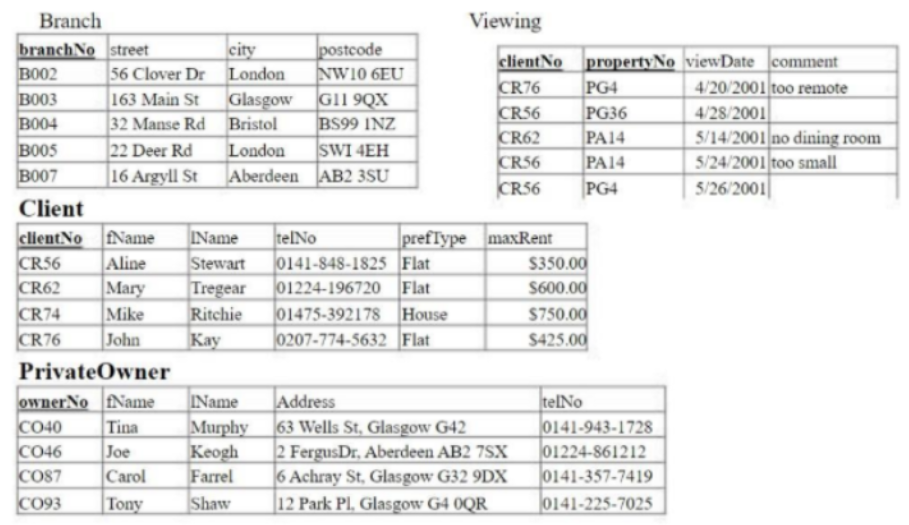
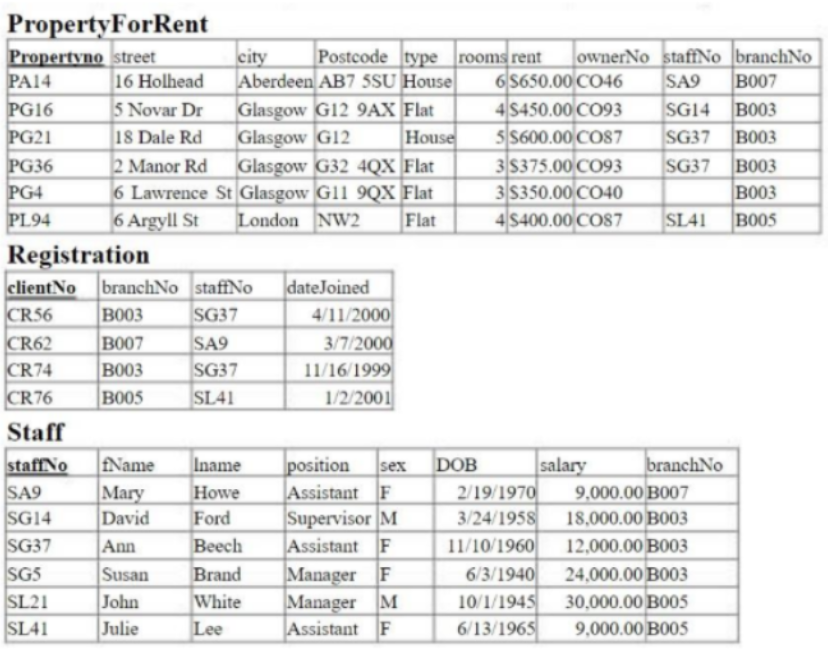




**Question:**





**Using the given tables, construct relational algebra expressions to answer the following queries and represent the answer as a relation:**

1. First and last names of the clients registered at Branch B003.

* **Temp = σ branchNo='B003' (Registration)**
* **Temp ⨝ Client** -> Natural join occurs on the common attribute clientNo
* **Result = π fName, lName (Temp)**

2. Which staff members were born after 1955?

* **σ DOB >= '1956-1-1' (Staff)**

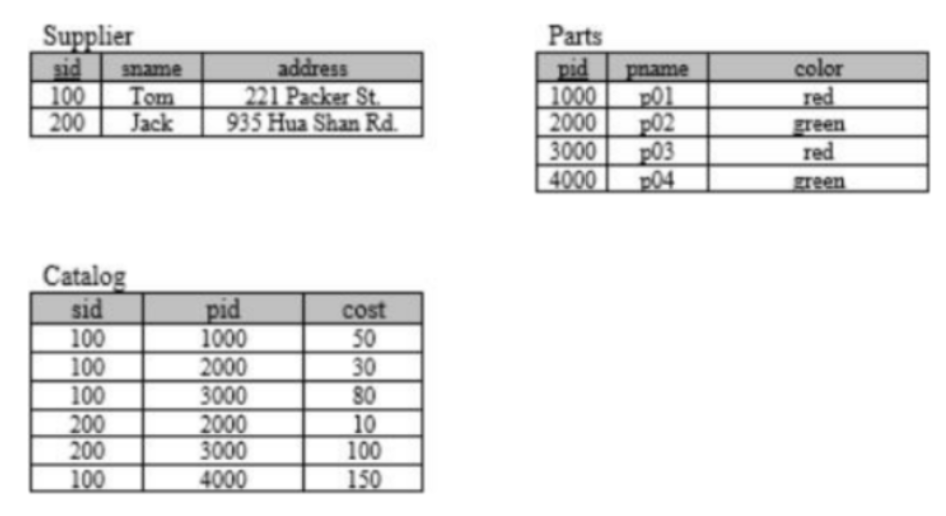
3. List the flats with rent less than $500.

* **π propertyNo, street, city, rent (σ type='Flat' ∧/AND rent < 500 (PropertyForRent))**

4. Telephone numbers of the owners who have property at London

* **Temp = σ city='London' (PropertyForRent)**
* **CommonRows = Temp ⨝ PrivateOwner** -> Performs a natural join on ownerNo (common attribute)
* **Result = π telNo (CommonRows)** OR **Result = π telNo (Temp)** -> Both are correct

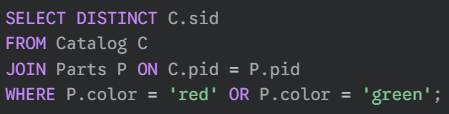
**Question:**



**The key fields are underlined, and the domain of each field is listed after the field name. Thus sid is the key for Suppliers, pid is the key for Parts, and sid and pid together form the key for Catalog. The Catalog relation lists the prices charged for parts by Suppliers.**

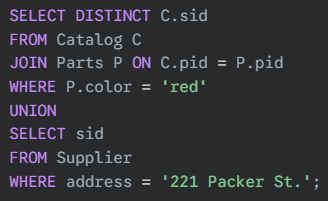
**Write the following queries in sql and then convert it into relational algebra:**

a) Find the sids of suppliers who supply some red **or** green part.



* **RedGreen\_Supplier = σ color='red' V/OR color='green' (Parts)**
* **Temp = Catalog ⨝ RedGreen\_Supplier**
* **Result = π sid (Temp)**

b) Find the sids of suppliers who supply some red part **or** are at 221 Packer Ave.



* **Red\_Part = σ color='red' (Parts)**
* **Packer\_Sup = σ address='221 Packer St.' (Supplier)**
* **Red\_Sup = Catalog ⨝ Red\_Part**
* **Red\_Sup\_Sid = π sid (Red\_Sup)**
* **Packer\_Sup\_Sid = π sid (Packer\_Sup)**
* **Result = Red\_Sup\_Sid ∪ Packer\_Sup\_Sid**

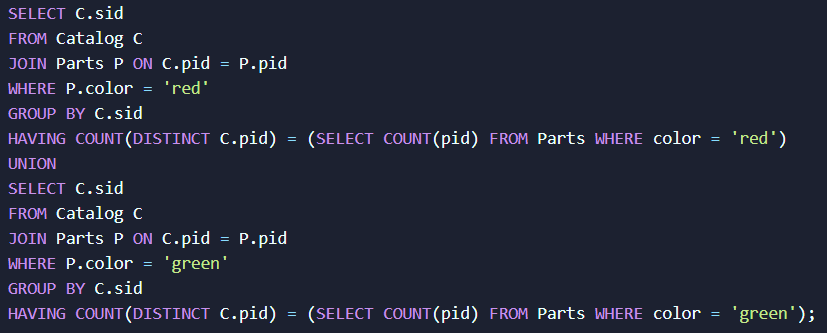
c) Find the sids of suppliers who supply **every** part.

A black background with white text

AI-generated content may be incorrect.

* **AllParts ← π pid (Parts)**
* **Temp ← π sid, pid (Catalog)**
* **Result ← Temp ÷ AllParts**

d) Find the sids of suppliers who supply every red part **or** supply every green part.

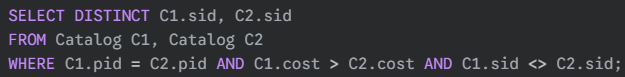


* **All\_Red\_Parts = σ color = 'red' (Parts)**
* **All\_Red\_Parts\_Pid = π pid (All\_Red\_Parts)**
* **All\_Green \_Parts = σ color = 'green' (Parts)**
* **All\_Green \_Parts\_Pid = π pid (All\_Green \_Parts)**
* **Temp = π sid, pid (Catalog)**
* **Result1 = Temp ÷ All\_Red\_Parts\_Pid**
* **Result2 = Temp ÷ All\_Green \_Parts\_Pid**
* **Result = Result1 ∪ Result2**

OR

* **Temp1 ← πpid (σcolor='red' (Parts))**
* **Temp2 ← πpid (σcolor='green' (Parts))**
* **Temp3 ← πsid, pid (Catalog) ÷ Temp1**
* **Temp4 ← πsid, pid (Catalog) ÷ Temp2**
* **Result ← Temp3 ∪ Temp4**

e) Find the pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second.



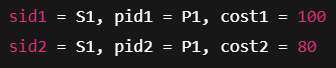
* **C1 ← ρ sid1/sid, pid1/pid, cost1/cost (Catalog)**
* **C2 ← ρ sid2/sid, pid2/pid, cost2/cost (Catalog)**
* **Temp1 ← C1 ⨝ C2**
* **Temp2 ← σ pid1 = pid2 ∧/AND cost1 > cost2 ∧/AND sid1 ≠ sid2 (Temp1)**
* **Result ← π sid1, sid2 (Temp2)**

Temp1 ← C1 ⨝ C2 is a **Cartesian product join** (cross join), producing **every possible pair** of supplier-part entries from C1 and C2.

So Temp1 contains:

* (sid1, pid1, cost1, sid2, pid2, cost2)  
  → All combinations of Catalog rows from both copies.

Without sid1 ≠ sid2 in Temp2 ← σ pid1 = pid2 ∧ cost1 > cost2 (Temp1)



But this isn't realistic — a **supplier shouldn't charge two different prices for the same part** in the same Catalog.

Result ← π sid1, sid2 (Temp2)

Finally, project only the supplier IDs:

* sid1: the supplier who charges **more**
* sid2: the supplier who charges **less**

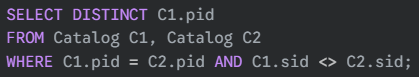
**All supplier pairs such that sid1 charges more than sid2 for the same part.**

f) Find the pids of parts that are supplied by at least two different suppliers.

A blue and white text

AI-generated content may be incorrect.

* **Temp1 = pid g count(distinct sid) as total\_supp (Catalog)**
* **Temp2 = σ total\_supp >= 2 (Temp1)**
* **Result = π pid (Temp2)**



* **C1 ← ρ sid1/sid, pid1/pid (Catalog)**
* **C2 ← ρ sid2/sid, pid2/pid (Catalog)**
* **Temp1 ← C1 ⨝ C2**
* **Temp2 ← σ pid1 = pid2 ∧ sid1 <> sid2 (Temp1)**
* **Result ← π pid1 (Temp2)**

g) Find the pids of parts supplied by every supplier at less than $200. (If any supplier either does not supply the part or charges more than $200 for it, the part is not selected)

A black background with white text

AI-generated content may be incorrect.

* **Temp1 ← σ cost < 200 (Catalog)**
* **Temp2 ← π sid, pid (Temp1)**
* **Temp3 ← π sid (Supplier)**
* **Temp4 ← Temp2 ÷ Temp3**
* **Result ← π pid (Temp4)**

**A diagram of a project operation

AI-generated content may be incorrect.**

**A diagram of a math problem

AI-generated content may be incorrect.**

* **C1 ← ρ sid1/sid, pid1/pid, cost1/cost (Catalog)**
* **C2 ← ρ sid2/sid, pid2/pid, cost2/cost (Catalog)**
* **Temp ← C1 ⨝ C2**
* **Inner join on self generates cartesian product. Every row is related with every other row.**

**A diagram of a number and a number

AI-generated content may be incorrect.**

**A diagram of a diagram

AI-generated content may be incorrect.**

* **Combination of Tuples:** For every single row in the first relation (R), it is paired with every single row in the second relation (S).
* **Number of Columns:** If R has n attributes and S has m attributes, then R×S will have n+m attributes.
* **Number of Rows:** If R has x rows and S has y rows, then R×S will have x×y rows.

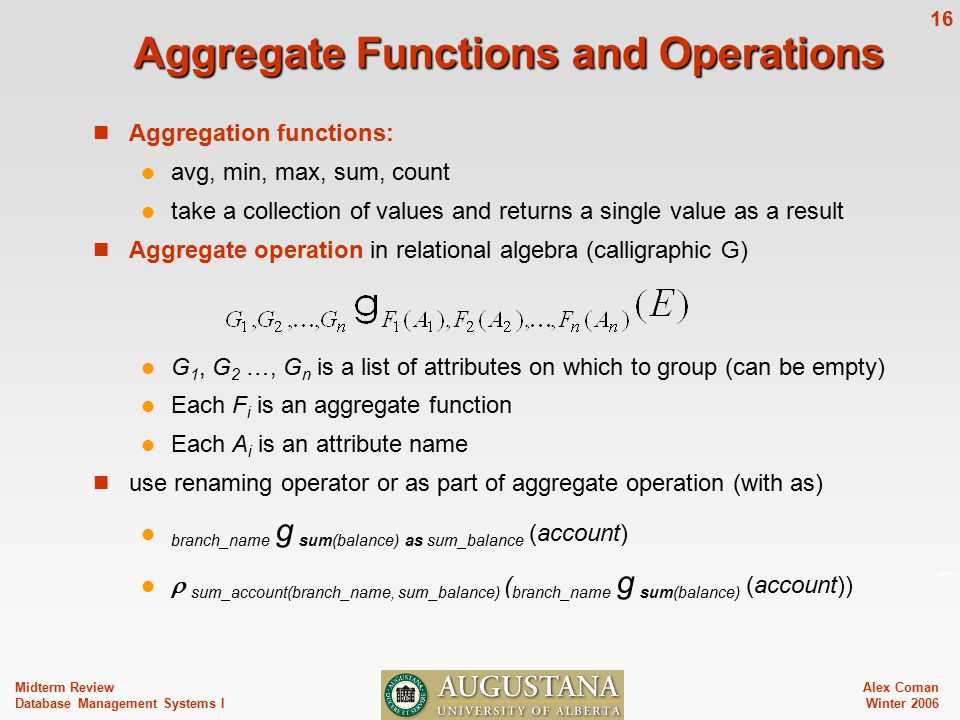
**A diagram of a mathematical equation

AI-generated content may be incorrect.**

1. **Gamma (G) is for Aggregation and Grouping:** It operates on a relation to produce summary values, potentially grouped by certain attributes.
2. **Sigma (σ) is for Tuple Selection (Row Filtering):** It filters individual rows *before* any aggregation happens.
3. We do not use **sigma** and **gamma** together in a single query statement. When you use the **gamma (G) symbol,** any columns listed before the **aggregate functions** are considered **grouping attributes.**

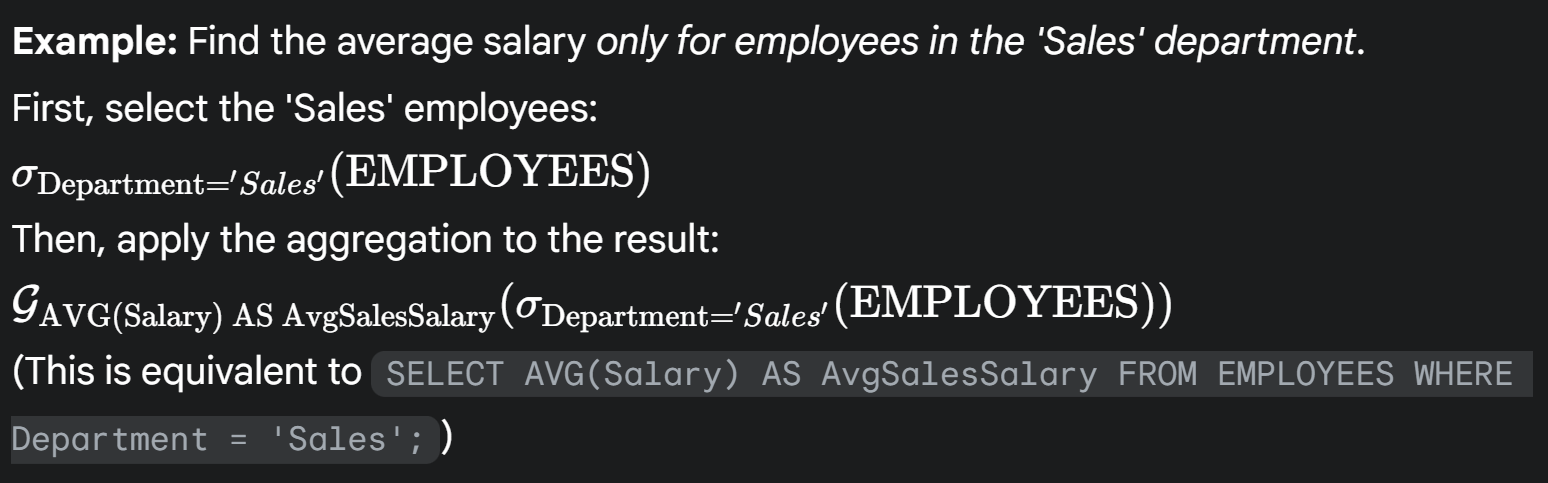
**A black and white text on a white background

AI-generated content may be incorrect.**



**A white paper with black text

AI-generated content may be incorrect.**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

* Department is the **grouping attribute**. It's written **before** the gamma symbol and no aggregate function is applied to it.
* The EMPLOYEES relation is logically grouped by unique Department values.
* The output relation will have two columns: **Department** and **AverageSalaryPerDept**. For each unique Department value, there will be a single row showing that department's name and the calculated average salary for employees in that department.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.