

## HW #4 7th Ed

### Homework Assignment #4:

All,

In HW4, you need to demonstrate your work using a RDBMS (e.g., MS Access, Oracle, MySQL, or others.)

Please use DDLs below to create the company database. The schema diagram for the company database is shown in Figure 5.5 (p. 161). You may encounter problems running the DDLs to create tables. Explain the reasons and how you fix them so that the table can be created

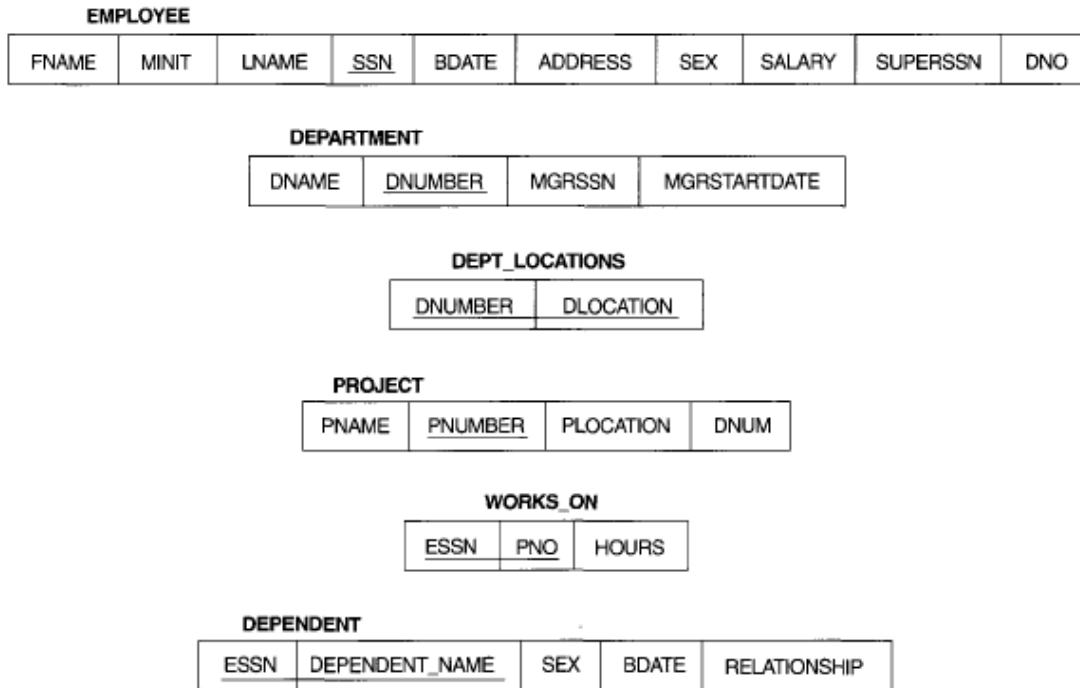


FIGURE 5.5 Schema diagram for the COMPANY relational database schema.

After you have created your database, please use DMLs below to populate all records shown in Figure 5.6 (p. 162) into your database. You may encounter problems running DMLs to insert data. Explain the reasons and fix them so that the data can be populated properly.

## HW #4 7th Ed

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	538 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 Barry, Belaire, 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

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Belaire
	5	Sugarland
		Houston

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1968-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

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

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Belaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1966-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

**FIGURE 5.6** One possible database state for the COMPANY relational database schema.

You can use screenshots to explain the problems and discuss how to fix them.

After you finish the first part above, demonstrate how the Division operation shown in Figure 8.8 (p. 256) works using the example: "Retrieve the SSNs of employees who work on all projects that John Smith works on" from the company database, using one SQL query statement that can retrieve the same results of the above example.

You may turn in your DDLs (create tables), DMLs (insert records) or your final SQL for the Division operation with your retrieved data. You may turn in your DBMS; you can have the log file, screenshots that include your DDLs and DMLs, and your findings in a Word document to prove your work is done properly.

Dar-Ning

### Modified DDLs for the Company Database:

-- Ref: Textbook Page 181

-- Figure 6.1 SQL CREATE TABLE data definition statements for defining the COMPANY schema from Figure 5.7.

-- Figure 6.2 Example illustrating how default attribute values and referential integrity triggered actions are specified in SQL.

```
CREATE TABLE EMPLOYEE (
    Fname          VARCHAR(15)          NOT NULL,
    Minit          CHAR,
    Lname          VARCHAR(15)          NOT NULL,
    Ssn            CHAR(9)              NOT NULL,
    Bdate          DATE,
    Address        VARCHAR(30),
    Sex            CHAR,
    Salary          DECIMAL(10, 2),
    Super_ssn      CHAR(9),
    Dno            INT                  NOT NULL,
    CONSTRAINT EMP_PK
        PRIMARY KEY (Ssn),
    CONSTRAINT EMP_SUPER_FK
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
            ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT EMP_DEPT_FK
        FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
            ON DELETE SET DEFAULT ON UPDATE CASCADE
);

CREATE TABLE DEPARTMENT (
    Dname          VARCHAR(15)          NOT NULL,
    Dnumber        INT                  NOT NULL,
    Mgr_ssn        CHAR(9)              NOT NULL DEFAULT '88866555',
    Mgr_start_date DATE,
    CONSTRAINT DEPT_PK
        PRIMARY KEY (Dnumber),
    CONSTRAINT DEPT_UK
        UNIQUE (Dname),
    CONSTRAINT DEPT_MGR_FK
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE (Ssn)
            ON DELETE SET DEFAULT ON UPDATE CASCADE
);

CREATE TABLE DEPT_LOCATIONS (
    Dnumber        INT                  NOT NULL,
    Dlocation      VARCHAR(15)          NOT NULL,
    CONSTRAINT DEPT_LOCATIONS_PK
        PRIMARY KEY (Dnumber , Dlocation),
    CONSTRAINT DEPT_LOCATIONS_DEPT_FK
        FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT (Dnumber)
            ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE PROJECT (
    Pname          VARCHAR(15)          NOT NULL,
    Pnumber        INT                  NOT NULL,
```

## HW #4 7th Ed

---

```
        Plocation          VARCHAR(15),
        Dnum               INT              NOT NULL,
CONSTRAINT PROJECT_PK
        PRIMARY KEY (Pnumber),
        UNIQUE (Pname),
CONSTRAINT PROJECT_DEPT_FK
        FOREIGN KEY (Dnum) REFERENCES DEPARTMENT (Dnumber)
);

CREATE TABLE WORKS_ON (
        Essn               CHAR(9)          NOT NULL,
        Pno                INT              NOT NULL,
        Hours              DECIMAL(3, 1),
CONSTRAINT WORKS_ON_PK
        PRIMARY KEY (Essn , Pno),
CONSTRAINT WORKS_ON_EMP_FK
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE (Ssn),
CONSTRAINT WORKS_ON_PROJ_FK
        FOREIGN KEY (Pno) REFERENCES PROJECT (Pnumber)
);

CREATE TABLE DEPENDENT (
        Essn               CHAR(9)          NOT NULL,
        Dependent_name     VARCHAR(15)      NOT NULL,
        Sex                CHAR,
        Bdate              DATE,
        Relationship        VARCHAR(8),
CONSTRAINT DEPENDENT_PK
        PRIMARY KEY (Essn , Dependent_name),
CONSTRAINT DEPENDENT_EMP_FK
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE (Ssn)
);
```

### DMLs for the Company Database:

In order to save your time, I have included the DMLs for you. However, you may need to modify them to properly import data with a DATE datatype of your RDBMS and a couple of minor typos.

```
INSERT INTO EMPLOYEE(FNAME,
MINIT,LNAME,SSN,BDATE,ADDRESS,SEX,SALARY,SUPERSSN,DNO) VALUES
('John','B','Smith',123456789,'09-JAN-65','731 Fondren, Houston,
TX','M',30000,333445555,5);
```

```
INSERT INTO EMPLOYEE(FNAME,
MINIT,LNAME,SSN,BDATE,ADDRESS,SEX,SALARY,SUPERSSN,DNO) VALUES
('Franklin','T','Wong',333445555,'08-DEC-55','635 Voss, Houston,
TX','M',40000,888665555,5);
```

```
INSERT INTO EMPLOYEE(FNAME,
MINIT,LNAME,SSN,BDATE,ADDRESS,SEX,SALARY,SUPERSSN,DNO) VALUES
('Alicia','J','Zelaya',999887777,'19-JAN-68','3321 Castle,
Spring','F',25000,987654321,4);
```

```
INSERT INTO EMPLOYEE(FNAME,
MINIT,LNAME,SSN,BDATE,ADDRESS,SEX,SALARY,SUPERSSN,DNO) VALUES
```

## HW #4 7th Ed

---

```
('Jennifer','S','Wallace',987654321,'20-JUN-41','291 Berry, Bellaire, TX','F',43000,888665555,4);
```

```
INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO) VALUES ('Ramesh','K','Narayan',666884444,'15-SEP-62','975 Fire Oak, Humble, TX','M',38000,333445555,5);
```

```
INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO) VALUES ('Joyce','A','English',453453453,'31-JUL-72','5631 Rice Houston','F',25000,333445555,5);
```

```
INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO) VALUES ('Ahmad','V','Jabbar',987987987,'29-MAR-69','980 Dallas, Houston','M',25000,987654321,4);
```

```
INSERT INTO EMPLOYEE (FNAME, MINIT, LNAME, SSN, BDATE, ADDRESS, SEX, SALARY, SUPERSSN, DNO) VALUES ('James','E','Borg',888665555,'10-NOV-37','450 Stone, Houston, TX','M',55000,NULL,1);
```

```
INSERT INTO DEPARTMENT (DNAME, DNUMBER, MGRSSN, MGRSTARTDATE) VALUES ('Research',5,333445555,'22-MAY-88');
```

```
INSERT INTO DEPARTMENT (DNAME, DNUMBER, MGRSSN, MGRSTARTDATE) VALUES ('Headquarters',1,888665555,'19-JUN-81');
```

```
INSERT INTO DEPARTMENT (DNAME, DNUMBER, MGRSSN, MGRSTARTDATE) VALUES ('Administration',4,987654321,'01-JAN-95');
```

```
INSERT INTO DEPT_LOCATIONS (DNUMBER, DLOCATION) VALUES (1, 'Houston');
```

```
INSERT INTO DEPT_LOCATIONS (DNUMBER, DLOCATION) VALUES (4, 'Stafford');
```

```
INSERT INTO DEPT_LOCATIONS (DNUMBER, DLOCATION) VALUES (5, 'Bellaire');
```

```
INSERT INTO DEPT_LOCATIONS (DNUMBER, DLOCATION) VALUES (5, 'Sugarland');
```

```
INSERT INTO DEPT_LOCATIONS (DNUMBER, DLOCATION) VALUES ( , 'Houston');
```

```
INSERT INTO PROJECT (PNAME, PNUMBER, PLOCATION, DNUM) VALUES ('ProductX',1, 'Bellaire',5);
```

```
INSERT INTO PROJECT (PNAME, PNUMBER, PLOCATION, DNUM) VALUES ('ProductY',2, 'Sugarland',5);
```

## HW #4 7th Ed

---

```
INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM) VALUES
('ProductZ',3,'Houston',5);
```

```
INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM) VALUES
('Computerization',10,'Stafford',4);
```

```
INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM) VALUES
('Reorganization',20,'Houston',1);
```

```
INSERT INTO PROJECT(PNAME,PNUMBER,PLOCATION,DNUM) VALUES
('Newbenefits',30,'Stafford',4);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(123456789,1,32.5);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(123456789,2,7.5);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(666884444,3,40);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(453453453,1,20);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(453453453,2,20);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(333445555,2,10);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(333445555,3,10);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(333445555,10,10);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(333445555,20,10);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(999887777,30,30);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(999887777,10,10);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(987987987,10,35);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(987987987,30,5);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
(987654321,30,20);
```

```
INSERT INTO WORKS_ON(ESSN,PNO,HOURS) VALUES
```

## HW #4 7th Ed

---

```
(987654321,20,15);
```

```
INSERT INTO WORKS_ON (ESSN,PNO,HOURS) VALUES  
(888665555,20,NULL);
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(123456789,'Alice','F','30-DEC-88','Daughter');
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(123456789,'Elizabeth','F','05-MAY-67','Spouse');
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(123456789,'Micheal','M','04-JAN-88','Son');
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(333445555,'Alice','F','05-APR-86','Daughter');
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(333445555,'Joy','F','03-MAY-58','Spouse');
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(333445555,'Theodore','M','25-OCT-83','Son');
```

```
INSERT INTO DEPENDENT (ESSN,DEPENDENT_NAME,SEX,BDATE,RELATIONSHIP)  
VALUES  
(987654321,'Abner','M','28-FEB-42','Spouse');
```

- The first issue that is encountered is that employee references department as a foreign key. The employee table should be created without that reference to the department, and once the department entity is created that foreign key relation can be added. The SQL code used to add the constraint after the employee relation was already created was:

```
ALTER TABLE employee  
ADD CONSTRAINT EMP_DEPT_FK  
FOREIGN KEY (Dno) REFERENCES DEPARTMENT (Dnumber)  
ON DELETE SET DEFAULT ON UPDATE CASCADE  
;
```

- For inserting into the employee relation, superssn is a typo, the column name is super\_ssn. Renaming super\_ssn to superssn was the easiest solution for this.
- Also for inserting an employee tuple, care should be taken to ensure that the manager is inserted before the employee. If not, you will violate the foreign key constraint emp\_super\_fk. For example, I inserted James Borg first, followed by Franklin Wong and Jennifer Wallace, etc.
- Again for inserting an employee tuple, the relationship between employee and department is problematic. Employee has a foreign key for department that they

## HW #4 7th Ed

- work in, and department has a foreign key for the employee manager. This makes it impossible to insert into either without violating the foreign key constraint. One solution for this is to remove the EMP\_DEPT\_FK constraint while inserting the employees and departments, then add the constraint after all the employees and departments are inserted.
- For inserting into department, two columns are misnamed, mgr\_ssn and mgr\_start\_date. These were changed to mgrssn and mgrstartdate respectively to make the insertion operations easier.
  - There was a typo in DEPT\_LOCATIONS, according to figure 5.6 Sugarland is supposed to be assigned to department number 5, while Houston does not have a department number. I was not exactly sure what to do about the Houston entry which did not seem to have a department number. The department number cannot be null since it is a primary key. I did notice that ProjectZ was located in Houston and had a department number of 5, so I figured that the missing value was supposed to be a 5 and gave department 5 a location in Houston.

After making the changes specified, the data was successfully loaded to the postgresql database. The results for each of the relations is shown below:

Query Editor

Query History

1

select \* from employee

Data Output

Explain

Messages

Notifications

	<div>fname</div> <div>character varying (15)</div>	<div>minit</div> <div>character (1)</div>	<div>lname</div> <div>character varying (15)</div>	<div>ssn</div> <div>[PK] character (9)</div>	<div>bdate</div> <div>date</div>	<div>address</div> <div>character varying (30)</div>	<div>sex</div> <div>character (1)</div>	<div>salary</div> <div>numeric (10,2)</div>	<div>superssn</div> <div>character (9)</div>	<div>dno</div> <div>integer</div>	
1	James	E	Borg	888665555	2037-11-...	450 Stone, Houston, TX	M	55000.00	[null]		1
2	Franklin	T	Wong	333445555	2055-12-...	635 Voss, Houston, TX	M	40000.00	888665555		5
3	Jennifer	S	Wallace	987654321	2041-06-...	291 Berry, Bellaire, TX	F	43000.00	888665555		4
4	John	B	Smith	123456789	2065-01-...	731 Fondren, Houston, TX	M	30000.00	333445555		5
5	Alicia	J	Zelaya	999887777	2068-01-...	3321 Castle, Spring	F	25000.00	987654321		4
6	Ramesh	K	Narayan	666884444	2062-09-...	975 Fire Oak, Humble, TX	M	38000.00	333445555		5
7	Joyce	A	English	453453453	1972-07-...	5631 Rice Houston	F	25000.00	333445555		5
8	Ahmad	V	Jabbar	987987987	2069-03-...	980 Dallas, Houston	M	25000.00	987654321		4

Query Editor

Query History

1

select \* from department

Data Output

Explain

Messages

Notifications

	<div>dname</div> <div>character varying (15)</div>	<div>dnumber</div> <div>[PK] integer</div>	<div>mgrssn</div> <div>character (9)</div>	<div>mgrstartdate</div> <div>date</div>
1	Research	5	333445555	1988-05-22
2	Headquarters	1	888665555	1981-06-19
3	Administration	4	987654321	1995-01-01



## HW #4 7th Ed

Query Editor

Query History

1

select \* from dept\_locations

Data Output

Explain

Messages

Notifications

	<div>dnumber</div> <div>[PK] integer</div>		<div>dlocation</div> <div>[PK] character varying (15)</div>
1		1	Houston
2		4	Stafford
3		5	Bellaire
4		5	Sugarland
5		5	Houston

Query Editor

Query History

1 select \* from project

2

Data Output

Explain

Messages

Notifications

	<b>pname</b> character varying (15)	<b>pnumber</b> [PK] integer	<b>plocation</b> character varying (15)	<b>dnum</b> integer
1	ProductX	1	Bellaire	5
2	ProductY	2	Sugarland	5
3	ProductZ	3	Houston	5
4	Computerization	10	Stafford	4
5	Reorganization	20	Houston	1
6	Newbenefits	30	Stafford	4

Query Editor

Query History

1 **select** \* **from** works\_on

2

Data Output

Explain

Messages

Notifications

	<b>essn</b> [PK] character (9)	<b>pno</b> [PK] integer	<b>hours</b> numeric (3,1)
1	123456789	1	32.5
2	123456789	2	7.5
3	666884444	3	40.0
4	453453453	1	20.0
5	453453453	2	20.0
6	333445555	2	10.0
7	333445555	3	10.0
8	333445555	10	10.0
9	333445555	20	10.0
10	999887777	30	30.0
11	999887777	10	10.0
12	987987987	10	35.0
13	987987987	30	5.0
14	987654321	30	20.0
15	987654321	20	15.0
16	888665555	20	[null]

## HW #4 7th Ed

Query Editor

Query History

1 select \* from dependent

2

Data Output

Explain

Messages

Notifications

	essn [PK] character (9)	dependent_name [PK] character varying (15)	sex character (1)	bdate date	relationship character varying (8)
1	123456789	Alice	F	1988-12-...	Daughter
2	123456789	Elizabeth	F	2067-05-...	Spouse
3	123456789	Micheal	M	1988-01-...	Son
4	333445555	Alice	F	1986-04-...	Daughter
5	333445555	Joy	F	2058-05-...	Spouse
6	333445555	Theodore	M	1983-10-...	Son
7	987654321	Abner	M	2042-02-...	Spouse

Division as relational algebra is not supported in SQL, however the same result can be represented using nested operations such as where not exists and except. One example of a DML which can accomplish this is:

```
SELECT ssn FROM employee as e
WHERE NOT EXISTS (
  (select pno from works_on where essn=(
    select ssn from employee where fname='John' and lname='Smith') )
EXCEPT(
  SELECT pno FROM works_on WHERE essn = e.ssn) );
```

Running this query against the created database produces:

Query Editor

Query History

```
1 SELECT ssn FROM employee as e
2 WHERE NOT EXISTS (
3     (select pno from works_on where essn=(
4         select ssn from employee where fname='John' and lname='Smith') )
5 EXCEPT(
6     SELECT pno FROM works_on WHERE essn = e.ssn) );
```

Data Output

Explain

Messages

Notifications

	ssn [PK] character (9)	
1	123456789	
2	453453453	

This matches what is expected. John Smith works on projects 1 and 2, and the only other employee who works on projects 1 and 2 is Joyce English, who has ssn=453453453.