Functions, Aggregate and Group-set functions

Functions

Functions fall into four categories:

- Sting functions
- Arithmetic functions
- Date functions
- Aggregate or Group-set functions

Many string functions work on arithmetic expressions and dates, and then automatically convert them to a string format. Note that Microsoft products typically uses operators and functions from Visual Basic/Excel rather than the SQL standard.

String functions

String concatenation: `string1 & string2`

Joins string1 with string2. Note that the AS command is needed to produce a sensible column name. Different symbols are used for differing applications – Microsoft maintains consistency with Visual Basic by using & whilst the SQL standard uses ||.

```
select ename & " " & job AS combined
from emp;
```

Length: `Length(string)` or `Len(string)` (Microsoft)

Determines the length of a string.

```
select ename, len(ename)
from emp;
```

Substring: `substr(string, x, y)` or `mid(string, x, y)` (Microsoft)

Extracts a string of length y from the string starting at position x.

E.g: `mid("hello",2,3)` would return “ell”

```
select ename, mid(ename,2,3)
from emp;
```

Arithmetic functions

Greatest value: `greatest(columnname)` or `max(columnname)` (Microsoft)

Returns the greatest value in a column.

```
SELECT max(sal)
FROM emp;
```

Least value: `least(columnname)` or `min(columnname)` (Microsoft)

Returns the least value in a column.

```
SELECT min(sal)
FROM emp;
```

Power: `power(x,y)` or `x^y` (Microsoft)

Returns x to the power y.

```
SELECT len(ename)^2
FROM emp;
```

Round: `round(x,y)`

Rounds the number x to y places (not available in versions of Access 97 or earlier)

```
SELECT round(sal/2.3, 2)
FROM emp;
```

Trunc: `trunc(x,y)`

Truncates the floating point number x to y decimal places (not available in Access).

```
SELECT sal/2.3 as example, trunc(example,2)
from emp;
```

Has to be implemented using a combination of Int and / in Access

```
SELECT sal/2.3 as example, (int(example*100))/100
from emp;
```

Date functions

Extract the day, month or year part of a date as an integer: Use `day()`, `month()` or `year()`

```
select day(hiredate), month(hiredate), year(hiredate)
from emp
```

Increase or decrease a date: Use + or – a number of days

```
select hiredate+20 from emp;
```

Subtract one date from another: Use date – date

Aggregate functions

Aggregate functions operate on a number of rows (for example a whole table)

Calculate and average: `avg (x)` where x is generally a column.

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

Total a column: `sum (x)` where x is generally a column.

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

Count a number of items: `count(x)` where x is a column.

```
select count(sal)
from emp;
```

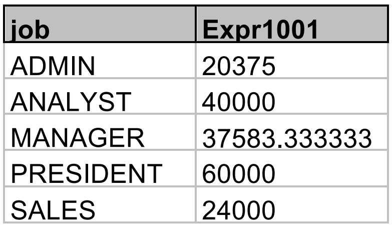
Grouping with Aggregate functions

Aggregate functions may be applied to several subsets of the table. The GROUP BY clause divides (sorts) a table into groups of rows with matching values in the same column (or columns).

Example: Calculate the average salary for each different type of job

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

Produces:

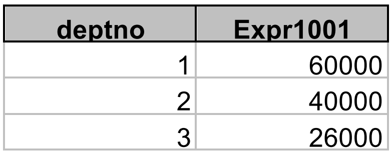


job	Expr1001
ADMIN	20375
ANALYST	40000
MANAGER	37583.333333
PRESIDENT	60000
SALES	24000

Example: Show the highest salary for each department excluding managers

```
select deptno, max(sal)
from emp
where job<>"Manager"
group by deptno;
```

Produces:



deptno	Expr1001
1	60000
2	40000
3	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]);
```

The sub-query must be enclosed in ( ) , as indeed must any levels of nested sub-query.

Example:

To find the person or people who earn the lowest salary in the company, the query must first calculate the lowest salary and then find the person who earns this salary.

This is a consequence of not being able to place both non-aggregated columns and aggregate functions in the same select list.

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

The nested query is executed first and the value retrieved is 'substituted' into the WHERE clause.

Note that sub-queries may be nested to any number of levels i.e.,

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

Exercises

For the following queries, write down the SQL and the result.

1. **Display the most recently employed person**

2. **Display the lowest salary earned by someone working in London.**

3. **Display the manager with the highest salary**

4. **Find the average admin salary**

5. **List the minimum and maximum salaries for each job**

6. **Find the average salary and average total compensation for each job type**

7. **Display the manager who most quickly employed a new person, along with that persons details (HARD)**