+Homework 4B

8.15. Show the result of each of the sample queries in Section 8.5 as it would apply to the database state in Figure 5.6.

Ans= Query 1)=Retrieve the name and address of all employees who work for the ‘Research’ department.

RESEARCH\_DEPT ← σDname=‘Research’(DEPARTMENT)

RESEARCH\_EMPS ← (RESEARCH\_DEPT Dnumber=DnoEMPLOYEE)

RESULT ← πFname, Lname, Address(RESEARCH\_EMPS)

As a single in-line expression, this query becomes:

πFname, Lname, Address (σDname=‘Research’(DEPARTMENT Dnumber=Dno(EMPLOYEE))

output :

|  |  |  |
| --- | --- | --- |
| Fname | Lname | Address |
| John | Smith | 731 Fondren, Houston, TX |
| Franklin | Wong | 638 Voss, Houston, TX |
| Ramesh | Narayan | 975 Fire Oak, Humble, TX |
| Joyce | English | 5631 Dallas, Houston, TX |

Query 2)= For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

STAFFORD\_PROJS ← σPlocation=‘Stafford’(PROJECT)

CONTR\_DEPTS ← (STAFFORD\_PROJS Dnum=DnumberDEPARTMENT)

PROJ\_DEPT\_MGRS ← (CONTR\_DEPTS Mgr\_ssn=SsnEMPLOYEE)

RESULT ← πPnumber, Dnum, Lname, Address, Bdate(PROJ\_DEPT\_MGRS)

Output:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pnumber | Dnum | Lname | Address | Bdate |
| 10 | 4 | Wallace | 291 Berry, Bellarie, TX | 1941-06-20 |
| 30 | 4 | Wallace | 291 Berry, Bellarie, TX | 1941-06-20 |

Query 3)= Find the names of employees who work on all the projects controlled by department number 5.

DEPT5\_PROJS ← ρ(Pno)(πPnumber(σDnum=5(PROJECT)))

EMP\_PROJ ← ρ(Ssn, Pno)(πEssn, Pno(WORKS\_ON))

RESULT\_EMP\_SSNS ← EMP\_PROJ ÷ DEPT5\_PROJS

RESULT\_EMP\_SSNS

RESULT ← πLname, Fname(RESULT\_EMP\_SSNS \* EMPLOYEE)

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |

Query 4)= Make a list of project numbers for projects that involve an employee whose last name is ‘Smith’, either as a worker or as a manager of the department that controls the project.

SMITHS(Essn) ← πSsn (σLname=‘Smith’(EMPLOYEE))

SMITH\_WORKER\_PROJS ← πPno(WORKS\_ON \* SMITHS)

MGRS ← πLname, Dnumber(EMPLOYEE Ssn=Mgr\_ssnDEPARTMENT)

SMITH\_MANAGED\_DEPTS(Dnum) ← πDnumber (σLname=‘Smith’(MGRS))

SMITH\_MGR\_PROJS(Pno) ← πPnumber(SMITH\_MANAGED\_DEPTS \* PROJECT)

RESULT ← (SMITH\_WORKER\_PROJS ∪ SMITH\_MGR\_PROJS)

Output:

|  |
| --- |
| PNO |
| 1 |
| 2 |

Query 5)= List the names of all employees with two or more dependents.

T1(Ssn, No\_of\_dependents)← Essn ℑ COUNT Dependent\_name(DEPENDENT)

T2 ← σNo\_of\_dependents>2(T1)

RESULT ← πLname, Fname(T2 \* EMPLOYEE)

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Smith | John |
| Wong | Franklin |

Query 6)= Retrieve the names of employees who have no dependents. This is an example of the type of query that uses the MINUS (SET DIFFERENCE) operation.

ALL\_EMPS ← πSsn(EMPLOYEE)

EMPS\_WITH\_DEPS(Ssn) ← πEssn(DEPENDENT)

EMPS\_WITHOUT\_DEPS ← (ALL\_EMPS – EMPS\_WITH\_DEPS)

RESULT ← πLname, Fname(EMPS\_WITHOUT\_DEPS \* EMPLOYEE)

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Zelaya | Alicia |
| Narayan | Ramesh |
| English | Joyce |
| Jabbar | Ahmad |
| Borg | James |

Query7)= . List the names of managers who have at least one dependent.

MGRS(Ssn) ← πMgr\_ssn(DEPARTMENT)

EMPS\_WITH\_DEPS(Ssn) ← πEssn(DEPENDENT)

MGRS\_WITH\_DEPS ← (MGRS ∩ EMPS\_WITH\_DEPS)

RESULT ← πLname, Fname(MGRS\_WITH\_DEPS \* EMPLOYEE)

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Wallace | Jennifer |
| Wong | Franklin |

8.16. Specify the following queries on the COMPANY relational database schema shown in Figure 5.5 using the relational operators discussed in this chapter. Also show the result of each query as it would apply to the database state in Figure 5.6.

a. Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.

Ans): EMP\_W\_X <-- ( s PNAME='ProductX' (PROJECT)) J (PNUMBER),(PNO)

(WORKS\_ON)

EMP\_WORK\_10 <-- (EMPLOYEE) J (SSN),(ESSN) ( s HOURS>10 (EMP\_W\_X))

RESULT <-- P LNAME,FNAME ( s DNO=5 (EMP\_WORK\_10))

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Smith | John |
| English | Joyce |

b. List the names of all employees who have a dependent with the same first name as themselves.

Ans): E <-- (EMPLOYEE) J (SSN,FNAME),(ESSN,DEPENDENT\_NAME) (DEPENDENT)

R <-- P LNAME,FNAME (E)

Output: (Empty)

c. Find the names of all employees who are directly supervised by ‘Franklin Wong’.

Ans): WONG\_SSN <-- P SSN ( s FNAME='Franklin' AND

LNAME='Wong' (EMPLOYEE))

WONG\_EMPS <-- (EMPLOYEE) J (SUPERSSN),(SSN) (WONG\_SSN)

RESULT <-- P LNAME,FNAME (WONG\_EMPS)

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Smith | John |
| Narayan | Ramesh |
| English | Joyce |

d. For each project, list the project name and the total hours per week (by all employees) spent on that project.

Ans): PROJ\_HOURS(PNO,TOT\_HRS) <-- PNO f SUM HOURS (WORKS\_ON)

RESULT <-- P PNAME,TOT\_HRS ( (PROJ\_HOURS) J (PNO),(PNUMBER)

(PROJECT) )

Output:

|  |  |
| --- | --- |
| PNAME | TOT\_HRS |
| ProductX | 52.5 |
| ProductY | 37.5 |
| ProductZ | 50.0 |
| Computerization | 55.0 |
| Reorganization | 25.0 |
| Newbenefits | 55.0 |

e. Retrieve the names of all employees who work on every project.

Ans): PROJ\_EMPS(PNO,SSN) <-- P PNO,ESSN (WORKS\_ON)

ALL\_PROJS(PNO) <-- P PNUMBER (PROJECT)

EMPS\_ALL\_PROJS <-- PROJ\_EMPS -:- ALLPROJS (\* DIVISION operation \*)

RESULT <-- P LNAME,FNAME (EMPLOYEE \* EMP\_ALL\_PROJS)

Output:

(empty)

f. Retrieve the names of all employees who do not work on any project.

Ans): ALL\_EMPS <-- P SSN (EMPLOYEE)

WORKING\_EMPS(SSN) <-- P ESSN (WORKS\_ON)

NON\_WORKING\_EMPS <-- ALL\_EMPS - WORKING\_EMPS (\* DIFFERENCE

\*)

RESULT <-- P LNAME,FNAME (EMPLOYEE \* NON\_WORKING\_EMPS)

Output:(empty)

g. For each department, retrieve the department name and the average salary of all employees working in that department.

Ans): DEPT\_AVG\_SALS(DNUMBER,AVG\_SAL) <-- DNO f AVG SALARY

(EMPLOYEE)

RESULT <-- P DNUMBER,AVG\_SAL ( DEPT\_AVG\_SALS \* DEPARTMENT )

Output:

|  |  |
| --- | --- |
| DNUMBER | AVG\_SAL |
| Research | 33250 |
| Administration | 31000 |
| Headquarters | 55000 |

h. Retrieve the average salary of all female employees.

Ans): RESULT(AVG\_F\_SAL) <-- f AVG SALARY ( s SEX='F' (EMPLOYEE) )

Output:

|  |
| --- |
| AVG\_F\_SAL |
| 31000 |

1. Find the names and addresses of all employees who work on at least one project located in Houston but whose department has no location in Houston.

Ans): ) E\_P\_HOU(SSN) <--

P ESSN (WORKS\_ON J(PNO),(PNUMBER) ( s PLOCATION='Houston' (PROJECT)))

D\_NO\_HOU <--

P DNUMBER (DEPARTMENT) - P DNUMBER ( s

DLOCATION='Houston' (DEPARTMENT))

E\_D\_NO\_HOU <-- P SSN (EMPLOYEE J(PNO),(DNUMBER) (D\_NO\_HOU))

RESULT\_EMPS <-- E\_P\_HOU - E\_D\_NO\_HOU (\* this is set DIFFERENCE \*)

RESULT <-- P LNAME,FNAME,ADDRESS (EMPLOYEE \* RESULT\_EMPS)

Output:

|  |  |  |
| --- | --- | --- |
| LNAME | FNAME | ADDRESS |
| Wallace | Jennifer | 291 Berry, Bellaire,TX |

j. List the last names of all department managers who have no dependents.

Ans): DEPT\_MANAGERS(SSN)<-- P MGRSSN (DEPARTMENT)

EMPS\_WITH\_DEPENDENTS(SSN) <-- P ESSN (DEPENDENT)

RESULT\_EMPS <-- DEPT\_MANAGERS - EMPS\_WITH\_DEPENDENTS

RESULT <-- P LNAME,FNAME (EMPLOYEE \* RESULT\_EMPS)

Output:

|  |  |
| --- | --- |
| LNAME | FNAME |
| Borg | James |

8.18. Consider the LIBRARY relational database schema shown in Figure 8.14, which is used to keep track of books, borrowers, and book loans. Referential integrity constraints are shown as directed arcs in Figure 8.14, as in the notation of Figure 5.7. Write down relational expressions for the following queries:

a. How many copies of the book titled The Lost Tribe are owned by the library branch whose name is ‘Sharpstown’?

Ans: (Note: We will use S for SELECT, P for PROJECT, \* for NATURAL JOIN, - for

SET DIFFERENCE, F for AGGREGATE FUNCTION)

Ans): A <-- BOOKCOPIES \* LIBRARY-BRANCH \* BOOK

RESULT <-- P No\_Of\_Copies ( S BranchName='Sharpstown' and Title='The Lost

Tribe'

(A) )

Note: A better query would be to do the SELECTs before the JOIN as follows:

A <-- P No\_Of\_Copies ( ( S BranchName='Sharpstown' (LIBRARY-BRANCH) ) \*

(BOOKCOPIES \* ( S Title='The Lost Tribe'

(BOOK) ) ) )

b. How many copies of the book titled The Lost Tribe are owned by each library branch?

Ans: P BranchID,No\_Of\_Copies ( ( S Title='The Lost Tribe' (BOOK)) \* BOOKCOPIES )

c. Retrieve the names of all borrowers who do not have any books checked out.

Ans= NO\_CHECKOUT\_B <-- P CardNo (BORROWER) - P CardNo (BOOK\_LOANS)

RESULT <-- P Name (BORROWER \* NO\_CHECKOUT\_B)

d. For each book that is loaned out from the Sharpstown branch and whose Due\_date is today, retrieve the book title, the borrower’s name, and the borrower’s address.

Ans: S <-- P BranchId ( S BranchName='Sharpstown' (LIBRARY-BRANCH) )

B\_FROM\_S <-- P BookId,CardNo ( ( S DueDate='today' (BOOKLOANS) ) \* S )

RESULT <-- P Title,Name,Address ( BOOK \* BORROWER \* B\_FROM\_S )

e. For each library branch, retrieve the branch name and the total number of books loaned out from that branch.

Ans: R(BranchId,Total) <-- BranchId FCOUNT(BookId,CardNo) (BOOK\_LOANS)

RESULT <-- P BranchName,Total (R \* LIBRARY\_BRANCH)

f. Retrieve the names, addresses, and number of books checked out for all borrowers who have more than five books checked out.

Ans: B(CardNo,TotalCheckout) <-- CardNo F COUNT(BookId) (BOOK\_LOANS)

B5 <-- S TotalCheckout > 5 (B)

RESULT <-- P Name,Address,TotalCheckout ( B5 \* BORROWER)

g. For each book authored (or coauthored) by Stephen King, retrieve the title and the number of copies owned by the library branch whose name is Central.

Ans: SK(BookId,Title) <-- ( sAuthorName='Stephen King' ( BOOK\_AUTHORS)) \* BOOK

CENTRAL(BranchId) <-- sBranchName='Central' ( LIBRARY\_BRANCH )

RESULT <-- P Title,NoOfCopies ( SK \* BOOKCOPIES \* CENTRAL )