all employees?

| | EmpID | EmpName | Title | PayRate | AveragePayRate |
|---|-------|-----------|----------------------|---------|----------------|
| 1 | 1 | Martinson | Manager | 75.00 | 39.5937 |
| 2 | 2 | Polanski | Database Designer | 45.00 | 39.5937 |
| 3 | 3 | Torquez | Manager | 85.00 | 39.5937 |
| 4 | 4 | Ling | Interface Programmer | 65.00 | 39.5937 |
| 5 | 8 | Cheng | Interface Programmer | 45.00 | 39.5937 |
| 6 | 9 | Fukamota | Web Programmer | 40.00 | 39.5937 |

Which customers have orders in the yOrd table?

| | CustID | CustomerName |
|---|--------|----------------------------------|
| 1 | 1234 | Reston Supplies |
| 2 | 2555 | Mountain Design |
| 3 | 6773 | Nothing to It Culinary Institute |
| 4 | 6899 | Opaka Sporting Goods |
| 5 | 8372 | CutGlass Tile Company |

Which customers do NOT have orders in the yOrd table?

| | CustID | CustomerName |
|---|--------|-------------------------|
| 1 | 3441 | Oasis Clean Gear Supply |
| 2 | 5711 | Rodriguez Markets |

Which Employees have accepted an order in the yOrd table?

| | empid | empname | payrate | empmgrid | jobtitleid |
|---|-------|---------|---------|----------|------------|
| 1 | 4 | Ling | 65.00 | 3 | 50 |
| 2 | 5 | Bassett | 25.00 | 1 | 45 |
| 3 | 7 | Johnson | 23.50 | 3 | 40 |
| 4 | 10 | Stein | 30.00 | 1 | 45 |

Which Employees have NOT accepted an order in the yOrd table?

| | empid | empname | payrate | empmgrid | jobtitleid |
|----|-------|-----------|---------|----------|------------|
| 1 | 1 | Martinson | 75.00 | NULL | 10 |
| 2 | 2 | Polanski | 45.00 | 1 | 40 |
| 3 | 3 | Torquez | 85.00 | 1 | 10 |
| 4 | 6 | Martinez | 35.00 | 1 | 45 |
| 5 | 8 | Cheng | 45.00 | 1 | 50 |
| 6 | 9 | Fukamota | 40.00 | 3 | 45 |
| 7 | 11 | Nguyen | 35.00 | 3 | 45 |
| 8 | 12 | Duong | 28.00 | 2 | 20 |
| 9 | 13 | Patel | 30.00 | 2 | 20 |
| 10 | 14 | Agarwal | 25.00 | 2 | 20 |
| 11 | 15 | Anand | 23.50 | 2 | 20 |
| 12 | 16 | Smith | 23.50 | 3 | 20 |

Which orders were accepted by an employee?

| | OrderID | OrderDate | CustID | DueDate | empid |
|---|---------|-------------------------|--------|-------------------------|-------|
| 1 | 100 | 2025-03-15 00:00:00.000 | 1234 | 2025-03-19 00:00:00.000 | 4 |
| 2 | 200 | 2025-03-24 00:00:00.000 | 6773 | 2025-04-18 00:00:00.000 | 5 |
| 3 | 300 | 2025-03-22 00:00:00.000 | 6899 | 2025-05-01 00:00:00.000 | 5 |
| 4 | 400 | 2025-03-27 00:00:00.000 | 2555 | 2025-04-16 00:00:00.000 | 7 |
| 5 | 500 | 2025-04-11 00:00:00.000 | 6899 | 2025-04-22 00:00:00.000 | 7 |
| 6 | 520 | 2025-03-31 00:00:00.000 | 6773 | 2025-05-12 00:00:00.000 | 5 |
| 7 | 600 | 2025-04-15 00:00:00.000 | 2555 | 2025-04-27 00:00:00.000 | 7 |
| 8 | 700 | 2025-04-11 00:00:00.000 | 2555 | 2025-06-04 00:00:00.000 | 10 |
| 9 | 900 | 2025-05-12 00:00:00.000 | 6899 | 2025-08-31 00:00:00.000 | 7 |

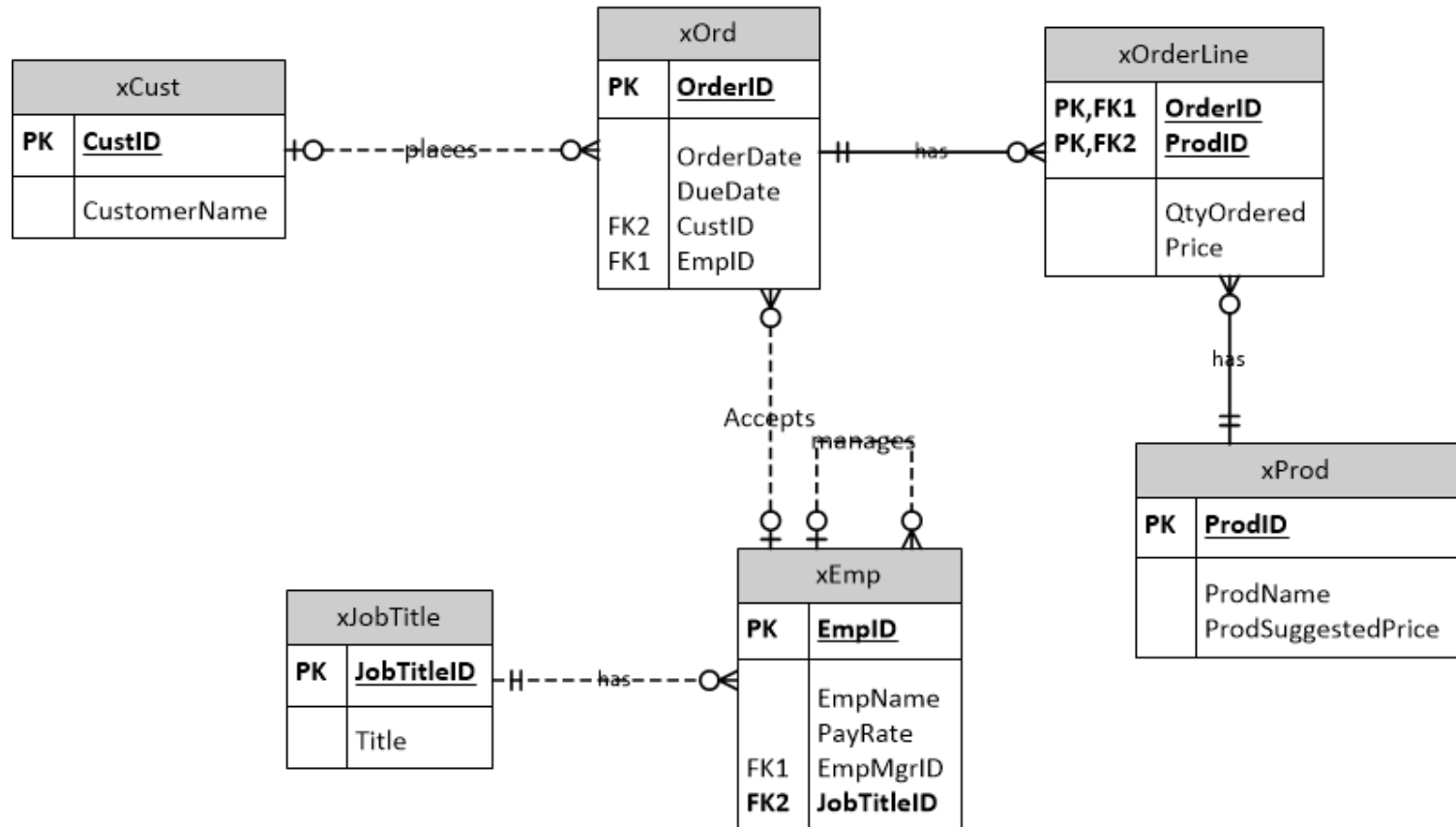
Which orders were NOT accepted by an employee?

| | OrderID | OrderDate | CustID | DueDate | empid |
|---|---------|-------------------------|--------|-------------------------|-------|
| 1 | 110 | 2025-03-30 00:00:00.000 | 8372 | 2025-04-12 00:00:00.000 | NULL |
| 2 | 800 | 2025-04-13 00:00:00.000 | 8372 | 2025-06-12 00:00:00.000 | NULL |

IS475/675 Agenda for 04/09/2025

- **Show** any requested answers for HW7
- **Do** SQL Lab Exercise 8
 - Contrast correlated and non-correlated sub-queries
 - Learn how to create and deploy a new database object – **SQL View** (ANSI-standard)
 - Learn how to create a memory-based SQL object – **Common Table Expression** (T-SQL)
- While waiting for class to start: Login to SQL Server Management Studio
- Create and populate six tables by opening and executing in SQL Server the file named:
K:\CoB\IS475\LabFiles\SQLLab8.sql

Here is the database design



Look at the content of the tables

- `SELECT` `*` `FROM` `xEmp;`
- `SELECT` `*` `FROM` `xProd;`
- `SELECT` `*` `FROM` `xOrd;`
- `SELECT` `*` `FROM` `xOrderLine;`
- `SELECT` `*` `FROM` `xCust`
- `SELECT` `*` `FROM` `xJobTitle;`

What is a sub-query?

- A sub-query is a query embedded inside another query.
- The sub-query is executed in the normal operation of the query in which it is embedded.
- The sub-query will return an “answer” result table to the query in which it is embedded.
- A sub-query can be placed in the SELECT list, FROM statement, WHERE clause &/or HAVING clause.
- Two types of sub-queries: non-correlated and correlated.

Non-correlated vs. correlated

- A **non-correlated** sub-query means that the inner query executes independently of the outer query.
- A **correlated** sub-query means that the inner query relies on data passed to it from the outer query.
- The outer query always relies on data generated from the inner query.

Let's start SQL Lab 8

- Run this script file if you didn't at the start of class: `K:\CoB\is475\labfiles\SQLLab8.sql`
- We are first going to create a non-correlated sub-query.
- Then we will change it to a correlated sub-query.
- Then we will change the code completely to make it more efficient with the use of a View.

This is the result table we are going to create

| | EmpID | EmpName | PayRate | Title | AveragePayRate | MaximumPayRate | MinimumPayRate |
|---|-------|-----------|---------|----------------------|----------------|----------------|----------------|
| 1 | 1 | Martinson | 75.00 | Manager | 39.5625 | 85.00 | 22.50 |
| 2 | 2 | Polanski | 45.00 | Database Designer | 39.5625 | 85.00 | 22.50 |
| 3 | 3 | Torquez | 85.00 | Manager | 39.5625 | 85.00 | 22.50 |
| 4 | 4 | Ling | 65.00 | Interface Programmer | 39.5625 | 85.00 | 22.50 |
| 5 | 8 | Cheng | 45.00 | Interface Programmer | 39.5625 | 85.00 | 22.50 |
| 6 | 9 | Fukamota | 40.00 | Web Programmer | 39.5625 | 85.00 | 22.50 |

The question we are answering is: Which employees have a payrate that is higher than the average payrate for all employees? The average payrate is calculated as the mean from the payrates in the xEmp table.

Which employees have a payrate that is less than or equal to the average payrate?

| | EmpID | EmpName | PayRate | Title | AveragePayRate | MaximumPayRate | MinimumPayRate |
|----|-------|----------|---------|-------------------|----------------|----------------|----------------|
| 1 | 5 | Bassett | 25.00 | Web Programmer | 39.5625 | 85.00 | 22.50 |
| 2 | 6 | Martinez | 35.00 | Web Programmer | 39.5625 | 85.00 | 22.50 |
| 3 | 7 | Johnson | 25.00 | Database Designer | 39.5625 | 85.00 | 22.50 |
| 4 | 10 | Stein | 30.00 | Web Programmer | 39.5625 | 85.00 | 22.50 |
| 5 | 11 | Nguyen | 35.00 | Web Programmer | 39.5625 | 85.00 | 22.50 |
| 6 | 12 | Duong | 28.00 | Business Analyst | 39.5625 | 85.00 | 22.50 |
| 7 | 13 | Patel | 30.00 | Business Analyst | 39.5625 | 85.00 | 22.50 |
| 8 | 14 | Agarwal | 25.00 | Business Analyst | 39.5625 | 85.00 | 22.50 |
| 9 | 15 | Anand | 22.50 | Business Analyst | 39.5625 | 85.00 | 22.50 |
| 10 | 16 | Smith | 22.50 | Business Analyst | 39.5625 | 85.00 | 22.50 |

Which employees have a payrate than is higher than the average payrate for their job title?

| | EmpID | EmpName | PayRate | Title | AveragePayRate |
|---|-------|----------|---------|----------------------|----------------|
| 1 | 2 | Polanski | 45.00 | Database Designer | 35.00 |
| 2 | 3 | Torquez | 85.00 | Manager | 80.00 |
| 3 | 4 | Ling | 65.00 | Interface Programmer | 55.00 |
| 4 | 6 | Martinez | 35.00 | Web Programmer | 33.00 |
| 5 | 9 | Fukamota | 40.00 | Web Programmer | 33.00 |
| 6 | 11 | Nguyen | 35.00 | Web Programmer | 33.00 |
| 7 | 12 | Duong | 28.00 | Business Analyst | 25.60 |
| 8 | 13 | Patel | 30.00 | Business Analyst | 25.60 |

The final query is very inefficient

```
SELECT emp.EmpID,  
       emp.EmpName,  
       emp.PayRate,  
       jt.Title,  
       (SELECT AVG(payrate)  
        FROM xEmp empSELECT  
        WHERE empSELECT.jobtitleid = emp.jobtitleid)  
       AveragePayRate  
FROM xEmp emp  
INNER JOIN xJobTitle jt  
ON emp.jobtitleid = jt.jobtitleid  
WHERE payrate >  
       (SELECT AVG(Payrate)  
        FROM xEmp empinner  
        WHERE empinner.jobtitleid = emp.jobtitleid)  
ORDER BY emp.empid
```

Correlated sub-queries

- The inner query relies on data from the outer query to create the result table produced by the inner query.
- They are inefficient and can slow down the processing of a SELECT.
- They can usually be replaced by a database object called a “view.”

What is a SQL view?

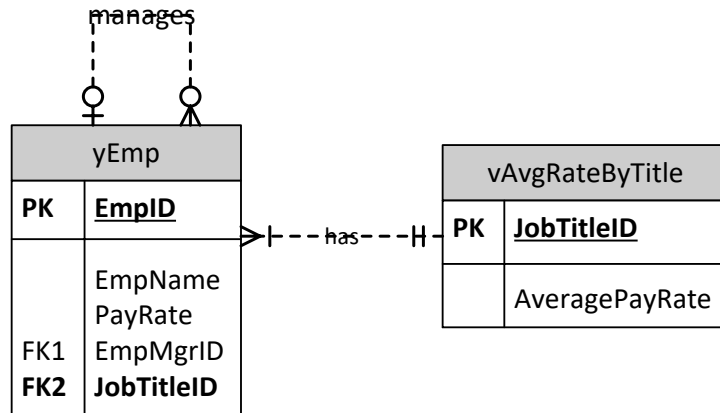
- A “virtual” table.
 - A set of SQL statements that creates a result table which can be accessed by other SQL statements.
- A database object.
 - The code for a view is stored in the database.
 - A view contains no data of its own.
 - A view relies on the data in the underlying/base tables used to create the view.
- A set of stored SQL code.
 - Stores code; not data.

We will solve the same problem with the use of a
SQL view.

Which employees have a payrate than is higher than the
average payrate for their job title?

| | EmpID | EmpName | PayRate | Title | AveragePayRate |
|---|-------|----------|---------|----------------------|----------------|
| 1 | 2 | Polanski | 45.00 | Database Designer | 35.00 |
| 2 | 3 | Torquez | 85.00 | Manager | 80.00 |
| 3 | 4 | Ling | 65.00 | Interface Programmer | 55.00 |
| 4 | 6 | Martinez | 35.00 | Web Programmer | 33.00 |
| 5 | 9 | Fukamota | 40.00 | Web Programmer | 33.00 |
| 6 | 11 | Nguyen | 35.00 | Web Programmer | 33.00 |
| 7 | 12 | Duong | 28.00 | Business Analyst | 25.60 |
| 8 | 13 | Patel | 30.00 | Business Analyst | 25.60 |

Join the view to the xEmp table



The view does not have a primary key because it does not contain data. As long as you include a field that can be used to join another table, the view acts like a table.

```
SELECT *
FROM xEmp
INNER JOIN vAvgRateByTitle
ON xEmp.jobtitleid =
vAvgRateByTitle.jobtitleid
```

| | empid | empname | payrate | empmgrid | jobtitleid | jobtitleID | AveragePayRate |
|----|-------|-----------|---------|----------|------------|------------|----------------|
| 1 | 1 | Martinson | 75.00 | NULL | 10 | 10 | 80.00 |
| 2 | 3 | Torquez | 85.00 | 1 | 10 | 10 | 80.00 |
| 3 | 12 | Duong | 28.00 | 2 | 20 | 20 | 25.60 |
| 4 | 13 | Patel | 30.00 | 2 | 20 | 20 | 25.60 |
| 5 | 14 | Agarwal | 25.00 | 2 | 20 | 20 | 25.60 |
| 6 | 15 | Anand | 22.50 | 2 | 20 | 20 | 25.60 |
| 7 | 16 | Smith | 22.50 | 3 | 20 | 20 | 25.60 |
| 8 | 2 | Polanski | 45.00 | 1 | 40 | 40 | 35.00 |
| 9 | 7 | Johnson | 25.00 | 3 | 40 | 40 | 35.00 |
| 10 | 5 | Bassett | 25.00 | 1 | 45 | 45 | 33.00 |
| 11 | 6 | Martinez | 35.00 | 1 | 45 | 45 | 33.00 |
| 12 | 9 | Fukamota | 40.00 | 3 | 45 | 45 | 33.00 |
| 13 | 10 | Stein | 30.00 | 1 | 45 | 45 | 33.00 |
| 14 | 11 | Nguyen | 35.00 | 3 | 45 | 45 | 33.00 |
| 15 | 8 | Cheng | 45.00 | 1 | 50 | 50 | 55.00 |
| 16 | 4 | Ling | 65.00 | 3 | 50 | 50 | 55.00 |

Why do SQL programmers use views?

- To break large queries into more manageable sections.
- To “hide” the underlying tables and make the structure of the database more secure.
- To create a virtual result that can be passed more easily to other programs.
- To cope with group functions.

What is a Common Table Expression (CTE)?

- A CTE is much like a view.
- A CTE creates a named virtual result table, just like a view.
- A CTE, however, is not a database object – it is only available in the session that is actively using the code.
- It is a temporary virtual result table, while a view is a more permanent virtual result table.
- A CTE is not ANSI-standard. It is available in MS SQL Server T-SQL.

View vs. CTE

| | View | Common Table Expression |
|---------------|--|---|
| Create/Store | Stored as a database object. | Not stored as a database object. Local to a single query. |
| Extent of Use | Use when the result table will be used in more than one query. | Use when the result table is local to single query. |
| Portability | Can be accessed by programs other than SQL. | Can only be used by SQL. |

“Group of a Group” type of query

- Usually trying to create a very small result table that answers a specific question.
- Identifying the MAX of a SUM, or the MIN of a COUNT.
- For example, maybe we want to know which customer placed the most orders with our company.
- Or we want to know which vendor delivered the most products that were damaged.

Task 6 in SQL Lab Exercise 8

The goal of this task is to determine which customer placed the most orders with our company based on a count of the orders.

| | CustID | CustomerName | CountOfOrders |
|---|--------|----------------------|---------------|
| 1 | 2555 | Mountain Design | 3 |
| 2 | 6899 | Opaka Sporting Goods | 3 |