Part II.B- #1

BREAK ON is used in this query

```
SQL> BREAK ON Employee
SQL> SELECT E.emp_num || ': ' || E.Fname || ' ' || E.Lname AS Employee,
  2         NVL(S.name,'No Skill') AS Skills_Acquired,
  3         REPLACE(TO_CHAR(count(Train_Num)),0,'-') AS Train_Count,
  4         NVL(TO_CHAR(MIN(T.Date_Acquired)),'---------') AS Earliest_Date,
  5         NVL(TO_CHAR(ROUND(Months_Between(SYSDATE,MAX(T.Date_Acquired)))),'---') AS Months_Passed
  6  FROM employee E LEFT OUTER JOIN training T ON (E.Emp_Num=T.Emp_Num)
  7                  LEFT OUTER JOIN skill    S ON (T.Code=S.Code)
  8  GROUP BY E.emp_num, E.lname, E.fname,S.name
  9  ORDER BY 1,3 DESC;
```

EMPLOYEE	SKILLS_ACQUIRED	TRAIN_COUNT	EARLIEST_DATE	MONTHS_PASSED
4001: Ken Mary	Python	3	10-JAN-03	201
4002: Lisa Wang	Sql	1	20-JAN-03	226
4003: Katty Li	Marketing	1	15-JUN-21	6
	Sql	1	20-FEB-21	9
4004: Melody Ran	PowerPoint	1	01-NOV-21	1
	Negotiation	1	15-OCT-21	2
4005: Ruby Smith	Word	1	10-SEP-09	147

4006: Leah Madore	Word	1	10-SEP-21	3
4007: Jassica Madore	Word	1	01-OCT-19	26
4008: David Novac	Python	1	10-JAN-19	35
4009: Lily Huang	Python	1	10-FEB-19	34
4010: Jewel Lin	Sql	3	10-MAR-09	129
	Marketing	1	15-JUN-09	150
4011: Mercy Hilton	Negotiation	1	15-NOV-21	1
	PowerPoint	1	01-OCT-21	2
4012: Sara Peng	Word	1	10-SEP-21	3
4013: Sara Kim	Word	1	10-SEP-02	231
4014: John Murry	Word	1	01-OCT-11	122
4015: Tony Murry	Python	1	10-SEP-21	3
4016: Toby Li	Python	2	10-FEB-06	177
4017: Zheng Ning	Sql	2	20-JAN-19	33
4018: Lucy Forbes	Marketing	1	15-JUN-19	30
4019: Victoria Hilar	Negotiation	1	15-NOV-07	169
4020: Ling Kevin	Marketing	1	01-OCT-11	122
4021: Hong Jason	No Skill	-	---------	---
4022: Milton Cassay	No Skill	-	---------	---

26 rows selected.

Part II.B- #2

In the employee tables, there is one CEO (#4013), 5 department managers (4001, 4007, 4010, 4018, 4020). CEO supervises 5 managers, and 5 managers supervises other employees.

```
SQL> SELECT LEVEL,
  2         LPAD(' ', 4*(LEVEL-1)) || emp_num || ': ' || Fname || ' ' || Lname AS Employee,
  3         Name AS Department
  4  FROM   employee JOIN department USING (dept_code)
  5  START WITH emp_num=4013
  6  CONNECT BY PRIOR emp_num=Super_ID;

     LEVEL EMPLOYEE                       DEPARTMENT
---------- ------------------------------ -------------------------
         1 4013: Sara Kim                 Technology Department
         2     4001: Ken Mary             Personnel Department
         3         4002: Lisa Wang        Personnel Department
         3         4003: Katty Li         Personnel Department
         2     4007: Jassica Madore       Technology Department
         3         4008: David Novac      Technology Department
         3         4009: Lily Huang       Technology Department
         3         4011: Mercy Hilton     Technology Department
         3         4012: Sara Peng        Technology Department
         3         4014: John Murry       Technology Department
```

```
3         4015: Tony Murry       Technology Department
2     4010: Jewel Lin            Marketing Department
3         4004: Melody Ran       Marketing Department
3         4005: Ruby Smith       Marketing Department
3         4006: Leah Madore      Marketing Department
2     4018: Lucy Forbes          Operational Department
3         4016: Toby Li          Operational Department
3         4017: Zheng Ning       Operational Department
3         4019: Victoria Hilary  Operational Department
2     4020: Ling Kevin           Administration Department
3         4021: Hong Jason       Administration Department
3         4022: Milton Cassay    Administration Department

22 rows selected.
```

Part II.B- #3

BREAK ON is used in this query

Here we our assuming that an ongoing project is the one which has an assignment in the month of decemeber,2021

The start date for the project G is in December,2021. Hence the number of hrs used are not available for this project

```
SQL> BREAK ON NAME
SQL> SELECT name, TO_CHAR(start_date) AS Start_Date,
  2         TO_CHAR(extract(month from date_assigned)) || '/' ||
  3                 to_CHAR(extract(year from date_assigned)) as Project_Month,
  4         COUNT(EMP_NUM) AS "No. Employees",
  5         NVL(TO_CHAR(SUM(HOURS_USED)),'Not Available(On Going)') AS "No. of Hours"
  6  FROM  project JOIN assignment USING(proj_number)
  7  WHERE total_cost IS NULL
  8  GROUP BY  name, start_date,
  9            extract(month from date_assigned),extract(year from date_assigned)
 10  UNION ALL
 11  SELECT 'Total of ' || NAME AS Project, '----','----', COUNT(DISTINCT(emp_num)),
 12          NVL(TO_CHAR(SUM(HOURS_USED)),'Not Available(On Going)')
 13  FROM project JOIN assignment USING(proj_number)
 14  WHERE total_cost IS NULL
 15  GROUP BY proj_number, name;
```

NAME	START_DATE	PROJECT_MONTH	No. Employees	No. of Hours
Project E	01-DEC-21	12/2021	3	Not Available(On Going)
Project G	01-SEP-20	9/2020	3	350
	01-SEP-20	12/2021	1	Not Available(On Going)
Total of Project G	----	----	4	350
Total of Project E	----	----	3	Not Available(On Going)

Part II.B- #4

```
SQL> ALTER TABLE employee
  2  ADD(Bonus_AMT VARCHAR(10));

Table altered.

SQL> UPDATE employee e1
  2  SET  Bonus_AMT = (SELECT TO_CHAR(NVL(Bonus,0),'$9999')
  3  FROM(
  4       SELECT * FROM
  5       employee LEFT OUTER JOIN (
  6                                SELECT Emp_num, count(proj_number)*200 AS Bonus
  7                                FROM assignment
  8                                WHERE proj_number IN(
  9                                                     SELECT DISTINCT Proj_number
 10                                                     FROM project JOIN assignment USING (proj_number)
 11                                                      WHERE EXTRACT(MONTH FROM start_date) IN (1,2,3)
 12                                                      AND
 13                                                      EXTRACT(YEAR FROM start_date) = 2021)
 14                               AND
 15                               Hours_used > 150
 16                               GROUP BY emp_num) USING (emp_num)) e2
 17  WHERE e1.emp_num=e2.emp_num);
22 rows updated.
```

```
SQL> SELECT * FROM employee;
```

EMP_NUM	LNAME	FNAME	DOB	HIRE_DATE	SUPER_ID	DEPT_CODE	BONUS_AMT
4013	Kim	Sara	03-JUL-73	01-SEP-01		3003	$0
4001	Mary	Ken	12-NOV-80	01-SEP-02	4013	3001	$0
4007	Madore	Jassica	04-MAY-89	01-JUN-18	4013	3003	$0
4010	Lin	Jewel	23-OCT-86	01-JUN-08	4013	3002	$0
4018	Forbes	Lucy	16-SEP-76	01-MAR-18	4013	3004	$0
4020	Kevin	Ling	01-SEP-88	01-MAR-07	4013	3005	$0
4002	Wang	Lisa	20-DEC-75	01-MAY-01	4001	3001	$0
4003	Li	Katty	21-MAR-95	01-MAY-20	4001	3001	$0
4004	Ran	Melody	06-JAN-00	01-MAY-21	4010	3002	$0
4005	Smith	Ruby	05-MAR-90	01-MAY-08	4010	3002	$0
4006	Madore	Leah	23-MAR-91	01-MAY-20	4010	3002	$0
4008	Novac	David	17-MAR-97	01-JUN-18	4007	3003	$0
4009	Huang	Lily	21-APR-87	01-JUN-18	4007	3003	$0
4011	Hilton	Mercy	12-NOV-97	01-SEP-21	4007	3003	$0
4012	Peng	Sara	07-DEC-98	01-SEP-21	4007	3003	$0
4014	Murry	John	21-JAN-86	01-SEP-10	4007	3003	$0
4015	Murry	Tony	04-JAN-01	01-SEP-21	4007	3003	$400
4016	Li	Toby	03-JUN-76	01-SEP-05	4018	3004	$200
4017	Ning	Zheng	23-MAY-94	01-MAR-18	4018	3004	$0
4019	Hilary	Victoria	03-AUG-87	01-MAR-06	4018	3004	$0
4021	Jason	Hong	23-MAY-99	01-SEP-21	4020	3005	$0
4022	Cassay	Milton	04-JAN-01	01-MAR-21	4020	3005	$0

```
22 rows selected.
```

Part II.B- #5

BREAK ON is used in this query

Quarter definition if: January, February, and March the Quarter is (1), April, May, and June the Quarter is (2) July, August, and September the Quarter is (3),October, November, and December the Quarter is (4)

```
SQL> BREAK ON EMPLOYEE ON Hire_Date ON Project_Count
SQL> SELECT emp_num || ': ' ||Employee AS Employee,
  2         Hire_Date,
  3         NVL(num_proj,0) AS Project_Count,
  4         Training_Name,Train_date,
  5         Days_Between_Hiring_Training
  6  FROM(
  7        (SELECT E.emp_num, E.Fname || ' ' || E.Lname AS Employee,
  8                E.hire_date AS Hire_Date, T.name AS Training_Name, T.date_acquired AS Train_date,
  9                TO_CHAR((T.Date_Acquired-E.hire_date),'9999') AS Days_Between_Hiring_Training
 10         FROM   employee E JOIN training T ON (E.emp_num=T.emp_num)
 11         WHERE EXTRACT(MONTH FROM E.hire_date) IN (4,5,6)
 12         AND EXTRACT(YEAR FROM E.hire_date) =2021
 13         ORDER BY E.emp_num)
 14         LEFT OUTER JOIN(
 15                         SELECT emp_num, TO_CHAR(NVL(count(proj_number),0),'99') AS num_proj
 16                         FROM assignment
 17                         GROUP by emp_num)
 18         USING (emp_num));
```

```
EMPLOYEE            HIRE_DATE PROJECT_COUNT   TRAINING_NAME                           TRAIN_DATE DAYS_BETWEEN_HIRING_TRAINING
------------------- --------- -------------- --------------------------------------- ---------- ------------------------------
4004: Melody Ran    01-MAY-21 0              Negotiation Strategy for Consulting     15-OCT-21    167
                                             PowerPoint Tips                         01-NOV-21    184
```

Part II.B- #6

"Discontinued" means the difference between date_assigned of one assignment and date_end of its previous assignment is larger than 30 days.

Here we our assuming that an ongoing project is the one which has an assignment in the month of December, 2021

```
SQL> COLUMN Discontinued_Project format a20
SQL> SELECT Proj_Number || ': ' || name AS Discontinued_Project, Start_date,
  2         CASE NVL(Total_Cost,0)
  3              WHEN 0 THEN 'on-going'
  4              ELSE    'completed'
  5         END  Status
  6  FROM project JOIN(
  7                    SELECT Proj_number,Assign_Num,Date_Assigned,Date_Ended,
  8                    (LAG(Date_Ended,1) OVER (
  9                                             PARTITION BY Proj_Number
  10                                            ORDER BY Date_Assigned)) AS Previous_end
  11                    FROM assignment
```

```
 12                    )
 13                USING(proj_number)
 14  WHERE Date_Assigned-Previous_end>30;

DISCONTINUED_PROJECT START_DATE STATUS
-------------------- ---------- ---------
6001: Project A      01-MAR-20  completed
6007: Project G      01-SEP-20  on-going
```

Part II.B- #7

Quarter definition if: January, February, and March the Quarter is(1), April, May, and June the Quarter is (2)

July, August, and September the Quarter is (3), October, November, and December the Quarter is (4)

Average hrs for the project in quarter is not available as there are ongoing projects in the quarter

```
SQL> SELECT FLOOR(((EXTRACT(month FROM START_DATE) - 1)/ 3) + 1) quarter,
  2         COUNT(DISTINCT(PROJ_NUMBER)) AS "#. of Projects",
  3         COUNT(DISTINCT(EMP_NUM)) AS "#. of Employees",
  4         NVL(TO_CHAR(FLOOR(SUM(HOURS_USED)/COUNT(PROJ_NUMBER))),'Not Available(On going)') AS "Average #. of Hours"
  5  FROM PROJECT JOIN ASSIGNMENT USING(PROJ_NUMBER)
  6  WHERE EXTRACT(year FROM START_DATE) = '2021'
  7  GROUP BY FLOOR(((EXTRACT(month FROM START_DATE) - 1)/ 3) + 1);

   QUARTER #. of Projects #. of Employees Average #. of Hours
---------- -------------- --------------- ----------------------------------------
         1              3               2 200
         2              1               2 150
         3              1               1 100
         4              1               3 Not Available(On going)
```

Part II.B- #8

```
SQL> column Employee_Name format a20
SQL> column ID format a4
SQL> column LD format a9
SQL> column Python format 99
SQL> column SQL format 99
SQL> column Marketing format 99
SQL> column Negotiation format 99
SQL> column Powerpoint format 99
SQL> column word format 99
SQL> column NumberOfSkills format a15
SQL> SELECT DECODE(E.Emp_num, NULL, '---', E.Emp_Num) "ID",
  2         DECODE(E.fname || ' ' || E.lname, NULL, 'Number of Trainings: ',E.fname || ' ' || E.lname) "Employee_Name",
  3         NVL(SUM(DECODE(S.code, 1001, 1,0)),0) "Python",
  4         NVL(TO_CHAR(MAX(DECODE(S.code, 1001, T.Date_Acquired,Null))),'------') "LD",
  5         NVL(SUM(DECODE(S.code, 1002, 1,0)),0) "SQL",
  6         NVL(TO_CHAR(MAX(DECODE(S.code, 1002, T.Date_Acquired,Null))),'------') "LD",
  7         NVL(SUM(DECODE(S.code, 1011, 1,0)),0) "Marketing",
  8         NVL(TO_CHAR(MAX(DECODE(S.code, 1011, T.Date_Acquired,Null))),'------') "LD",
  9         NVL(SUM(DECODE(S.code, 1012, 1,0)),0) "Negotiation",
 10         NVL(TO_CHAR(MAX(DECODE(S.code, 1012, T.Date_Acquired,Null))),'------') "LD",
 11         NVL(SUM(DECODE(S.code, 1021, 1,0)),0) "Powerpoint",
 12         NVL(TO_CHAR(MAX(DECODE(S.code, 1021, T.Date_Acquired,Null))),'------') "LD",
 13         NVL(SUM(DECODE(S.code, 1022, 1,0)),0) "word",
 14         NVL(TO_CHAR(MAX(DECODE(S.code, 1022, T.Date_Acquired,Null))),'------') "LD",
 15         TO_CHAR(COUNT(DISTINCT S.code)) "NumberOfSkills"
```

```
 16  FROM   employee E LEFT OUTER JOIN training T ON (E.Emp_Num=T.Emp_Num)
 17                    LEFT OUTER JOIN skill S ON (T.Code=S.Code)
 18  GROUP BY GROUPING SETS((E.Emp_num, E.lname, E.fname))
 19  UNION ALL
 20  SELECT '----', 'Number of Trainings: ',
 21         SUM(DECODE(code, 1001, 1,0)),'------',
 22         SUM(DECODE(code, 1002, 1,0)),'------',
 23         SUM(DECODE(code, 1011, 1,0)),'------',
 24         SUM(DECODE(code, 1012, 1,0)),'------',
 25         SUM(DECODE(code, 1021, 1,0)),'------',
 26         SUM(DECODE(code, 1022, 1,0)),'------',
 27         '---------'
 28  FROM training
 29  GROUP BY GROUPING SETS (());
```

ID	Employee_Name	Python	LD	SQL	LD	Marketing	LD	Negotiation	LD	Powerpoint	LD	word	LD	NumberOfSkills
4001	Ken Mary	3	10-MAR-05	0	------	0	------	0	------	0	------	0	------	1
4002	Lisa Wang	0	------	1	20-JAN-03	0	------	0	------	0	------	0	------	1
4003	Katty Li	0	------	1	20-FEB-21	1	15-JUN-21	0	------	0	------	0	------	2
4004	Melody Ran	0	------	0	------	0	------	1	15-OCT-21	1	01-NOV-21	0	------	2
4005	Ruby Smith	0	------	0	------	0	------	0	------	0	------	1	10-SEP-09	1
4006	Leah Madore	0	------	0	------	0	------	0	------	0	------	1	10-SEP-21	1
4007	Jassica Madore	0	------	0	------	0	------	0	------	0	------	1	01-OCT-19	1
4008	David Novac	1	10-JAN-19	0	------	0	------	0	------	0	------	0	------	1
4009	Lily Huang	1	10-FEB-19	0	------	0	------	0	------	0	------	0	------	1
4010	Jewel Lin	0	------	3	20-FEB-11	1	15-JUN-09	0	------	0	------	0	------	2

4011 Mercy Hilton	0 ------	0 ------	0 ------	1 15-NOV-21	1 01-OCT-21	0 ------	2
4012 Sara Peng	0 ------	0 ------	0 ------	0 ------	0 ------	1 10-SEP-21	1
4013 Sara Kim	0 ------	0 ------	0 ------	0 ------	0 ------	1 10-SEP-02	1
4014 John Murry	0 ------	0 ------	0 ------	0 ------	0 ------	1 01-OCT-11	1
4015 Tony Murry	1 10-SEP-21	0 ------	0 ------	0 ------	0 ------	0 ------	1
4016 Toby Li	2 10-MAR-07	0 ------	0 ------	0 ------	0 ------	0 ------	1
4017 Zheng Ning	0 ------	2 20-FEB-19	0 ------	0 ------	0 ------	0 ------	1
4018 Lucy Forbes	0 ------	0 ------	1 15-JUN-19	0 ------	0 ------	0 ------	1
4019 Victoria Hilary	0 ------	0 ------	0 ------	1 15-NOV-07	0 ------	0 ------	1
4020 Ling Kevin	0 ------	0 ------	1 01-OCT-11	0 ------	0 ------	0 ------	1
4021 Hong Jason	0 ------	0 ------	0 ------	0 ------	0 ------	0 ------	0
4022 Milton Cassay	0 ------	0 ------	0 ------	0 ------	0 ------	0 ------	0
---- Number of Trainings:	8 ------	7 ------	4 ------	3 ------	2 ------	6 ------	---------

23 rows selected.

Part II.B- #9

BREAK ON is used in this query

```
SQL> BREAK ON Department
SQL> SELECT D.name AS Department,S.name AS Skill,count(*) AS Number_Trainings,
  2         RANK() OVER (PARTITION BY D.dept_code ORDER BY count(*) DESC) AS rank
  3  FROM
  4  Skill S JOIN training T on (S.code=T.code)
  5          JOIN employee E on (T.Emp_Num=E.Emp_Num)
  6          JOIN department D on (E.dept_code=D.dept_code)
  7  GROUP BY D.dept_code,D.name,S.name
  8  ORDER BY D.dept_code;
```

DEPARTMENT	SKILL	NUMBER_TRAININGS	RANK
Personnel Department	Python	3	1
	Sql	2	2
	Marketing	1	3
Marketing Department	Sql	3	1
	Word	2	2
	Marketing	1	3
	Negotiation	1	3
	PowerPoint	1	3

Technology Department	Word	4	1
	Python	3	2
	Negotiation	1	3
	PowerPoint	1	3
Operational Department	Python	2	1
	Sql	2	1
	Marketing	1	3
	Negotiation	1	3
Administration Department	Marketing	1	1

17 rows selected.

Part II.B- #10

Assuming that the question is asking for projects that have at least five monthly and then the total no of days in the first

3 assignments (assignments ranked on the basis of days b/w date_ended and date_assigned) of such projects is more than or equal to 60

```
SQL> SELECT PROJ_NUMBER AS Project_Number, SUM((DATE_ENDED-DATE_ASSIGNED)) "Total_Working_Days"
  2  FROM ASSIGNMENT
  3  WHERE PROJ_NUMBER IN (SELECT PROJ_NUMBER FROM (SELECT PROJ_NUMBER, SUM((DATE_ENDED-DATE_ASSIGNED))
  4                        FROM(SELECT PROJ_NUMBER, ASSIGN_NUM, DATE_ASSIGNED, DATE_ENDED
  5                              FROM(
  6                                   SELECT PROJ_NUMBER, ASSIGN_NUM, DATE_ASSIGNED, DATE_ENDED,
  7                                   ROW_NUMBER() OVER (ORDER BY (Date_Ended - Date_Assigned)) RANKING
  8                                   FROM ASSIGNMENT
  9                                   WHERE PROJ_NUMBER IN ( SELECT Proj_Number AS "Project Number"
 10                                                          FROM assignment
 11                                                          GROUP BY Proj_Number
 12                                                          HAVING COUNT(*) >= 5
 13                                                         )
 14                                    )
 15                                                         WHERE RANKING <4)
 16                                                         GROUP BY Proj_Number
 17                                                         HAVING SUM((DATE_ENDED-DATE_ASSIGNED))>=60))
 18  GROUP BY Proj_Number;

PROJECT_NUMBER Total_Working_Days
-------------- ------------------
          6004                146
```

Part II.B- #11

Most senior employees are those with earliest hire_date

```
SQL> SELECT T1.Emp_num, T1.Employee AS LastName, T1.hire_date, NVL(T1.Name,'-----') AS manage_dep,
  2         NVL(TO_CHAR(T2.supervising),'----') AS Supervising_Count
  3  FROM(
  4        (SELECT Emp_num, Lname AS Employee, hire_date, Name
  5         FROM
  6         (SELECT *FROM (SELECT* FROM employee ORDER BY Hire_date)
  7           WHERE ROWNUM<5)
  8         LEFT OUTER JOIN department  ON (Emp_Num = Manager_ID)
  9       ) T1
 10      LEFT OUTER JOIN
 11      (
 12        SELECT super_id,count(*)AS supervising
 13        FROM Employee
 14        GROUP BY super_id
 15        Having super_id IN  (SELECT emp_num FROM(SELECT *FROM(SELECT* FROM employee ORDER BY Hire_date)
 16        WHERE ROWNUM<5))
 17       ) T2 ON (T1.Emp_num=T2.super_id)
 18  );
```

EMP_NUM	LASTNAME	HIRE_DATE	MANAGE_DEP	SUPERVISING_COUNT
4013	Kim	01-SEP-01	-----	5
4001	Mary	01-SEP-02	Personnel Department	2
4002	Wang	01-MAY-01	-----	----
4016	Li	01-SEP-05	-----	----

Part II.B- #12

```
SQL> SELECT
  2        CASE NVL(substr(Web_Address, LENGTH(Web_Address)-2, 3),'XXX')
  3             WHEN 'edu' THEN 'Education Institution'
  4             WHEN 'gov' THEN 'Government Agency'
  5             WHEN 'org' THEN 'Non-Profit Organization'
  6             WHEN 'com' THEN 'For-Profit Company'
  7     WHEN 'XXX' THEN 'Not Available'
  8             ELSE            'Other'
  9        END "Client_Type",
 10        Count(DISTINCT Client_ID) AS Number_Of_Cients, Count(DISTINCT Proj_number) AS Number_Of_Proj
 11  FROM client LEFT OUTER JOIN project USING (Client_ID)
 12  GROUP BY
 13  ( CASE NVL(substr(Web_Address, LENGTH(Web_Address)-2, 3),'XXX')
 14             WHEN 'edu' THEN 'Education Institution'
 15             WHEN 'gov' THEN 'Government Agency'
 16             WHEN 'org' THEN 'Non-Profit Organization'
 17             WHEN 'com' THEN 'For-Profit Company'
 18             WHEN 'XXX' THEN 'Not Available'
 19             ELSE            'Other'
 20        END ) ;
```

Client_Type	NUMBER_OF_CIENTS	NUMBER_OF_PROJ
Education Institution	2	4
For-Profit Company	2	3
Non-Profit Organization	2	1
Not Available	1	0
Other	1	2

Part II.B- #13

```
SQL> SELECT E.Emp_num || ': ' || E.Fname || ' ' || E.Lname AS Employee,
  2         D.name AS Department, NVL(P.name, 'No Project Assigned') AS Last_Project
  3  FROM employee E LEFT OUTER JOIN department D ON (E.dept_code=D.dept_code)
  4                  LEFT OUTER JOIN assignment A ON (E.emp_num=A.emp_num)
  5                  LEFT OUTER JOIN project P ON (A.proj_number=P.proj_number)
  6  WHERE E.emp_num IN (
  7  SELECT emp_num
  8  FROM employee JOIN department USING (dept_code)
  9  MINUS
  10  SELECT emp_num
  11  FROM employee JOIN department USING (dept_code)
  12                JOIN assignment USING (emp_num)
  13  WHERE Date_Assigned > '01-July-2021'
  14  )
  15  AND (A.date_assigned IS NULL OR (E.emp_num, A.date_assigned) IN (SELECT emp_num,
  16                                                                   MAX(date_assigned)
  17                                                                   FROM assignment
  18                                                                   GROUP BY emp_num))
  19  ORDER BY D.name, E.Lname;

EMPLOYEE                       DEPARTMENT              LAST_PROJECT
```

4022: Milton Cassay	Administration Department	No Project Assigned
4021: Hong Jason	Administration Department	No Project Assigned
4020: Ling Kevin	Administration Department	No Project Assigned
4010: Jewel Lin	Marketing Department	Project F
4006: Leah Madore	Marketing Department	Project B
4004: Melody Ran	Marketing Department	No Project Assigned
4005: Ruby Smith	Marketing Department	Project B
4018: Lucy Forbes	Operational Department	Project G
4017: Zheng Ning	Operational Department	Project D
4003: Katty Li	Personnel Department	Project A
4001: Ken Mary	Personnel Department	Project F
4011: Mercy Hilton	Technology Department	No Project Assigned
4009: Lily Huang	Technology Department	No Project Assigned
4007: Jassica Madore	Technology Department	Project F
4015: Tony Murry	Technology Department	Project H
4014: John Murry	Technology Department	No Project Assigned
4008: David Novac	Technology Department	No Project Assigned
4012: Sara Peng	Technology Department	No Project Assigned

18 rows selected.

Part II.B- #14

```
SQL> (SELECT T1.skillname, trainingnumber AS Training_Count,projectnumber AS Project_Count
  2   FROM
  3     (
  4       (SELECT S.name AS skillname, count(*) AS trainingnumber
  5        FROM  skill S JOIN training T ON (S.code=T.code)
  6        GROUP by S.name
  7        ORDER BY S.name) T1
  8       JOIN
  9      (SELECT SK.name AS Skillname, count(*) projectnumber
 10       FROM skill SK JOIN project P ON (SK.code=P.code)
 11       GROUP by SK.name
 12       ORDER by SK.name) T2 ON (T1.skillname=T2.skillname)))
 13  UNION ALL
 14  (SELECT  'total number',trainingnumber,projectnumber
 15   FROM
 16    (
 17      (SELECT count(*) AS trainingnumber
 18       FROM skill S JOIN training T ON (S.code=T.code)) T1
 19       CROSS JOIN
 20      (SELECT count(*) projectnumber
 21       FROM skill SK JOIN project P ON (SK.code=P.code))
```

```
 22    )
 23  );
```

SKILLNAME	TRAINING_COUNT	PROJECT_COUNT
Marketing	4	1
Negotiation	3	2
PowerPoint	2	1
Python	8	3
Sql	7	2
Word	6	1
total number	30	10

7 rows selected.

Part II.B- #15

```
SQL> set pagesize 200
SQL> Column TABLE_NAME format a10
SQL> Column COLUMN_NAME format a20
SQL> Column CONSTRAINT_NAME format a30
SQL> Column CONSTRAINT_TYPE format a2
SQL> Column SEARCH_CONDITION format a39
SQL> Column  TABLE_FK_REFERENCES format a20
SQL> Column COLUMN_FK_REFERENCES format a20
SQL>
SQL> SELECT  u.table_name, u.column_name,
  2          NVL(u.constraint_name, '--------') AS "CONSTRAINT_NAME",
  3          NVL(u.constraint_type,'---')AS "CONSTRAINT_TYPE",
  4          v.search_condition AS "SEARCH_CONDITION",
  5          NVL(t.table_name,'--------') AS "TABLE_FK_REFERENCES",
  6          NVL(t.column_name,'--------') AS "COLUMN_FK_REFERENCES"
  7  FROM (
  8         SELECT a.table_name, a.column_name, column_id, c.constraint_name,
  9         substr(c.constraint_name, length(c.constraint_name)-1, length(c.constraint_name)) AS "CONSTRAINT_TYPE"
 10         FROM user_tab_columns a JOIN user_constraints t
 11                                 ON a.table_name= t.table_name
 12                                 LEFT OUTER JOIN user_cons_columns c
 13                                 ON a.column_name=c.column_name
 14         GROUP BY a.table_name, a.column_name, column_id, c.constraint_name
```

```
 15         ORDER BY a.table_name) u
 16  LEFT OUTER JOIN
 17         (SELECT constraint_name, search_condition
 18          FROM user_constraints
 19          WHERE constraint_type='C') v
 20          ON u.constraint_name=v.constraint_name
 21  LEFT OUTER JOIN
 22         (SELECT t.constraint_name, t.r_constraint_name, c.table_name, c.column_name
 23          FROM user_cons_columns c, user_constraints t
 24          WHERE t.constraint_type= 'R'
 25          AND c.constraint_name=t.r_constraint_name ) t
 26  ON u.constraint_name=t.constraint_name
 27  ORDER BY u.table_name, u.column_id;
```

TABLE_NAME	COLUMN_NAME	CONSTRAINT_NAME	CO	SEARCH_CONDITION	TABLE_FK_REFERENCES	COLUMN_FK_REFERENCES
ASSIGNMENT	ASSIGN_NUM	ASSIGN_NUM_PK	PK		--------	--------
ASSIGNMENT	PROJ_NUMBER	PROJECT_NUMBER_PK	PK		--------	--------
ASSIGNMENT	PROJ_NUMBER	ASSIGNMENT_PROJNUM_FK	FK		PROJECT	PROJ_NUMBER
ASSIGNMENT	EMP_NUM	ASSIGNMENT_EMP_NUM_FK	FK		EMPLOYEE	EMP_NUM
ASSIGNMENT	EMP_NUM	TRAINING_EMP_NUM_FK	FK		EMPLOYEE	EMP_NUM
ASSIGNMENT	EMP_NUM	EMPLOYEE_EMP_NUM_PK	PK		--------	--------
ASSIGNMENT	DATE_ASSIGNED	ASSIGNMENT_DATEASSIGNEND_CK	CK	Date_Assigned <= Date_Ended	--------	--------
ASSIGNMENT	DATE_ENDED	ASSIGNMENT_DATEASSIGNEND_CK	CK	Date_Assigned <= Date_Ended	--------	--------
ASSIGNMENT	HOURS_USED	--------	--		--------	--------
CLIENT	CLIENT_ID	PROJECT_CLIENT_ID_FK	FK		CLIENT	CLIENT_ID

CLIENT	CLIENT_ID	CLIENT_ID_PK	PK	--------	--------
CLIENT	NAME	--------	--	--------	--------
CLIENT	STREET	--------	--	--------	--------
CLIENT	CITY	--------	--	--------	--------
CLIENT	STATE	--------	--	--------	--------
CLIENT	ZIP_CODE	--------	--	--------	--------
CLIENT	INDUSTRY	--------	--	--------	--------
CLIENT	WEB_ADDRESS	--------	--	--------	--------
CLIENT	PHONE	--------	--	--------	--------
CLIENT	CONTACT_LNAME	--------	--	--------	--------
CLIENT	CONTACT_FNAME	--------	--	--------	--------
DEPARTMENT	DEPT_CODE	PROJECT_DEPT_DODE_FK	FK	DEPARTMENT	DEPT_CODE
DEPARTMENT	DEPT_CODE	EMPLOYEE_DEPT_CODE_FK	FK	DEPARTMENT	DEPT_CODE
DEPARTMENT	DEPT_CODE	DEPARTMENT_DEPT_CODE_PK	PK	--------	--------
DEPARTMENT	NAME	--------	--	--------	--------
DEPARTMENT	LOCATION	--------	--	--------	--------
DEPARTMENT	PHONE	--------	--	--------	--------
DEPARTMENT	MANAGER_ID	DEPARTMENT_MANAGER_ID_FK	FK	EMPLOYEE	EMP_NUM
EMPLOYEE	EMP_NUM	EMPLOYEE_EMP_NUM_PK	PK	--------	--------
EMPLOYEE	EMP_NUM	ASSIGNMENT_EMP_NUM_FK	FK	EMPLOYEE	EMP_NUM
EMPLOYEE	EMP_NUM	TRAINING_EMP_NUM_FK	FK	EMPLOYEE	EMP_NUM
EMPLOYEE	LNAME	--------	--	--------	--------
EMPLOYEE	FNAME	--------	--	--------	--------
EMPLOYEE	DOB	EMPLOYEE_DOB_HIRE_CK	CK MONTHS_BETWEEN(Hire_date,DOB)>18*12	--------	--------
EMPLOYEE	HIRE_DATE	EMPLOYEE_DOB_HIRE_CK	CK MONTHS_BETWEEN(Hire_date,DOB)>18*12	--------	--------
EMPLOYEE	SUPER_ID	EMPLOYEE_SUPER_ID_FK	FK	EMPLOYEE	EMP_NUM
EMPLOYEE	DEPT_CODE	PROJECT_DEPT_DODE_FK	FK	DEPARTMENT	DEPT_CODE
EMPLOYEE	DEPT_CODE	DEPARTMENT_DEPT_CODE_PK	PK	--------	--------

EMPLOYEE	DEPT_CODE	EMPLOYEE_DEPT_CODE_FK	FK	DEPARTMENT	DEPT_CODE
EMPLOYEE	BONUS_AMT	--------	--	--------	-------
PROJECT	PROJ_NUMBER	ASSIGNMENT_PROJNUM_FK	FK	PROJECT	PROJ_NUMBER
PROJECT	PROJ_NUMBER	PROJECT_NUMBER_PK	PK	--------	--------
PROJECT	NAME	--------	--	--------	--------
PROJECT	START_DATE	--------	--	--------	--------
PROJECT	TOTAL_COST	PROJECT_TOTAL_COST_CK	CK Total_Cost>=0	--------	--------
PROJECT	DEPT_CODE	PROJECT_DEPT_DODE_FK	FK	DEPARTMENT	DEPT_CODE
PROJECT	DEPT_CODE	DEPARTMENT_DEPT_CODE_PK	PK	--------	--------
PROJECT	DEPT_CODE	EMPLOYEE_DEPT_CODE_FK	FK	DEPARTMENT	DEPT_CODE
PROJECT	CLIENT_ID	PROJECT_CLIENT_ID_FK	FK	CLIENT	CLIENT_ID
PROJECT	CLIENT_ID	CLIENT_ID_PK	PK	--------	--------
PROJECT	CODE	TRAINING_CODE_FK	FK	SKILL	CODE
PROJECT	CODE	SKILL_CODE_PK	PK	--------	--------
PROJECT	CODE	PROJECT_CODE_FK	FK	SKILL	CODE
SKILL	CODE	SKILL_CODE_PK	PK	--------	--------
SKILL	CODE	TRAINING_CODE_FK	FK	SKILL	CODE
SKILL	CODE	PROJECT_CODE_FK	FK	SKILL	CODE
SKILL	NAME	--------	--	--------	--------
SKILL	CATEGORY	--------	--	--------	--------
TRAINING	TRAIN_NUM	TRAIN_NUM_PK	PK	--------	--------
TRAINING	CODE	SKILL_CODE_PK	PK	--------	--------
TRAINING	CODE	PROJECT_CODE_FK	FK	SKILL	CODE
TRAINING	CODE	TRAINING_CODE_FK	FK	SKILL	CODE
TRAINING	EMP_NUM	ASSIGNMENT_EMP_NUM_FK	FK	EMPLOYEE	EMP_NUM
TRAINING	EMP_NUM	EMPLOYEE_EMP_NUM_PK	PK	--------	--------
TRAINING	EMP_NUM	TRAINING_EMP_NUM_FK	FK	EMPLOYEE	EMP_NUM
TRAINING	NAME	--------	--	--------	--------

TRAINING	DATE_ACQUIRED	--------	--	--------	--------
TRAINING	COMMENTS	TRAINING_COMMENTS_CK	CK comments IN('Pass','Good','Excellent')	--------	--------

68 rows selected.