

Functions - Recap

Transform a value or set of values using some rule

Built into the SQL standard

Problem: Microsoft uses many of its own function names to maintain compatibility with Excel and Word functions

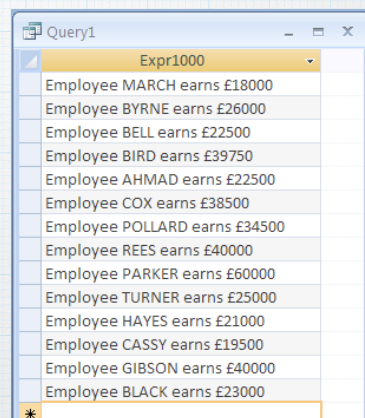
Functions Categories

String	concatenation, length, substring
Arithmetic	max, min, power, round, trunc
Date	add, subtract dates
Aggregate or group	average, sum, count

String Function - String Concatenation &

Combines fields with additional text if required

```
select "Employee "&ename&"  
     earns £"&sal  
from emp
```

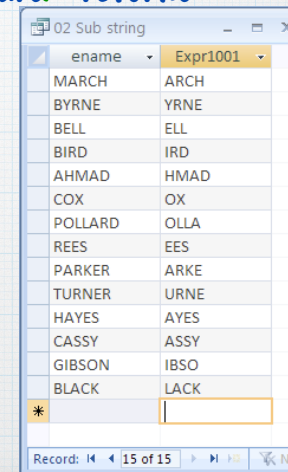


Expr1000
Employee MARCH earns £18000
Employee BYRNE earns £26000
Employee BELL earns £22500
Employee BIRD earns £39750
Employee AHMAD earns £22500
Employee COX earns £38500
Employee POLLARD earns £34500
Employee REES earns £40000
Employee PARKER earns £60000
Employee TURNER earns £25000
Employee HAYES earns £21000
Employee CASSY earns £19500
Employee GIBSON earns £40000
Employee BLACK earns £23000

String Function - Substring

mid (string, starting point, no of chars) - returns part of a string

```
select ename, mid (ename, 2, 4)  
from emp
```

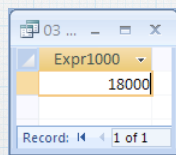


ename	Expr1001
MARCH	ARCH
BYRNE	YRNE
BELL	ELL
BIRD	IRD
AHMAD	HMAD
COX	OX
POLLARD	OLLA
REES	EES
PARKER	ARKE
TURNER	URNE
HAYES	AYES
CASSY	ASSY
GIBSON	IBSO
BLACK	LACK

Arithmetic Function - min

min () - returns smallest value in a column

```
select min (sal)
from emp
```

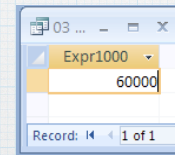


SAL
18000
26000
22500
39750
22500
38500
34500
40000
60000
25000
21000
19500
40000
23000

Arithmetic Function - max

max () - returns largest value in a column

```
select max (sal)
from emp
```

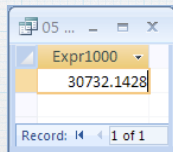


SAL
18000
26000
22500
39750
22500
38500
34500
40000
60000
25000
21000
19500
40000
23000

Aggregate Function - avg

avg () - returns mean value in a column

```
select avg (sal)
from emp
```

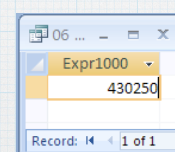


SAL
18000
26000
22500
39750
22500
38500
34500
40000
60000
25000
21000
19500
40000
23000

Aggregate Function - sum

sum () - returns total of all values in a column

```
select sum (sal)
from emp
```



SAL
18000
26000
22500
39750
22500
38500
34500
40000
60000
25000
21000
19500
40000
23000

Aggregate Function - count

`count ()` - returns total number of values in a column

`select count (sal)
from emp`

Expr1000
14

Record: 14 of 1

SAL
18000
26000
22500
39750
22500
38500
34500
40000
60000
25000
21000
19500
40000
23000

Group by

Aggregate functions can be applied to subsets of the table by using the `group by` syntax

`select * or expression
from relations
[where expression]
[group by expression]`

Group by

Aggregate functions can be applied to subsets of the table by using the `group by` syntax

e.g.

Calculate `avg()`

Calculate `avg()`

Calculate `avg()`

ENAME	JOB	MGR	HIREDATE	SAL
BLACK	ADMIN	818	21/11/1997	23000
CASSY	ADMIN	734	23/07/2002	19500
HAYES	ADMIN	824	04/06/2001	21000
MARCH	ADMIN	938	13/06/1997	18000
GIBSON	ANALYST	602	05/12/1997	40000
REES	ANALYST	602	05/03/2000	40000
POLLARD	MANAGER	875	14/05/2000	34500
COX	MANAGER	875	11/06/2002	38500
BIRD	MANAGER	875	31/10/1997	39750
PARKER	PRESIDENT		09/07/2002	60000
TURNER	SALES	734	04/06/2001	25000
AHMAD	SALES	734	05/12/1997	22500
BELL	SALES	734	26/03/2000	22500
BYRNE	SALES	734	15/08/1997	26000

Group by

Aggregate functions can be applied to subsets of the table by using the `group by` syntax

`select job, avg(sal)
from emp
group by job`

job	Expr1001
ADMIN	20375
ANALYST	40000
MANAGER	37583.333333
PRESIDENT	60000
SALES	24000

Record: 14 of 5

Group by

Find the highest salary for each job category

```
select job, max(sal)
from emp
group by job
```

job	Expr1001
ADMIN	23000
ANALYST	40000
MANAGER	39750
PRESIDENT	60000
SALES	26000

Nested sub-queries

When one of the conditions of a WHERE clause is a query itself, this is called a nested sub-query, i.e.,

```
SELECT select-list
FROM table(s)
WHERE object operator (SELECT select-list
FROM table(s)
[WHERE condition]);
```

Nested sub-queries example

Find all the employees who earn more than the average salary

Start by finding the average salary

```
select avg(sal)
from emp
```

Expr1000
30732.1428

Nested sub-queries example

Find all the employees who earn more than the average salary

Now find all the employees that earn more than this:

```
select ename, sal
from emp
where sal >= (select avg(sal)
from emp)
```

ename	sal
BIRD	39750
COX	38500
POLLARD	34500
REES	40000
PARKER	60000
GIBSON	40000

Nested sub-queries example (2)

Typically, you need to use a nested sub-query where you need to use an aggregate function and an attribute at the same time

Find the employee who earns the least money

Attribute required

Aggregate function required

Nested sub-queries example (2)

Find the employee who earns the least money

First attempt may try something like this:

```
select ename, min(sal)
from ...
where ...
```

many values

single value

This can't work

Nested sub-queries example (2)

Find the employee who earns the least money

Better attempt:

```
select min(sal)
from emp
```

Find the minimum salary

Nested sub-queries example (2)

Find the employee who earns the least money

Better attempt:

```
select ename, sal
from emp
where sal = (select min(sal)
              from emp)
```

Find the person who has the minimum salary

ename	sal
MARCH	18000
*	

Record: 14 | 2 of 2