Lecture 08: Queries against one table

1. *SQL*
2. *Queries against one table*
3. *Order by*
4. *The functions upper, lower and initcap*
5. *Concatenation*
6. *Column alias*
7. *Distinct*
8. *NULL values and the nvl function*
9. *Between*
10. *Group by*
11. *Aggregate functions in SQL: max, min, sum and avg*
12. *Sub queries*

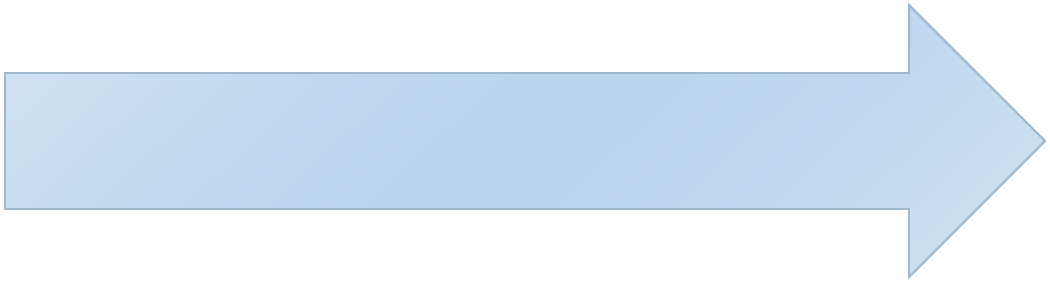
Pär Douhan, pdo@du.se

Exercise

Exercise table with data:

```
SQL> desc emp
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 EMPNO                                     NOT NULL NUMBER(3)
 FNAME                                              VARCHAR2(15)
 ENAME                                              VARCHAR2(15)
 DEPT                                               VARCHAR2(10)
 SAL                                                NUMBER(6)
 BONUS                                              NUMBER(4)
 HIREDATE                                           DATE
```

Let's begin



The select statement

SQL, *Data Retrieval Language* = select

Used when we want to read data from the database

The select statement

```
select = which columns should be included in the result.
from = from which table/tables data should be retrieved
where = conditions for rows to be fulfilled.
group by = groups columns that aggregate functions do not work with.
having = conditions for columns that aggregate functions work with.
order by = sort result ascending (A-Z) or descending (Z-A).
```

E1

E1: Show all rows, all columns.

```
select *
from emp;


-- * = all columns, not optimal! The no "cashing" of the data blocks.


select empno,fname,ename,dept,sal,bonus,hiredate
from emp;
```

E2

E2: Show all rows, and the columns FNAME, ENAME and DEPT

```
select fname,ename,dept
from emp;
```

```
FNAME           ENAME           DEPT
--------------- --------------- ---------
Bo              Ek              Engineering
Ewa             Ek              Engineering
Rolf            Svensson        Engineering
Raoul           Ortiz           Advertisement
lena            Olsson          HR
Arman           Trebic          Advertising
Ola             Ek              Engineering
Bosse           Karlmats        hr
Lenita          Ekström         ENGINEERING
bo              larsson         finance
```

E3

E3: Same as E2, but order by DEPT descending

```
-- data in tables are allways unsorted!

asc = a-z, 1-1000, i.e. ascending (is default)
desc = z-a, 1000-1, i.e. descending


select fname,ename,dept
from emp
order by dept desc;
```

FNAME	ENAME	DEPT
lena	Olsson	hr
Bosse	Karlmats	hr
bo	larsson	finance
Bo	Ek	Engineering
Ewa	Ek	Engineering
Ola	Ek	Engineering
Rolf	Svensson	Engineering
Lenita	Ekström	ENGINEERING
Raoul	Ortiz	Advertisement
Arman	Trebic	Advertisement

z-a

E4

E4: Show EMPNO,ENAME,SAL for those who works on the engineering department.

```
select empno,ename,sal
from emp
where dept = 'Engineering';    -- Problem!

The functions upper() or lower() or initcap() solves the problem!

select empno,ename,sal
from emp
where lower(dept) = 'engineering';
```

```
EMPNO ENAME           SAL
----- --------------- ----------
    1 Ek              22400
    2 Ek              24400
    3 Svensson        22400
    7 Ek              22400
    9 Ekström         29400
```

E5

E5: Show FNAME concatenated with ENAME under the heading name. Capitalize the first letter of the names. Order by ENAME ascending.

```
select initcap(fname)||' '||initcap(ename) as name
from emp
order by ename asc;
```

```
NAME
---------------
Bo Ek
Ewa Ek
Ola Ek
Lenita Ekström
Bosse Karlmats
Bo Larsson
Lena Olsson
Raoul Ortiz
Rolf Svensson
Arman Trebic
```

`name` is column alias for the column to the left.

`as` is optional!

`||` = concatenate operator in Oracle

`'Kalle'||'Andersson' = 'KalleAndersson'`

`'Anna'||' '||'Ekholm' = 'Anna Ekholm'`

`|` = Alt Gr + > < |

(the key to the right of left shift)

E6

E6: Show all the departments with capital letters, without duplicates.

```
select distinct upper(dept) departments
from emp;
```

`distinct` removes duplicates!

```
DEPARTMENTS
----------
FINANCE
HR
ADVERTISING
ENGINEERING
```

Column alias for 'distinct upper(dept)' without as.

E7

E7: Show FNAME, ENAME, SAL + bonus with the heading Total Sal for everyone who belongs to the engineering department, order by TOTAL SAL descending.

```
select fname,ename,sal + nvl(bonus,0) as "Total Sal"
from emp
where upper(dept) = 'ENGINEERING'
order by sal + nvl(bonus,0) desc;
```

`nvl(bonus,0)` = if the column bonus contain a `null` values, the `null` value will be replaced with 0 (zero).

FNAME	ENAME	Total Sal:
Lenita	Ekström	30500
Ola	Ek	27150
Ewa	Ek	24400
Bo	Ek	23700
Rolf	Svensson	23700

"Alias" within double quotation marks gives exact matching of the column header.

E8

E8: Show all rows and columns for those without any bonus

```
select *
from emp
where bonus = null;
where bonus is null;
```

EMPNO	FNAME	ENAME	DEPT	SAL	BONUS	HIREDATE
2	Ewa	Ek	Engineering	24400		1987-08-01
5	lena	Olsson	HR	27540		1991-02-25
6	Arman	Trebic	Advertisement	19850		2005-06-01
8	Bosse	Karlmats	HR	23560		1999-02-25
10	bo	larsson	finance	25400		

E9

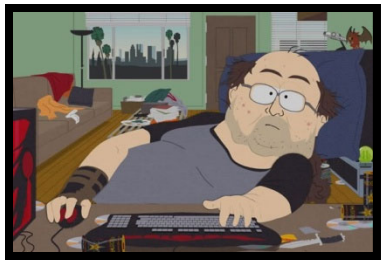
E9: Show FNAME, ENAME, HIREDATE. Replace every null value in the column hiredate with the string 'missing'.

```
select fname,ename, nvl(to_char(hiredate,'YYYY-MM-DD'),'missing') hiredate
from emp;
```

for

alias

FNAMN	ENAMN	HIREDATE
Bo	Ek	1991-02-25
Ewa	Ek	1987-08-01
Rolf	Svensson	1991-02-25
Raoul	Ortiz	2002-01-01
lena	Olsson	1991-02-25
Arman	Trebic	2005-06-01
Ola	Ek	1997-02-01
Bosse	Karlmats	1999-02-25
Lenita	Ekström	1991-02-25
bo	larsson	missing



E10

E10: Show EMPNO, FNAME, ENAME on the staff who have EMPNO between 3 and 6.

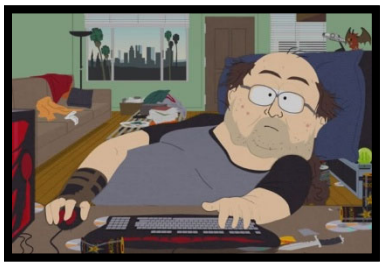
```
select empno,fname,ename
from emp
where empno between 3 and 6;
```

```
EMPNO FNAME           ENAME
----- --------------- --------
    3 Rolf            Svensson
    4 Raoul           Ortiz
    5 lena            Olsson
    6 Arman           Trebic
```

E11

E11: Show EMPNO for those in the staff who earns more than 24000 per month and are called 'Ek' in last name.

```
select empno
from emp
where lower(ename) = 'ek'
and sal > 24000;
```



```
EMPNO
-----
  2
```

E12

E12: Show ENAME for those in the staff that are named 'rolf', 'lena' or 'raoul' in first name.

```
select ename
from emp
where initcap(fname) in('Rolf','Lena','Raoul');
```

```
ENAMN
--------
Svensson
Ortiz
Olsson
```

E13

E13: Show DEPT, number of employees on every department under the heading 'number'.

```
select initcap(dept) department,count(empno) number
from emp
group by initcap(detp);
```

```
DEPT            NUMBER
--------------- ----------
Finance          1
HR               2
Advertisement    2
Engineering      5
```

Aggregate functions

Aggregate functions or group functions, perform calculations.

```
select initcap(dept) department,count(empno) number
from emp
group by initcap(dept);
```

DEPARTMENT	NUMBER
Finance	1
HR	2
HR	1
Advertisement	2
Advertisement	1
Engineering	5
Engineering	1
Engineering	1
Engineering	1
Engineering	1

`count` returns the number of rows.
`max` returns the maximum value.
`min` returns the minimum value.
`avg` returns the average value.
`sum` returns the sum.

Group functions returns a single value.

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

31200

E14

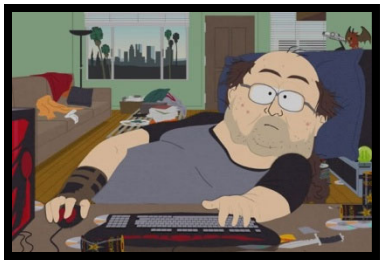
E14: Same as E13, but only the departments with more than 3 employees.

```
select initcap(dept) department,count(empno) number
from emp
group by initcap(dept)
having count(empno) > 3;
```

Condition for group functions

having

```
DEPT          NUMBER
------------ ----------
Engineering   5
```

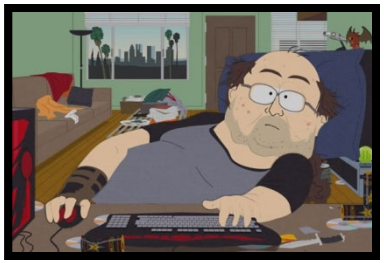


E15

E15: Show how many rows there are in the emp table.

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

```
  COUNT(*)
---------
        10
```



E16

E16: Show total salary cost for each department during a year.

```
select initcap(dept),sum(sal) * 12
from emp
group by initcap(dept);
```

```
INITCAP(DEPT)    SUM(SAL)*12
--------------   -----------
Finance          304800
HR               613200
Advertisement    612600
Engineering      1452000
```

E17

E17: Show FNAME, ENAME, SAL for the worker with the highest salary.

The nested(=inner) select-statement will execute first! (sub-query)

```
select fname,ename,sal
from emp
where sal = (select max(sal)
             from emp);
```

```
select fname,ename,sal
from emp
where sal = 31200;
```

FNAME	ENAME	SAL
Raoul	Ortiz	31200

E18

E18: Show EMPNO, FNAME for those employees who have a last name which start with the letter 'e' and ends with the letter 'k' and was hired before the year of 2000.

```
select empno,fname
from emp
where upper(fname) like 'E%'
and upper(ename) like '%K'
and hiredate < to_date('2000-01-01','YYYY-MM-DD');

to_char(hiredate,'YYYY') < '2000';
```

```
EMPNO FNAME
----- -----
    1 Bo
    2 Ewa
    7 Ola
```

% = "wild card" operator

Used along with the like operator

The End