5차시 Quiz 정답

홍형경
chariehong@gmail.com
2020.11

Quiz

1. Jobs 테이블에는 min_salary와 max_salary란 컬럼이 있는데, 이는 해당 job_id에 대한 최소와 최대급여 금액을 담고 있습니다. Jobs 테이블과 employees 테이블을 조인하고 사원의 급여가 최소와 최대급여 금액을 벗어난 사원이 있는지 조회하는 쿼리를 작성해 보세요.

정답 :

```
SELECT a.employee_id, a.first_name || ' ' || a.last_name emp_names
  FROM employees a,
            jobs b
 WHERE a.job_id = b.job_id
   AND     a.salary NOT BETWEEN b.min_salary AND b.max_salary
 ORDER BY 1;
```


Quiz

2. 아래 외부조인 문장을 실행하면 내부조인을 한 것과 결과가 같습니다. 왜 이런 결과가 나왔는지 설명해 보세요.

```
SELECT a.employee_id, a.first_name || ' ' || a.last_name emp_names, b.*
  FROM employees a,
       job_history b
 WHERE a.employee_id(+) = b.employee_id
 ORDER BY 1;
```

정답 :

a쪽 테이블 조인조건에 (+)가 붙어 있으므로 b, 즉 job_history 테이블쪽 데이터 중 조인조건에 부합하지 않는 데이터가 조회되어야 합니다.

그런데 job_history 테이블에 있는 employee_id 값은 모두 employees 테이블에 존재하므로 내부조인을 한 것과 같은 결과가 조회되는 것입니다.

Quiz

3. 실습시간 마지막에 배웠던 셀프조인의 경우 사번이 100번인 Steven King은 조회되지 않습니다. 그 이유는 뭘까요?

정답 :

사번이 100번인 Steven King의 경우 manager_id 값이 null 이라서 조인조건에서 누락되어 조회되지 않습니다.

Quiz

4. 실습시간 마지막에 배웠던 셀프조인에서 누락된 사번이 100번인 Steven King 까지 조회되도록 쿼리를 작성해 보세요.

정답 :

```
SELECT a.employee_id
      ,a.first_name || ' ' || a.last_name emp_name
      ,a.manager_id
      ,b.first_name || ' ' || b.last_name manager_name
 FROM employees a
 LEFT JOIN employees b
   ON a.manager_id = b.employee_id
ORDER BY 1;
```


Quiz

5. Quiz 2-2-6번 문제인 EMPLOYEES 테이블에서 FIRST_NAME이 'David'이고 급여가 6000이상인 사람이 속한 부서가 위치한 도시를 찾는 쿼리를 3문장이 아닌 1문장으로 작성해 보세요.

정답 :

```
SELECT a.employee_id, a.first_name, a.last_name, a.salary, b.department_name, c.city
  FROM employees a,
       departments b,
       locations c
 WHERE a.first_name = 'David'
   AND a.salary > 6000
   AND a.department_id = b.department_id
   AND b.location_id = c.location_id  ;
```


Quiz

6. ORDERS, CUSTOMERS, STORES, STAFFS 테이블을 조인해 2018년 1월 주문 내역에 대해 다음 결과처럼 조회하는 쿼리를 작성해 보세요.

ORDER_ID	ORDER_DATE	CUSTOMER_NAME	STORE_NAME	STAFF_NAME
1325	2018-01-01 00:00:00	Jayne Kirkland	Rowlett Bikes	Kali Vargas
1324	2018-01-01 00:00:00	Mellie Puckett	Baldwin Bikes	Marcelene Boyer
1326	2018-01-01 00:00:00	Sheila Travis	Rowlett Bikes	Layla Terrell
1327	2018-01-02 00:00:00	Jenine Dawson	Baldwin Bikes	Marcelene Boyer
1328	2018-01-02 00:00:00	Cher Alston	Baldwin Bikes	Marcelene Boyer
1329	2018-01-04 00:00:00	Ayana Keith	Santa Cruz Bikes	Genna Serrano
1330	2018-01-04 00:00:00	Rod Hatfield	Baldwin Bikes	Venita Daniel
1331	2018-01-05 00:00:00	Cicely Deleon	Baldwin Bikes	Marcelene Boyer
1332	2018-01-06 00:00:00	Erma Salinas	Baldwin Bikes	Venita Daniel
1333	2018-01-06 00:00:00	Minerva Decker	Rowlett Bikes	Kali Vargas
1334	2018-01-07 00:00:00	Augustina Joyner	Baldwin Bikes	Marcelene Boyer
1335	2018-01-07 00:00:00	Delfina Gilliam	Baldwin Bikes	Venita Daniel
1336	2018-01-09 00:00:00	Jana Thomas	Santa Cruz Bikes	Mireya Copeland
1337	2018-01-09 00:00:00	Ruth Horton	Baldwin Bikes	Venita Daniel
1338	2018-01-10 00:00:00	Hae Ramirez	Baldwin Bikes	Marcelene Boyer
1339	2018-01-11 00:00:00	Mellisa Kim	Santa Cruz Bikes	Genna Serrano
1340	2018-01-12 00:00:00	Raeann Duncan	Santa Cruz Bikes	Genna Serrano
1341	2018-01-12 00:00:00	Todd Waters	Baldwin Bikes	Venita Daniel
1342	2018-01-12 00:00:00	Vivian Deleon	Baldwin Bikes	Venita Daniel

```
SELECT  a.order_id , a.order_date
      ,b.first_name || ' ' || b.last_name customer_name
      ,c.store_name
      ,d.first_name || ' ' || d.last_name staff_name
FROM orders a,
    customers b,
    stores c,
    staffs d
WHERE a.order_date BETWEEN TO_DATE('2018-01-01','YYYY-MM-DD')
                       AND TO_DATE('2018-01-31','YYYY-MM-DD')
  AND a.customer_id = b.customer_id
  AND a.store_id = c.store_id
  AND a.staff_id = d.staff_id
ORDER BY 2;
```


Quiz

7. ORDERS, ORDER_ITEMS 테이블을 조인해 2018년 월별 주문금액 합계를 조회하는 쿼리를 ANSI 조인으로 작성해 보세요. (주문금액 = order_items 의 quantity * list_price)

	MONTHS	ORDER_AMT
1	2018-01	426301.72
2	2018-02	223941.44
3	2018-03	406701.2
4	2018-04	909179.47
5	2018-06	209.99
6	2018-07	12949.89
7	2018-08	10256.91
8	2018-09	9949.96
9	2018-10	4219.92
0	2018-11	12278.93
1	2018-12	7999.96

```
SELECT TO_CHAR(a.order_date, 'YYYY-MM') months
     ,SUM(b.quantity * b.list_price) order_amt
FROM ORDERS A
INNER JOIN ORDER_ITEMS B
  ON A.ORDER_ID = B.ORDER_ID
WHERE a.order_date BETWEEN TO_DATE('2018-01-01','YYYY-MM-DD')
                       AND TO_DATE('2018-12-31','YYYY-MM-DD')
GROUP BY TO_CHAR(a.order_date, 'YYYY-MM')
ORDER BY 1;
```


Quiz

8. ORDERS, ORDER_ITEMS, PROUCTS, BRANDS 테이블을 조인해 2018년 분기별, 브랜드별 주문금액 합계를 조회하는데, 주문금액이 10000 이상인 데이터를 조회하는 쿼리를 ANSI 조인으로 작성해 보세요. (주문금액 = order_items 의 quantity * list_price)

QUARTER	BRAND_NAME	ORDER_AMT
2018-1	Trek	736187.1
2018-1	Electra	186717.45
2018-1	Surly	77605.7
2018-1	Sun Bicycles	26882.46
2018-1	Heller	14315.99
2018-2	Trek	618357.74
2018-2	Electra	170697.85
2018-2	Surly	57719.73
2018-2	Sun Bicycles	22965.59
2018-2	Haro	22819.68
2018-2	Heller	10396
2018-3	Trek	22509.9
2018-4	Trek	16699.94

```
SELECT TO_CHAR(a.order_date, 'YYYY-Q') quarter
     ,d.brand_name
     ,SUM(b.quantity * b.list_price) order_amt
FROM ORDERS A
INNER JOIN ORDER_ITEMS B
  ON A.ORDER_ID = B.ORDER_ID
INNER JOIN PRODUCTS C
  ON B.PRODUCT_ID = C.PRODUCT_ID
INNER JOIN BRANDS D
  ON C.BRAND_ID = D.BRAND_ID
WHERE a.order_date BETWEEN TO_DATE('2018-01-01','YYYY-MM-DD')
                       AND TO_DATE('2018-12-31','YYYY-MM-DD')
GROUP BY TO_CHAR(a.order_date, 'YYYY-Q'), D.BRAND_NAME
HAVING SUM(b.quantity * b.list_price) >= 10000
ORDER BY 1, 3 DESC;
```


Quiz

9. 년도별 매장별 주문금액 합계를 조회하는 쿼리를 ANSI 조인으로 작성해 보세요. (주문금액 = order_items 의 quantity * list_price)

QUARTER	STORE_NAME	ORDER_AMT
2016	Baldwin Bikes	1781131.68
2016	Rowlett Bikes	299407.72
2016	Santa Cruz Bikes	628945.07
2017	Baldwin Bikes	2764466.2
2017	Rowlett Bikes	450966.48
2017	Santa Cruz Bikes	630082.34
2018	Baldwin Bikes	1280644.33
2018	Rowlett Bikes	212226.56
2018	Santa Cruz Bikes	531118.5

```
SELECT TO_CHAR(a.order_date, 'YYYY') years
     ,e.store_name
     ,SUM(b.quantity * b.list_price) order_amt
FROM ORDERS A
INNER JOIN ORDER_ITEMS B
  ON A.ORDER_ID = B.ORDER_ID
INNER JOIN STORES E
 ON A.STORE_ID = E.STORE_ID
WHERE a.order_date BETWEEN TO_DATE('2016-01-01','YYYY-MM-DD')
                       AND TO_DATE('2019-12-31','YYYY-MM-DD')
GROUP BY TO_CHAR(a.order_date, 'YYYY'),  e.store_name
ORDER BY 1;
```


Quiz

10. 분석함수를 사용해 다음과 같이 누적합계를 구하는 쿼리를 작성해 보세요.
(힌트: SUM 함수 사용)

DEPARTMENT_ID	DEPARTMENT_NAME	EMP_NAME	HIRE_DATE	SALARY	누적합계
110	Accounting	William Gietz	2002-06-07 00:00:00	8300	8300
110	Accounting	Shelley Higgins	2002-06-07 00:00:00	12008	20308
10	Administration	Jennifer Whalen	2003-09-17 00:00:00	4400	4400
90	Executive	Lex De Haan	2001-01-13 00:00:00	17000	17000
90	Executive	Steven King	2003-06-17 00:00:00	24000	41000
90	Executive	Neena Kochhar	2005-09-21 00:00:00	17000	58000
100	Finance	Daniel Faviet	2002-08-16 00:00:00	9000	9000
100	Finance	Nancy Greenberg	2002-08-17 00:00:00	12008	21008
100	Finance	John Chen	2005-09-28 00:00:00	8200	29208
100	Finance	Ismael Sciarra	2005-09-30 00:00:00	7700	36908
100	Finance	Jose Manuel Urman	2006-03-07 00:00:00	7800	44708
100	Finance	Luis Popp	2007-12-07 00:00:00	6900	51608
40	Human Resources	Susan Mavris	2002-06-07 00:00:00	6500	6500
60	IT	David Austin	2005-06-25 00:00:00	4800	4800
60	IT	Alexander Hunold	2006-01-03 00:00:00	9000	13800
60	IT	Valli Pataballa	2006-02-05 00:00:00	4800	18600
60	IT	Diana Lorentz	2007-02-07 00:00:00	4200	22800
60	IT	Bruce Ernst	2007-05-21 00:00:00	6000	28800

Quiz

10. 분석함수를 사용해 다음과 같이 누적합계를 구하는 쿼리를 작성해 보세요.
(힌트: SUM 함수 사용)

정답 :

```
SELECT b.department_id, b.department_name,
       a.first_name || ' ' || a.last_name as emp_name,
       a.hire_date,
       a.salary ,
       sum(salary) OVER (PARTITION BY b.department_id
                         ORDER BY a.hire_date, a.salary ) 누적합계
  FROM employees a,
       departments b
 WHERE a.department_id = b.department_id
 ORDER BY 2, 4, 5 ;
```

