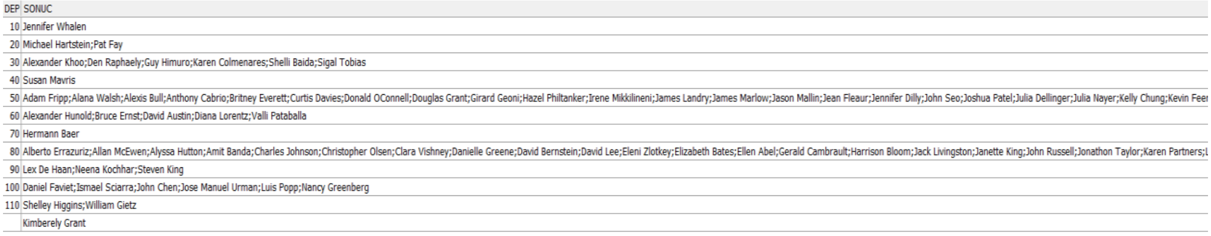
Topic	ORACLE Analytic SQL
Document Name	AnalyticSQL01-EX-01-05

Enes Tahtacı

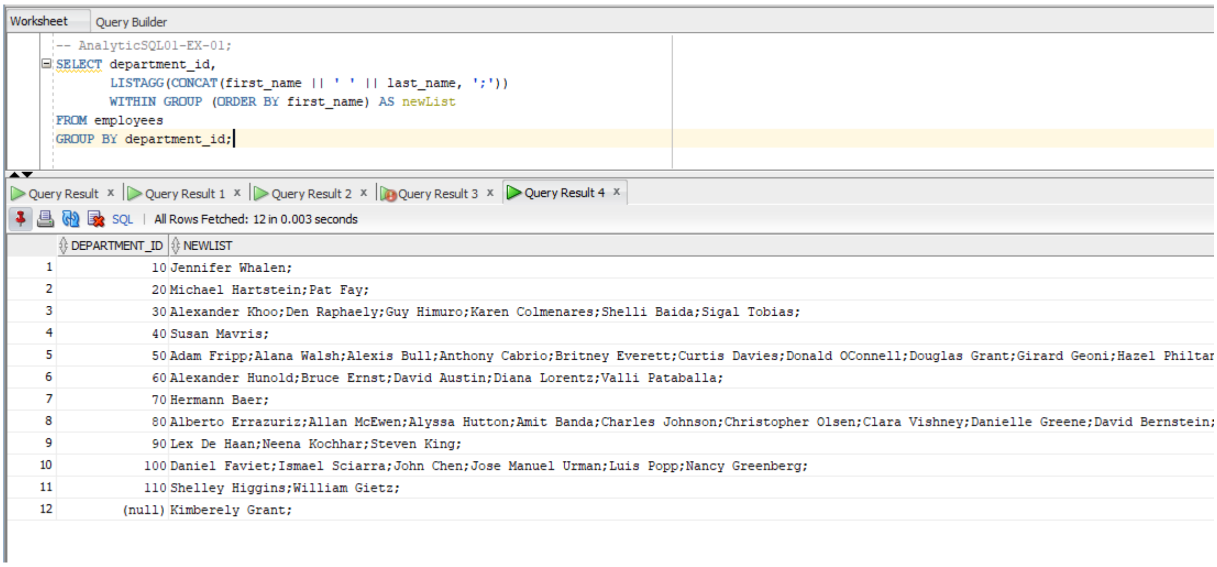
Exercise EX -01:

Definiton : 1-List employees’ first name and last name in same department in one row.
Firstname Lastname; Firstname Lastname ;



DEP	SONUC
10	Jennifer Whalen
20	Michael Hartstein;Pat Fay
30	Alexander Khoo;Den Raphaely;Guy Himuro;Karen Colmenares;Shelli Baida;Sigal Tobias
40	Susan Mavris
50	Adam Fripp;Alana Walsh;Alexis Bull;Anthony Cabrio;Britney Everett;Curtis Davies;Donald OConnell;Douglas Grant;Girard Geoni;Hazel Philtanker;Irene Mikkilineni;James Landry;James Marlow;Jason Mallin;Jean Fleaur;Jennifer Dilly;John Seo;Joshua Patel;Julia Dellinger;Julia Nayer;Kelly Chung;Kevin Feer
60	Alexander Hunold;Bruce Ernst;David Austin;Diana Lorentz;Valli Pataballa
70	Hermann Baer
80	Alberto Errazuriz;Allan McEwen;Alyssa Hutton;Amit Banda;Charles Johnson;Christopher Olsen;Clara Vishney;Danielle Greene;David Bernstein;David Lee;Eleni Zlotkey;Elizabeth Bates;Ellen Abel;Gerald Cambrault;Harrison Bloom;Jack Livingston;Janette King;John Russell;Jonathon Taylor;Karen Partners;
90	Lex De Haan;Neena Kochhar;Steven King
100	Daniel Faviet;Ismael Sciarra;John Chen;Jose Manuel Urman;Luis Popp;Nancy Greenberg
110	Shelley Higgins;William Gietz
	Kimberely Grant

Screenshot:



```sql
-- AnalyticSQL01-EX-01;
SELECT department_id,
       LISTAGG(CONCAT(first_name || ' ' || last_name, ';'))
       WITHIN GROUP (ORDER BY first_name) AS newList
FROM employees
GROUP BY department_id;
```

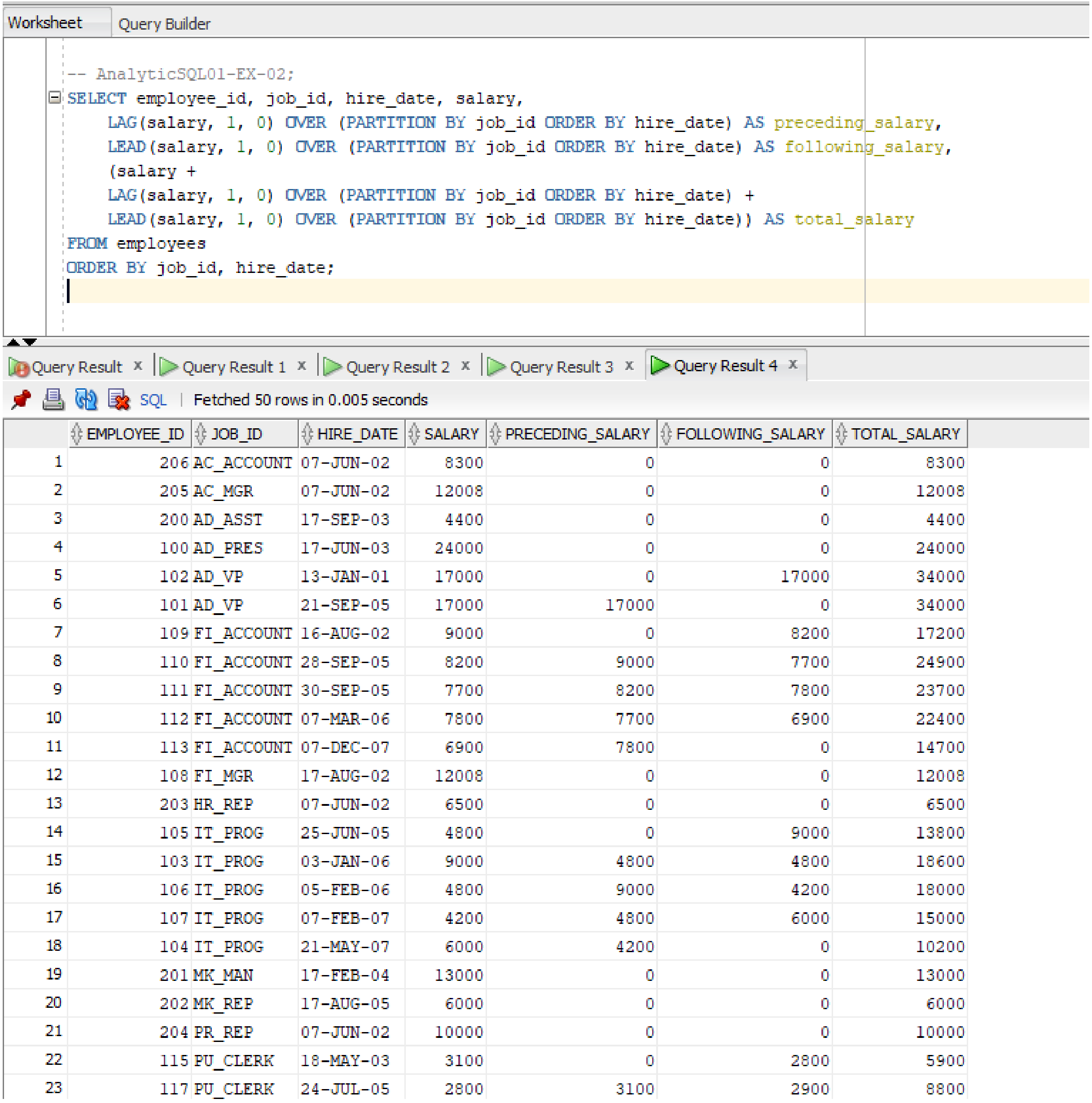
All Rows Fetched: 12 in 0.003 seconds

	DEPARTMENT_ID	NEWLIST
1	10	Jennifer Whalen;
2	20	Michael Hartstein;Pat Fay;
3	30	Alexander Khoo;Den Raphaely;Guy Himuro;Karen Colmenares;Shelli Baida;Sigal Tobias;
4	40	Susan Mavris;
5	50	Adam Fripp;Alana Walsh;Alexis Bull;Anthony Cabrio;Britney Everett;Curtis Davies;Donald OConnell;Douglas Grant;Girard Geoni;Hazel Philtar
6	60	Alexander Hunold;Bruce Ernst;David Austin;Diana Lorentz;Valli Pataballa;
7	70	Hermann Baer;
8	80	Alberto Errazuriz;Allan McEwen;Alyssa Hutton;Amit Banda;Charles Johnson;Christopher Olsen;Clara Vishney;Danielle Greene;David Bernstein;
9	90	Lex De Haan;Neena Kochhar;Steven King;
10	100	Daniel Faviet;Ismael Sciarra;John Chen;Jose Manuel Urman;Luis Popp;Nancy Greenberg;
11	110	Shelley Higgins;William Gietz;
12	(null)	Kimberely Grant;

Exercise EX -02:

Definiton : Write a SQL statement that shows employees' preceeding 1 and following 1 salary totals in same job_id and ordered by hiredate.

Screenshot:



```
-- AnalyticSQL01-EX-02;
SELECT employee_id, job_id, hire_date, salary,
    LAG(salary, 1, 0) OVER (PARTITION BY job_id ORDER BY hire_date) AS preceding_salary,
    LEAD(salary, 1, 0) OVER (PARTITION BY job_id ORDER BY hire_date) AS following_salary,
    (salary +
    LAG(salary, 1, 0) OVER (PARTITION BY job_id ORDER BY hire_date) +
    LEAD(salary, 1, 0) OVER (PARTITION BY job_id ORDER BY hire_date)) AS total_salary
FROM employees
ORDER BY job_id, hire_date;
```

Fetched 50 rows in 0.005 seconds

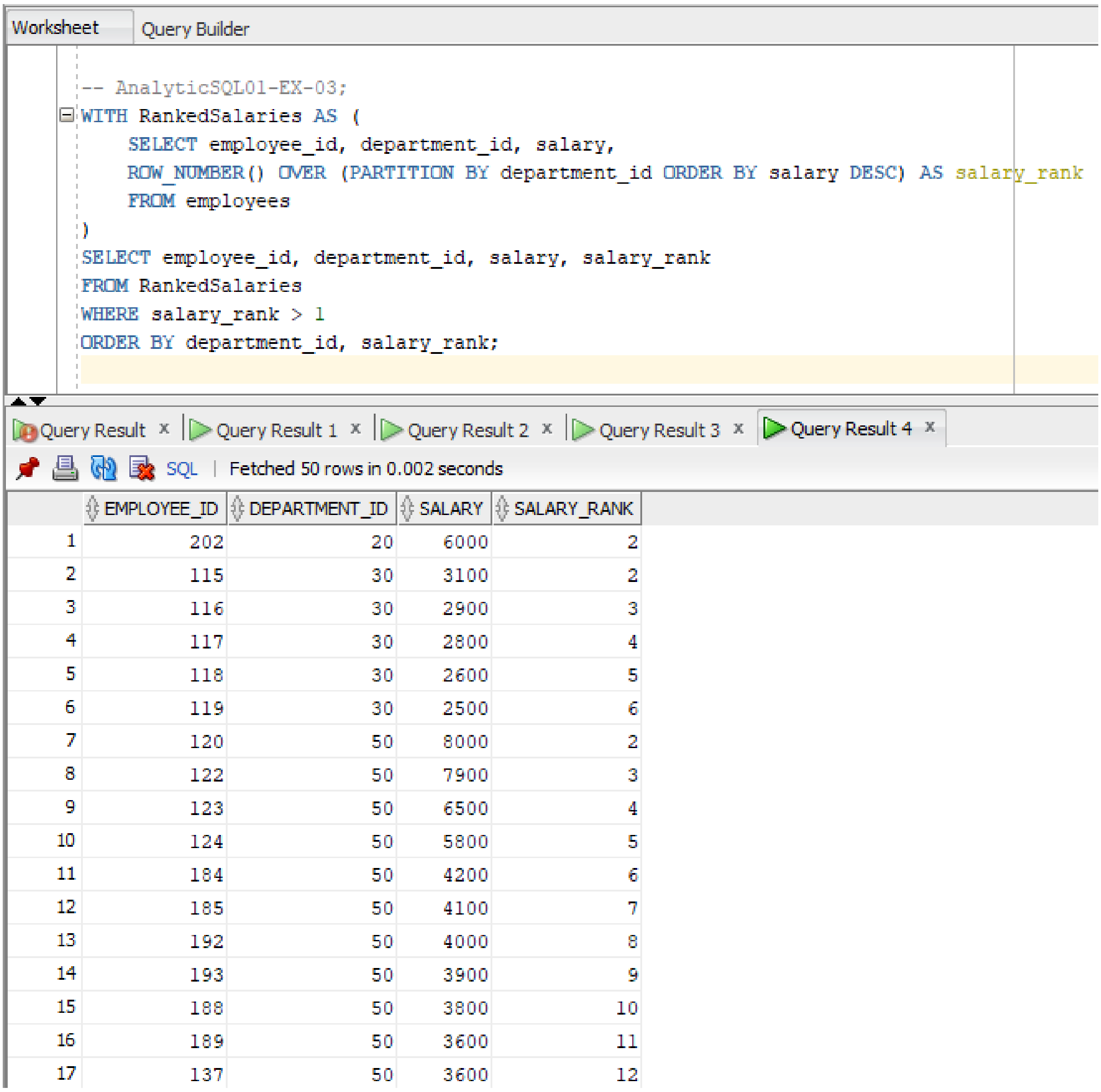
	EMPLOYEE_ID	JOB_ID	HIRE_DATE	SALARY	PRECEDING_SALARY	FOLLOWING_SALARY	TOTAL_SALARY
1	206	AC_ACCOUNT	07-JUN-02	8300	0	0	8300
2	205	AC_MGR	07-JUN-02	12008	0	0	12008
3	200	AD_ASST	17-SEP-03	4400	0	0	4400
4	100	AD_PRES	17-JUN-03	24000	0	0	24000
5	102	AD_VP	13-JAN-01	17000	0	17000	34000
6	101	AD_VP	21-SEP-05	17000	17000	0	34000
7	109	FI_ACCOUNT	16-AUG-02	9000	0	8200	17200
8	110	FI_ACCOUNT	28-SEP-05	8200	9000	7700	24900
9	111	FI_ACCOUNT	30-SEP-05	7700	8200	7800	23700
10	112	FI_ACCOUNT	07-MAR-06	7800	7700	6900	22400
11	113	FI_ACCOUNT	07-DEC-07	6900	7800	0	14700
12	108	FI_MGR	17-AUG-02	12008	0	0	12008
13	203	HR_REP	07-JUN-02	6500	0	0	6500
14	105	IT_PROG	25-JUN-05	4800	0	9000	13800
15	103	IT_PROG	03-JAN-06	9000	4800	4800	18600
16	106	IT_PROG	05-FEB-06	4800	9000	4200	18000
17	107	IT_PROG	07-FEB-07	4200	4800	6000	15000
18	104	IT_PROG	21-MAY-07	6000	4200	0	10200
19	201	MK_MAN	17-FEB-04	13000	0	0	13000
20	202	MK_REP	17-AUG-05	6000	0	0	6000
21	204	PR_REP	07-JUN-02	10000	0	0	10000
22	115	PU_CLERK	18-MAY-03	3100	0	2800	5900
23	117	PU_CLERK	24-JUL-05	2800	3100	2900	8800

Exercise EX -03:

Definiton : List employees’ salary orders in their department and exclude higest salaried employee.

(Hint: ROW_NUMBER)

Screenshot:



```
-- AnalyticSQL01-EX-03;
WITH RankedSalaries AS (
    SELECT employee_id, department_id, salary,
    ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY salary DESC) AS salary_rank
    FROM employees
)
SELECT employee_id, department_id, salary, salary_rank
FROM RankedSalaries
WHERE salary_rank > 1
ORDER BY department_id, salary_rank;
```

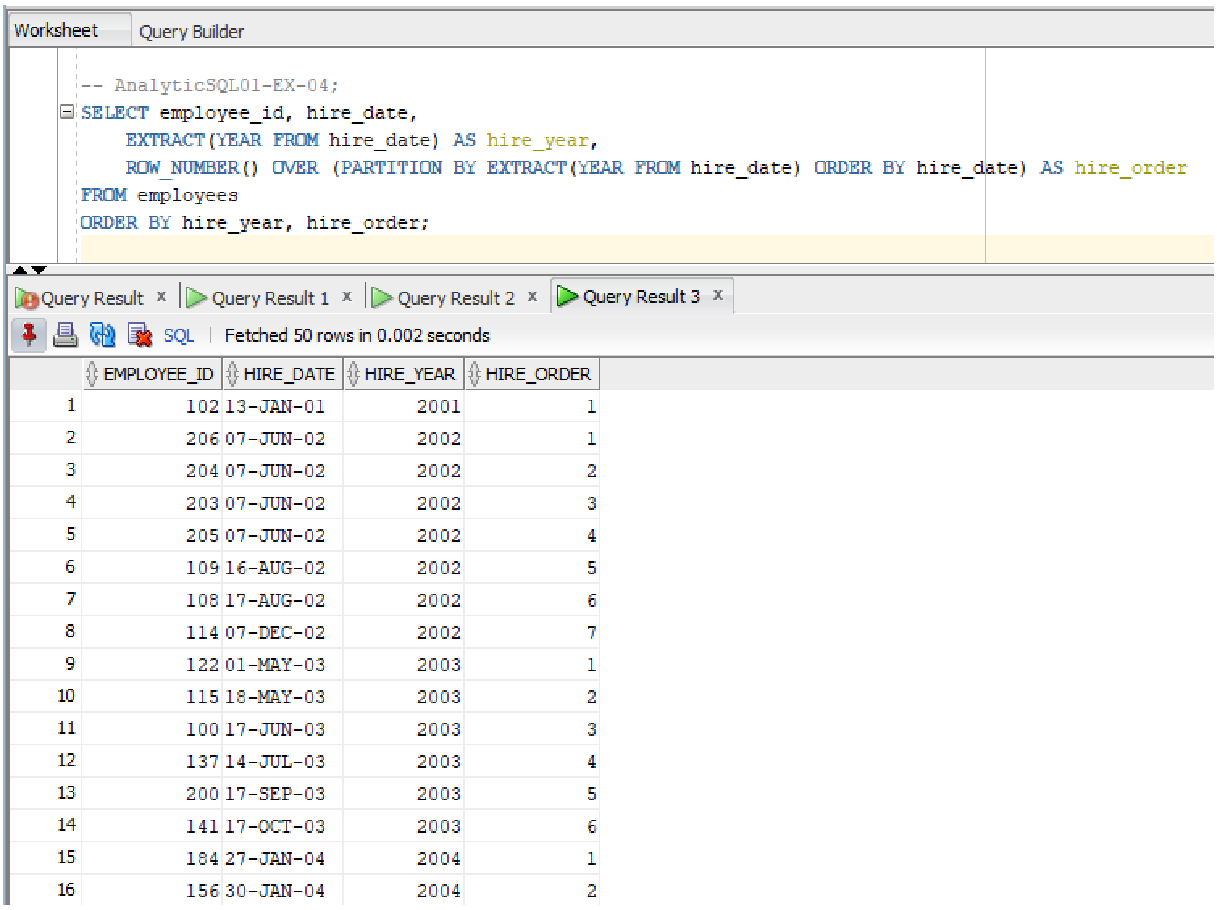
Fetched 50 rows in 0.002 seconds

	EMPLOYEE_ID	DEPARTMENT_ID	SALARY	SALARY_RANK
1	202	20	6000	2
2	115	30	3100	2
3	116	30	2900	3
4	117	30	2800	4
5	118	30	2600	5
6	119	30	2500	6
7	120	50	8000	2
8	122	50	7900	3
9	123	50	6500	4
10	124	50	5800	5
11	184	50	4200	6
12	185	50	4100	7
13	192	50	4000	8
14	193	50	3900	9
15	188	50	3800	10
16	189	50	3600	11
17	137	50	3600	12

Exercise EX -04:

Definiton : List employees’ hire order according to hiredate year value.

Screenshot:



```
-- AnalyticSQL01-EX-04;
SELECT employee_id, hire_date,
    EXTRACT(YEAR FROM hire_date) AS hire_year,
    ROW_NUMBER() OVER (PARTITION BY EXTRACT(YEAR FROM hire_date) ORDER BY hire_date) AS hire_order
FROM employees
ORDER BY hire_year, hire_order;
```

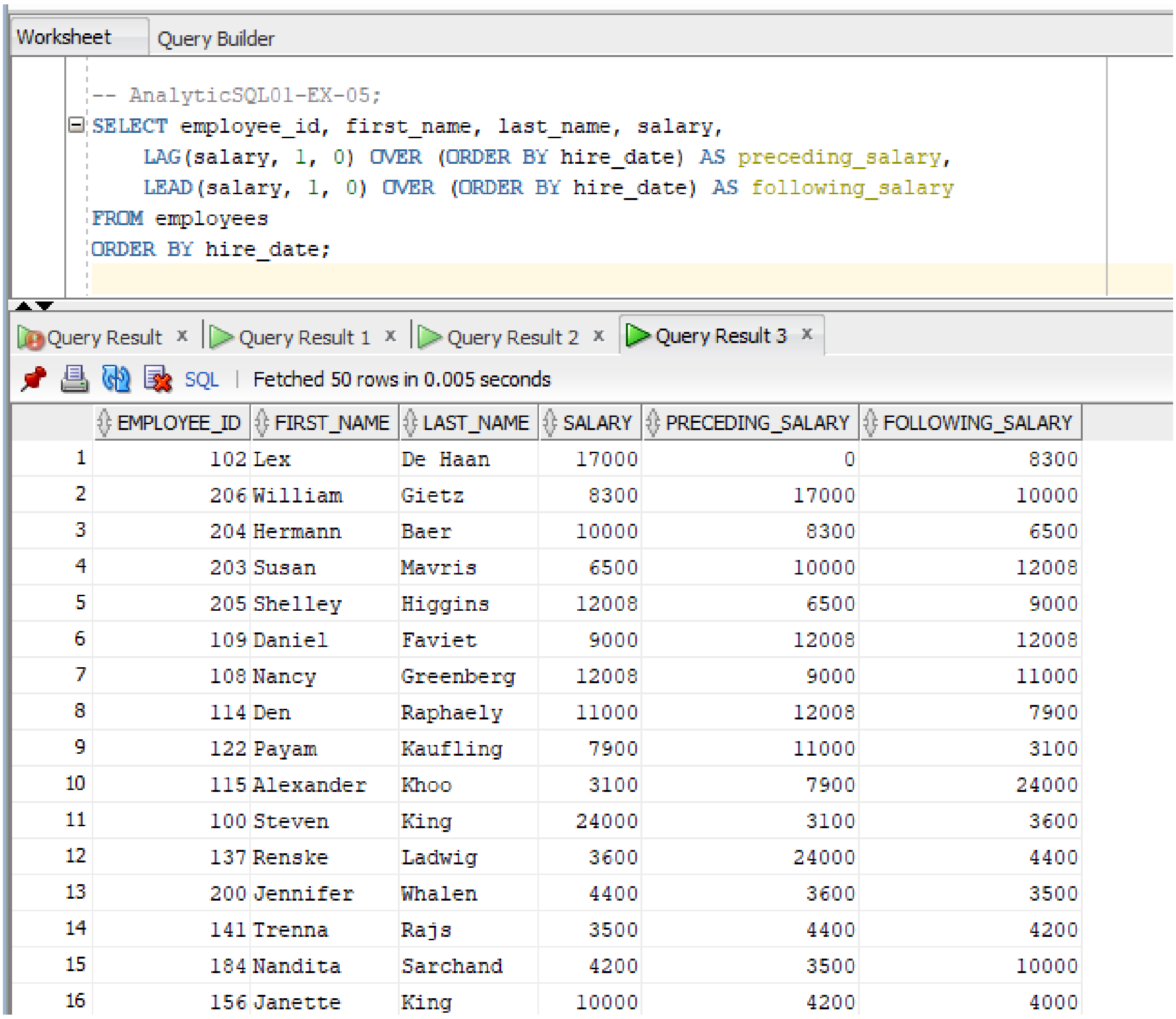
Fetched 50 rows in 0.002 seconds

	EMPLOYEE_ID	HIRE_DATE	HIRE_YEAR	HIRE_ORDER
1	102	13-JAN-01	2001	1
2	206	07-JUN-02	2002	1
3	204	07-JUN-02	2002	2
4	203	07-JUN-02	2002	3
5	205	07-JUN-02	2002	4
6	109	16-AUG-02	2002	5
7	108	17-AUG-02	2002	6
8	114	07-DEC-02	2002	7
9	122	01-MAY-03	2003	1
10	115	18-MAY-03	2003	2
11	100	17-JUN-03	2003	3
12	137	14-JUL-03	2003	4
13	200	17-SEP-03	2003	5
14	141	17-OCT-03	2003	6
15	184	27-JAN-04	2004	1
16	156	30-JAN-04	2004	2

Exercise EX -05:

Definiton : List employees' firstname, lastname, salary and salaries of employees' hired before and after this employee according to hiredate.

Screenshot:



```
-- AnalyticSQL01-EX-05;
SELECT employee_id, first_name, last_name, salary,
    LAG(salary, 1, 0) OVER (ORDER BY hire_date) AS preceding_salary,
    LEAD(salary, 1, 0) OVER (ORDER BY hire_date) AS following_salary
FROM employees
ORDER BY hire_date;
```

Fetched 50 rows in 0.005 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	PRECEDING_SALARY	FOLLOWING_SALARY
1	102	Lex	De Haan	17000	0	8300
2	206	William	Gietz	8300	17000	10000
3	204	Hermann	Baer	10000	8300	6500
4	203	Susan	Mavris	6500	10000	12008
5	205	Shelley	Higgins	12008	6500	9000
6	109	Daniel	Faviet	9000	12008	12008
7	108	Nancy	Greenberg	12008	9000	11000
8	114	Den	Raphaely	11000	12008	7900
9	122	Payam	Kaufling	7900	11000	3100
10	115	Alexander	Khoo	3100	7900	24000
11	100	Steven	King	24000	3100	3600
12	137	Renske	Ladwig	3600	24000	4400
13	200	Jennifer	Whalen	4400	3600	3500
14	141	Trenna	Rajs	3500	4400	4200
15	184	Nandita	Sarchand	4200	3500	10000
16	156	Janette	King	10000	4200	4000