SQL QUERIES

Retrieve the name of all employees who earn more than the Average Salary

```
SELECT      FNAME, LNAME
FROM        EMPLOYEE
WHERE       Salary >  AVG(SALARY)
```

✗

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

NESTED QUERIES

A complete SELECT query, called a ***nested query***, can be specified within the WHERE-clause of another query, called the *outer query*

Retrieve the name of all employees who earn more than the Average Salary

```
SELECT       FNAME, LNAME
FROM         EMPLOYEE
WHERE        Salary >  (SELECT  AVG(SALARY)
                        FROM       EMPLOYEE)
```

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

EMPLOYEE

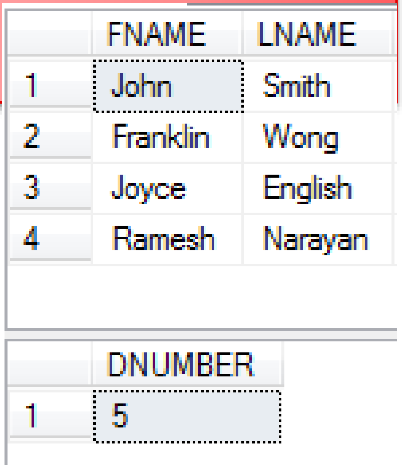
FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

NESTING OF QUERIES

A complete SELECT query, called a ***nested query*** , can be specified within the WHERE-clause of another query

Retrieve the name of all employees who work for the 'Research' department.

```
SELECT      FNAME, LNAME
FROM        EMPLOYEE
WHERE       DNO =  (SELECT  DNUMBER
                     FROM    DEPARTMENT
                     WHERE DNAME='Research' )
```



	FNAME	LNAME
1	John	Smith
2	Franklin	Wong
3	Joyce	English
4	Ramesh	Narayan

	DNUMBER
1	5

If `=` is used the inner query must return one value

If more than one value is returned then an error msg is generated

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

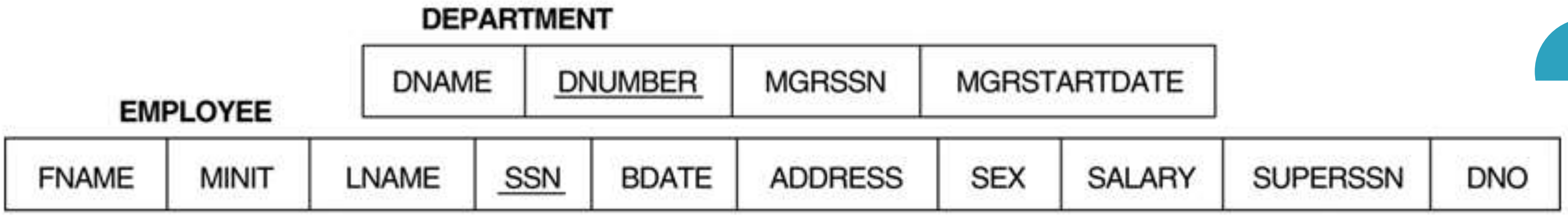
NESTING OF QUERIES

Retrieve the name of all employees who work for the 'Research’ or Administration ‘department .

```
SELECT        FNAME, LNAME
FROM          EMPLOYEE
WHERE         DNO IN  (SELECT  DNUMBER
                       FROM DEPARTMENT
                       WHERE DNAME='Research'  OR
                                    DNAME=‘Administration’ )
```

If `=` is used the inner query must return one value.

If inner query returns more than one value then **use IN**



DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

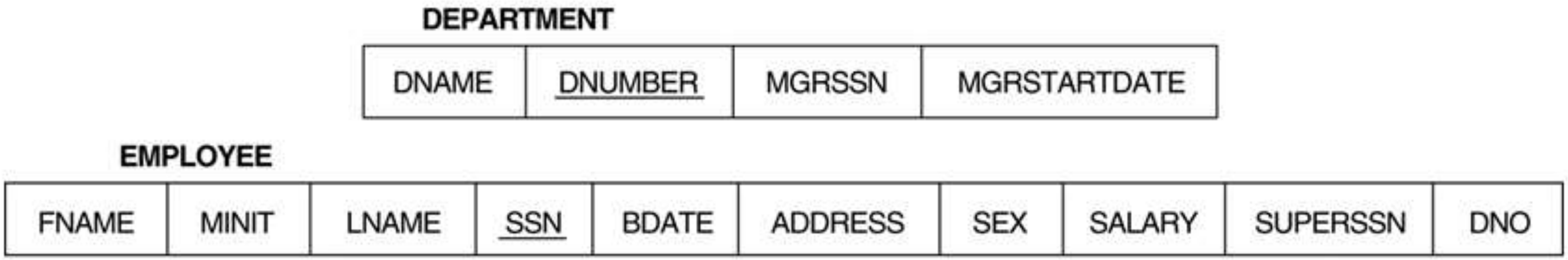
EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

NESTING OF QUERIES

Retrieve the name of all employees who do not work for the 'Research' department.

```
SELECT     FNAME, LNAME
FROM       EMPLOYEE
WHERE      DNO NOT IN  (SELECT  DNUMBER
                          FROM DEPARTMENT
                          WHERE  DNAME='Research' )
```



DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

EMPLOYEE

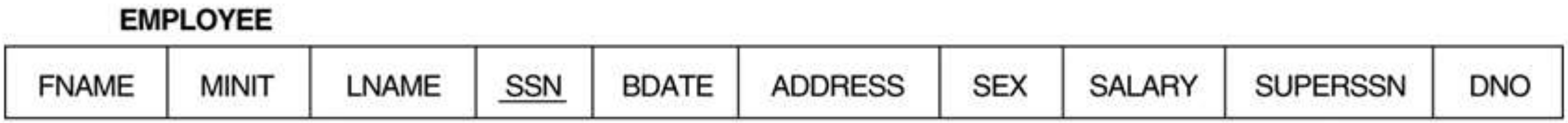
FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

NESTED QUERIES

You can also use: s > ALL R (means greater than every value)
s > ANY R (means greater than any value)

= ANY is same as IN , <>ALL is same as NOT IN

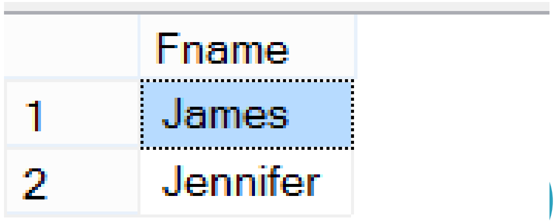
Find name of employees whose salary is greater than the salary of all employees in department 5



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

```
SELECT  Fname
FROM    Employee
WHERE  Salary >  ALL (SELECT Salary
                FROM    Employee
                where Dno=5)
```



	Fname
1	James
2	Jennifer

NESTED QUERIES & TSQL

A subquery can be nested inside the **WHERE or HAVING** clause of an outer SELECT, INSERT, UPDATE, or DELETE statement.

Statements with subquery usually take one of these formats:

- WHERE expression [NOT] IN (subquery)
- WHERE expression **comparison_operator** [ANY | ALL] (subquery)
- WHERE [NOT] EXISTS (subquery)

comparison_operator { = | <> | != | > | >= | !> | < | <= | !< }

- Up to **32 levels** of nesting is possible,
  - This limit depends on available memory and the complexity of other expressions in the query.

Why NESTED QUERIES ?

- Many Transact-SQL statements that include subqueries can be alternatively formulated as joins.

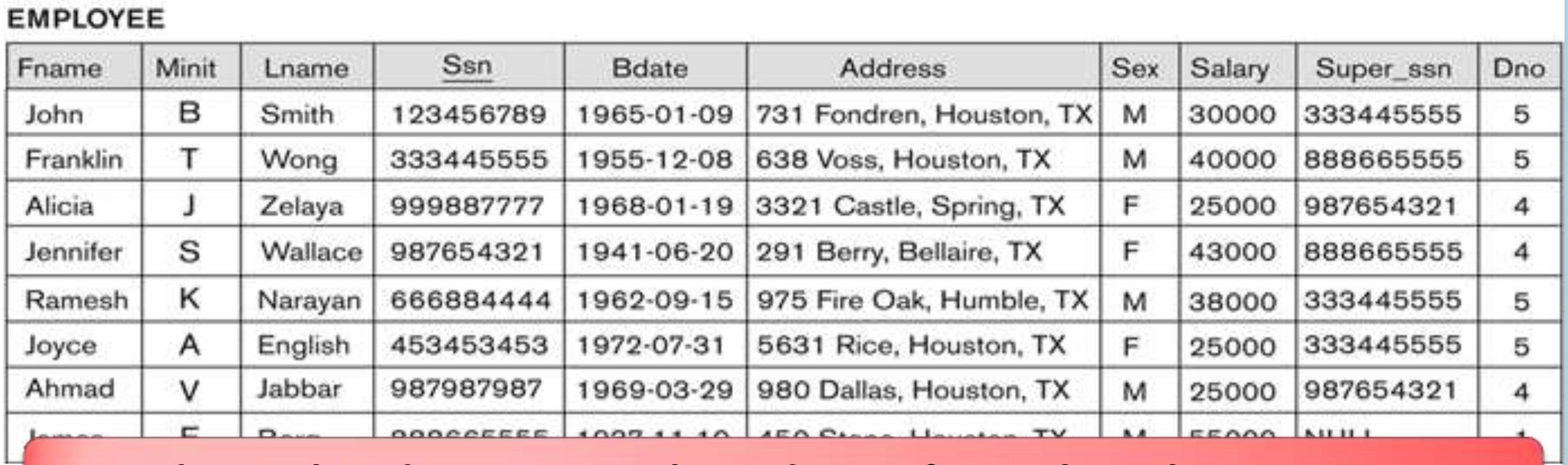
- **Other questions can be posed only with subqueries.**

- An aggregate may not appear in the **WHERE clause**
  - unless it is in a subquery contained in a HAVING clause or a select list, and the column being aggregated is an outer reference

CORRELATED NESTED QUERIES

- If a condition in the ***nested query* references** an attribute of a relation **declared in the *outer query*** **=>**
  - **Then two queries are said to be correlated**

Retrieve the name of each employee who has a dependent with the same first name as the employee.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Nested Correlated query is evaluated once for each tuple in outer query

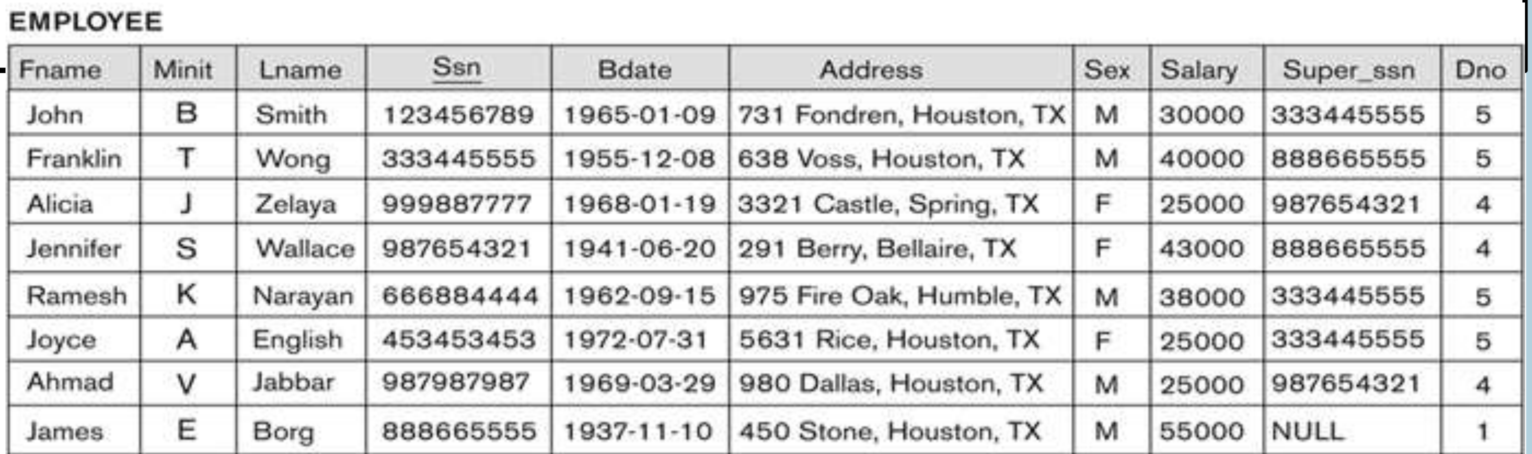
CORRELATED NESTED QUERIES

A **correlated subquery** (*also called repeating subquery*) depends on the outer query for its values.

- This means that the nested subquery is executed **repeatedly**, once for each row that might be selected by the *outer query*.

Retrieve the name of each employee who has a dependent with the same first name as the employee.

```
SELECT  E.FNAME, E.LNAME
FROM    EMPLOYEE AS E
WHERE  E.SSN IN (SELECT ESSN       FROM DEPENDENT
                        WHERE SSN = ESSN AND FNAME=DEPENDENT_NAME)
```



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

NESTED QUERIES

- A query written with nested SELECT... FROM... WHERE... blocks and using the = or IN comparison operators can ***always*** be expressed as a single block query.

Retrieve the name of each employee who has a dependent with the same first name as the employee.

```
SELECT    E.FNAME, E.LNAME
FROM      EMPLOYEE E, DEPENDENT D
WHERE     E.SSN=D.ESSN AND
          E.FNAME=D.DEPENDENT_NAME
```

EXISTS FUNCTION

EXISTS Function checks whether the result of a nested query is empty or not

- Retrieve the name of each employee who has a dependent with the same first name as the employee.

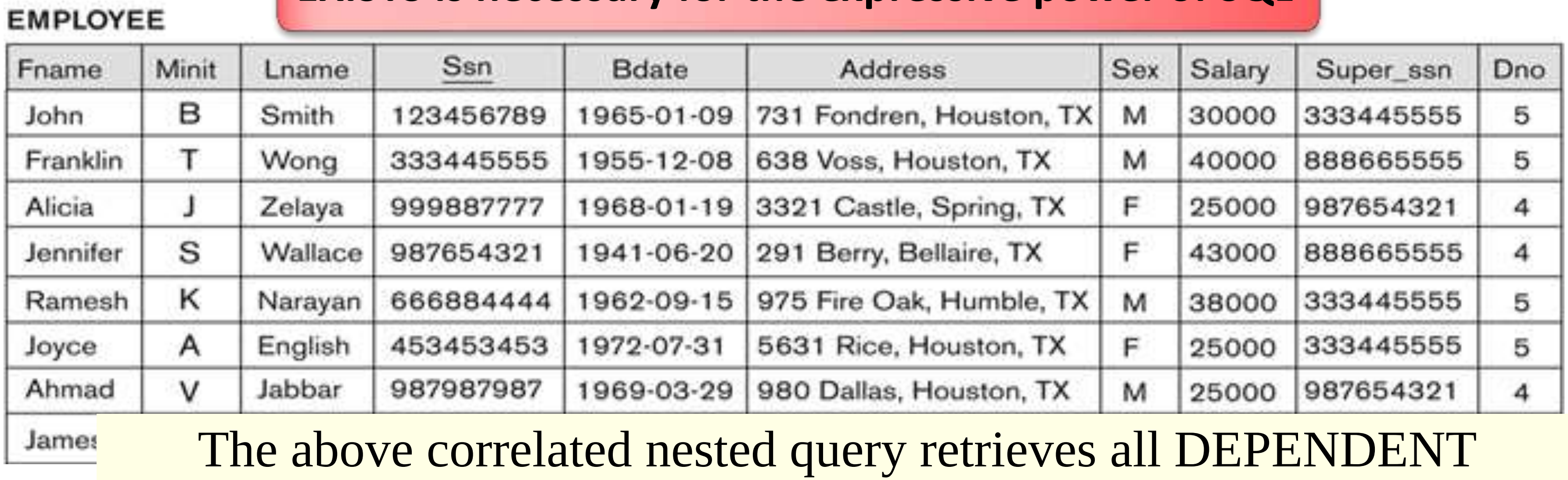
```
SELECT      FNAME, LNAME
FROM        EMPLOYEE
WHERE       EXISTS  (SELECT      *
                     FROM        DEPENDENT
                     WHERE        SSN=ESSN AND
                                  FNAME=DEPENDENT_NAME)
```

EXISTS FUNCTION

Retrieve the names of employees who have no dependents.

```
SELECT      FNAME, LNAME
FROM        EMPLOYEE
WHERE       NOT EXISTS   (SELECT  *
                          FROM  DEPENDENT
                           WHERE SSN=ESSN)
```

EXISTS is necessary for the expressive power of SQL



EMPLOYEE

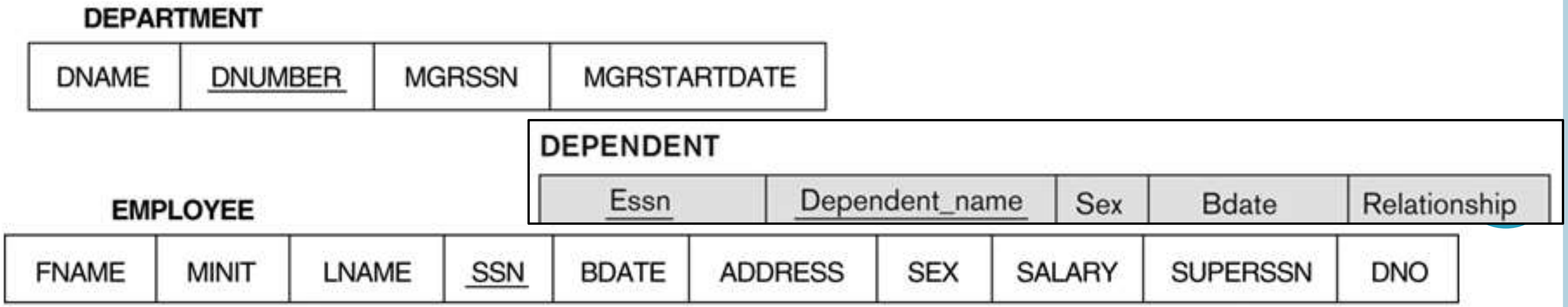
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James									

The above correlated nested query retrieves all DEPENDENT tuples related to an EMPLOYEE tuple.
If *none exist* , the EMPLOYEE tuple is selected

EXISTS FUNCTION

Find the names of managers who have at least one dependents.

```
SELECT   FNAME, LNAME
FROM     EMPLOYEE
WHERE  EXISTS   (SELECT   *
                     FROM  DEPARTMENT
                     WHERE SSN=Mgr_SSN)
         AND
        EXISTS   (SELECT  *
                     FROM  DEPENDENT
                     WHERE SSN=ESSN)
```



DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

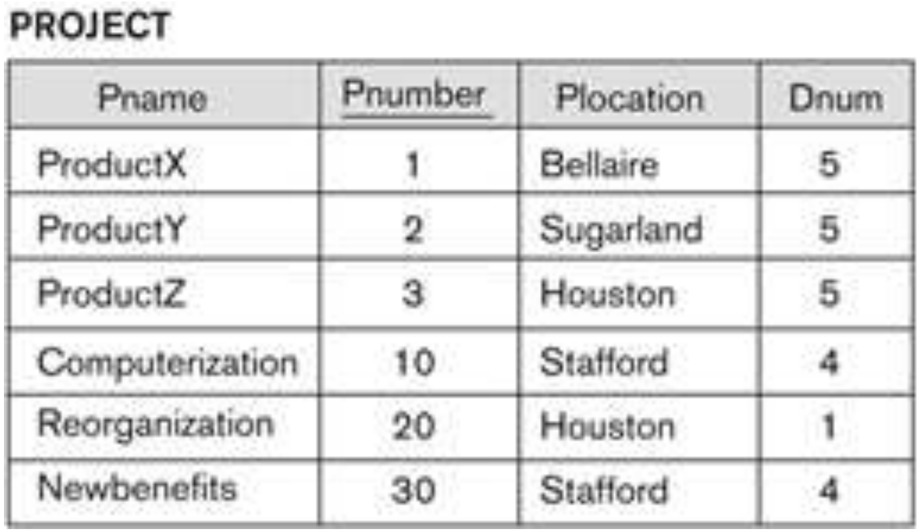
EXISTS FUNCTION

Retrieve the name of each employee who works on *all* the projects controlled by department number 4.

Set theory: S1 contains S2 if (S2 – S1 = 0)

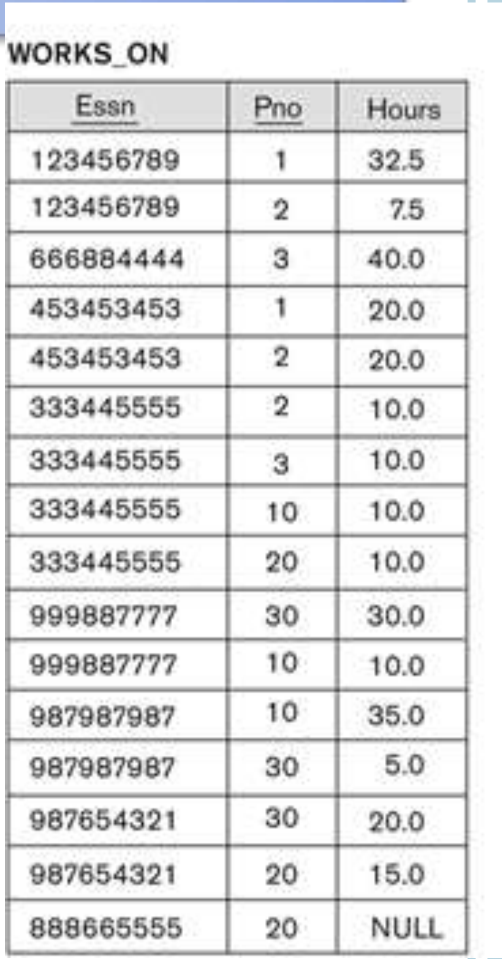
S1 = set of projects of each employee

S2 = set of Dept 4 projects



PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4



WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

EXISTS FUNCTION

Retrieve the name of each employee who works on *all* the projects controlled by department number 4.

Set theory: S1 contains S2 if (S2 – S1 = 0)

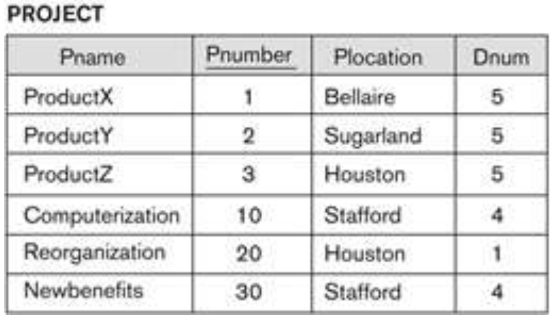
```
SELECT   FNAME, LNAME
FROM     EMPLOYEE
WHERE    NOT EXISTS (
         (SELECT  PNUMBER
          FROM PROJECT
          WHERE     DNUM=4)

         EXCEPT

        (SELECT      PNO
         FROM        WORKS_ON
         WHERE       SSN=ESSN)
)
```

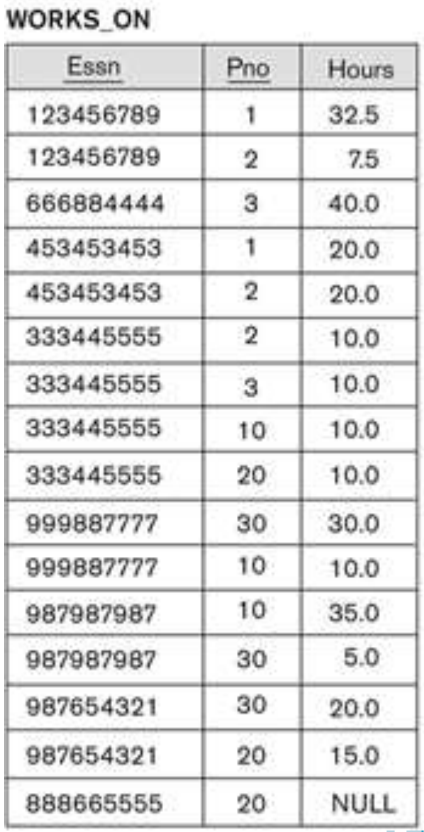
S1 = set of projects of each employee

S2 = set of dept 4 projects



PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4



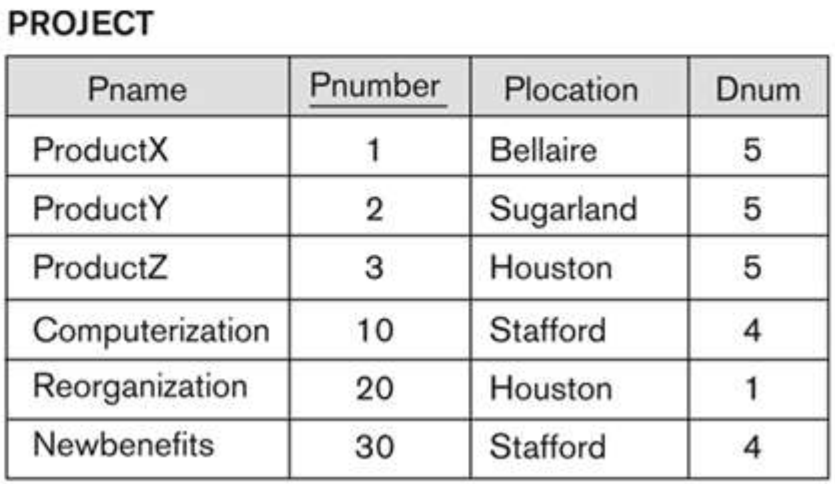
WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

HOW to do this in Relational Algebra?

Yet another Example

Find SSN of employees who work on all the projects of Dnum= 4

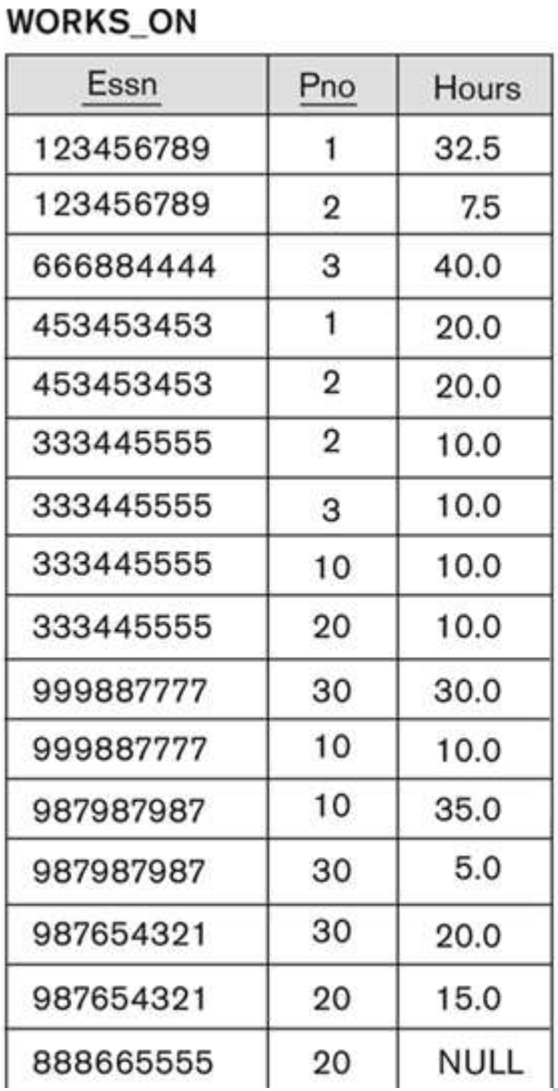


PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

- PD4(Pno) ← π_{Pnumber} ($\sigma_{\text{DNUM=4}}$ Project)
- Ssn_Pnos ← $\pi_{\text{Essn,Pno}}$ (Works_on)
- SSNS(ssn)← Ssn_Pnos ??? PD4

DIVISION

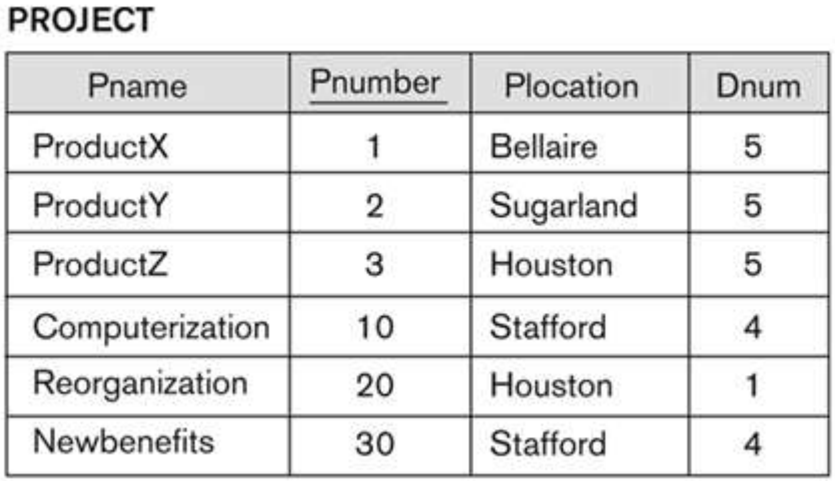


WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

Yet an other Example

Find SSN of employees who work on all the projects of Dnum= 4

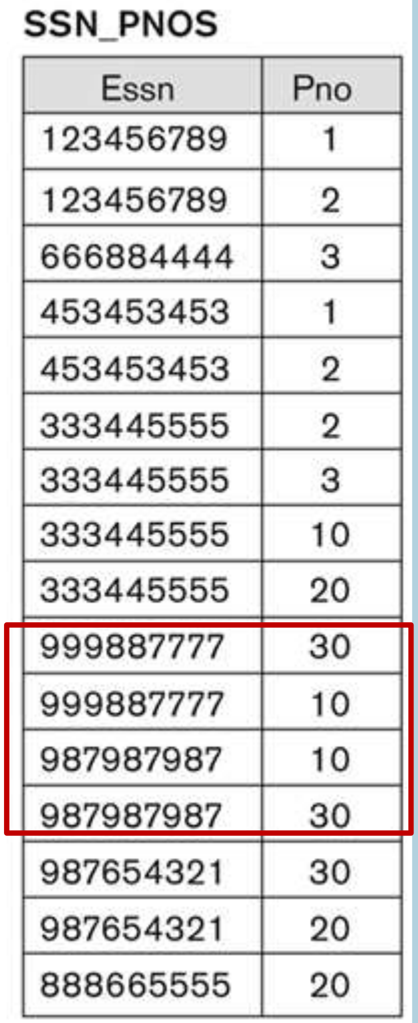


PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

PD4

Pno
10
30



SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

- PD4(Pno) $\leftarrow \pi_{\text{Pnumber}} (\sigma_{\text{DNUM=4}}$ Project)
- Ssn_Pnos $\leftarrow \pi_{\text{Essn,Pno}}$ (Works_on)
- **SSNS(ssn)← Ssn_Pnos ÷ PD4**

DIVISION

DIVISION (Binary Operation)

Division operation is applied to two relations R1 and R2

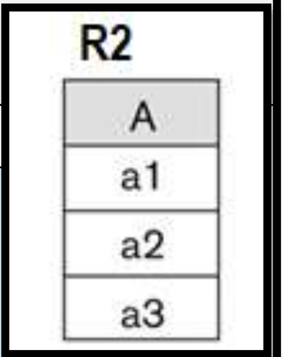
R1(Attributes_R1) ÷ R2(Attributes_R2)

- where Attributes_R2 ⊂ Attributes_R1.

Let **Result = R1 ÷ R2**

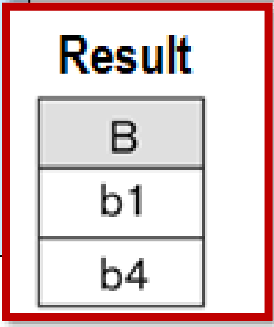
Attr_Res = Attributes_R1 - Attributes_R2

- Attr_Res is a set of attributes of R1 that are not the attributes of R2.



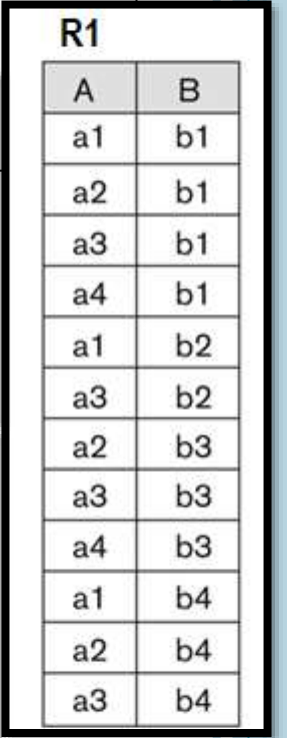
R2

A
a1
a2
a3



Result

B
b1
b4



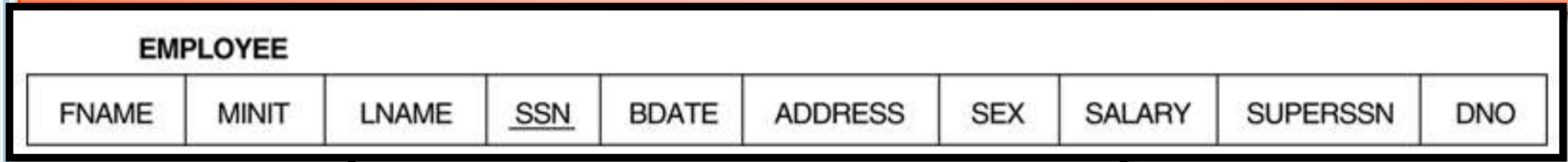
R1

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

For a **tuple t** to appear in the result of the DIVISION, the values in t must appear in R1 in combination with *every* tuple in R2.

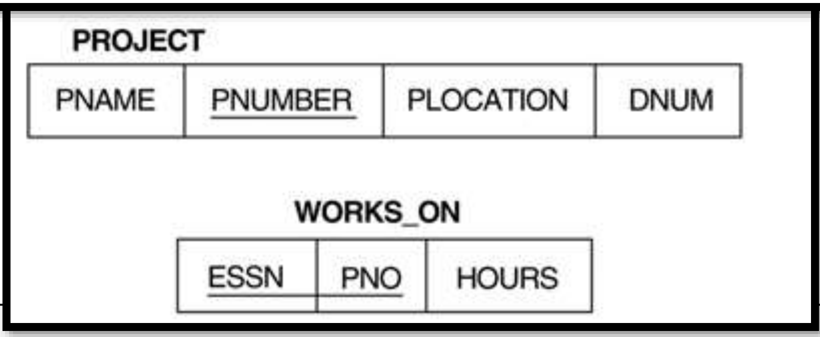
Example of DIVISION

Find **SSN** of employees who work on all the projects that *John Smith* works on



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO



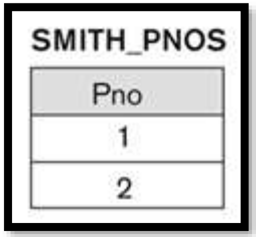
PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

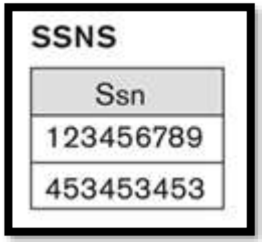
ESSN	PNO	HOURS

- Smith $\leftarrow \sigma_{\text{fname='John' and lname='Smith'}}$ (Employee)
- Smith_Pnos $\leftarrow \pi_{\text{Pno}}$ (Works_on $\bowtie_{\text{essn=ssn}}$ Smith)
- Ssn_Pnos $\leftarrow \pi_{\text{Essn,Pno}}$ (Works_on)
- SSNS(ssn)$\leftarrow$ Ssn_Pnos $\div$ Smith_Pnos



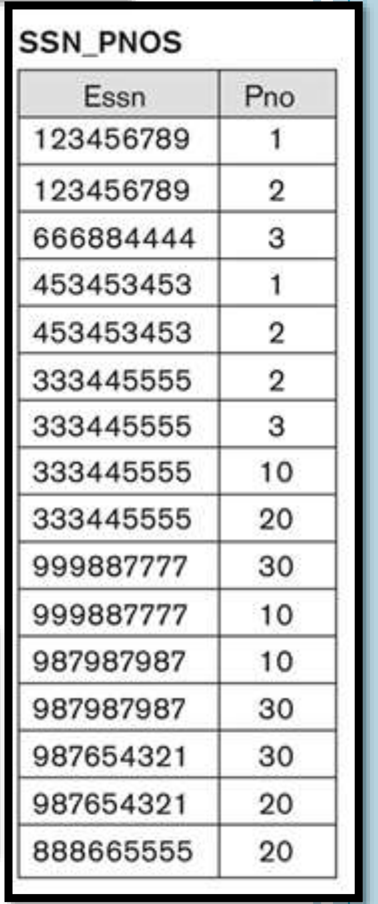
SMITH_PNOS

Pno
1
2



SSNS

Ssn
123456789
453453453



SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

Examples of Queries in RA

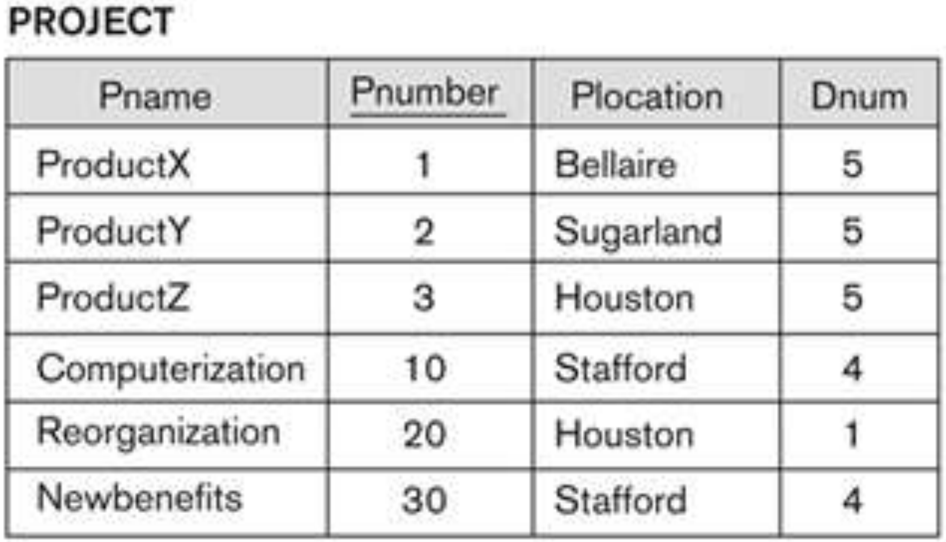
Find the names of employees who work on *all* the projects controlled by department number 5.

$$T1(Pno) \leftarrow \pi_{\text{Pnumber}} (\sigma_{\text{Dnum}= 5} (\text{Project}))$$

$$T2 \leftarrow \pi_{\text{Essn, Pno}} (\text{Work\_On})$$

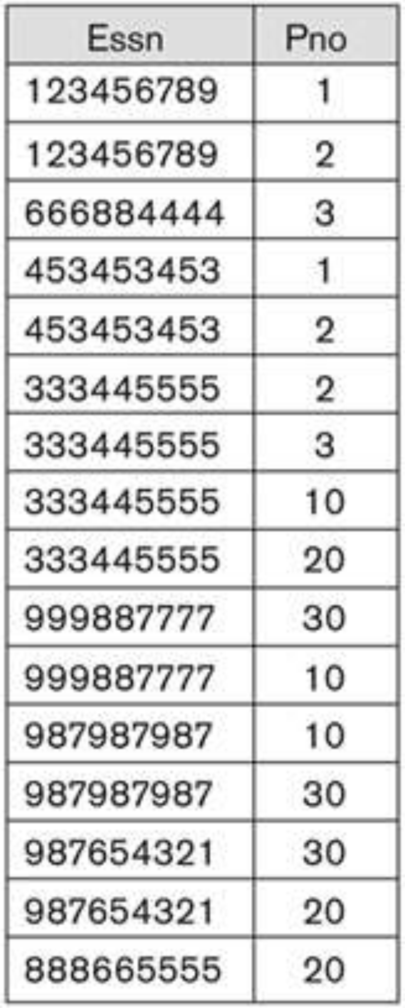
$$T3 \leftarrow (T2 \div T1)$$

$$R \leftarrow \pi_{\text{LNAME, FNAME}} (T3 * \text{Employee})$$



PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4



Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20