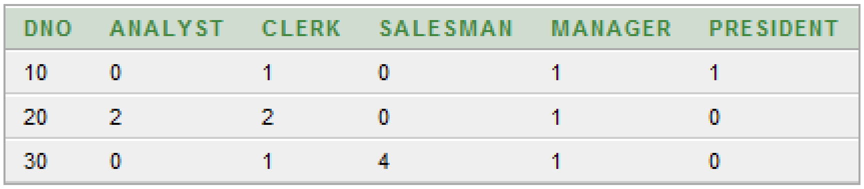
- Print the summary report of the employees in the following manner.



DNO	ANALYST	CLERK	SALESMAN	MANAGER	PRESIDENT
10	0	1	0	1	1
20	2	2	0	1	0
30	0	1	4	1	0

Solution 1: WITH A SUBQUERY

```
SELECT dno,     SUM(analyst)          AS ANALYST,
                SUM(clerk)            AS CLERK,
                SUM(salesman)         AS SALESMAN,
                SUM(manager)          AS MANAGER,
                SUM(president)        AS PRESIDENT
FROM   (
                SELECT dno,     CASE WHEN job = 'ANALYST'     THEN COUNT(*)        END ANALYST,
                                CASE WHEN job = 'CLERK'       THEN COUNT(*)        END CLERK,
                                CASE WHEN job = 'SALESMAN'    THEN COUNT(*)        END SALESMAN,
                                CASE WHEN job = 'MANAGER'     THEN COUNT(*)        END MANAGER,
                                CASE WHEN job = 'PRESIDENT'   THEN COUNT(*)        END PRESIDENT
                FROM employees
                GROUP BY dno,job
       )
GROUP BY dno
ORDER BY dno;
```

(**NOTE:** You will get '-' printed instead of zeros by the above query. To get '0', use NVL)

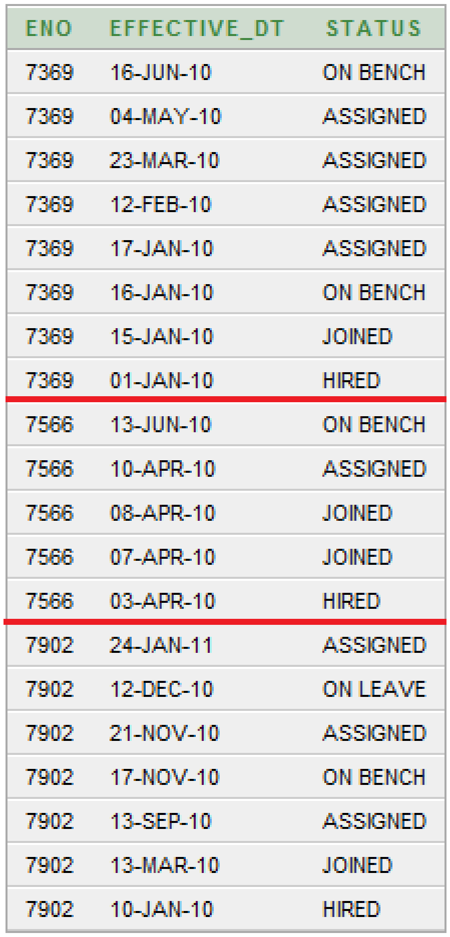
Solution 1: WITHOUT A SUBQUERY

```
SELECT          dno,    COUNT(DECODE(job,'ANALYST',eno))          AS ANALYST,
                        COUNT(DECODE(job,'CLERK',eno))            AS CLERK,
                        COUNT(DECODE(job,'SALESMAN',eno))         AS SALESMAN,
                        COUNT(DECODE(job,'MANAGER',eno))          AS MANAGER,
                        COUNT(DECODE(job,'PRESIDENT',eno))        AS PRESIDENT
FROM            employees
GROUP BY        dno
ORDER BY        dno;
```

- HEWITT ASSOCIATES QUERY

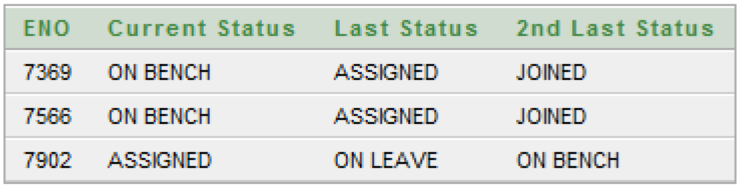
Let there be a table for status of the employees. Print the previous **three distinct status** of the employees as follows.

emp_status table



ENO	EFFECTIVE_DT	STATUS
7369	16-JUN-10	ON BENCH
7369	04-MAY-10	ASSIGNED
7369	23-MAR-10	ASSIGNED
7369	12-FEB-10	ASSIGNED
7369	17-JAN-10	ASSIGNED
7369	16-JAN-10	ON BENCH
7369	15-JAN-10	JOINED
7369	01-JAN-10	HIRED
7566	13-JUN-10	ON BENCH
7566	10-APR-10	ASSIGNED
7566	08-APR-10	JOINED
7566	07-APR-10	JOINED
7566	03-APR-10	HIRED
7902	24-JAN-11	ASSIGNED
7902	12-DEC-10	ON LEAVE
7902	21-NOV-10	ASSIGNED
7902	17-NOV-10	ON BENCH
7902	13-SEP-10	ASSIGNED
7902	13-MAR-10	JOINED
7902	10-JAN-10	HIRED

Desired Output



ENO	Current Status	Last Status	2nd Last Status
7369	ON BENCH	ASSIGNED	JOINED
7566	ON BENCH	ASSIGNED	JOINED
7902	ASSIGNED	ON LEAVE	ON BENCH

1st SOLUTION : WITH JOINS

```
SELECT e1.eno,  e1.status          AS "Current Status",
                e2.status          AS "Last Status",
                e3.status          AS "2nd Last Status"
FROM (
        SELECT eno,status
        FROM   (
                SELECT eno, status,RANK()OVER(PARTITION BY eno ORDER BY effective_dt DESC) rank
                FROM   (
                        SELECT eno,status,MAX(effective_dt) effective_dt
                        FROM emp_status
                        GROUP BY eno,status
                        )
                )
        WHERE rank=3
       ) e3

RIGHT JOIN
        (
        SELECT eno, status
        FROM   (
                SELECT eno, status,RANK()OVER(PARTITION BY eno ORDER BY effective_dt DESC)  rank
                FROM   (
                        SELECT eno,status,MAX(effective_dt) effective_dt
                        FROM emp_status
                        GROUP BY eno,status
                        )
         )
        WHERE rank=2
        ) e2

        ON (e2.eno=e3.eno)

RIGHT JOIN
        (
        SELECT eno, status
        FROM   (
                SELECT eno, status,RANK()OVER(PARTITION BY eno ORDER BY effective_dt DESC)  rank
                FROM emp_status
                )
        WHERE rank=1
        ) e1

        ON (e1.eno=e2.eno)

ORDER BY e1.eno;
```

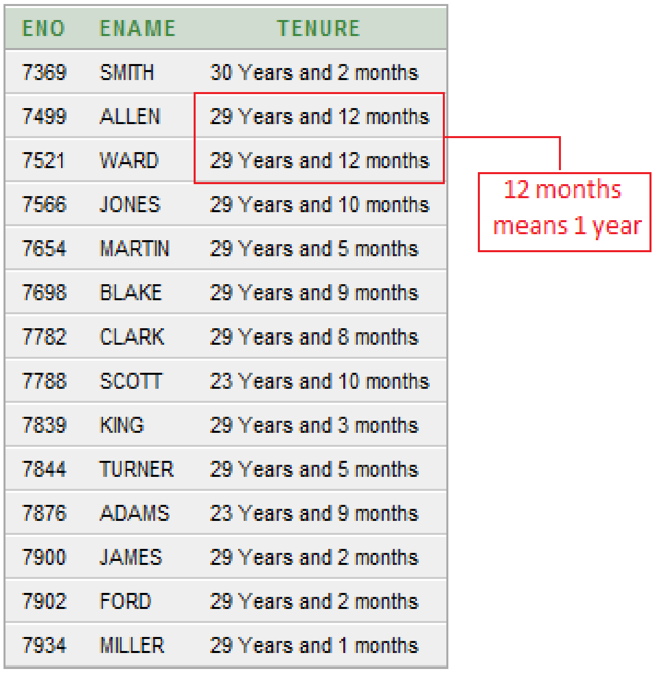
2nd SOLUTION : WITHOUT JOINS

```
SELECT eno,     MAX(DECODE(rank,1,status))   AS "Current Status",
                MAX(DECODE(rank,2,status))   AS "Last Status",
                MAX(DECODE(rank,3,status))   AS "2nd Last Status"

FROM   (
                SELECT eno, status,RANK()OVER(PARTITION BY eno ORDER BY effective_dt DESC) rank
                FROM   (
                                SELECT eno,status,MAX(effective_dt) effective_dt
                                FROM emp_status
                                GROUP BY eno,status
                       )
       )

GROUP BY eno;
```

- Find the employee no., name & the following attributes.

  Tenure – to be displayed as the no. of years & the months (rounded off) from the time the employee has been hired,
  Induction date – as the first Friday of the next month from the hire date.
  Review date – as the exact date when the employee has completed 6 months.
  Salary date – as the last date of the next month from the hire date.
  Picnic date – as the next 1st February after the hire date

```
SELECT          eno,ename,
                FLOOR(MONTHS_BETWEEN(SYSDATE,doj)/12) ||' Years and ' ||
                ROUND(MOD(MONTHS_BETWEEN(SYSDATE,doj),12)) ||' months'      AS Tenure
FROM employees;
```



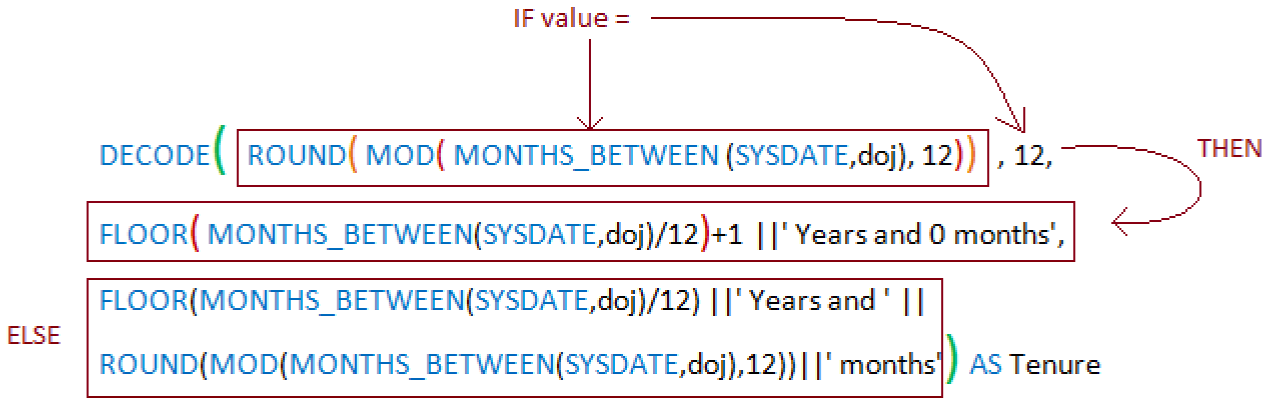
ENO	ENAME	TENURE
7369	SMITH	30 Years and 2 months
7499	ALLEN	29 Years and 12 months
7521	WARD	29 Years and 12 months
7566	JONES	29 Years and 10 months
7654	MARTIN	29 Years and 5 months
7698	BLAKE	29 Years and 9 months
7782	CLARK	29 Years and 8 months
7788	SCOTT	23 Years and 10 months
7839	KING	29 Years and 3 months
7844	TURNER	29 Years and 5 months
7876	ADAMS	23 Years and 9 months
7900	JAMES	29 Years and 2 months
7902	FORD	29 Years and 2 months
7934	MILLER	29 Years and 1 months

12 months means 1 year

This problem can be solved by following query.

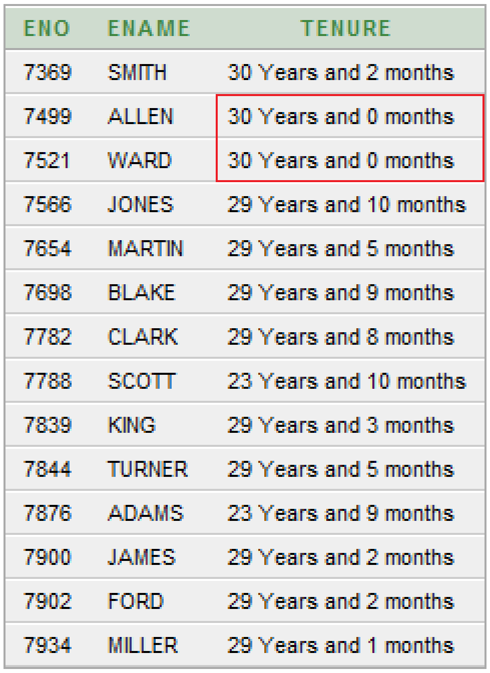
```
SELECT eno, ename,      DECODE( ROUND( MOD( MONTHS_BETWEEN (SYSDATE,doj), 12))  , 12,
                        FLOOR( MONTHS_BETWEEN(SYSDATE,doj)/12)+1 ||' Years and 0 months',
                        FLOOR(MONTHS_BETWEEN(SYSDATE,doj)/12) ||' Years and ' ||
                        ROUND(MOD(MONTHS_BETWEEN(SYSDATE,doj),12))||' months') AS Tenure
FROM employees
;
```

- The above **decode** is well depicted below.



```
DECODE( ROUND( MOD( MONTHS_BETWEEN (SYSDATE,doj), 12)) , 12,
FLOOR( MONTHS_BETWEEN(SYSDATE,doj)/12)+1 ||' Years and 0 months',
FLOOR(MONTHS_BETWEEN(SYSDATE,doj)/12) ||' Years and ' ||
ROUND(MOD(MONTHS_BETWEEN(SYSDATE,doj),12))||' months') AS Tenure
```

OUTPUT

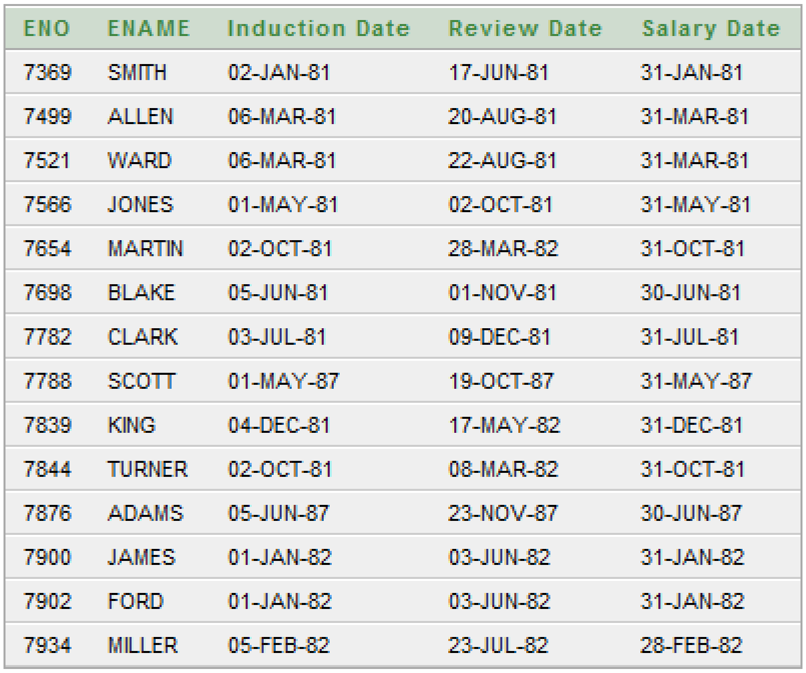


ENO	ENAME	TENURE
7369	SMITH	30 Years and 2 months
7499	ALLEN	30 Years and 0 months
7521	WARD	30 Years and 0 months
7566	JONES	29 Years and 10 months
7654	MARTIN	29 Years and 5 months
7698	BLAKE	29 Years and 9 months
7782	CLARK	29 Years and 8 months
7788	SCOTT	23 Years and 10 months
7839	KING	29 Years and 3 months
7844	TURNER	29 Years and 5 months
7876	ADAMS	23 Years and 9 months
7900	JAMES	29 Years and 2 months
7902	FORD	29 Years and 2 months
7934	MILLER	29 Years and 1 months

- Query for Induction Date, Review Date, Salary Date

```
SELECT          eno, ename,
                        NEXT_DAY(LAST_DAY(doj),6)          AS "Induction Date",

                        ADD_MONTHS(doj,6)                  AS "Review Date",

                        LAST_DAY(ADD_MONTHS(doj,1))        AS "Salary Date"
FROM            employees
;
```

<u>OUTPUT</u>

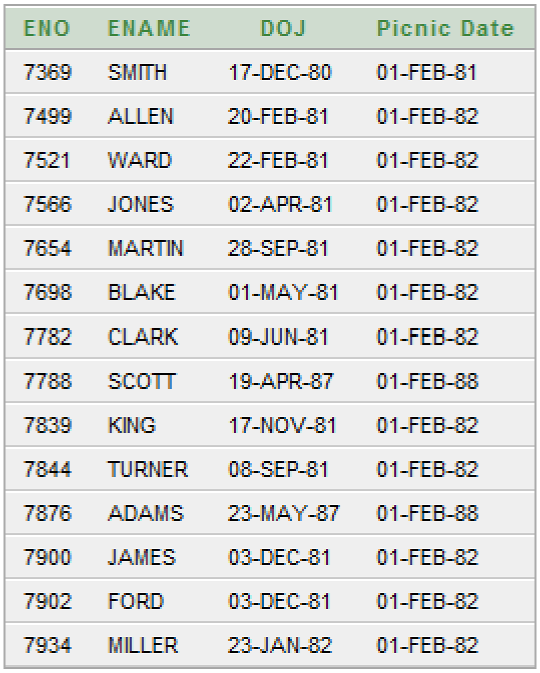


ENO	ENAME	Induction Date	Review Date	Salary Date
7369	SMITH	02-JAN-81	17-JUN-81	31-JAN-81
7499	ALLEN	06-MAR-81	20-AUG-81	31-MAR-81
7521	WARD	06-MAR-81	22-AUG-81	31-MAR-81
7566	JONES	01-MAY-81	02-OCT-81	31-MAY-81
7654	MARTIN	02-OCT-81	28-MAR-82	31-OCT-81
7698	BLAKE	05-JUN-81	01-NOV-81	30-JUN-81
7782	CLARK	03-JUL-81	09-DEC-81	31-JUL-81
7788	SCOTT	01-MAY-87	19-OCT-87	31-MAY-87
7839	KING	04-DEC-81	17-MAY-82	31-DEC-81
7844	TURNER	02-OCT-81	08-MAR-82	31-OCT-81
7876	ADAMS	05-JUN-87	23-NOV-87	30-JUN-87
7900	JAMES	01-JAN-82	03-JUN-82	31-JAN-82
7902	FORD	01-JAN-82	03-JUN-82	31-JAN-82
7934	MILLER	05-FEB-82	23-JUL-82	28-FEB-82

- Query for Picnic Date

```
SELECT          eno,ename,doj,

                CASE WHEN EXTRACT(MONTH FROM doj)>=2 THEN '01-FEB-'||
                MOD(EXTRACT(YEAR FROM doj)+1,1900) ELSE '01-FEB-'||MOD(EXTRACT(YEAR FROM doj),1900)
                END
                AS "Picnic Date"

FROM            employees;
```

OUTPUT

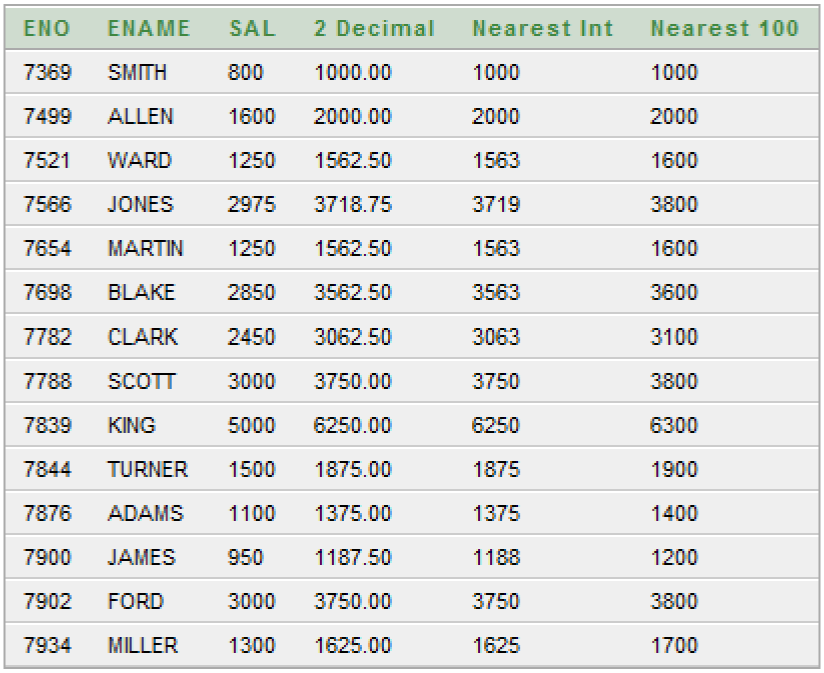


ENO	ENAME	DOJ	Picnic Date
7369	SMITH	17-DEC-80	01-FEB-81
7499	ALLEN	20-FEB-81	01-FEB-82
7521	WARD	22-FEB-81	01-FEB-82
7566	JONES	02-APR-81	01-FEB-82
7654	MARTIN	28-SEP-81	01-FEB-82
7698	BLAKE	01-MAY-81	01-FEB-82
7782	CLARK	09-JUN-81	01-FEB-82
7788	SCOTT	19-APR-87	01-FEB-88
7839	KING	17-NOV-81	01-FEB-82
7844	TURNER	08-SEP-81	01-FEB-82
7876	ADAMS	23-MAY-87	01-FEB-88
7900	JAMES	03-DEC-81	01-FEB-82
7902	FORD	03-DEC-81	01-FEB-82
7934	MILLER	23-JAN-82	01-FEB-82

- Print the salary of all employees incremented by 25%, accurate upto
  1. Two decimal places
  2. To the nearest integer
  3. To the nearest hundred

NOTE: You cannot reduce the salary of any employee to bring it to the nearest integer or to nearest hundred, but you are allowed to increase.

```
SELECT  eno,ename,sal,
        TO_CHAR(ROUND(sal*1.25,2),'9999999.00')    AS "2 decimal",
        CEIL(sal*1.25)                              AS "Nearest Int",
        CEIL(sal*1.25/100)*100                      AS "Nearest 100"

FROM employees;
```



ENO	ENAME	SAL	2 Decimal	Nearest Int	Nearest 100
7369	SMITH	800	1000.00	1000	1000
7499	ALLEN	1600	2000.00	2000	2000
7521	WARD	1250	1562.50	1563	1600
7566	JONES	2975	3718.75	3719	3800
7654	MARTIN	1250	1562.50	1563	1600
7698	BLAKE	2850	3562.50	3563	3600
7782	CLARK	2450	3062.50	3063	3100
7788	SCOTT	3000	3750.00	3750	3800
7839	KING	5000	6250.00	6250	6300
7844	TURNER	1500	1875.00	1875	1900
7876	ADAMS	1100	1375.00	1375	1400
7900	JAMES	950	1187.50	1188	1200
7902	FORD	3000	3750.00	3750	3800
7934	MILLER	1300	1625.00	1625	1700