CS561WS Midterm Exam Solutions

Exam solutions must be submitted as a single MS Word or MS powerpoint file. (No other file types will be accepted, nor will multiple files.)

Each of the nine questions is worth 11 points.

1. We are given a database schema with two table schemas: Table1(A, B, C, D, E, F) and Table2(G, H, I, J, K, L), with a foreign key reference from IJK of Table2 to ABC of Table1.

Which row(s), if any, of the following instances of Table1 and Table2 violate the foreign key reference. For each row that violates the foreign key reference, explain why it violates it.

Table1

A	B	C	D	E	F
1	1	2	2	1	1
1	1	3	3	1	1
3	2	1	1	2	1
2	1	2	2	1	2
3	2	1	1	2	3

Table2

G	H	I	J	K	L
1	2	2	1	2	3
2	3	1	2	1	2
3	1	1	1	2	1
3	3	3	2	1	1
1	3	3	2	1	2

Table 2 Row 2 violates the foreign key reference because (A=1, B=2, C=1) does not exist in Table 1.

There also happens to be a key constraint violation in Table 1: rows 3 and 5, but this wasn't asked for, so no credit is lost if this isn't noticed.

2. (Multiple company database from SQL chapter.) How many rows are there in the value of the relational algebra expression

$$\sigma(\text{site} \times \text{division} \times \text{company})$$

(site.co_id = division.co_id)
AND (site.div_name = division.div_name)
AND (division.co_id = company.govt_id)

on the assumptions that the instance of site has m rows, the instance of division has n rows, and the instance of company has k rows. (Be sure to explain how you arrived at your answer.)

There are m rows because for each site there is a unique division and for each division there is a unique company.

3. (Multiple company database from SQL chapter.) Write the following query in **un-extended** relational algebra: Find every product at whose manufacture no employee of the company that manufactures the product is skilled.

Strategy:
(set of all products) – (set of products where at least one employee of the company that manufactures the product is skilled at producing that product)

π (product) - π (σ (skill X works_for))

product.prod_id

skill.prod_id

(skill.ssn = works_for.ssn) AND
(works_for.co_id = skill.manuf_co)

4. (Multiple company database from SQL chapter.) Write the following query in SQL: Find every company none of whose divisions has any sites. You may not use set union, set intersection or set difference in this query. You may also not use IN, NOT IN, EXISTS, NOT EXISTS, GROUP BY, or HAVING

```
select company.govt_id
from company
where
        (select count (*)
        from site
        where  company.govt_id =site.co_id) = 0
```

Note: None of the company's divisions has a site = the company has no sites

5. (Multiple company database from SQL chapter) Write the following query in **extended** relational algebra: Find every product at whose manufacture every person who works for the company that produces it is skilled.

$$\pi_{\text{product.prod\_id,manuf\_co}} (\sigma \text{ (product)}) \; \pi \text{ (COUNT(*))} \; \sigma \text{ (works\_for)} \; = \pi \text{ (COUNT(*))} \sigma \text{ (skill)}$$

works_for.co_id = product.manuf_co (skill.manuf_co = product.manuf_co)
AND (skill.prod_id = product.prod_id)

6. (Airline database from Lecture/homework 1) Write the following query in SQL: Find every airplane with more than 100 seats that can land at all the airports. You may not use set union, set intersection or set difference in this query. You may also not use IN, NOT IN, EXISTS, NOT EXISTS, GROUP BY, or HAVING

Strategy:

```
Select airplane.airplane_id
from airplane
where total_number_of_seats > 100
        AND (count of airports where plane can land = number of airports)
```

```
select airplane.airplane_id from airplane
where total_number_of_seats > 100 AND
(
        (select count(*)
        from can_land
        where airplane.airplane_type = airplane_type_name)
        =
(select count(*) from airport)
```

7. Write SQL for the query – based on the SQL Chapter's multiple-company database – "Find every company that produces no products." This time you may have an SFW embedded in the outer SFW, and it must be embedded using either the EXITS or NOT EXISTS operator, whichever makes the complete SFW simpler. You may not use set union, set intersection or set difference in this query. You may not use IN, NOT IN, GROUP BY or HAVING anywhere in your answer.

```
SELECT company.govt_id
FROM company
WHERE NOT EXISTS (SELECT *
                        FROM product
                        WHERE company.govt_id = product.manuf_co)
```

8. (Multiple company database from SQL chapter.) Write the following query in SQL using either the IN or NOT IN, embedding operator, whichever is more appropriate: Find every person who works for every company that has a division in Atlanta at which at least 10 products are manufactured. You may not use set union, set intersection or set difference in this query. You may also not use EXISTS, NOT EXISTS, GROUP BY, or HAVING.

It's important to think through how one is going to construct a query; one way is to use the kind of "strategy" that is used in the lectures. Here's one way to do that in order to develop the query:

```
select person.ssn
from person
where (the person is IN (      the set of persons such that
                               the person works for every company
                               such that the company has a division in Atlanta at which at
                               least 10 products are manufactured)
```

which is

```
select person.ssn
from employee
where (the person is IN (      the set of persons such that
                               (the number of companies that the employee works for that
                               have a division in Atlanta that manufactures at least ten
                               products)
                               =
                               (the number of companies that have a division in Atlanta
                               that manufactures at least ten products)
```

9. (Single company database from Elmasri & Navathe book) Write the following query in SQL using GROUP BY but not HAVING: Find every supervisor who supervises more than 50 employees and the number of projects on which the supervisor works.

```
select supervisor.ssn, count(works_on)
from employee AS supervisor
where  (count(supervisee.ssn)
        from employee as supervisee
        where (supervisee.superssn = supervisor.ssn)) > 50
        and
        works_on.essn = supervisor.essn
group by supervisor.ssn
```