1. (2 pts each) Consider the two tables $T_1$ and $T_2$. Show the results of the following relational algebra operations:

<table>
<thead>
<tr>
<th>Table $T_1$</th>
<th>Table $T_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Q</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
</tr>
<tr>
<td>15</td>
<td>b</td>
</tr>
<tr>
<td>25</td>
<td>a</td>
</tr>
</tbody>
</table>

(a) $T_1 \bowtie_{T_1.P=T_2.A} T_2$

(b) $T_1 \bowtie_{T_1.Q=T_2.B} T_2$

(c) $T_1 \bowtie_{T_1.P=T_2.A} T_2$

(d) $T_1 \bowtie_{T_1.Q=T_2.B} T_2$

(e) $T_1 \cup T_2$

(f) $T_1 \bowtie_{T_1.P=T_2.A \land T_1.R=T_2.C} T_2$
2. Refer to figure 4.6, a schema diagram for a library database, for these questions:

(a) (6 pts) Write the SQL DDL statements to define this database. Include appropriate domains, constraints and referential triggered actions.

```sql
CREATE TABLE Book (  
    Book_id Int PRIMARY KEY,  
    Title Varchar(200),  
    Publisher_name Varchar(200),  
    FOREIGN KEY (Publisher_name) REFERENCES Publisher(Name) 
        ON DELETE SET NULL ON UPDATE CASCADE
);

CREATE TABLE Book_Authors (  
    Book_id Int NOT NULL,  
    Author_name Varchar(200) NOT NULL,  
    PRIMARY KEY (Book_id, Author_name),  
    FOREIGN KEY (Book_id) REFERENCES Book(Book_id) 
        ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE Publisher (  
    Name Varchar(200) PRIMARY KEY,  
    Address Varchar(400),  
    Phone Decimal(20)
);  

CREATE TABLE Book_Copies (  
    Book_id Int NOT NULL,  
    Branch_id Char(4) NOT NULL,  
    No_of_copies Int DEFAULT 1,  
    PRIMARY KEY (Book_id, Branch_id),  
    FOREIGN KEY (Book_id) REFERENCES Book(Book_id) 
        ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (Branch_id) REFERENCES Library_Branch(Branch_id) 
        ON DELETE CASCADE ON UPDATE CASCADE
);  

CREATE TABLE Book_Loans (  
    Book_id Int NOT NULL,  
    Branch_id Char(4) NOT NULL,  
    Card_no Int NOT NULL,  
    Date_out Date,  
    Due_date Date,  
    PRIMARY KEY (Book_id, Branch_id, Card_no),
);
CREATE TABLE Library_Branch (  
    Branch_id Char(4) PRIMARY KEY,  
    Branch_name Varchar(200) NOT NULL,  
    Address Varchar(400)  
);  

CREATE TABLE Borrower (  
    Card_no Int PRIMARY KEY,  
    Name Varchar(200) NOT NULL,  
    Address Varchar(400),  
    Phone Decimal(20)  
);  

(b) (4 pts) Write the SQL DML statement to insert a new borrower, (328820001, 'Marten Fisher', '123 Fake St, Springfield', 406 582 2400), in the database.  

INSERT INTO Borrower  
VALUES (328820001, 'Martin Fisher', '123 Fake St, Springfield', 4065822400);  

c) (4 pts) The Bozeman branch has acquired a second copy of the book Here Comes a Candle. Write the SQL to update the database to increase the number of copies for that book by one.  

UPDATE Book_Copies SET No_of_copies = No_of_copies+1  
WHERE Book_id IN (  
    SELECT Book_id FROM Book WHERE Title='Here Comes a Candle'  
) AND Branch_id IN (  
    SELECT Branch_id FROM Library_Branch WHERE Branch_name='Bozeman'  
);  

3. (4 pts each) Refer to figure 3.5, the schema diagram for the COMPANY database, for these questions:  

(a) Write the SQL query to retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.
SELECT Fname, Minit, Lname
FROM Employee
WHERE Dno = (
    SELECT Dno
    FROM Employee
    WHERE Salary = (
        SELECT max(Salary)
        FROM Employee
    )
);

(b) Write the SQL query to retrieve the names of all employees whose supervisor’s supervisor has ‘888665555’ for Ssn.

SELECT Fname, Minit, Lname
FROM Employee
WHERE Super_ssn IN (
    SELECT Ssn FROM Employee WHERE Super_ssn = 888665555
);

(c) Write the SQL query to retrieve the names of employees who make at least $10,000 more than the employee who is paid the least in the company.

SELECT Fname, Minit, Lname
FROM Employee
WHERE Salary >= 10000+(SELECT min(Salary) FROM Employee);

4. (4 pts each) Refer again to figure 4.6, this time give relational algebra 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’?

\[
\text{Sharps}_\text{id} \leftarrow \pi_{\text{Branch}_\text{id}}(\sigma_{\text{Branch}_\text{name}=\text{Sharpstown}}(\text{Library}\_\text{Branch}))
\]
\[
\text{Tribe}_\text{id} \leftarrow \pi_{\text{Book}_\text{id}}(\sigma_{\text{Title}=\text{The Lost Tribe}}(\text{Book}))
\]
\[
\text{Answer} \leftarrow \pi_{\text{No}_\text{of}_\text{copies}}(\text{Book}\_\text{Copies} \ast \text{Sharps}_\text{id} \ast \text{Tribe}_\text{id})
\]

-or-

\[
\text{Answer} \leftarrow \pi_{\text{No}_\text{of}_\text{copies}}(\sigma_{\text{Branch}_\text{name}=\text{Sharpstown} \wedge \text{Title}=\text{The Lost Tribe}}(\text{Book}\_\text{Copies} \ast \text{Library}\_\text{Branch} \ast \text{Book}))
\]
(b) Retrieve the names of all borrowers who do not have any books checked out.

\[
\text{None}_\text{id} \leftarrow \pi_{\text{Card}_\text{no}}(\text{Borrower}) - \pi_{\text{Card}_\text{no}}(\text{Book}_\text{Loans})
\]
\[
\text{Answer} \leftarrow \pi_{\text{Name}}(\text{Borrower} \ast \text{None}_\text{id})
\]

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

\[
\text{Loan}_\text{counts} \leftarrow \rho(\text{Card}_\text{no}, \text{Book}_\text{count})(\text{Card}_\text{no} \triangleright \text{Count}(\text{Book}_\text{id})(\text{Book}_\text{Loans}))
\]
\[
\text{Big}_\text{borrowers} \leftarrow \sigma_{\text{Book}_\text{count} > 5}(\text{Loan}_\text{counts})
\]
\[
\text{Answer} \leftarrow \pi_{\text{Name}, \text{Address}, \text{Book}_\text{count}}(\text{Big}_\text{borrowers} \ast \text{Borrower})
\]

5. **(6 pts)** In relational models, primary keys based on existing, meaningful attributes of the tuples are known as *natural keys*. Some database designers prefer to add an extra attribute that doesn’t model anything about the miniworld; it is specifically generated solely to be a primary key. These are called *surrogate keys*. Discuss the advantages and disadvantages of both approaches.
A relational database schema for a LIBRARY database.
Figure 3.5

<table>
<thead>
<tr>
<th>EMPLOYEE</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fname</td>
<td>Minit</td>
<td>Lname</td>
<td>Ssn</td>
<td>Bdate</td>
<td>Address</td>
<td>Sex</td>
<td>Salary</td>
<td>Super ssn</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPARTMENT</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dname</td>
<td>Dnumber</td>
<td>Mgr ssn</td>
<td>Mgr start date</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEPT_LOCATIONS</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dnumber</td>
<td>Location</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PROJECT</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pname</td>
<td>Pnumber</td>
<td>Plocation</td>
<td>Dnum</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WORKS_ON</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>n</td>
<td></td>
<td>P</td>
<td>n</td>
</tr>
</tbody>
</table>

| DEPENDENT |    |    |    |    |    |
|-----------|----|----|----|----|
| E|n| | Dependent name | Sex | Bdate | Relationship |

Figure 3.5
Schema diagram for the COMPANY relational database schema.