CSE 344 Section 2

1. Outer Joins

Given tables A and B,

A:

a1	a2
1	7
2	5
3	3
4	6

B:

b1	b2
3	2
4	1
5	0
6	3

Write down the output of each of the following queries:

```
SELECT * FROM A INNER JOIN B ON A.a1 = B.b1;
```

a1	a2	b1	b2
3	3	3	2
4	6	4	1

```
SELECT * FROM A LEFT OUTER JOIN B ON A.a1 = B.b1;
```

a1	a2	b1	b2
1	7	null	null
2	5	null	null
3	3	3	2
4	6	6	3

```
SELECT * FROM A RIGHT OUTER JOIN B ON A.a1 = B.b1;
```

a1	a2	b1	b2
3	3	3	2
4	6	4	1
null	null	5	0
null	null	6	3

```
SELECT * FROM A FULL OUTER JOIN B ON A.a1 = B.b1;
```

a1	a2	b1	b2
1	7	null	null
2	5	null	null
3	3	3	2
4	6	4	1
null	null	5	0
null	null	6	3

2. Grouping and Aggregation

```
CREATE TABLE Movies (
    id int,
    name varchar(30),
    budget int,
    gross int,
    rating int,
    year int,
    PRIMARY KEY (id)
);
```

```
CREATE TABLE Actors (
    id int,
    name varchar(30),
    age int,
    PRIMARY KEY (id)
);

CREATE TABLE ActsIn (
    mid int,
    aid int,
    FOREIGN KEY (mid) REFERENCES Movies (id),
    FOREIGN KEY (aid) REFERENCES Actors (id)
);
```

Write queries to answer the following:

(a) For each movie, find the number of actors who acted in it, ordered by descending number of actors. Make sure to include movies with no actors!

```
SELECT m.id, count(ai.aid)
FROM Movies m LEFT OUTER JOIN ActsIn ai ON m.id = ai.mid
GROUP BY m.id
ORDER BY count(ai.aid) DESC;
```

(b) What is the number of movies and the average rating of all the movies that the actor "Kit Harington" has appeared in?

```
        SELECT count(*), avg(m.rating)
        FROM Movies m, ActsIn ai, Actors a
        WHERE m.id = ai.mid AND a.id = ai.aid
        AND a.name = 'Kit Harington'
␣␣␣␣␣␣␣␣
```

(c) What is the age of the youngest actor who has appeared in a movie that grossed over $1,000,000,000?

```
SELECT min(age)
FROM Movies m, ActsIn ai, Actors a
WHERE m.id = ai.mid AND a.id = ai.aid
AND m.gross > 1000000000;
```

The following relations track the classes taught by instructors at the UW.

```
CREATE TABLE Class (
    dept varchar(6),
    number int,
    title varchar(75),
    PRIMARY KEY (dept, number)
);

CREATE TABLE Instructor (
    username varchar(8),
    fname varchar (50),
    lname varchar (50),
    PRIMARY KEY (username)
);

CREATE TABLE Teaches (
    username varchar(8),
    dept varchar(6),
    number int,
    PRIMARY KEY (username, dept, number),
    FOREIGN KEY (username) REFERENCES Instructor(username),
    FOREIGN KEY (dept, number) REFERENCES Class(dept, number)
);
```

Write queries to answer the following:

(d) How many classes are taught by at least 1 instructor?

```
SELECT count(*)
FROM Teaches t
GROUP BY t.dept, t.number;
```

(e) Find the username, first name, and last name of the instructors who teach more than 1 class.

```
SELECT I.username, I.fname, I.lname
FROM Instructor I, Teaches T
WHERE I.username = T.username
GROUP BY I.username, I.fname, I.lname
HAVING count(*) > 1;
```

(f) What CSE courses do neither Dr.Suciu ('su') nor Dr.Balazinska ('bal') teach? Find the number, and title of the courses.

```
SELECT c.number, c.title
FROM Class c
WHERE c.dept = 'CSE'
AND c.number NOT IN (
    SELECT c2.number
    FROM Class c2, Teaches t
    WHERE c2.dept = 'CSE'
    AND c2.dept = t.dept
    AND c2.number = t.number
    AND (T.username = 'su' OR T.username = 'bal')
);
```

3. Subqueries

```
CREATE TABLE Company (
    cid int,
    cname varchar(20),
    PRIMARY KEY cname
)
```

```
CREATE TABLE Product (
    pname varchar(20),
    price int,
    cid int,
    PRIMARY KEY pname,
    FOREIGN KEY cid REFERENCES Company(cid)
);
```

Write queries to find the following information:

(a) Find all the companies that only sell products that cost over $200.

```
SELECT c.cid
FROM Company c
WHERE 200 < ALL (
    SELECT p.price
    FROM Product p
    WHERE p.cid = c.cid
);
```

(b) Find all companies that do not sell any products.

```
(SELECT c.cid
 FROM Company c)
 EXCEPT
(SELECT c.cid
 FROM Company c, Product p
 WHERE c.cid = p.pid
 );
```

Another way to write the query:

```
SELECT c.cid
FROM Company c
WHERE NOT EXISTS (
    SELECT * FROM Product p
    WHERE p.cid = c.cid)
)
```

(c) For each company, find the name of its most expensive product. If multiple products are tied for highest price, return all of them.

```
WITH CompanyMax AS (
    SELECT c.cid AS cid, max(p.price) AS max_price
    FROM Company c INNER JOIN Product p ON c.cid = p.cid)
SELECT cm.cid
FROM CompanyMax cm, Product p2
WHERE cm.cid = p2.cid
AND cm.max_price = p2.price;
```

Another way to write the query:

```
SELECT c.cid, v.pname
FROM Company c, Product v
WHERE c.cid = v.cid
AND v.price >= ALL (
    SELECT v2.price
    FROM Product v2
    WHERE c.cid = v2.cid)
);
```