HW #3

- The SQL parts of this homework allow for, if you prefer, using MongoDB instead.

Part甲

- **The donation relation was defined based on the following SQL statement:**

```
CREATE TABLE donations
        (recipientName CHAR(20) NOT NULL,
        donorOrganization CHAR(20) NOT NULL,
        organizationType CHAR(20),
        amount REAL,
        PRIMARY KEY (recipientName, donorOrganization))
```

- **Determine whether each of the following four pairs of SQL statements is equivalent. If the pair is equivalent, just say yes and no explanation is needed.**

HW #3 (2)

- **If you do not think the pair is equivalent, construct an instance of the donations relation to illustrate the difference between the pair of statements.**

甲1:

```
select distinct recipientName from donations A
where not exists
  (select B.donorOrganization from donations B
   where recipientName = "Campbell"
   and A.donorOrganization ≠ B.donorOrganization)
```

vs

```
select distinct recipientName from donations A
where not exists
   ( (select donorOrganization from donations
      where recipientName = "Campbell" )
      except
     (select donorOrganization from donations B
      where B.recipientName = A.recipientName))
```

HW #3 (3)

甲2:

```
(select distinct recipientName from donations
 where amount ≥ 500)
union
(select distinct recipientName from donations
 where amount < 500)
```

vs

```
select distinct recipientName from donations
```

甲3:

```
select distinct recipientName from donations A, donations B
where A.amount ≥ 1000
and A.recipientName = B.recipientName
and A.donorOrganization ≠ B.donorOrganization
```

vs

```
select distinct recipientName from donations
  where amount ≥ 1000
  group by recipientName
  having count(donorOrganization) ≥ 2
```

HW #3 (4)

甲4:

```
select distinct recipientName from donations
where recipientName not in
    (select recipientName from donations
     where organizationType = tobacco)
```

vs

```
select distinct recipientName from donations A
where exists
    (select * from donations B
     where B.recipientName = A.recipientName
     and organizationType  ≠ tobacco)
```

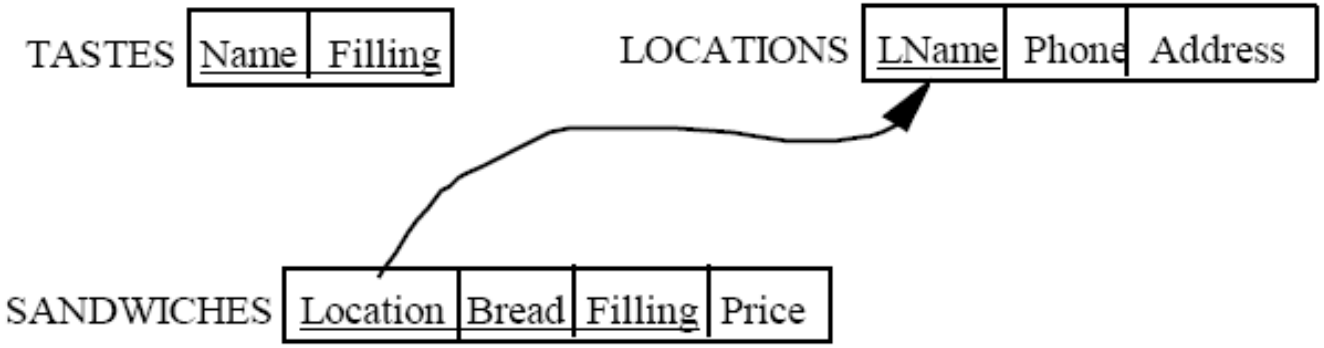
HW #3 (5)

Part乙

- **Consider the following relational database schema. It is intended to represent who will eat what kinds of sandwiches and the places which serve the various kinds of sandwiches. A sample database is also given.**
- **Write SQL statements to retrieve the following information:**

乙1. Places where Jones can eat.

乙2. For each location the number of people who can eat there.



TASTES	NAME	FILLING
	Brown	Turkey
	Brown	Beef
	Brown	Ham
	Jones	Cheese
	Green	Beef
	Green	Turkey
	Green	Cheese

SANDWICHES	LOCATION	BREAD	FILLING	PRICE
	Lincoln	Rye	Ham	1.25
	O'Neill's	White	Cheese	1.20
	O'Neill's	Whole	Ham	1.25
	Old Nag	Rye	Beef	1.35
	Buttery	White	Cheese	1.00
	O'Neill's	White	Turkey	1.35
	Buttery	White	Ham	1.10
	Lincoln	Rye	Beef	1.35
	Lincoln	White	Ham	1.30
	Old Nag	Rye	Ham	1.40

LOCATIONS	LNAME	PHONE	ADDRESS
	Lincoln	683 4523	Lincoln Place
	O'Neill's	674 2134	Pearse St
	Old Nag	767 8132	Dame St
	Buttery	702 3421	College St

HW #3 (6)

Part丙

- **Consider an Internet bookshop which maintains information about authors, publications, publishers, readers and the relations between them.**
- **A publication (e.g., a book or a magazine) is published by one and only one publisher, although a publisher can publish multiply publications.**
- **Moreover, a publication may be made by one or more authors and each author may make more than one publication.**
- **The publication can be sold to readers online.**

HW #3 (7)

- **This system contains the following tables: Authors, Publications, Publishers, Readers, PubAuthors, and Selling. The underline fields are the primary keys. These tables are defined by the following schema:**

  Authors (aname, age, affiliation, nationality);

  Publications (ISBN, title, pdate, category, pid);

  Publishers (pid, pname);

  Readers (rid, rname, email);

  PubAuthors (ISBN, aname);

  Selling (ISBN, rid, quantity);

HW #3 (8)

Table details:

- **Table “Authors” contains an “aname” which can be uniquely identify an author (we assume that every author have a unique name). The age of an author is a number with maximum 3 digits, which does not exceed 999. The “affiliation” attribute records institutes or companies where authors are currently working and the “nationality” shows where authors are.**

HW #3 (9)

- **The “Publications” “ISBN” is the primary key of this table and it can uniquely identify a publication. The “pdate” is the publication date of a publication, and is the date type. The “category” of a publication can be information technology, sciences, history, etc. The “pid” is the identifier of the publisher.**

HW #3 (10)

- **There is a unique “pid” in “Publishers” which is the primary key of this table.**
- **Table “Readers” records information of readers, and the “rid” can uniquely identify a reader.**
- **The “ISBN” and the “aname” are the primary key for the table “PubAuthors”.**

HW #3 (11)

- **Table “Selling” “quantity” is an integer between 0 and 999999, indicating how many copies of a certain publication (a book) a reader buys. The primary key is the “ISBN” of the publication and the “rid” of the reader.**

- **The title of publications, the name of publishers and the email address of readers are of string type with maximum 50 characters each. Other attributes, unless explicitly defined above, are of string type with the maximum 25 characters.**

HW #3 (12)

- **Queries:**

丙1. Find the names of authors, who have made at least 2 publications after the year 2000 (i.e., since January 1, 2001). The results should be listed in descending order of the names of authors.

丙2. For each publisher, find the total sales of each publication. The result should show the publisher's "pid", the name of the publisher, the "ISBN" and the title of its each publication and the total sales of each publication. Please sort the result by the "pid", "pname", "ISBN", "title" and the number of total sales in ascending order.

HW #3 (13)

丙**3.** **Find the most popular publication for each publisher after the year 2000. The most popular publication has the most sales in term of quantity. Please list the title of the publication and the name of the publisher in ascending order. (If two or more publications have the same sales, please list all of them.)**

丙**4.** **Find the most popular publication in each category, which has the most sales. Please list the titles of publications, the total sales and the category (in ascending order).**

HW #3 (14)

丙**5. Find the titles of the technical publications (i.e., the category is “technology”) that are published after the year 2000 and none of authors are older than 50 (“age > 50”). The result should be listed in ascending order of publication title.**

HW #3 (15)

Part丁

- Consider the following relational database schema:
  Earthquake (qname, qdate, latitude, longitude, magnitude, fault斷層)
  Damage (cityID, qname, cost, liveslost)
  Cities (cID, cname, latitude, longitude, population)

  -- [city names may **not** be unique]
- Write SQL statements to retrieve the following information:

丁**1.** Find the names and dates of earthquakes whose magnitude was greater than 7.0.

丁**2.** Find the average population of all the cities whose names start with "San".

HW #3 (16)

- **3.** List the lives lost, quake names and city names for all quakes that occurred before 1960.
- **4.** List the city identifiers and total cost, per city, of all quakes with magnitude greater than 7.0.
- **5.** List the average magnitude of quakes for each fault (grouped by fault).
- **6.** List all earthquakes and, where applicable, also list city IDs and lives lost in that city for quakes that affected cities.
- **7.** List the names of cities whose populations are greater than the average city population.

HW #3 (17)

Part戊

- Use the following schema, write SQL queries to solve the problems.

Product (maker, model, type) **--**maker is foreign key to **Manufacturer** (maker)

PC (model, speed, ram, hd, price)

Laptop (model, speed, ram, hd, screen, price)

Printer (model, color, type, price)

Manufacturer (maker) --no other non-key attribute (assume okay)

HW #3 (18)

- You should use at least one subquery in each of your answers.

戊**1.** Find the makers of PC's with a speed of at least 3.0.

戊**2.** Find the printers with the highest price.

戊**3.** Find the laptops whose speed is slower than that of any PC.

HW #3 (19)

Part已

- Use the following schema, write SQL queries to solve the problems.

  Actor (pid, fname, lname, gender)

  Movie (mid, name, year, revenue營收) --name represents the movie name

  Directors (did, fname, lname)

  Casts (pid, mid, role)

  -- pid is FK to **Actor** (pid), mid is FK to **Movie** (mid)

  -- an actor can play multiple roles in one movie, thus (pid, mid) is not a PrimaryKey

  Movie_directors (did, mid) -- mid is FK to **Movie** (mid) did is FK to **Directors** (did)

HW #3 (20)

- You should use at least one subquery in 已**2**.

已**1.** List all directors who directed at least 200 movies, in descending order of the number of movies they directed. Display the directors' first and last names and the number of movies each of them directed.

已**2.** For each year, count the number of movies in that year that had only female actors. Display the year and the number of movies in that year. **[Hint]** consider a universal quantifier: a movie without any actors is also a movie with only female actors (since there are no male actors in that movie).