CS2102 Tutorial 2: RA & SQL 1 **Wk 4, Sem 2, 2020/21**

1. **Solution:**

   (a) $\pi_{rname,pizza}(Sells)/\pi_{pizza}(\sigma_{cname='Maggie'}(Likes)) - \pi_{rname}(Sells \bowtie \sigma_{cname='Ralph'}(Likes))$

   (b) Let $\pi_A(R) = Q \cup Q'$, where $Q = R/S$ and Q' are the remaining tuples in $\pi_A(R)$ that are not in quotient of R/S.

   The expression $\pi_A(R) \times S$ computes all the combinations of $\pi_A(R)$ and S.

   Thus, $Q' = \pi_A((\pi_A(R) \times S) - R)$.

   Therefore, $Q = \pi_A(R) - Q' = \pi_A(R) - \pi_A((\pi_A(R) \times S) - R)$.

2. **Solution:**

```
drop table if exists Offices, Employees cascade;

create table Offices (
    office_id   integer,
    building    text not null,
    level       integer not null,
    room_number integer not null,
    area        integer,
    primary key (office_id),
    unique (building, level, room_number)
);

create table Employees (
    emp_id      integer,
    name        text not null,
    office_id   integer not null,
    manager_id  integer,
    primary key (emp_id),
    foreign key (office_id) references Offices (office_id)
        on update cascade,
    foreign key (manager_id) references Employees (emp_id)
        on update cascade
);
```

   Note that the constraint that each employee must be assigned to exactly one office and the constraint that each employee is managed by at most one manager are both enforced by the primary key constraint in `Employees` which ensures that there can't be two two Employees records with the same primary key value and different values for office_id / manager_id.

3. **Solution:**

   Each table in a database schema must have a primary key.

   (a)

```
drop table if exists
    Books, Customers, Carts, Purchase, Purchased_Items cascade;

create table Books (
    isbn    text,
    title   text not null,
    authors text not null,
    year integer,
    edition text not null
        check (edition in (’hardcopy’, ’paperback’, ’ebook’)),
    publisher text,
    number_pages integer
        check (number_pages > 0),
    price  numeric not null
        check (price > 0),
    primary key (isbn)
);

create table Customers (
    cust_id integer,
    name    text not null,
    email   text,
    primary key (cust_id)
);

create table Carts (
    cust_id integer,
    isbn    text,
    primary key (cust_id, isbn),
    foreign key (cust_id) references Customers,
    foreign key (isbn) references Books
);

create table Purchase (
    pid integer,
    purchase_date   date not null,
    cust_id integer not null,
    primary key (pid),
    foreign key (cust_id) references Customers
);

create table Purchased_Items (
    pid integer,
    isbn    text,
    primary key (pid, isbn),
    foreign key (pid) references Purchase,
    foreign key (isbn) references Books
);
```

(b) (purchase_timestamp, cust_id) is a candidate key of `Purchase`.

```
create table Purchase (
    pid integer,
    purchase_timestamp timestamp not null,
    cust_id integer not null,
    primary key (pid),
    foreign key (cust_id) references Customers,
    unique (purchase_timestamp, cust_id)
);
```

(c) The constraint of the form "$p \implies q$" is equivalent to "(not p) or q".

1. This constraint can be expressed using a table constraint on `Books`:

   check ((edition <> 'hardcover') or (price >= 30))

2. This constraint can't be expressed using the constructs learned so far.
3. This constraint can be expressed using a table constraint on `Books`:

   check ((number_of_pages <= 1000) or (edition = 'ebook') or (price >= 100))

4. This constraint can be expressed using a table constraint on `Books`:

   check ((publisher <> 'Acme') or (pub_year < 2010) or (edition = 'ebook'))

(d)

```
drop table if exists Books, Customers, Carts, Purchase,
    Purchased_Items cascade;

create table Books (
    isbn    text,
    title   text not null,
    authors text not null,
    year    integer,
    edition text not null
        check (edition in ('hardcover', 'paperback', 'ebook')),
    publisher text,
    number_pages integer
        check (number_pages > 0),
    price  numeric
        check (price > 0),
    primary key (isbn)
);

create table Customers (
    cust_id integer,
    name    text not null,
    email   text,
    primary key (cust_id)
);

create table Carts (
    cust_id integer,
    isbn    text,
    primary key (cust_id, isbn),
    foreign key (cust_id) references Customers
        on delete cascade
        on update cascade,
    foreign key (isbn) references Books
        on delete cascade
        on update cascade
);

create table Purchase (
    pid integer,
    purchase_date   date not null,
    cust_id integer not null,
    primary key (pid),
    foreign key (cust_id) references Customers
        on delete cascade
        on update cascade
);

create table Purchased_Items (
    pid integer,
    isbn    text default '0',
    primary key (pid, isbn),
    foreign key (pid) references Purchase
        on delete cascade
        on update cascade,
    foreign key (isbn) references Books
        on delete set default
        on update cascade
);
```

For the "on delete set default" action to work in `Purchased_items` when a referenced book in `Books` is deleted, there must exist a record in `Books` with isbn = '0'. If not, the deletion operation will be rejected.