Michael Longstreth
Module 5
November 5, 2017

1.

```
SQL> SELECT MIN(salary), AVG(salary), MAX(salary), job_id
  2  FROM employees
  3  GROUP BY job_id
  4  ORDER BY job_id;

MIN(SALARY) AVG(SALARY) MAX(SALARY) JOB_ID
----------- ----------- ----------- ----------
       8300        8300        8300 AC_ACCOUNT
      12000       12000       12000 AC_MGR
       4400        4400        4400 AD_ASST
      24000       24000       24000 AD_PRES
      17000       17000       17000 AD_VP
       6900        7920        9000 FI_ACCOUNT
      12000       12000       12000 FI_MGR
       6500        6500        6500 HR_REP
       4200        5760        9000 IT_PROG
      13000       13000       13000 MK_MAN
       6000        6000        6000 MK_REP

MIN(SALARY) AVG(SALARY) MAX(SALARY) JOB_ID
----------- ----------- ----------- ----------
      10000       10000       10000 PR_REP
       2500        2780        3100 PU_CLERK
      11000       11000       11000 PU_MAN
      10500       12200       14000 SA_MAN
       6100        8350       11500 SA_REP
       2500        3215        4200 SH_CLERK
       2100        2785        3600 ST_CLERK
       5800        7280        8200 ST_MAN

19 rows selected.
```

2.

```
SQL> SELECT CASE
  2  WHEN max_salary < '10000' THEN 'Low max salary'
  3  WHEN max_salary < '20000' THEN 'Medium max salary'
  4  WHEN max_salary >= '20000' THEN 'High max salary' END AS max_sal_cat, COUNT(job_id)
  5  FROM jobs
  6  GROUP BY CASE
  7  WHEN max_salary < '10000' THEN 'Low max salary'
  8  WHEN max_salary < '20000' THEN 'Medium max salary'
  9  WHEN max_salary >= '20000' THEN 'High max salary' END;

MAX_SAL_CAT       COUNT(JOB_ID)
----------------- -------------
Medium max salary             7
Low max salary                9
High max salary               3
```

3.

```
SQL> SELECT SUBSTR(job_id,1,2) AS job_cat, AVG(min_salary), AVG(max_salary)
  2  FROM jobs
  3  GROUP BY SUBSTR(job_id,1,2)
  4  ORDER BY job_cat DESC;

JOB_CAT  AVG(MIN_SALARY) AVG(MAX_SALARY)
-------- --------------- ---------------
ST                  3750            6750
SH                  2500            5500
SA                  8000           16000
PU                  5250           10250
PR                  4500           10500
MK                  6500           12000
IT                  4000           10000
HR                  4000            9000
FI                  6200           12500
AD             12666.6667      25333.3333
AC                  6200           12500

11 rows selected.
```

4.

```
SQL> SELECT 'Sales Employees' AS emp_category, SUBSTR(hire_date,8,9) AS year, AVG(salary) AS avg_salary
  2  FROM employees
  3  WHERE commission_pct > 0
  4  GROUP BY SUBSTR(hire_date,8,9)
  5  HAVING SUBSTR(hire_date,8,9) > 97
  6  UNION
  7  SELECT 'Non-Sales Employees' AS emp_category, SUBSTR(hire_date,8,9) AS year, AVG(salary) AS avg_salary
  8  FROM employees
  9  WHERE commission_pct IS NULL
 10  GROUP BY SUBSTR(hire_date,8,9)
 11  HAVING SUBSTR(hire_date,8,9) > 97
 12  ORDER BY year ASC, emp_category ASC;

EMP_CATEGORY       YEAR                                AVG_SALARY
------------------ ---------------------------------- ----------
Non-Sales Employees 98                                     3312.5
Sales Employees     98                                   8442.85714
Non-Sales Employees 99                                   3308.33333
Sales Employees     99                                         8200
```

5.

```
SQL> SELECT job_id, manager_id, ROUND(AVG(salary),2) AS average_sal
  2  FROM employees
  3  WHERE commission_pct > 0
  4  GROUP BY job_id, manager_id
  5  ORDER BY job_id, manager_id;

JOB_ID     MANAGER_ID AVERAGE_SAL
---------- ---------- -----------
SA_MAN            100       12200
SA_REP            145        8500
SA_REP            146        8500
SA_REP            147     7766.67
SA_REP            148        8650
SA_REP            149     8333.33

6 rows selected.
```

6.

```
SQL> SELECT country_id, COUNT(country_id) AS count_loc
  2  FROM locations
  3  WHERE country_id IN('US', 'DE', 'IT', 'UK', 'CA', 'NL')
  4  GROUP BY country_id
  5  HAVING COUNT(country_id) > 1
  6  ORDER BY count_loc DESC, country_id ASC;

CO  COUNT_LOC
-- ----------
US          4
UK          3
CA          2
IT          2
```