COMP 5120/6120 Database Systems I

Fall 2024
Homework #2
Due: 10/7/2024

1. What is a foreign key constraint? Why are such constraints important? What is referential integrity? (10 pts)

   i A foreign key constraint is a rule in a relational database that establishes a relationship between two relations based on a value (the "foreign key") in their columns. The foreign key in the "referencing relation" must match the primary key in the "referenced relation"; this means they must have the same number of columns and compatible data types, but the column names can be different. Foreign key constraints are important because they enforce referential integrity, and help ensure the accuracy and consistency of the data within the database. Referential integrity means that every value in the foreign key column of the referencing table must exist as a corresponding value in the primary key column of the referenced table. This prevents orphaned records and ensures that relationships between tables remain valid.

2. Explain the difference between external, internal, and conceptual schemas. How are these different schema layers related to the concepts of logical and physical data independence? (10 pts)

   i A conceptual schema describes the stored data in terms of the data model of the DBMS, it describes all relations that are stored in the database, it abstracts away how the data is physically stored and focuses on the logical structure. An external schema allows data access to be customized or authorized at the level of individual users or groups, multiple external schemas can exist for the same conceptual schema, giving different users different perspectives. An internal schema, also known as a physical schema, describes how the data is physically stored and organized within the storage medium of the database.

   ii Physical data independence refers to the ability to modify the internal schema while not affecting the conceptual or external schemas. For example, rearranging the files or changing indices should not require changes to the conceptual or external schemas (logical structure or authorizations)

   iii Logical data independence refers to the ability to modify the conceptual schema without affecting the external schemas. For example, adding new attributes in tables should not affect the parts of which authorized users are able to see.

3. Consider the following schema:

   Suppliers (<u>sid: integer</u>, sname: string, address: string)
   Parts (<u>pid: integer</u>, pname: string, color: string)
   Catalog (<u>sid: integer, pid: integer</u>, cost: real)

   The Catalog relation lists the prices charged for parts by suppliers. Write the following queries in **SQL** (40 pts):

   NOTE: I'm sorry about the spacing here, word would not allow me to change it for some reason

   a.) Find the pnames of parts for which there is some supplier.

   SELECT DISTINCT P.pname

   FROM Parts P, Catalog C

   WHERE P.pid = C.pid

   b.) For each part, find the sname of the supplier who charges the most for that part.

   SELECT P.pid, S.sname

   FROM Parts P, Suppliers S, Catalog C

   WHERE C.pid = P.pid

   AND C.sid = S.sid

   AND C.cost = (SELECT MAX(C1.cost)

   FROM Catalog C1

   WHERE C1.pid = P.pid)

   c.) Find the sids of suppliers who supply only red parts.

   SELECT DISTINCT C.sid

   FROM Catalog C

   WHERE NOT EXISTS (SELECT *

   FROM Parts P

```sql
        WHERE P.pid = C.pid AND P.color <> 'Red')
```

d.) Find the snames of suppliers who supply every part.

```sql
SELECT S.sname

FROM Suppliers S

WHERE NOT EXISTS (( SELECT P.pid

            FROM Parts P)

      EXCEPT

      (SELECT C.pid

      FROM Catalog C

      WHERE C.sid = S.sid))
```

4. Consider the following schema:

   Employee (person-name, street, city)
   Works (person-name, company-name, salary)
   Company (company-name, city)
   Manages (person-name, manager-name)

   Write the following queries in **relational algebra** (40 pts):

a.) Find the names of all employees who work for Auburn Bank.

$\pi_{\text{person-name}}(\sigma_{\text{company-name='Auburn Bank'}}(\text{Works}))$

b.) Find the names and cities of residence of all employees who work for Auburn Bank.

$\pi_{\text{person-name, city}}((\sigma_{\text{company-name='Auburn Bank'}}(\text{Works})) \bowtie \text{Employee})$

c.) Find the names, street address, and cities of residence of all employees who work for Auburn Bank and earn more than \$50,000 per year.

$\pi_{\text{person-name, street, city}}((\sigma_{\text{company-name='Auburn Bank'} \wedge \text{salary}>50000}(\text{Works})) \bowtie \text{Employee})$

d.) Find the names of all employees in this database who live in the same city as the company for which they work.

$\pi_{\text{person-name}}(\sigma_{\text{Employee.city=Company.city}}(\text{Employee} \bowtie \text{Works} \bowtie \text{Company})$