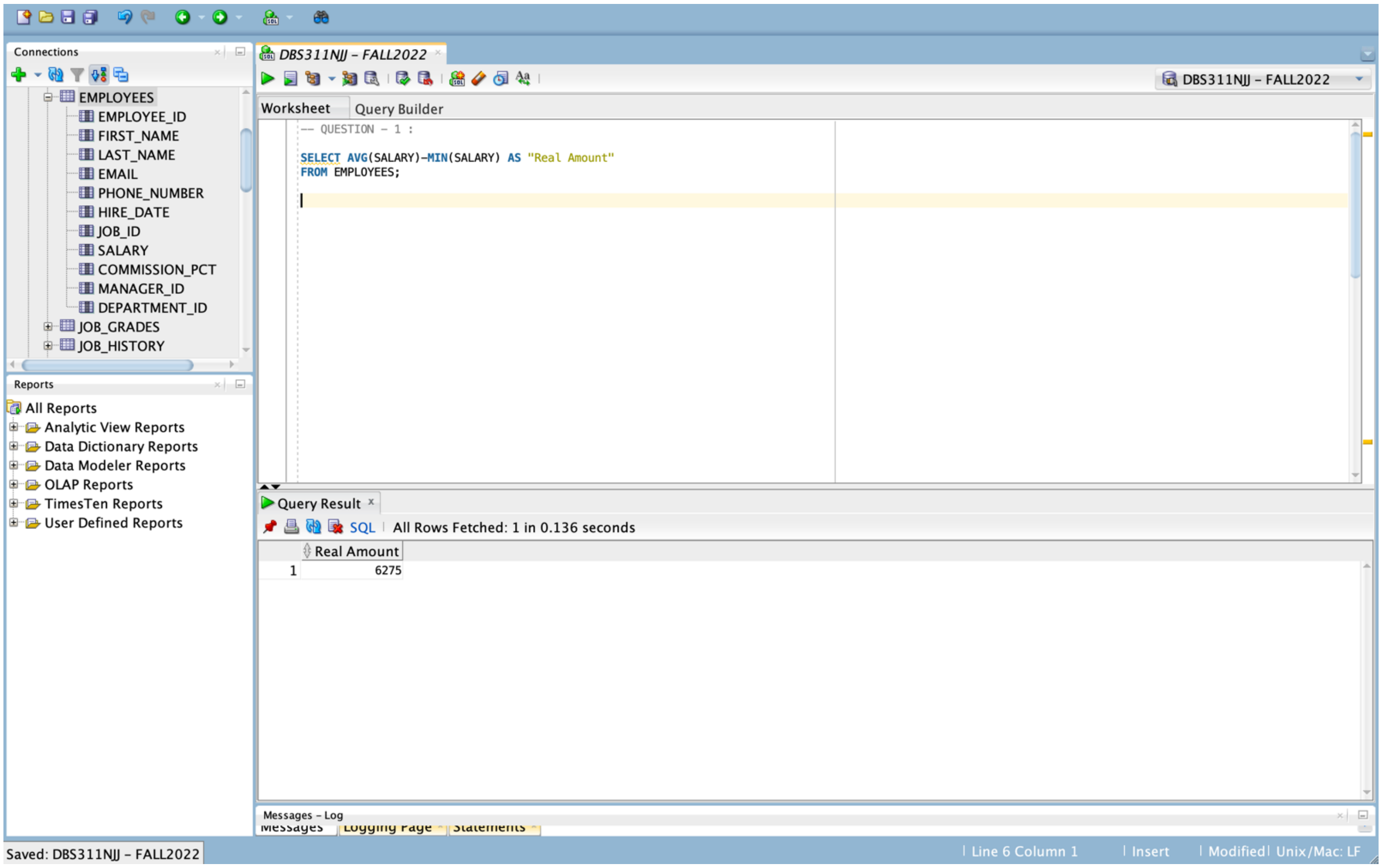
Lab 3

Please remember to upload the correct file before submitting - you only get one chance to submit. The Lab 3 submission link will not be available after deadline in the Assignments section.

Allow enough time to upload or deal with unexpected issues. Do not wait for the last moment since there are transmission time/queuing delay/processing time etc. from your machine to the blackboard server.

Late penalty is 100%.

1. Display the difference between the Average pay and Lowest pay in the company among employees. Name this result *Real Amount*.

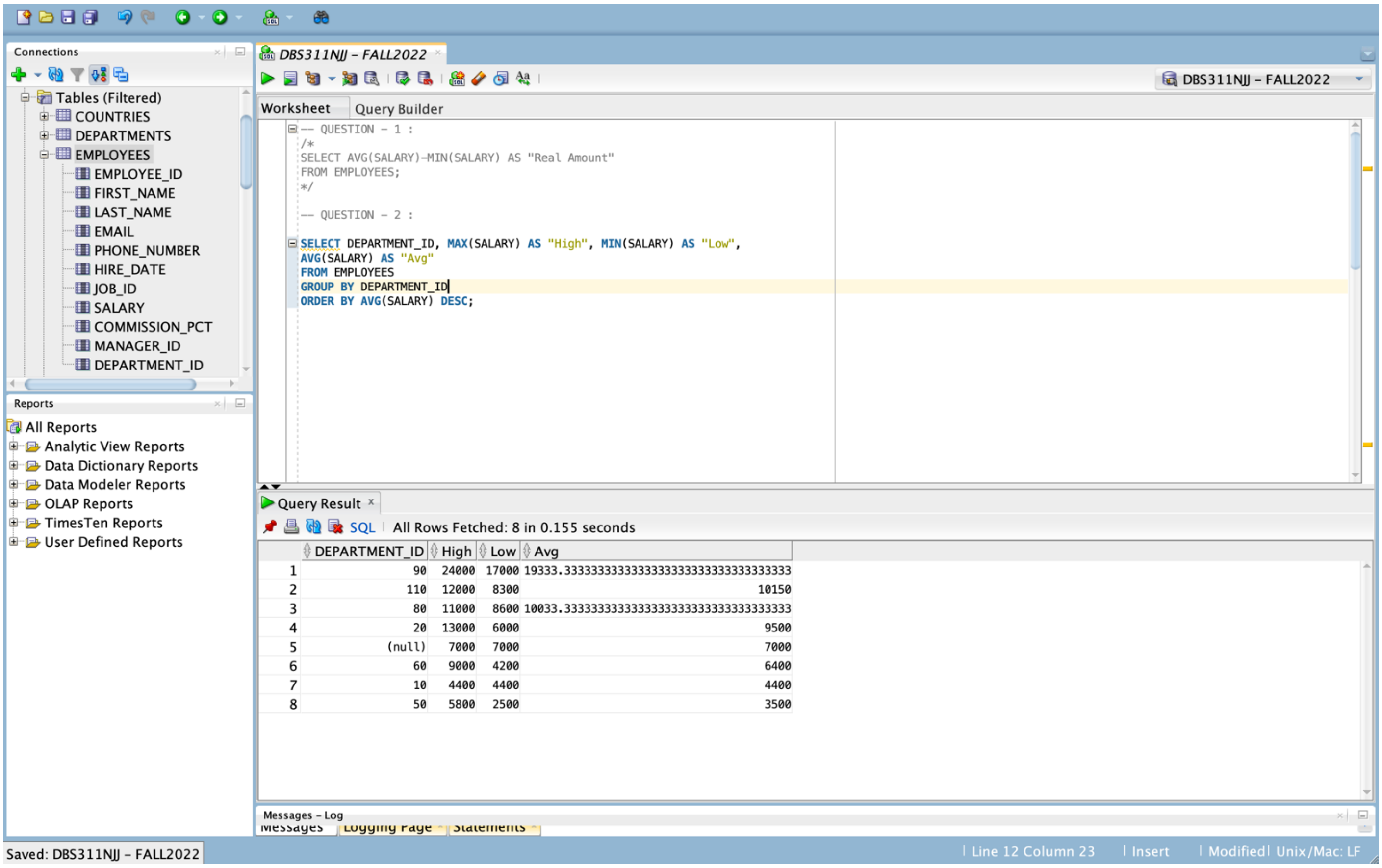


```
-- QUESTION - 1 :

SELECT AVG(SALARY)-MIN(SALARY) AS "Real Amount"
FROM EMPLOYEES;
```

All Rows Fetched: 1 in 0.136 seconds

```
   Real Amount
1         6275
```

2. Display the department number and Highest, Lowest and Average pay per each department. Name these results *High, Low* and *Avg.*
   Sort the output so that department with highest average salary are shown first.

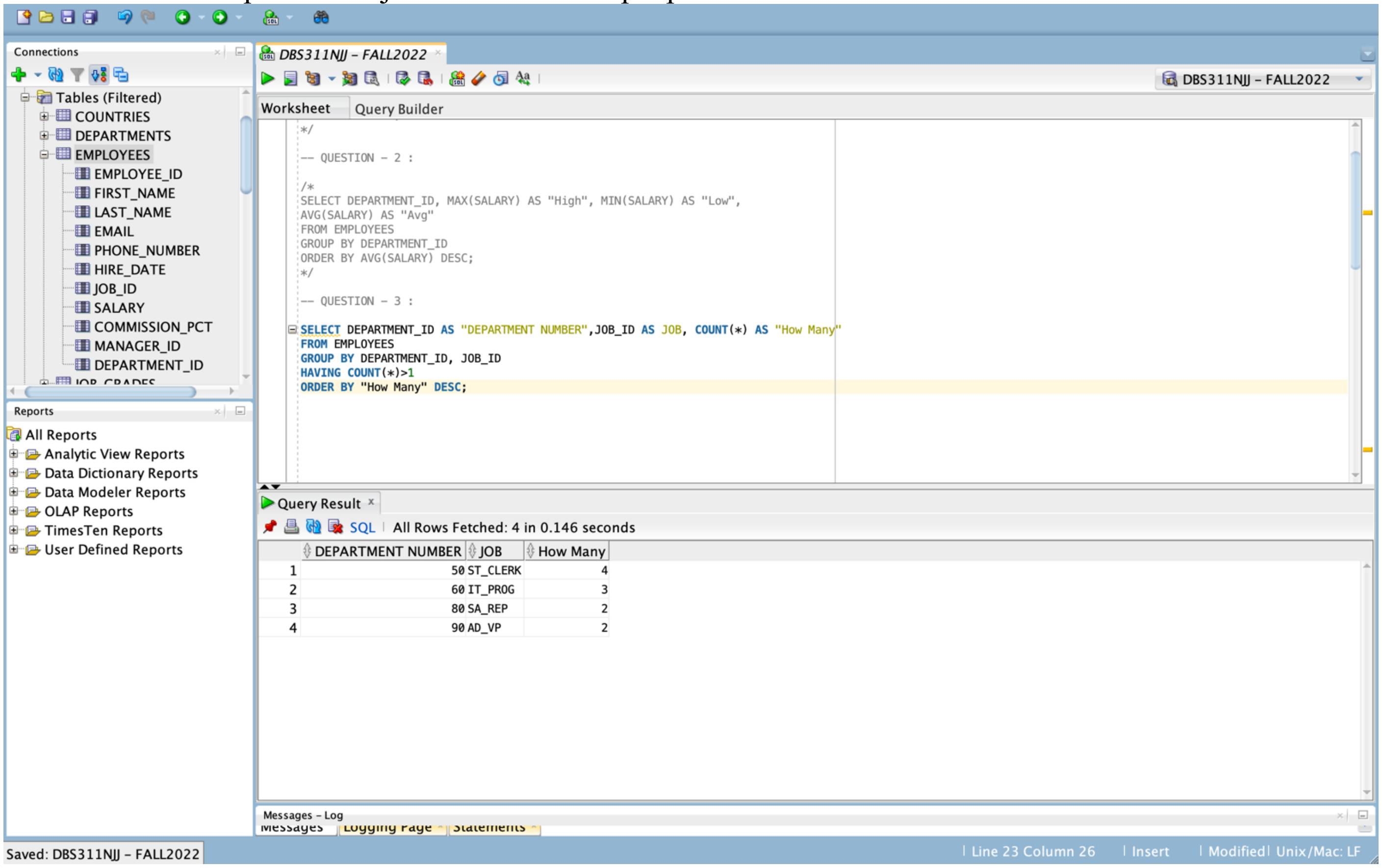


```sql
-- QUESTION - 1 :
/*
SELECT AVG(SALARY)-MIN(SALARY) AS "Real Amount"
FROM EMPLOYEES;
*/

-- QUESTION - 2 :

SELECT DEPARTMENT_ID, MAX(SALARY) AS "High", MIN(SALARY) AS "Low",
AVG(SALARY) AS "Avg"
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID
ORDER BY AVG(SALARY) DESC;
```

All Rows Fetched: 8 in 0.155 seconds

	DEPARTMENT_ID	High	Low	Avg
1	90	24000	17000	19333.3333333333333333333333333333333333
2	110	12000	8300	10150
3	80	11000	8600	10033.3333333333333333333333333333333333
4	20	13000	6000	9500
5	(null)	7000	7000	7000
6	60	9000	4200	6400
7	10	4400	4400	4400
8	50	5800	2500	3500

3. Display how many people work the same job in the same department. Name these results *Dept#, Job* and *HowMany*. Include only jobs that involve more than one person.
Sort the output so that jobs with the most people involved are shown first.

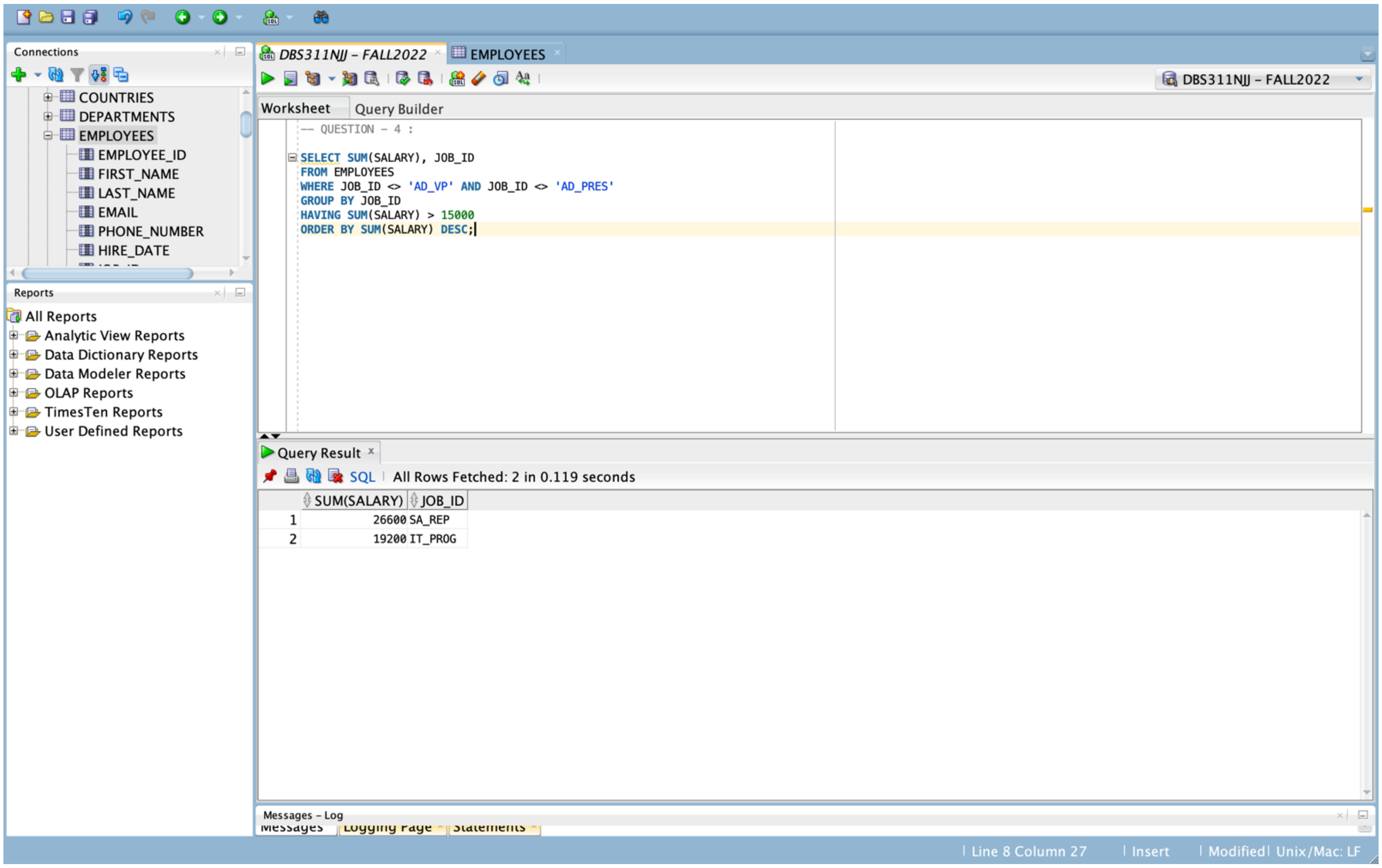


```sql
*/

-- QUESTION - 2 :

/*
SELECT DEPARTMENT_ID, MAX(SALARY) AS "High", MIN(SALARY) AS "Low",
AVG(SALARY) AS "Avg"
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID
ORDER BY AVG(SALARY) DESC;
*/

-- QUESTION - 3 :

SELECT DEPARTMENT_ID AS "DEPARTMENT NUMBER",JOB_ID AS JOB, COUNT(*) AS "How Many"
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID, JOB_ID
HAVING COUNT(*)>1
ORDER BY "How Many" DESC;
```

All Rows Fetched: 4 in 0.146 seconds

	DEPARTMENT NUMBER	JOB	How Many
1	50	ST_CLERK	4
2	60	IT_PROG	3
3	80	SA_REP	2
4	90	AD_VP	2

4. For each job id display the job id and total amount paid each month for this type of the job. Exclude job_id *AD_PRES* and *AD_VP* and also include only jobs that require more than $15,000 in total. Sort the output so that top paid jobs are shown first.

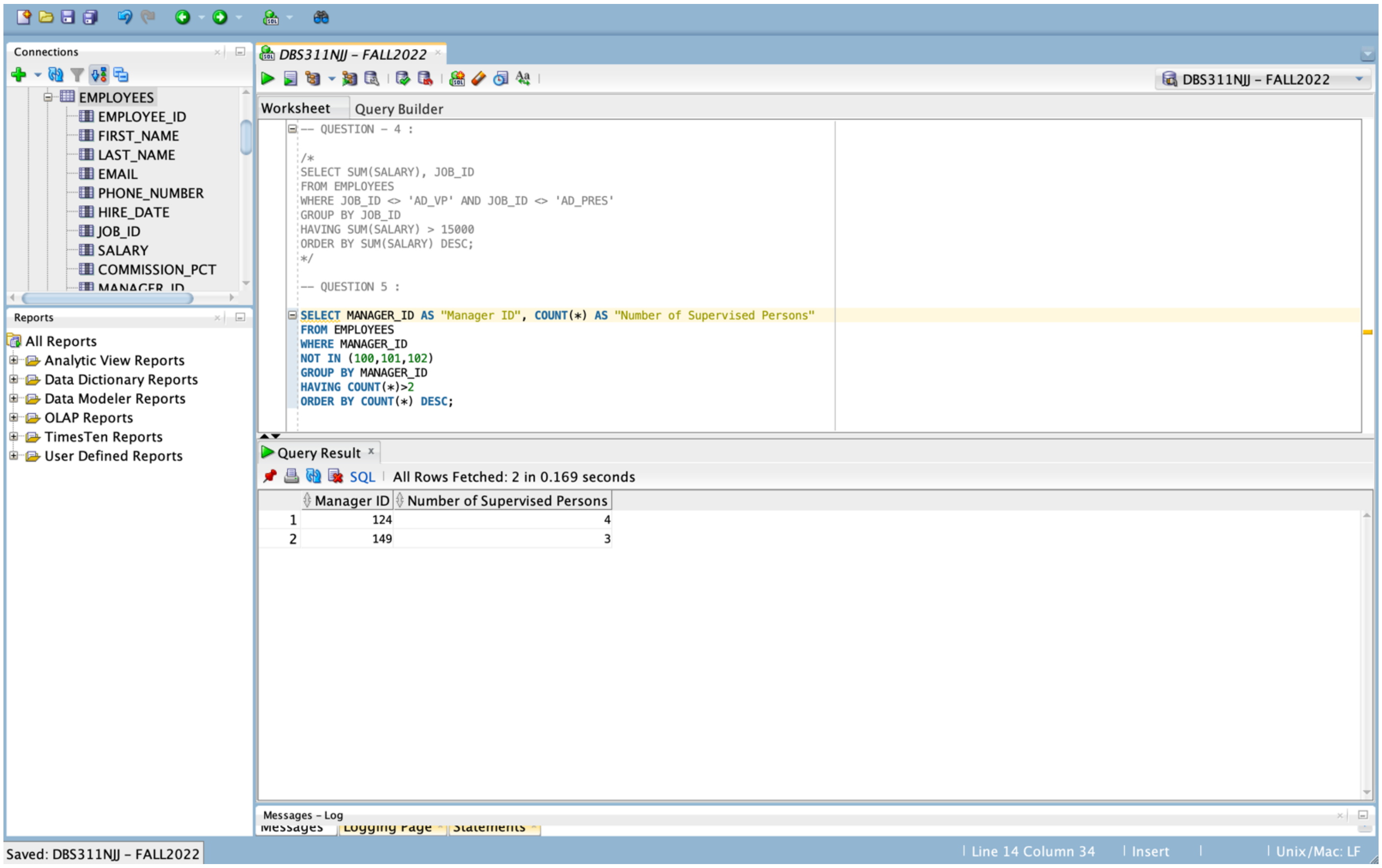


```
-- QUESTION - 4 :

SELECT SUM(SALARY), JOB_ID
FROM EMPLOYEES
WHERE JOB_ID <> 'AD_VP' AND JOB_ID <> 'AD_PRES'
GROUP BY JOB_ID
HAVING SUM(SALARY) > 15000
ORDER BY SUM(SALARY) DESC;
```

All Rows Fetched: 2 in 0.119 seconds

```
   SUM(SALARY) JOB_ID
1        26600 SA_REP
2        19200 IT_PROG
```

5. For each manager number display how many persons he / she supervises. Exclude managers with numbers 100, 101 and 102 and also include only those managers that supervise more than 2 persons. Sort the output so that manager numbers with the most supervised persons are shown first.

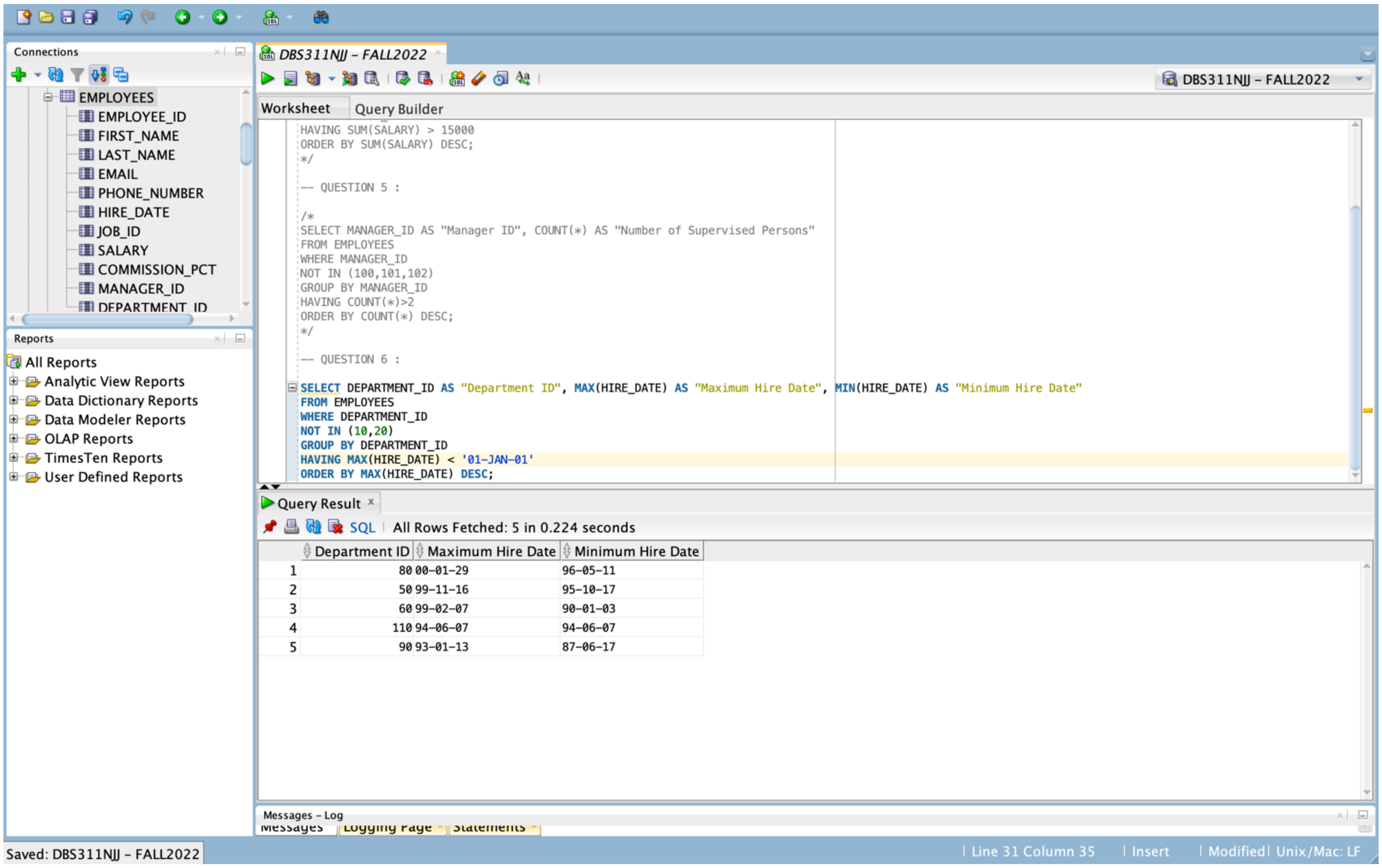


```sql
-- QUESTION - 4 :

/*
SELECT SUM(SALARY), JOB_ID
FROM EMPLOYEES
WHERE JOB_ID <> 'AD_VP' AND JOB_ID <> 'AD_PRES'
GROUP BY JOB_ID
HAVING SUM(SALARY) > 15000
ORDER BY SUM(SALARY) DESC;
*/

-- QUESTION 5 :

SELECT MANAGER_ID AS "Manager ID", COUNT(*) AS "Number of Supervised Persons"
FROM EMPLOYEES
WHERE MANAGER_ID
NOT IN (100,101,102)
GROUP BY MANAGER_ID
HAVING COUNT(*)>2
ORDER BY COUNT(*) DESC;
```

All Rows Fetched: 2 in 0.169 seconds

	Manager ID	Number of Supervised Persons
1	124	4
2	149	3

6. For each department show the latest and earliest hire date, but exclude departments 10 and 20 and also exclude those departments where the last person was hired in this century. Sort the output so that most recent latest hire dates are shown first.



```sql
HAVING SUM(SALARY) > 15000
ORDER BY SUM(SALARY) DESC;
*/

-- QUESTION 5 :

/*
SELECT MANAGER_ID AS "Manager ID", COUNT(*) AS "Number of Supervised Persons"
FROM EMPLOYEES
WHERE MANAGER_ID
NOT IN (100,101,102)
GROUP BY MANAGER_ID
HAVING COUNT(*)>2
ORDER BY COUNT(*) DESC;
*/

-- QUESTION 6 :

SELECT DEPARTMENT_ID AS "Department ID", MAX(HIRE_DATE) AS "Maximum Hire Date", MIN(HIRE_DATE) AS "Minimum Hire Date"
FROM EMPLOYEES
WHERE DEPARTMENT_ID
NOT IN (10,20)
GROUP BY DEPARTMENT_ID
HAVING MAX(HIRE_DATE) < '01-JAN-01'
ORDER BY MAX(HIRE_DATE) DESC;
```

All Rows Fetched: 5 in 0.224 seconds

	Department ID	Maximum Hire Date	Minimum Hire Date
1	80	00-01-29	96-05-11
2	50	99-11-16	95-10-17
3	60	99-02-07	90-01-03
4	110	94-06-07	94-06-07
5	90	93-01-13	87-06-17