**Kate Moreland**

**Kem0149**

**904239271**

**COMP 5120**

**Homework 2**

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

The foreign key constraint is a rule that will connect a table’s column with the primary key of another table. This makes sure that values match real values, otherwise it is null. This will keep the relationship between tables up and running. Referential integrity is when the database will make sure that the foreign key will always reference a real primary key value.

1. 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)

The external schema is called the view level. This shows how each person will see the data. This is simplifying the view and putting up restrictions on access. The conceptual schema shows the entire logic of the database. This will define entities, relationships, and constraints. The internal schema is the physical level. This will show how the data is stored. When there are changes to the internal schema this demonstrates physical data independence because the schema will remain constant even if any part of the physical storage is altered. Logical data independence is shown when something is added to the conceptual schema and nothing is altered in the external schema.

1. Consider the following schema:

Suppliers (sid: integer, sname: string, address: string)

Parts (pid: integer, pname: string, color: string)

Catalog (sid: integer, pid: integer, cost: real)

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

* 1. Find the pnames of parts for which there is some supplier.

SELECT DISTINCT Parts.pname

FROM Parts, Catalog

WHERE Parts.pid = Catalog.pid;

* 1. For each part, find the sname of the supplier who charges the most for that part.

SELECT Parts.pname, Suppliers.sname

FROM Catalog

JOIN Suppliers ON Catalog.sid = Suppliers.sid

JOIN Parts ON Catalog.pid = Parts.pid

WHERE (Catalog.pid, Catalog.cost) IN (

SELECT Catalog.pid, MAX(Catalog.cost)

FROM Catalog

GROUP BY Catalog.pid

);

* 1. Find the sids of suppliers who supply only red parts.

SELECT Catalog.sid

FROM Catalog

JOIN Parts ON Catalog.pid = Parts.pid

GROUP BY Catalog.sid

HAVING COUNT(\*) = SUM(CASE WHEN Parts.color = 'red' THEN 1 ELSE 0 END);

* 1. Find the snames of suppliers who supply every part.

SELECT Suppliers.sname

FROM Suppliers

WHERE NOT EXISTS (

SELECT Parts.pid

FROM Parts

WHERE NOT EXISTS (

SELECT \*

FROM Catalog

WHERE Catalog.sid = Suppliers.sid

AND Catalog.pid = Parts.pid

)

);

1. 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):

* + 1. Find the names of all employees who work for Auburn Bank.

π\_person-name (σ\_company-name = ‘Auburn Bank’ (Works))

* + 1. Find the names and cities of residence of all employees who work for Auburn Bank.

π\_person-name, city (

σ\_company-name = ‘Auburn Bank’ (Works ⨝ Employee)

)

* + 1. 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.

π\_person-name, street, city (

σ\_company-name = ‘Auburn Bank’ ˄ salary > 50000 (Works ⨝ Employee)

)

* + 1. Find the names of all employees in this database who live in the same city as the company for which they work.

π\_person-name (

σ\_Employee.city = Company.city (Employee ⨝ Works ⨝ Company)

)