# CS2102 Tutorial 2

1.

$R/S$ computes largest set of tuples $Q \subseteq \pi_{A1 \ldots Am} R$

such that for every tuple $(a_1, \ldots a_m) \in Q$

$$\pi_L \left( \{(a_1, \ldots a_m)\} \times S \right) \subseteq R$$

b)

To find all possible $(A, B)$ pairs $\Rightarrow$ $\pi_A(R) \times S$

To find tuples that are in $\pi_A(R) \times S$ but not in $R$ $\Rightarrow$ $\pi_A(R) \times S - R$

These tuples that violate definition cannot be part of $R/S$

$$\boxed{R/S = \pi_A(R) - \pi_A\left(\pi_A(R) \times S - R\right)}$$

a)

* quotient is used when we wish to express queries with "all"

based on $S$, find all $\pi_A(R)$ which is related to "all" of $S$

pizzas maggie likes $Q_1 = \pi_{pizza}\left(\sigma_{cname = 'Maggie'} Likes\right)$

pizzas Ralph likes $Q_2 = \pi_{pizza}\left(\sigma_{cname = 'Ralph'} Likes\right)$

pizzas maggie likes but not Ralph $= Q_3 = Q_1 - Q_2$

find all restaurants related to $Q_3$

$$\pi_{rname}(sells / Q_3)$$

2.

```
CREATE TABLE Offices (
    office_id   INTEGER PRIMARY KEY,
    building    VARCHAR(50) NOT NULL,
    level       INTEGER NOT NULL,
    room_number INTEGER NOT NULL

    UNIQUE ( building, level, room_number )
);
```

— office_id is the primary key

(building, level, room_number) is a candidate key ⇒ unique and not null

```
CREATE TABLE Employees (
    emp_id      INTEGER PRIMARY KEY,
    name        VARCHAR(50) NOT NULL,
    office_id   INTEGER NOT NULL,
    manager_id  INTEGER

    FOREIGN KEY ( manager_id ) REFERENCES Employees ( emp_id )
        ON DELETE NO ACTION ON UPDATE CASCADE

    FOREIGN KEY ( office_id ) REFERENCES Offices ( office_id )
        ON DELETE NO ACTION ON UPDATE CASCADE
);
```

— emp_id is the primary key

— name is not null

— exactly 1 office ⇒ not null

— at most 1 manager, can be null

— reject delete if managing someone

— propagate update

— reject delete if employee using office

— propagate update

ids, level and room number are usually integers.

names are usually characters, 50 should be sufficient

3. a)

```
CREATE TABLE Books (
    isbn        INTEGER PRIMARY KEY,
    title       VARCHAR(50) NOT NULL,
    authors     VARCHAR(50) NOT NULL,
    year        INTEGER,
    edition     VARCHAR(50) NOT NULL,
    publisher   VARCHAR(50),
    number_pages INTEGER,
    price       FLOAT(8) NOT NULL,
    CHECK ( edition = 'paperback' OR edition = 'hardcover' OR edition = 'ebook' ),
    CHECK ( price > 0 ),
    CHECK ( number_pages IS NULL OR number_pages > 0 )
);

CREATE TABLE Customers (
    cust_id INTEGER PRIMARY KEY,
    name    VARCHAR(50) NOT NULL,
    email   VARCHAR(50)
);

CREATE TABLE Cards (
    cust_id INTEGER,
    isbn    INTEGER,
    FOREIGN KEY ( cust_id ) REFERENCES Customers ( cust_id ),
    FOREIGN KEY ( isbn ) REFERENCES Books (isbn)
);

CREATE TABLE Purchase (
    pid         INTEGER PRIMARY KEY,
    purchase_date DATE,
    cust_id     INTEGER,
    FOREIGN KEY (cust_id) REFERENCES Customers ( cust_id)
```

enum CREATE TYPE ... ?

```
CREATE TABLE purchased_items (
    pid     INTEGER,
    isbn    INTEGER,
    FOREIGN KEY (pid) REFERENCES purchase (pid),
    FOREIGN KEY (isbn) REFERENCES Books (isbn)
);
```

b)

Data type for purchase_timestamp becomes TIMESTAMP

There is also a new unique constraint for cust_id and purchase_timestamp

```
UNIQUE (cust_id, purchase_timestamp)
```

c) i)

```
CHECK ( edition = 'hardcover' AND price >= 30
        OR edition = 'paperback' OR edition = 'ebook' )
```

ii)

scope of check constraint is single row

it cannot check for books that have multiple editions

need use assertions or triggers

iii)

```
CHECK ( number_pages > 1000 AND ( edition = 'ebook' OR price >= 100)
        OR number_pages IS NULL OR (number_pages <= 1000 AND
                                    number_pages > 0 ) )
```

iv)

```
CHECK ( publisher = 'ACME' AND year >= 2010 AND edition = 'ebook'
        OR publisher = 'ACME'
```

d) i) ON DELETE CASCADE for foreign key constraint in Cards and Purchases for cust_id

ii) ON DELETE CASCADE for foreign key constraint in Cards for isbn

ON DELETE SET DEFAULT for foreign key constraint in Purchased_items for isbn

include DEFAULT 0 for isbn also

iii) ONDELETE CASCADE for FKC in Purchased_items for p_id

iv)

v) ON UPDATE CASCADE for all FKC

vi)