

## Complex Queries:

Complex queries have 3 or more tables joined, a custom scalar function and use built-in SQL functions and group by summarization.

**Premise 01:** How many vacation days have been taken by each department?

**Detailed explanation of the problem:** Return all Departments and the calculated amount of vacation days each has taken.

**Query:**

```
CREATE FUNCTION dbo.Ufngetvacationdays (@VacationHours AS INT)
returns INT
AS
BEGIN
    RETURN @VacationHours / 24
END;

go

USE adventureworks2014;

SELECT humanresources.department.NAME,
       Sum(dbo.Ufngetvacationdays(Employee_1.vacationhours)) AS VacationDays
FROM   humanresources.employee AS Employee_1
       INNER JOIN humanresources.employeedepartmenthistory
                ON Employee_1.businessentityid =
                   humanresources.employeedepartmenthistory.businessentityid
       INNER JOIN humanresources.department
                ON humanresources.employeedepartmenthistory.departmentid =
                   humanresources.department.departmentid
GROUP BY humanresources.department.NAME
ORDER BY vacationdays;
```

**Sample Output:**

	Name	VacationDays
1	Engineering	2
2	Tool Design	2
3	Executive	4
4	Research and Development	6
5	Production Control	7
6	Sales	12
7	Human Resources	12
8	Marketing	12
9	Document Control	15
10	Finance	20
11	Quality Assurance	20
12	Shipping and Receiving	21
13	Facilities and Maintenance	21
14	Information Services	23
15	Purchasing	25
16	Production	284

**Premise 02:** Provide the Email Addresses for those who have taken the most vacation days.

**Detailed explanation of the problem:** Return the first name, last name, and e-mail address of the top 10 employees with the most vacation days.

**Query:**

```
SELECT TOP (10) person.person.firstname,
               person.person.lastname,
               person.emailaddress.emailaddress,
               dbo.Ufngetvacationdays(Employee_1.vacationhours) AS VacationDays
FROM   humanresources.employee AS Employee_1
       INNER JOIN person.person
              ON Employee_1.businessentityid = person.person.businessentityid
       INNER JOIN person.emailaddress
              ON person.person.businessentityid =
                 person.emailaddress.businessentityid
ORDER  BY vacationdays DESC;
```

**Sample Output:**

	FirstName	LastName	EmailAddress	VacationDays
1	Ken	Sánchez	ken0@adventure-works.com	4
2	Betsy	Stadick	betsy0@adventure-works.com	4
3	Patrick	Wedge	patrick0@adventure-works.com	4
4	Danielle	Tiedt	danielle0@adventure-works.com	4
5	Tom	Vande Velde	tom0@adventure-works.com	4
6	Linda	Moschell	linda0@adventure-works.com	4
7	Mindaugas	Krapauskas	mindaugas0@adventure-works.com	4
8	Michael	Patten	michael2@adventure-works.com	4
9	Chad	Niswonger	chad0@adventure-works.com	4
10	Susan	Eaton	susan0@adventure-works.com	4

**Premise 03:** Which states/regions have the most orders shipping within 2 weeks?

**Detailed explanation of the problem:** Return all the States and Countries, and the amount of Orders that are shipping in rates less than 2 weeks.

### Query:

```
CREATE FUNCTION dbo.Getshippingdays (@shipdate AS DATE,  
                                     @duedate AS DATE)  
  
returns INT  
AS  
BEGIN  
    RETURN Datediff(year, @shipdate, @duedate)  
END;  
  
go  
  
SELECT person.stateprovince.stateprovincecode,  
       person.stateprovince.countryregioncode,  
       Count(salesorderid) AS OrderCount  
FROM   person.stateprovince  
       INNER JOIN person.address  
           ON person.stateprovince.stateprovinceid =  
              person.address.stateprovinceid  
       INNER JOIN sales.salesorderheader  
           ON person.address.addressid =  
              sales.salesorderheader.billtoaddressid  
           AND person.address.addressid =  
              sales.salesorderheader.shiptoaddressid  
WHERE  dbo.Ufngetshippingdays(salesorderheader.shipdate,  
                               salesorderheader.duedate) < 14  
GROUP BY person.stateprovince.stateprovincecode,  
          person.stateprovince.countryregioncode  
ORDER BY ordercount DESC;
```

### Sample Output:

	StateProvinceCode	CountryRegionCode	OrderCount
1	CA	US	5882
2	BC	CA	3472
3	ENG	GB	3219
4	NSW	AU	3009
5	WA	US	2990
6	VIC	AU	1710
7	QLD	AU	1461
8	OR	US	1380
9	SL	DE	658
10	75	FR	575
11	NW	DE	570
12	HE	DE	560
13	SA	AU	466
14	HH	DE	441
15	93	FR	391
16	59	FR	387
17	BY	DE	344
18	ON	CA	339
19	92	FR	295
20	TX	US	241

**Premise 04:** Which sales country has taken the most vacation days?

**Detailed explanation of the problem:** Return the top country with the most vacation days taken.

**Query:**

```
SELECT TOP (1) sales.salesterritory.countryregioncode,
               Count(dbo.Ufngetvacationdays(Employee_1.vacationhours)) AS
               VacationDays
FROM   humanresources.employee AS Employee_1
       INNER JOIN sales.salesperson
               ON Employee_1.businessentityid =
                  sales.salesperson.businessentityid
       INNER JOIN sales.salesterritory
               ON sales.salesperson.territoryid =
                  sales.salesterritory.territoryid
GROUP BY salesterritory.countryregioncode
ORDER BY vacationdays DESC;
```

**Sample Output:**

	CountryRegionCode	VacationDays
1	US	8

**Premise 05:** What is the average amount of sales per month for each store?

**Detailed explanation of the problem:** Return all store names, and the calculated average of their sales per month.

**Query:**

```
CREATE FUNCTION dbo.Avgsalespermonth(@SalesLastYear AS INT)
returns INT
AS
BEGIN
    RETURN @SalesLastYear / 12
END

go

SELECT sales.store.NAME,
       dbo.Ufnavgsalespermonth(salesterritory.saleslastyear) AS AvgSalesPerMonth
FROM   sales.store
       INNER JOIN sales.customer
               ON sales.store.businessentityid = sales.customer.storeid
       INNER JOIN sales.salesterritory
               ON sales.customer.territoryid = sales.salesterritory.territoryid
ORDER BY sales.store.NAME;
```

**Sample Output:**

	Name	AvgSalesPerMonth
1	A Bicycle Association	300595
2	A Bike Store	274891
3	A Bike Store	274891
4	A Cycle Shop	274891
5	A Great Bicycle Company	267084
6	A Great Bicycle Company	267084
7	A Typical Bike Shop	447214
8	A Typical Bike Shop	447214
9	Acceptable Sales & Service	474499
10	Acceptable Sales & Service	474499
11	Accessories Network	199711
12	Accessories Network	199711
13	Acclaimed Bicycle Company	327089
14	Acclaimed Bicycle Company	327089
15	Ace Bicycle Supply	199711
16	Ace Bicycle Supply	199711
17	Action Bicycle Specialists	136318
18	Action Bicycle Specialists	136318
19	Active Cycling	300595
20	Active Cycling	300595

**Premise 06:** How many employees were hired in January for each department?

**Detailed explanation of the problem:** Return all Departments and the count of employees hired in January.

**Query:**

```
CREATE FUNCTION dbo.Gethiremonth (@HireDate AS DATE)
returns INT
AS
BEGIN
    RETURN Month(@HireDate)
END;

go

SELECT humanresources.department.NAME,
       Count(dbo.Gethiremonth(Employee_1.hiredate)) AS HiredinMonth
FROM   humanresources.employeehistory
       INNER JOIN humanresources.employee AS Employee_1
           ON humanresources.employeehistory.businessentityid =
              Employee_1.businessentityid
       INNER JOIN humanresources.department
           ON humanresources.employeehistory.departmentid =
              humanresources.department.departmentid
WHERE  dbo.Gethiremonth(Employee_1.hiredate) = 1
GROUP BY humanresources.department.NAME
ORDER BY hiredinmonth DESC;
```

**Sample Output:**

	Name	HiredinMonth
1	Production	53
2	Finance	5
3	Purchasing	5
4	Marketing	4
5	Engineering	4
6	Information Services	3
7	Document Control	2
8	Production Control	2
9	Executive	2
10	Shipping and Receiving	2
11	Quality Assurance	1
12	Research and Development	1
13	Sales	1
14	Facilities and Maintenance	1
15	Human Resources	1

**Premise 07:** Provide the E-Mail Addresses for employees hired in June.

**Detailed explanation of the problem:** Return all Email Addresses for Employees hired in June.

**Query:**

```

SELECT person.person.firstname,
       person.person.lastname,
       person.emailaddress.emailaddress
FROM   humanresources.employee AS Employee_1
       INNER JOIN person.person
           ON Employee_1.businessentityid = person.person.businessentityid
           AND Employee_1.businessentityid =
               person.person.businessentityid
           AND Employee_1.businessentityid =
               person.person.businessentityid
           AND Employee_1.businessentityid =
               person.person.businessentityid
       INNER JOIN person.emailaddress
           ON person.person.businessentityid =
               person.emailaddress.businessentityid
WHERE  dbo.Gethiremonth(Employee_1.hiredate) = 5
GROUP BY person.person.firstname,
         person.person.lastname,
         person.emailaddress.emailaddress
ORDER BY person.firstname;

```

**Sample Output:**

	FirstName	LastName	EmailAddress
1	David	Campbell	david8@adventure-works.com
2	Garrett	Vargas	garrett1@adventure-works.com
3	Jae	Pak	jae0@adventure-works.com
4	Jillian	Carson	jillian0@adventure-works.com
5	José	Saraiva	josé1@adventure-works.com
6	Linda	Mitchell	linda3@adventure-works.com
7	Lynn	Tsoflias	lynn0@adventure-works.com
8	Michael	Blythe	michael9@adventure-works.com
9	Michael	Raheem	michael6@adventure-works.com
10	Pamela	Ansman-Wolfe	pamela0@adventure-works.com
11	Rachel	Valdez	rachel0@adventure-works.com
12	Ranjit	Varkey Chudukatil	ranjit0@adventure-works.com
13	Shu	Ito	shu0@adventure-works.com
14	Tsvi	Reiter	tsvi0@adventure-works.com

**Premise 08:** Provide the same information as above for employees hired on the same month as their birthday.

**Detailed explanation of the problem:** Return all Email Addresses for employees hired on their birth month.

**Query:**

```

SELECT person.person.firstname,
       person.person.lastname,
       person.emailaddress.emailaddress
FROM   humanresources.employee AS Employee_1
       INNER JOIN person.person
           ON Employee_1.businessentityid = person.person.businessentityid
           AND Employee_1.businessentityid =
               person.person.businessentityid
           AND Employee_1.businessentityid =
               person.person.businessentityid
           AND Employee_1.businessentityid =
               person.person.businessentityid
       INNER JOIN person.emailaddress
           ON person.person.businessentityid =
               person.emailaddress.businessentityid
WHERE   dbo.Gethiremonth(Employee_1.hiredate) = Month(Employee_1.birthdate)
GROUP BY person.person.firstname,
         person.person.lastname,
         person.emailaddress.emailaddress

```

**Sample Output:**

	FirstName	LastName	EmailAddress
1	Alejandro	McGuel	alejandro0@adventure-works.com
2	Annik	Stahl	annik0@adventure-works.com
3	Benjamin	Martin	benjamin0@adventure-works.com
4	Betsy	Stadick	betsy0@adventure-works.com
5	Brian	Goldstein	brian2@adventure-works.com
6	Dan	Wilson	dan1@adventure-works.com
7	David	Ortiz	david2@adventure-works.com
8	Diane	Glimp	diane0@adventure-works.com
9	Dragan	Tomic	dragan0@adventure-works.com
10	Dylan	Miller	dylan0@adventure-works.com
11	Eric	Gubbels	eric0@adventure-works.com
12	Erin	Hagens	erin0@adventure-works.com
13	Eugene	Kogan	eugene1@adventure-works.com
14	Gigi	Matthew	gigi0@adventure-works.com
15	Janet	Sheperdigian	janet0@adventure-works.com
16	Jean	Trenary	jean0@adventure-works.com
17	Jeff	Hay	jeff0@adventure-works.com
18	John	Frum	john3@adventure-works.com
19	Ken	Sánchez	ken0@adventure-works.com
20	Kevin	Lin	kevin1@adventure-works.com

**Premise 09:** Who was hired in 2009, and what are their phone numbers?

**Detailed explanation of the problem:** Return the Last Name and Phone Numbers associated with the employees hired in 2009.

**Query:**

```

CREATE FUNCTION dbo.Gethireyear (@HireDate AS DATE)
returns INT
AS
BEGIN
    RETURN Year(@HireDate)
END;

go

SELECT person.person.lastname,
       person.personphone.phonenumber
FROM   person.personphone
       INNER JOIN humanresources.employee AS Employee_1
           ON person.personphone.businessentityid =
              Employee_1.businessentityid
       INNER JOIN person.person
           ON person.personphone.businessentityid =
              person.person.businessentityid
       AND person.personphone.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
       AND Employee_1.businessentityid =
           person.person.businessentityid
WHERE  ( dbo.Gethireyear(Employee_1.hiredate) ) = 2009
ORDER BY lastname;

```



### Sample Output:

	LastName	PhoneNumber
1	Abolrous	869-555-0125
2	Ackerman	577-555-0185
3	Adams	407-555-0165
4	Ajenstat	785-555-0110
5	Altman	110-555-0112
6	Anderson	970-555-0118
7	Arfin	204-555-0115
8	Bacon	166-555-0159
9	Baker	712-555-0113
10	Baker	283-555-0185
11	Barbariol	150-555-0194
12	Barber	477-555-0132
13	Berg	654-555-0177
14	Berge	746-555-0164
15	Berglund	181-555-0124
16	Berndt	139-555-0120
17	Bischoff	927-555-0168
18	Brewer	438-555-0172
19	Caron	238-555-0116
20	Cassini	207-555-0182

**Premise 10:** Provide the average pay rates of employees hired in 2010 for each department?

**Detailed explanation of the problem:** Return all Departments and the average pay rate for each one. Make sure it is for 2010 employees only.

### Query:

```
SELECT humanresources.department.NAME,  
       Avg(humanresources.employeepayhistory.rate) AS AvgPayRate  
FROM   humanresources.employee AS Employee_1  
       INNER JOIN humanresources.employeedepartmenthistory  
                 ON Employee_1.businessentityid =  
                    humanresources.employeedepartmenthistory.businessentityid  
       INNER JOIN humanresources.department  
                 ON humanresources.employeedepartmenthistory.departmentid =  
                    humanresources.department.departmentid  
       INNER JOIN humanresources.employeepayhistory  
                 ON Employee_1.businessentityid =  
                    humanresources.employeepayhistory.businessentityid  
WHERE  dbo.Gethireyear(Employee_1.hiredate) = 2010  
GROUP BY NAME  
ORDER BY avgpayrate DESC;
```

### Sample Output:

	Name	AvgPayRate
1	Engineering	36.0577
2	Tool Design	26.9231
3	Purchasing	16.8894
4	Production	13.70
5	Quality Assurance	10.5769
6	Facilities and Maintenance	9.25