**CS561 Midterm Exam**

**Midterms must be submitted as single Word or PPT files; other types of files will not be graded or given credit**

**Also, midterms must be submitted by the stated deadline or they will not be graded or given credit**

Name: Jiayin Huang

CWID: 10477088

All questions on this exam refer to the following relational database schema

            company(co\_name, govt\_id, ceo\_ssn, hq\_loc)

            division(co\_id, div\_name, subdiv\_of, dir\_ssn, div\_hq)

            site(co\_id, div\_name, loc)

            product(prod\_id, manuf\_co, manuf\_div, loc, prod\_descr)

            person(ssn, name, address)

            works\_for(ssn, co\_id, div\_name, salary, emp\_id, sup\_ssn)

            skill(ssn, prod\_id, manuf\_co)

The various keys and fkrs may be found in the ADK SQL Chapter

1. a) (3 points) Construct an instance of the skill table that contains a violation of the table's key constraint - or explain why it is impossible to construct such an instance

The below is part of the skill table:

|  |  |  |
| --- | --- | --- |
| **ssn** | **prod\_id** | **manuf\_co** |
| 777-00-0005 | FA-S/1 | 333-44-5555 |
| 777-00-0005 | FA-S/2 | 333-44-5555 |
| 777-00-0006 | FA-L/1 | 333-44-5555 |

Violation reason: Because (ssn, prod\_id, manuf\_co) is a key of table schema SKILL means that an *instance* of SKILL is a *legal* instance only if no two rows agree on the values of all the attributes in (ssn,prod\_id,manuf\_co).

b) (3 points) Construct an instance of the product table that contains a violation of the table's key constraint - or explain why it is impossible to construct such an instance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **product\_id** | **manuf\_co** | **manuf\_div** | **loc** | **prod\_descr** |
| FA-S/1 | 333-44-5555 | front axle | Valdosta | small light-weight front axle |
| FA-S/1 | 333-44-5555 | front axle | Valdosta | small heavy weight front axle |

Violation reason: (prod\_id, manuf\_co) is a key of table schema PRODUCT means that an *instance* of PRODUCT is a *legal* instance only if no two rows agree on the values of all the attributes in (prod\_id,manuf\_co).

c) (4 points) Construct instances of both the skill and product tables that contain violations of all the relevant foreign key reference constraints -- or explain why it is impossible to construct such instances

Skill table:

|  |  |  |
| --- | --- | --- |
| **ssn** | **prod\_id** | **manuf\_co** |
| 777-00-0005 | FA-S/1 | 333-44-5555 |

Product table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **product\_id** | **manuf\_co** | **manuf\_div** | **loc** | **prod\_descr** |
| FA-L/2 | 333-44-5555 | front axle | New York | small light-weight front axle |
| FA-M/3 | 333-44-5555 | front axle | Valdosta | small heavy weight front axle |

Violation: The prod\_id, manuf\_co of SKILL table is a foreign key referring to prod\_id, manuf\_co in PRODUCT. But the prod\_id(FA-S/1) in skill table does not exist in product table, which means that the foreign key is conficted.

In all cases in which there is a violation, specify in which row the violation is to be found and why it’s a violation.

2. (10 points) Write the following query in the pseudo-code used in the lectures:

Find every division (of any company) that has at least two subdivisions

for i = 1 to no\_rows(division) do

for j = 1 to no\_rows(division) do

if (division [i][subdiv\_of] = division [j][subdiv\_of]

and division [i][subdiv\_of] <> null

and division [i][co\_id] = division [j][co\_id])

and division [i][div\_name] <> division [j][div\_name])

then output(division [i][co\_id]],division [i][subdiv\_of]])

3. (10 points) Write the following query in unextended relational algebra:

Find every person who works for at least two different divisions of the same company

Text

Description automatically generated

4. (10 points) Write the following query in extended relational algebra:

Find the number of products manufactured by the company with govt\_id 123-45-6789

**The purpose of questions 5-8 is to test your knowledge of SQL operations that were discussed in the lectures. So, using operations, for example, JOIN and its variations is not permitted, and will not get credit.**

5. (10 points) Write the following query in SQL without any embedded sub-queries:

Find every person who works for a company that has at least two divisions each of which has at least two sites.

SELECT DISTINCT works\_for.ssn

FROM division d1, division d2, site s1, site s2, site s3, site s4, works\_for

WHERE d1.co\_id = d2.co\_id

AND d1.div\_name != d2.div\_name

AND s1.loc != s2.loc

AND s3.loc != s4.loc

AND d1.div\_name = s1.div\_name AND d1.co\_id = s1.co\_id

AND d1.div\_name = s2.div\_name AND d1.co\_id = s2.co\_id

AND d2.div\_name = s3.div\_name AND d2.co\_id = s3.co\_id

AND d2.div\_name = s4.div\_name AND d2.co\_id = s4.co\_id

AND works\_for.co\_id = d1.co\_id;

6. Consider the following query:

Find every company that employs at least one person who has no skills at any of the products that the company manufactures.

a) (10 points) Write the query in SQL using an embedded query, but without using any of the embedding operators (IN, NOT IN, EXISTS, NOT EXISTS).

SELECT company.govt\_id

FROM company c

WHERE company.govt\_id IN (

SELECT works\_for.co\_id

FROM works\_for wf

WHERE (

SELECT COUNT(\*)

FROM skill sk

JOIN product pr ON skill.prod\_id = product.prod\_id

WHERE sk.ssn = wf.ssn AND product.manuf\_co = works\_for.co\_id

) = 0

);

b) (10 points) Write the query in SQL with an embedded query that uses either IN or NOT IN, whichever is appropriate. (You may not use EXISTS or NOT EXISTS).

SELECT company.govt\_id

FROM company c

WHERE company.govt\_id IN (

SELECT DISTINCT works\_for.co\_id

FROM works\_for wf

WHERE works\_for.ssn NOT IN (

SELECT skill.ssn

FROM skill sk

JOIN product pr ON skill.prod\_id = product.prod\_id

WHERE product.manuf\_co = works\_for.co\_id

)

);

 c) (10 points) Write the query in SQL using an embedded query that uses either EXISTS or NOT EXISTS, whichever is appropriate. (You may not use IN or NOT IN).

SELECT company.govt\_id

FROM company c

WHERE EXISTS (

SELECT 1

FROM works\_for wf

WHERE company.govt\_id = works\_for.co\_id

AND NOT EXISTS (

SELECT 1

FROM skill sk

JOIN product pr ON skill.prod\_id = product.prod\_id

WHERE skill.ssn = works\_for.ssn AND product.manuf\_co = works\_for.co\_id

)

);

7. (10 points) Write the following query in SQL using both a GROUP BY and a HAVING clause. (You may not use any of the following operators: IN, NOT IN, EXISTS, NOT EXISTS, <rel op> ALL, <rel op> ANY, <rel op SOME>)

For each division (of any company) that employs at least 200 people, find the average salary of all the people the division employs.

SELECT works\_for. co\_id, works\_for. div\_name, AVG (works\_for.salary)

FROM works\_for, division

WHERE (works\_for. co\_id = division. co\_id)

AND (works\_for. div\_name = division. div\_name)

GROUP BY works\_for. co\_id, works\_for. div\_name

HAVING COUNT(DISTINCT works\_for. emp\_id) >= 200

8. (10 points) Write the query of Question 7 in SQL using a GROUP BY but no HAVING clause. (You may not use any of the following operators: IN, NOT IN, EXISTS, NOT EXISTS, <rel op> ALL, <rel op> ANY, <rel op SOME>)

SELECT division.co\_id, division.div\_name, AVG(works\_for.salary)

FROM division, works\_for

WHERE (SELECT COUNT(works\_for.emp\_id)

FROM works\_for

WHERE division.div\_name = works\_for.div\_name

AND division.co\_id = works\_for.co\_id) >= 200

AND division.div\_name = works\_for.div\_name

AND division.co\_id = works\_for.co\_id

GROUP BY division.co\_id, division.div\_name