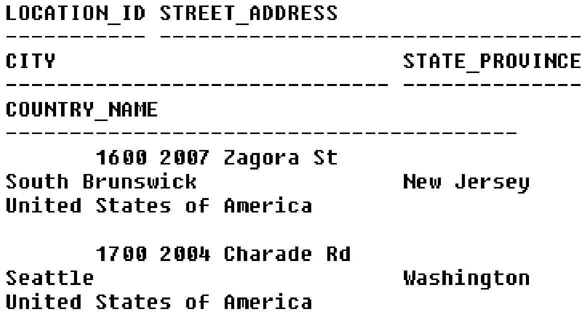
PRACTISE 5

1. Menampilkan query dari 2 buah table menggunakan operator natural join. Hal ini dimungkinkan karena ke-dua table memiliki kunci yang sama, yaitu location_id. Dimana berperan sebagai primary key dan foreign key.

```
SELECT location_id, street_address, city, state_province,
country_name FROM locations NATURAL JOIN countries;
```

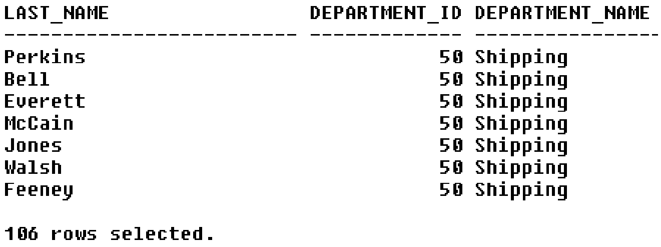


```
LOCATION_ID STREET_ADDRESS
----------- ----------------------------------
CITY                           STATE_PROVINCE
------------------------------ --------------
COUNTRY_NAME
----------------------------------------
       1600 2007 Zagora St
South Brunswick                New Jersey
United States of America

       1700 2004 Charade Rd
Seattle                        Washington
United States of America
```

2. Menampilkan last_nama, department_id dan department_name dari table employees dan departments menggunakan join (using (key))/natural join.

```
SELECT last_name, department_id, department_name
FROM employees JOIN departments USING (department_id);
```

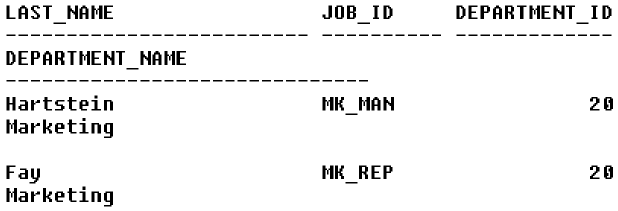


```
LAST_NAME                 DEPARTMENT_ID DEPARTMENT_NAME
------------------------- ------------- ----------------
Perkins                              50 Shipping
Bell                                 50 Shipping
Everett                              50 Shipping
McCain                               50 Shipping
Jones                                50 Shipping
Walsh                                50 Shipping
Feeney                               50 Shipping

106 rows selected.
```

3. Menampilkan last_name, job_id, department_id dan department_name untuk semua employees yang bekerja di Toronto.

```
SELECT e.last_name, e.job_id, e.department_id, d.department_name
FROM employees e JOIN departments d
ON (e.department_id = d.department_id) JOIN locations l
ON (d.location_id = l.location_id)
WHERE LOWER(l.city) = 'Toronto';
```

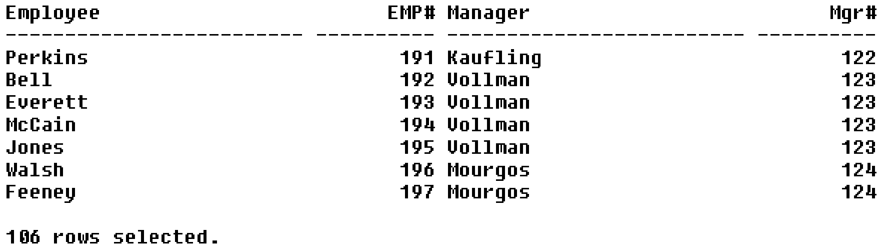


```
LAST_NAME                 JOB_ID     DEPARTMENT_ID
------------------------- ---------- -------------
DEPARTMENT_NAME
------------------------------
Hartstein                 MK_MAN                20
Marketing

Fay                       MK_REP                20
Marketing
```

4. Menampilkan last_name sebagai Employee, employee_id sebagai EMP#, dan last_name manager sebagai Manager, manager_id sebagai Mgr# dari tiap manager mereka. Hal ini berarti kita menggunakan self join, karena data-data pegawai menyangkut manager dan pegawai biasa yang diberi nama table Employees.

```
SELECT e.last_name "Employee", e.employee_id "EMP#", m.last_name
"Manager", m.employee_id "Mgr#" FROM employees e join employees m
ON (e.manager_id = m.employee_id);
```

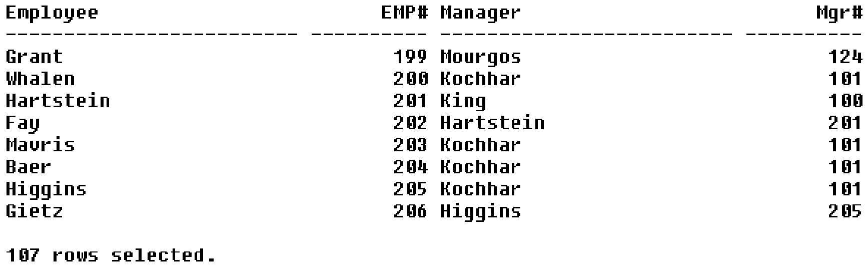


```
Employee                        EMP# Manager                         Mgr#
------------------------- ---------- ------------------------- ----------
Perkins                          191 Kaufling                         122
Bell                             192 Vollman                          123
Everett                          193 Vollman                          123
McCain                           194 Vollman                          123
Jones                            195 Vollman                          123
Walsh                            196 Mourgos                          124
Feeney                           197 Mourgos                          124

106 rows selected.
```

5. Menampilkan semua employees termasuk employees yang tidak memiliki manager. Pengerjaannya memakai left outer join, yaitu tampilkan semua value dari queqy di sebelah kiri.

```
SELECT e.last_name "Employee", e.employee_id "EMP#", m.last_name
"Manager", m.employee_id "Mgr#" FROM employees e
LEFT OUTER JOIN employees m ON (e.manager_id = m.employee_id)
Order By e.employee_id;
```

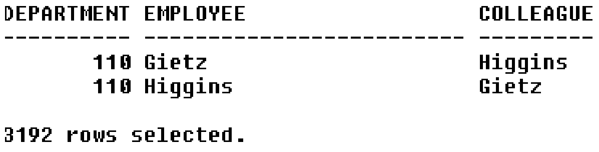


```
Employee                        EMP# Manager                         Mgr#
------------------------- ---------- ------------------------- ----------
Grant                            199 Mourgos                          124
Whalen                           200 Kochhar                          101
Hartstein                        201 King                             100
Fay                              202 Hartstein                        201
Mavris                           203 Kochhar                          101
Baer                             204 Kochhar                          101
Higgins                          205 Kochhar                          101
Gietz                            206 Higgins                          205

107 rows selected.
```

6. Menampilkan last_name, department_id dan employees yang bekerja satu department.

```
SELECT e.department_id department, e.last_name employee,
c.last_name colleague FROM employees e JOIN employees c
ON (e.department_id = c.department_id)
WHERE e.employee_id <> c.employee_id
ORDER BY e.department_id, e.last_name, c.last_name;
```



```
DEPARTMENT EMPLOYEE                  COLLEAGUE
---------- ------------------------- ---------
       110 Gietz                     Higgins
       110 Higgins                   Gietz

3192 rows selected.
```

7. Mendeskripsikan struktur table job_grades

```
DESC JOB_GRADES;
```

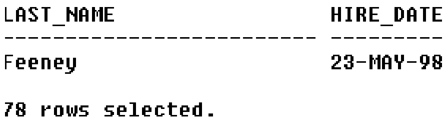
```
Name         Null?      Type
------------ ---------- ------------
GRADE_LEVEL             VARCHAR2(3)
LOWEST_SAL              NUMBER
HIGHEST_SAL             NUMBER
```

Menampilkan last_name, job_id, department_name, salary, serta gradenya.

```
SELECT e.last_name, e.job_id, d.department_name, e.salary,
j.grade_level FROM employees e JOIN departments d
ON (e.department_id = d.department_id) JOIN job_grades j
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```

8. Menampilkan last_name, hire_date dari employees yang bekerja setelah davies.

```
SELECT e.last_name, e.hire_date
FROM employees e JOIN employees davies
ON (davies.last_name = 'Davies')
WHERE davies.hire_date < e.hire_date;
```

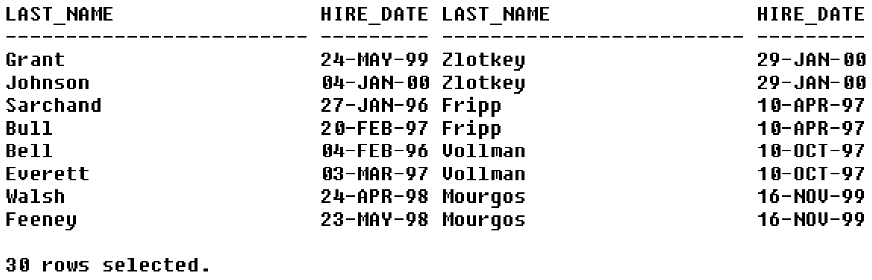


```
LAST_NAME                 HIRE_DATE
------------------------- ---------
Feeney                    23-MAY-98

78 rows selected.
```

9. Menampilkan last_name, hire_date dari employees dan manager dimana employees lebih dulu bekerja dibandingkan dengan manager mereka.

```
SELECT w.last_name, w.hire_date, m.last_name, m.hire_date
FROM employees w JOIN employees m ON (w.manager_id =
m.employee_id)
WHERE w.hire_date < m.hire_date;
```



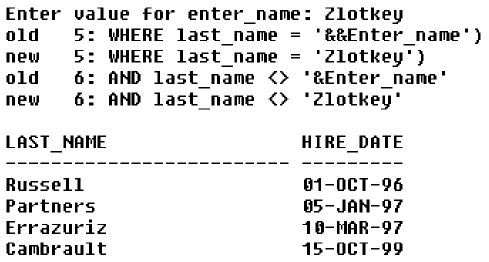
```
LAST_NAME                 HIRE_DATE LAST_NAME                 HIRE_DATE
------------------------- --------- ------------------------- ---------
Grant                     24-MAY-99 Zlotkey                   29-JAN-00
Johnson                   04-JAN-00 Zlotkey                   29-JAN-00
Sarchand                  27-JAN-96 Fripp                     10-APR-97
Bull                      20-FEB-97 Fripp                     10-APR-97
Bell                      04-FEB-96 Vollman                   10-OCT-97
Everett                   03-MAR-97 Vollman                   10-OCT-97
Walsh                     24-APR-98 Mourgos                   16-NOV-99
Feeney                    23-MAY-98 Mourgos                   16-NOV-99

30 rows selected.
```

PRACTISE 6

1. Menampilkan last_name dan hire_date dari employees, dimana last_name di input melalui prompt.

```
UNDEFINE Enter_name;
SELECT last_name, hire_date FROM employees
WHERE department_id = (SELECT department_id FROM employees
                       WHERE last_name = '&&Enter_name')
AND last_name <> '&Enter_name';
```

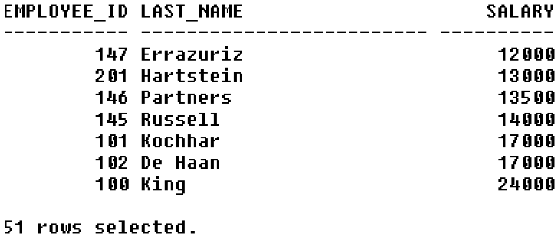


```
Enter value for enter_name: Zlotkey
old   5: WHERE last_name = '&&Enter_name')
new   5: WHERE last_name = 'Zlotkey')
old   6: AND last_name <> '&Enter_name'
new   6: AND last_name <> 'Zlotkey'

LAST_NAME                 HIRE_DATE
------------------------- ---------
Russell                   01-OCT-96
Partners                  05-JAN-97
Errazuriz                 10-MAR-97
Cambrault                 15-OCT-99
```

2. Menampilkan employee_id, last_name, salary dari employees yang salary nya > dari rata-rata salary. Urutkan berdasarkan salary secara ascending

```
SELECT employee_id, last_name, salary FROM employees
WHERE salary > (SELECT AVG(salary)FROM employees)
ORDER BY salary asc;
```

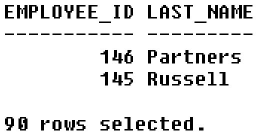


```
EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        147 Errazuriz                      12000
        201 Hartstein                      13000
        146 Partners                       13500
        145 Russell                        14000
        101 Kochhar                        17000
        102 De Haan                        17000
        100 King                           24000

51 rows selected.
```

3. Menampilkan employee_id dan last_name di department_id yang kebanyakan employees last_name mengandung huruf a atua u.

```
SELECT employee_id, last_name FROM employees
WHERE department_id IN (SELECT department_id FROM employees
                        WHERE last_name like '%u%');
```

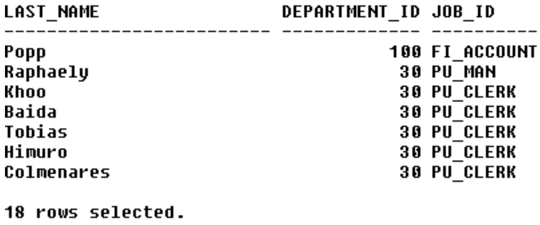


```
EMPLOYEE_ID LAST_NAME
----------- ---------
        146 Partners
        145 Russell

90 rows selected.
```

4. Menampilkan last_name, department_id, job_id dari employees di department_id yang location_id=1700.

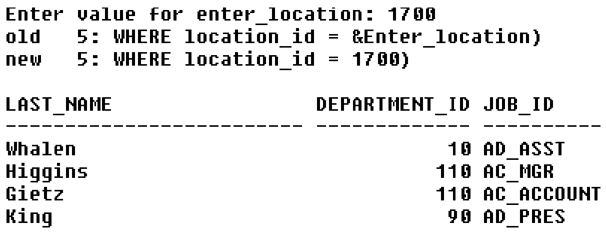
```
SELECT last_name, department_id, job_id FROM employees
WHERE department_id IN (SELECT department_id FROM departments
                        WHERE location_id = 1700);
```



```
LAST_NAME                 DEPARTMENT_ID JOB_ID
------------------------- ------------- ----------
Popp                                100 FI_ACCOUNT
Raphaely                             30 PU_MAN
Khoo                                 30 PU_CLERK
Baida                                30 PU_CLERK
Tobias                               30 PU_CLERK
Himuro                               30 PU_CLERK
Colmenares                           30 PU_CLERK

18 rows selected.
```

```
SELECT last_name, department_id, job_id FROM employees
WHERE department_id IN (SELECT department_id FROM departments
                        WHERE location_id = &Enter_location);
```

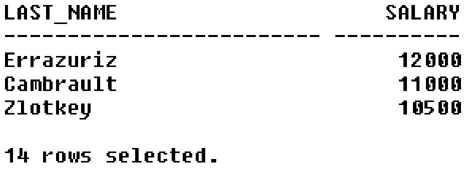


```
Enter value for enter_location: 1700
old   5: WHERE location_id = &Enter_location)
new   5: WHERE location_id = 1700)

LAST_NAME                DEPARTMENT_ID JOB_ID
------------------------ ------------- ----------
Whalen                              10 AD_ASST
Higgins                            110 AC_MGR
Gietz                              110 AC_ACCOUNT
King                                90 AD_PRES
```

5. Menampilkan last_name dan salary dari employees yang managernya King.

```
SELECT last_name, salary FROM employees
WHERE manager_id in (SELECT employee_id FROM employees
                    WHERE last_name = 'King');
```

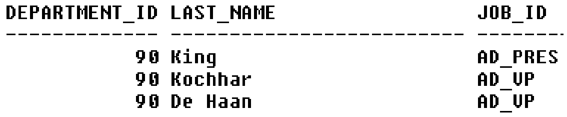


```
LAST_NAME                    SALARY
------------------------- ----------
Errazuriz                      12000
Cambrault                      11000
Zlotkey                        10500

14 rows selected.
```

6. Menampilkan department_id, last_name, job_id dari employees di department yang department_name = Executive.

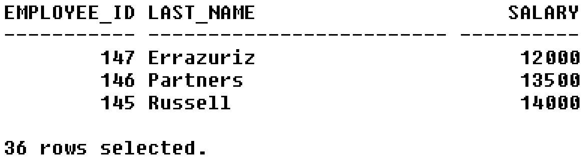
```
SELECT department_id, last_name, job_id FROM employees
WHERE department_id IN (SELECT department_id FROM departments
                        WHERE department_name = 'Executive');
```



```
DEPARTMENT_ID LAST_NAME                JOB_ID
------------- ------------------------ --------
           90 King                     AD_PRES
           90 Kochhar                  AD_VP
           90 De Haan                  AD_VP
```

7. Menampilkan employee_id, last_name, salary dari employee di department yang memiliki huruf u di last_name-nya.

```
SELECT employee_id, last_name, salary FROM employees
WHERE department_id IN (SELECT department_id FROM employees
                        WHERE last_name like '%u%')
AND salary > (SELECT AVG(salary) FROM employees);
```



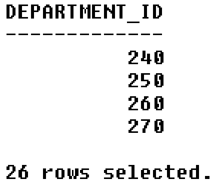
```
EMPLOYEE_ID LAST_NAME                    SALARY
----------- ------------------------- ----------
        147 Errazuriz                      12000
        146 Partners                       13500
        145 Russell                        14000

36 rows selected.
```

PRACTISE 7

1. Menampilkan semua department_id, tanpa mengikutkan department_id yang memiliki job_id=”ST_CLERK”.

```
SELECT department_id FROM departments
MINUS
SELECT department_id FROM employees WHERE job_id = 'ST_CLERK';
```

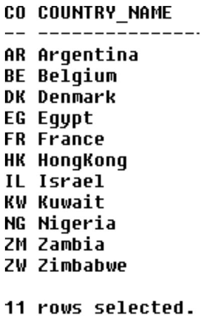


```
DEPARTMENT_ID
-------------
          240
          250
          260
          270

26 rows selected.
```

2. Menampilkan country_id, country_name dari countries yang tidak mempunyai department perwakilan di Negara tersebut.

```
SELECT country_id, country_name FROM countries
MINUS
SELECT country_id, country_name FROM countries
NATURAL JOIN locations NATURAL JOIN departments;
```

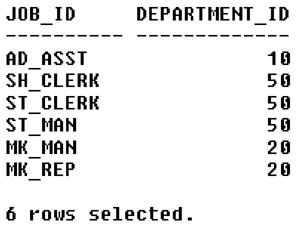


```
CO COUNTRY_NAME
-- ---------------
AR Argentina
BE Belgium
DK Denmark
EG Egypt
FR France
HK HongKong
IL Israel
KW Kuwait
NG Nigeria
ZM Zambia
ZW Zimbabwe

11 rows selected.
```

3. Menampilkan job_id dan department_id yang ditentukan urutannya.

```
COLUMN dummy NOPRINT;
SELECT job_id, department_id, 'x' dummy FROM employees
WHERE department_id = 10 UNION
SELECT job_id, department_id, 'y' dummy FROM employees
WHERE department_id = 50 UNION
SELECT job_id, department_id, 'z' dummy FROM employees
WHERE department_id = 20 ORDER BY dummy;
```

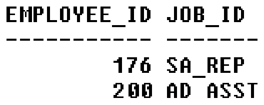


```
JOB_ID     DEPARTMENT_ID
---------- -------------
AD_ASST               10
SH_CLERK              50
ST_CLERK              50
ST_MAN                50
MK_MAN                20
MK_REP                20

6 rows selected.
```

4. Menampilkan employee_id, job_id dari employees yang memiliki job_id yang sama ketika mereka kembali bekerja setelah sebelumnya pernah keluar atau berhenti.

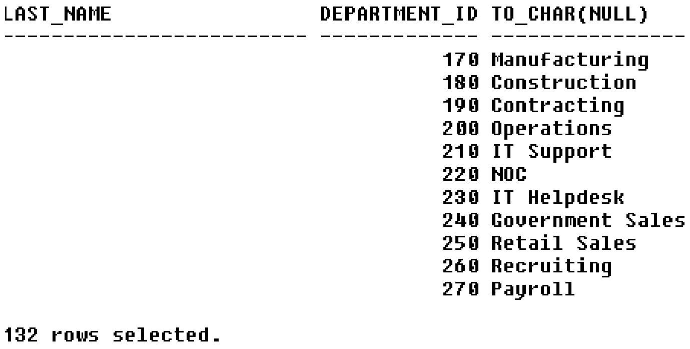
```
SELECT employee_id,job_id FROM employees
INTERSECT
SELECT employee_id,job_id FROM job_history;
```



```
EMPLOYEE_ID JOB_ID
----------- -------
        176 SA_REP
        200 AD ASST
```

5. Menampilkan last_name, department_id dari employees walaupun tidak memilki departments dan department_id dan department_name dari departments walaupun tidak memiliki employees.

```
SELECT last_name,department_id,TO_CHAR(null) FROM employees
UNION
SELECT TO_CHAR(null),department_id,department_name
FROM departments;
```



```
LAST_NAME                 DEPARTMENT_ID TO_CHAR(NULL)
------------------------- ------------- ----------------
                                    170 Manufacturing
                                    180 Construction
                                    190 Contracting
                                    200 Operations
                                    210 IT Support
                                    220 NOC
                                    230 IT Helpdesk
                                    240 Government Sales
                                    250 Retail Sales
                                    260 Recruiting
                                    270 Payroll

132 rows selected.
```