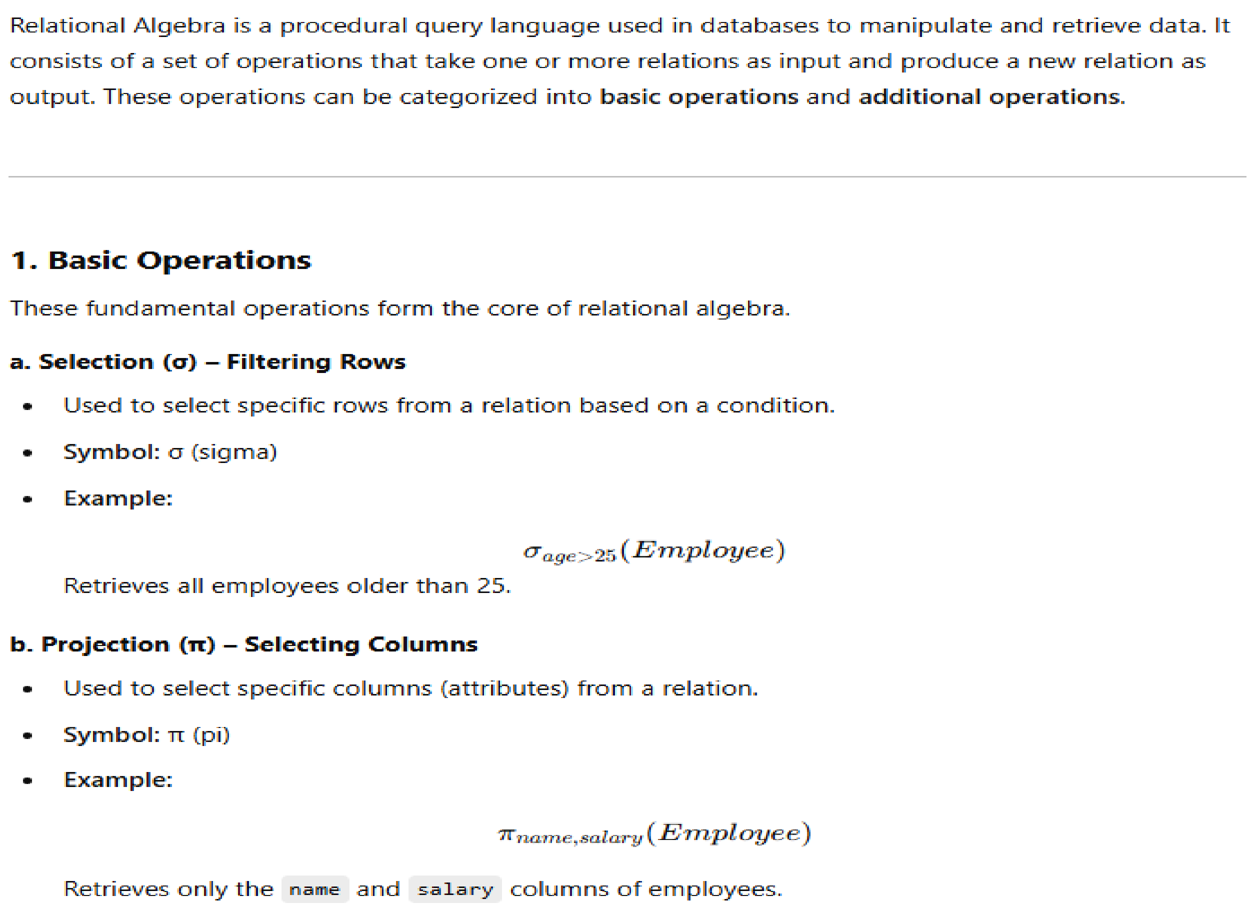
Relational Algebra –

Example: https://www.exploredatabase.com/2020/04/relational-algebra-in-database-solved-exercise.html



Relational Algebra is a procedural query language used in databases to manipulate and retrieve data. It consists of a set of operations that take one or more relations as input and produce a new relation as output. These operations can be categorized into **basic operations** and **additional operations**.

---

1. Basic Operations

These fundamental operations form the core of relational algebra.

a. Selection (σ) – Filtering Rows

- Used to select specific rows from a relation based on a condition.
- **Symbol:** σ (sigma)
- **Example:**

$$\sigma_{age>25}(Employee)$$

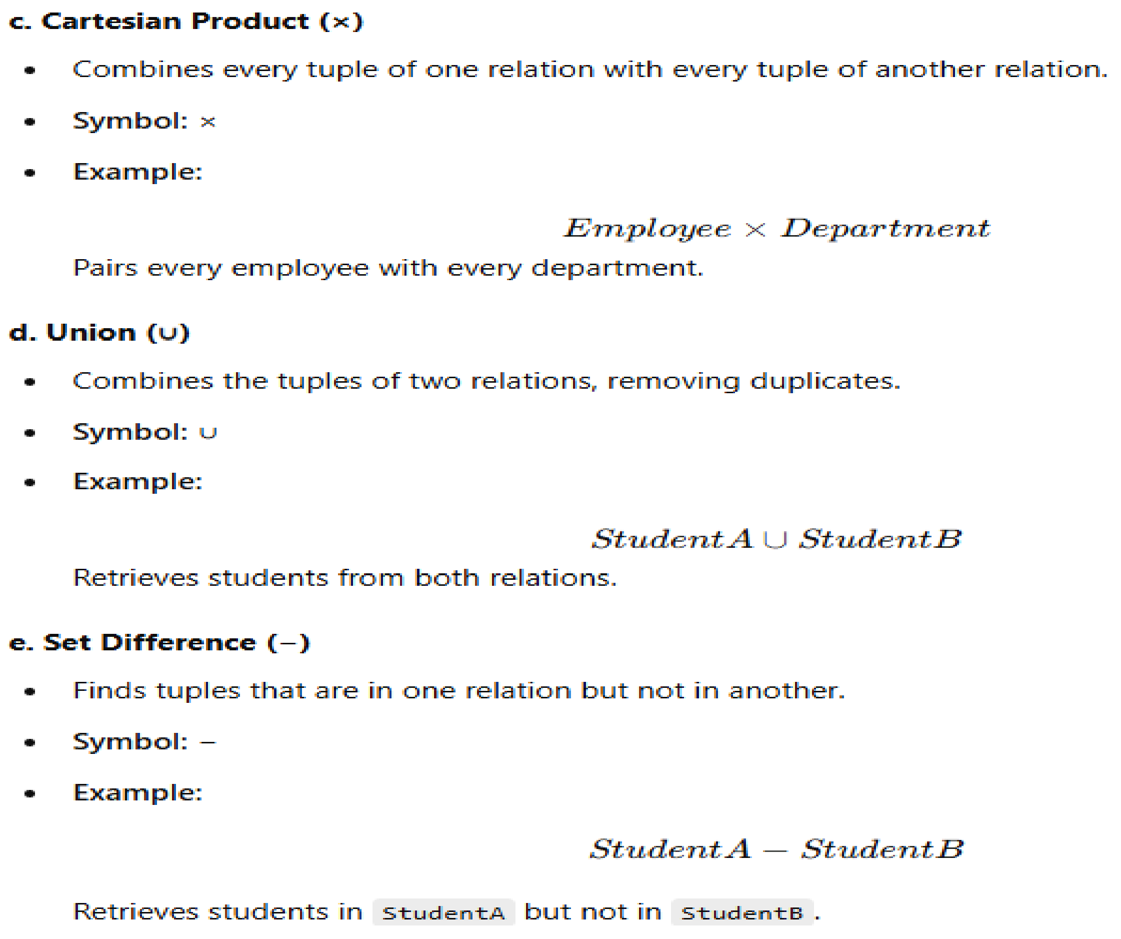
Retrieves all employees older than 25.

b. Projection (π) – Selecting Columns

- Used to select specific columns (attributes) from a relation.
- **Symbol:** π (pi)
- **Example:**

$$\pi_{name,salary}(Employee)$$

Retrieves only the `name` and `salary` columns of employees.



c. Cartesian Product (×)

- Combines every tuple of one relation with every tuple of another relation.
- **Symbol:** ×
- **Example:**

$$Employee \times Department$$

Pairs every employee with every department.

d. Union (∪)

- Combines the tuples of two relations, removing duplicates.
- **Symbol:** ∪
- **Example:**

$$StudentA \cup StudentB$$

Retrieves students from both relations.

e. Set Difference (−)

- Finds tuples that are in one relation but not in another.
- **Symbol:** −
- **Example:**

$$StudentA - StudentB$$

Retrieves students in `StudentA` but not in `StudentB`.

2. Additional Operations

These operations are derived from basic ones but enhance query capabilities.

a. Join (⋈)

- Combines tuples from two relations based on a related attribute.
- Types of Join:
  - Theta Join (⋈ condition): Uses a condition (e.g., `=`, `>`, `<`).
  - Equi-Join: Special case of Theta Join where condition is `=`.
  - Natural Join (⋈): Implicit equi-join on common attributes.

  Example (Natural Join):

  $$Employee \bowtie Department$$

  Combines employees with their respective departments.

b. Division (÷)

- Used to find tuples in one relation that are associated with all tuples in another.
- Example:

  $$A \div B$$

  Retrieves students who have taken all courses.

c. Aggregation Functions

- Used for calculations like COUNT, SUM, AVG, MIN, MAX.
- Example:

  $$\gamma_{dept, AVG(salary)}(Employee)$$

  Finds the average salary per department.

Scenario 1: University Faculty and Courses Database

Relations:

- **Faculty(FID, FName, Department, Age, Gender):** Stores information about faculty members.
- **Courses(CID, CName, Department, Credits, InstructorID):** Details of courses, including which faculty member is teaching the course (InstructorID refers to Faculty).
- **Teaches(FID, CID):** Links faculty members to the courses they teach.

Queries:

1. Retrieve the names of all faculty members in the "Mathematics" department.
   Relational Algebra:
   $\pi_{\text{FName}}(\sigma_{\text{Department}='Mathematics'}(\text{Faculty}))$
2. Find the names of all courses taught by "Dr. John Doe."
   Relational Algebra:
   $\pi_{\text{CName}}((\sigma_{\text{FName}='Dr.JohnDoe'}(\text{Faculty})) \bowtie_{\text{FID}} \text{Teaches}) \bowtie_{\text{CID}} \text{Courses}$

Scenario 2: Library Management Database

Relations:

- **Books(BID, BName, Author, Publisher, Category):** Stores information about books in the library.
- **Members(MID, MName, Department, Age, MembershipDate):** Stores information about library members.
- **Borrowings(MID, BID, BorrowDate, ReturnDate):** Tracks which members have borrowed which books.

Queries:

1. Retrieve the names of all members who have borrowed books authored by "J.K. Rowling."
   Relational Algebra:
   $\pi_{\text{MName}}((\sigma_{\text{Author}='J.K.Rowling'}(\text{Books})) \bowtie_{\text{BID}} \text{Borrowings}) \bowtie_{\text{MID}} \text{Members}$
2. Find all books borrowed by members from the "Computer Science" department.
   Relational Algebra:
   $\pi_{\text{BName}}((\sigma_{\text{Department}='ComputerScience'}(\text{Members})) \bowtie_{\text{MID}} \text{Borrowings}) \bowtie_{\text{BID}} \text{Books}$

Scenario 3: Hospital Patient Management

Relations:

- **Patients(PID, PName, Age, Gender, Disease, DoctorID)**: Stores information about patients.
- **Doctors(DID, DName, Specialization, Department)**: Stores information about doctors and their specializations.
- **Appointments(PID, DID, AppointmentDate)**: Tracks patient appointments with doctors.

Queries:

1. Retrieve the names of all patients who have appointments with doctors in the "Cardiology" department.
   Relational Algebra:
   $\pi_{\text{PName}}\left((\sigma_{\text{Department}='Cardiology'}(\text{Doctors})) \bowtie_{\text{DID}} \text{Appointments}\right) \bowtie_{\text{PID}} \text{Patients}$
2. List the names of doctors treating patients diagnosed with "Diabetes."
   Relational Algebra:
   $\pi_{\text{DName}}\left((\sigma_{\text{Disease}='Diabetes'}(\text{Patients})) \bowtie_{\text{PID}} \text{Appointments}\right) \bowtie_{\text{DID}} \text{Doctors}$

Scenario 4: Online Shopping System

Relations:

- **Customers(CID, CName, Age, Gender, City)**: Stores information about customers.
- **Products(PID, PName, Category, Price)**: Details about products in the online store.
- **Orders(CID, PID, OrderDate, Quantity)**: Tracks which customers ordered which products and the quantities.

Queries:

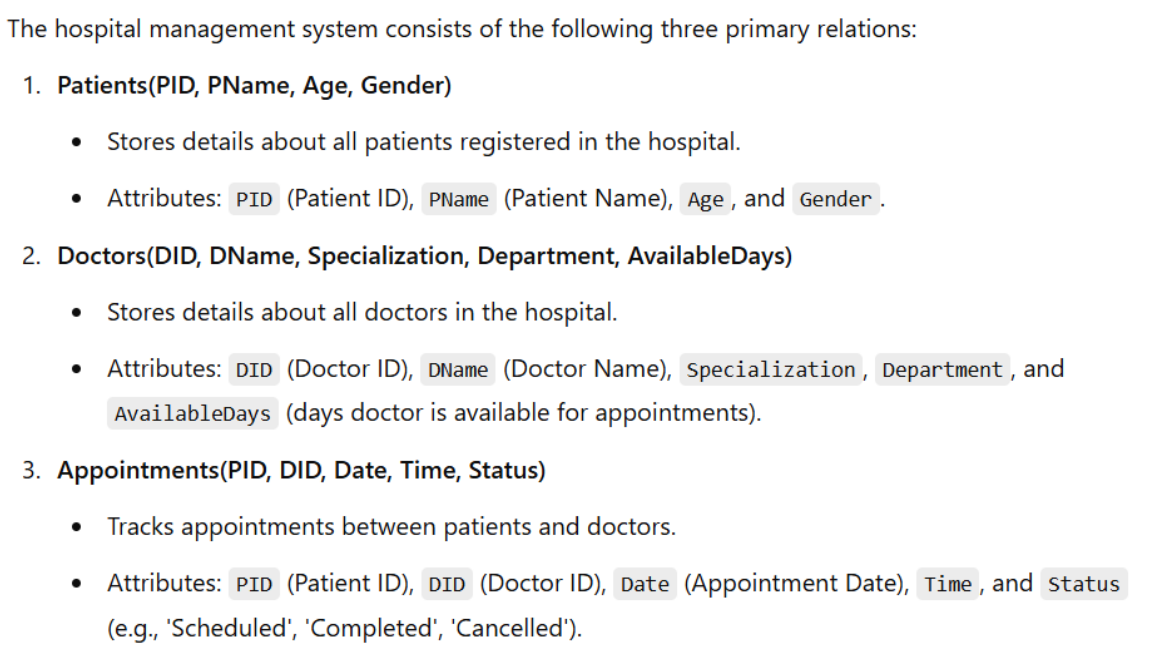
1. Retrieve the names of all customers who have purchased products in the "Electronics" category.
   Relational Algebra:
   $\pi_{\text{CName}}\left((\sigma_{\text{Category}='Electronics'}(\text{Products})) \bowtie_{\text{PID}} \text{Orders}\right) \bowtie_{\text{CID}} \text{Customers}$
2. Find all products purchased by customers from the city "New York."
   Relational Algebra:
   $\pi_{\text{PName}}\left((\sigma_{\text{City}='NewYork'}(\text{Customers})) \bowtie_{\text{CID}} \text{Orders}\right) \bowtie_{\text{PID}} \text{Products}$

Example:

Hospital Management System Scenario:

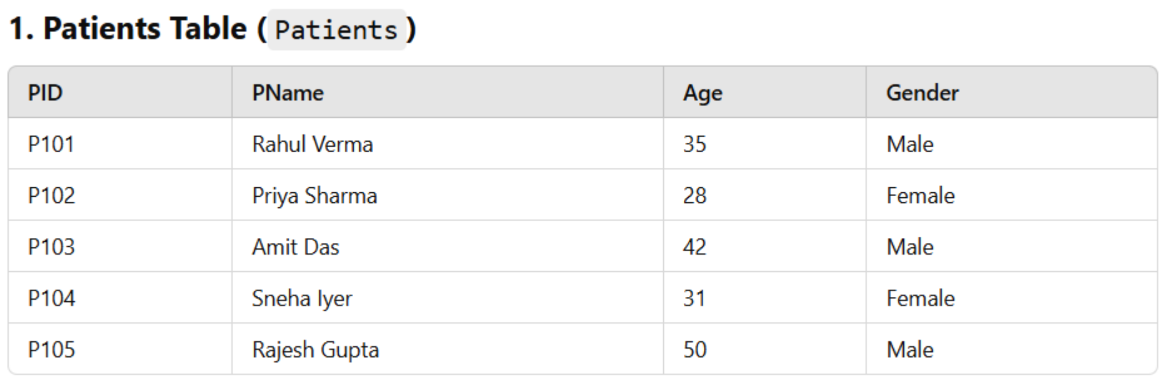
A hospital maintains a relational database to efficiently manage its operations, including patient records, doctor schedules, and appointment bookings. The hospital consists of multiple departments, each handling different specializations. Patients can book appointments with doctors, and their medical records are tracked for diagnosis and treatment history.

The hospital management system consists of the following three primary relations:



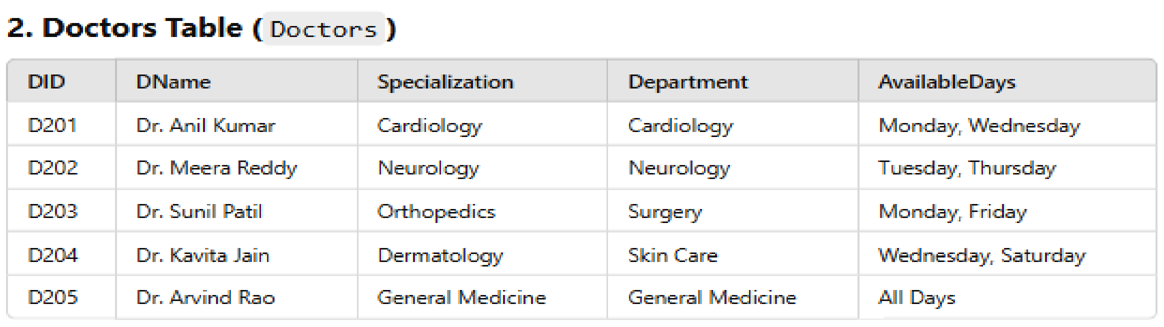
The hospital management system consists of the following three primary relations:

1. **Patients(PID, PName, Age, Gender)**
   - Stores details about all patients registered in the hospital.
   - Attributes: `PID` (Patient ID), `PName` (Patient Name), `Age`, and `Gender`.
2. **Doctors(DID, DName, Specialization, Department, AvailableDays)**
   - Stores details about all doctors in the hospital.
   - Attributes: `DID` (Doctor ID), `DName` (Doctor Name), `Specialization`, `Department`, and `AvailableDays` (days doctor is available for appointments).
3. **Appointments(PID, DID, Date, Time, Status)**
   - Tracks appointments between patients and doctors.
   - Attributes: `PID` (Patient ID), `DID` (Doctor ID), `Date` (Appointment Date), `Time`, and `Status` (e.g., 'Scheduled', 'Completed', 'Cancelled').



1. Patients Table (`Patients`)

PID	PName	Age	Gender
P101	Rahul Verma	35	Male
P102	Priya Sharma	28	Female
P103	Amit Das	42	Male
P104	Sneha Iyer	31	Female
P105	Rajesh Gupta	50	Male



2. Doctors Table (`Doctors`)

DID	DName	Specialization	Department	AvailableDays
D201	Dr. Anil Kumar	Cardiology	Cardiology	Monday, Wednesday
D202	Dr. Meera Reddy	Neurology	Neurology	Tuesday, Thursday
D203	Dr. Sunil Patil	Orthopedics	Surgery	Monday, Friday
D204	Dr. Kavita Jain	Dermatology	Skin Care	Wednesday, Saturday
D205	Dr. Arvind Rao	General Medicine	General Medicine	All Days

3. Appointments Table (`Appointments`)

PID	DID	Date	Time	Status
P101	D201	2025-03-10	10:00	Completed
P102	D202	2025-03-15	12:00	Scheduled
P103	D201	2025-03-10	11:00	Completed
P104	D204	2025-03-12	14:00	Completed
P105	D205	2025-03-13	09:00	Scheduled
P101	D202	2025-03-15	10:30	Scheduled
P102	D203	2025-03-10	15:00	Completed
P103	D202	2025-03-15	12:30	Scheduled
P104	D201	2025-03-14	16:00	Scheduled
P105	D204	2025-03-12	11:00	Completed

1. **Retrieve the names of all patients who have booked an appointment with a doctor specializing in 'Cardiology'.**
   - Step 1: Select all doctors with `Specialization = 'Cardiology'`.
   - Step 2: Join the result with `Appointments` on `DID` to find patients with appointments.
   - Step 3: Join with `Patients` using `PID` and project the patient names.

   Relational Algebra Expression:
   $\pi_{PName}(Patients \bowtie_{\text{Patients.PID} = \text{Appointments.PID}}$
   $(Appointments \bowtie_{\text{Appointments.DID} = \text{Doctors.DID} \wedge \text{Doctors.Specialization} = \text{'Cardiology'}} Doctors))$

---

2. **Find all patients who have booked an appointment in the 'Neurology' department.**
   - Step 1: Select all doctors in the `Neurology` department.
   - Step 2: Join the result with `Appointments` to find patient bookings.
   - Step 3: Join with `Patients` and project the patient names.

   Relational Algebra Expression:
   $\pi_{PName}(Patients \bowtie_{\text{Patients.PID} = \text{Appointments.PID}}$
   $(Appointments \bowtie_{\text{Appointments.DID} = \text{Doctors.DID} \wedge \text{Doctors.Department} = \text{'Neurology'}} Doctors))$

---

3. **List the names of all patients who booked appointments on a specific date (e.g., '2025-03-10').**
   - Step 1: Select all appointments with `Date = '2025-03-10'`.
   - Step 2: Join with `Patients` on `PID` to get patient names.

   Relational Algebra Expression:
   $\pi_{PName}(\sigma_{\text{Date} = \text{'2025-03-10'}}(Patients \bowtie Appointments))$

4. **Find the names of all doctors available on 'Monday'.**
   - Step 1: Select all doctors who are available on `Monday`.
   - Step 2: Project the `DName`.

   Relational Algebra Expression:
   $\pi_{DName}(\sigma_{\text{AvailableDays = 'Monday'}}(Doctors))$

---

5. **Retrieve the names of patients who have booked appointments with more than 2 different doctors.**
   - Step 1: Group `Appointments` by `PID` and count the number of distinct `DID`.
   - Step 2: Select patients with a count greater than 2.
   - Step 3: Join with `Patients` to retrieve the names.

   Relational Algebra Expression:
   $\pi_{PName}(\sigma_{\text{Count(DID)} > 2}(\gamma_{PID,PName;\text{Count(DID)}}(Appointments \bowtie Patients)))$

6. **Find the names of all patients who have at least one completed appointment.**
   - Step 1: Select all appointments where `Status = 'Completed'`.
   - Step 2: Join with `Patients` to get patient names.

   Relational Algebra Expression:
   $\pi_{PName}(\sigma_{\text{Status = 'Completed'}}(Patients \bowtie Appointments))$

---

7. **List all doctors who have at least one appointment scheduled for '2025-03-15'.**
   - Step 1: Select all appointments on `Date = '2025-03-15'`.
   - Step 2: Join with `Doctors` to get doctor names.

   Relational Algebra Expression:
   $\pi_{DName}(\sigma_{\text{Date = '2025-03-15'}}(Doctors \bowtie Appointments))$

---

8. **Retrieve the names of patients who have appointments with both a 'Cardiologist' and a 'Neurologist'.**
   - Step 1: Select `DID` of doctors where `Specialization = 'Cardiology'` and `Specialization = 'Neurology'`.
   - Step 2: Find `PID` s of patients who have booked appointments with both.
   - Step 3: Join with `Patients` to get their names.

   Relational Algebra Expression:
   $\pi_{PName}(Patients \bowtie_{\text{Patients.PID = A.PID}} ((\pi_{PID}(\sigma_{\text{Specialization = 'Cardiology'}}(Doctors \bowtie Appointments))) \cap (\pi_{PID}(\sigma_{\text{Specialization = 'Neurology'}}(Doctors \bowtie Appointments)))))$

9. **Find the names of doctors who have not scheduled any appointments.**
    - Step 1: Find all doctors in the `Doctors` table.
    - Step 2: Find all doctors who have at least one appointment in `Appointments`.
    - Step 3: Subtract to get doctors with no appointments.

    Relational Algebra Expression:

    $\pi_{DName}(Doctors) - \pi_{DName}(Doctors \bowtie Appointments)$

---

10. **Retrieve the names of patients who have booked more than one appointment on the same day.**

- Step 1: Group `Appointments` by `PID` and `Date`, counting occurrences.
- Step 2: Select patients with count greater than 1.
- Step 3: Join with `Patients` to get names.

Relational Algebra Expression:

$\pi_{PName}(\sigma_{\text{Count}(*) > 1}(\gamma_{PID, Date; \text{Count}(*)}(Appointments) \bowtie Patients))$