# Group 20 Project

## Bhavajna Kallakuri (202003046)

## Divyansh Jain (202003047)

## Lab 10 (Date: 29-10-2021)

## Bus Management System

## Queries:

**1.** Find number of seats available in each bus.

**Ans:**

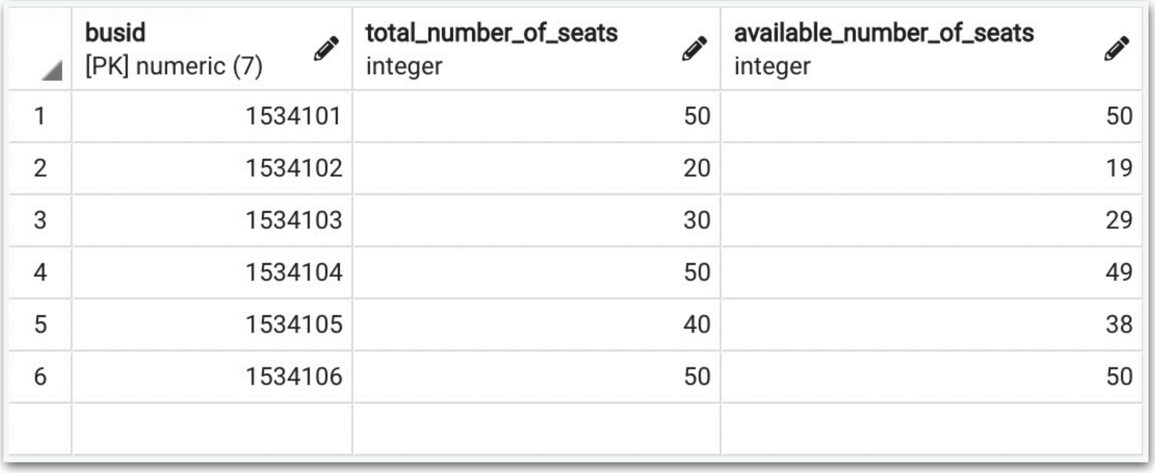
**Relation algebra Expression:**

$$\pi_{busid,total\_number\_of\_seats,\ available\_num\_of\_seats}(Bus)$$

**SQL Statements:**

```
SELECT busid,total_number_of_seats, available_number_of_seats

FROM Bus;
```

**Output:**



| | busid<br>[PK] numeric (7) | total_number_of_seats<br>integer | available_number_of_seats<br>integer |
|---|---|---|---|
| 1 | 1534101 | 50 | 50 |
| 2 | 1534102 | 20 | 19 |
| 3 | 1534103 | 30 | 29 |
| 4 | 1534104 | 50 | 49 |
| 5 | 1534105 | 40 | 38 |
| 6 | 1534106 | 50 | 50 |

**2.** List the details of all AC buses.

**Ans:**

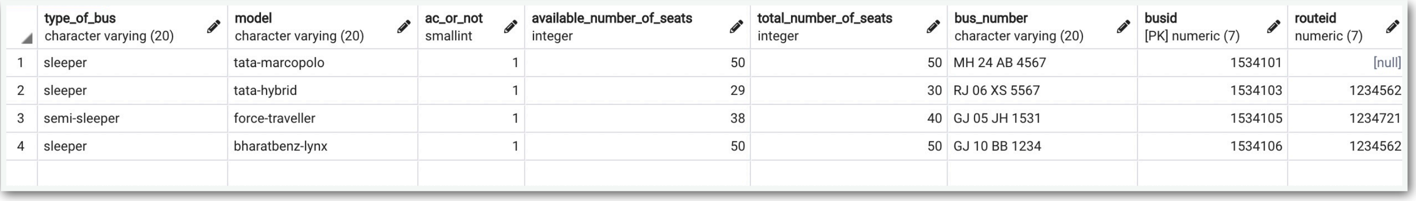
**Relation algebra Expression:**

$\sigma_{ac\_or\_not=1}(Bus)$

**SQL Statements:**

```
SELECT * FROM Bus

WHERE ac_or_not = 1;
```

**Output:**



| | type_of_bus<br>character varying (20) | model<br>character varying (20) | ac_or_not<br>smallint | available_number_of_seats<br>integer | total_number_of_seats<br>integer | bus_number<br>character varying (20) | busid<br>[PK] numeric (7) | routeid<br>numeric (7) |
|---|---|---|---|---|---|---|---|---|
| 1 | sleeper | tata-marcopolo | 1 | 50 | 50 | MH 24 AB 4567 | 1534101 | [null] |
| 2 | sleeper | tata-hybrid | 1 | 29 | 30 | RJ 06 XS 5567 | 1534103 | 1234562 |
| 3 | semi-sleeper | force-traveller | 1 | 38 | 40 | GJ 05 JH 1531 | 1534105 | 1234721 |
| 4 | sleeper | bharatbenz-lynx | 1 | 50 | 50 | GJ 10 BB 1234 | 1534106 | 1234562 |

**3.** Find the names of all customers or passengers along with their gender.

**Ans:**

**Relation algebra Expression:**

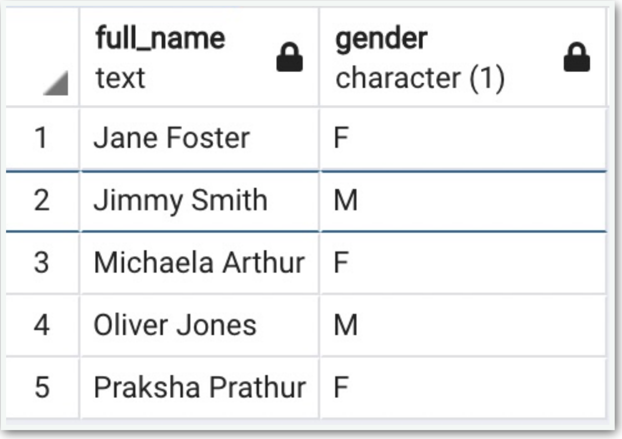
$\pi_{(\rho(full\_name, fname||lname), gender)}(Customer \bowtie User\_login)$

Here, natural join is used.

**SQL Statements:**

```
SELECT fname || ' ' || lname AS full_name,gender FROM Customer
NATURAL JOIN User_login

ORDER BY fname,lname;
```

**Output:**



| | full_name<br>text | gender<br>character (1) |
|---|---|---|
| 1 | Jane Foster | F |
| 2 | Jimmy Smith | M |
| 3 | Michaela Arthur | F |
| 4 | Oliver Jones | M |
| 5 | Praksha Prathur | F |

**4.** Display the BusId, Model and Bus number of bus which are available for a particular route.

**Ans:**

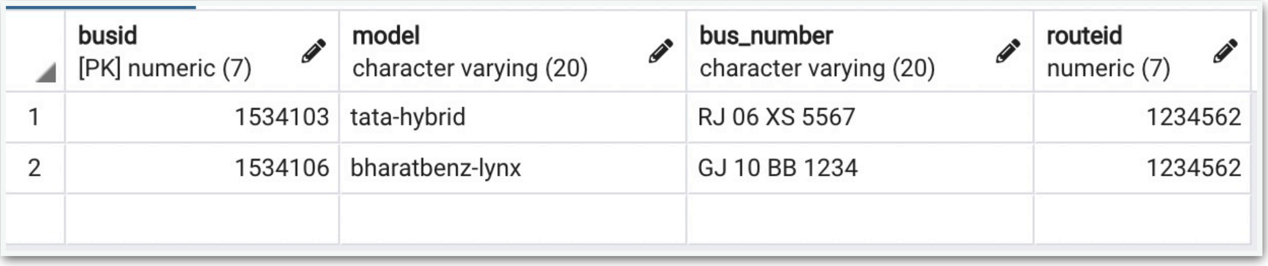
**Relation algebra Expression:**

$$\pi_{(busid,model,bus\_number,routeid)}(\sigma_{(routeid=1234562)}(Bus))$$

**SQL Statements:**

```
SELECT busid, model, bus_number,routeid

FROM Bus

WHERE routeid = 1234562;
```

**Output:**



| | busid<br>[PK] numeric (7) | model<br>character varying (20) | bus_number<br>character varying (20) | routeid<br>numeric (7) |
|---|---|---|---|---|
| 1 | 1534103 | tata-hybrid | RJ 06 XS 5567 | 1234562 |
| 2 | 1534106 | bharatbenz-lynx | GJ 10 BB 1234 | 1234562 |
| | | | | |

**5.** Display the name and info of all the Drivers.

**Ans:**

**Relation algebra Expression:**

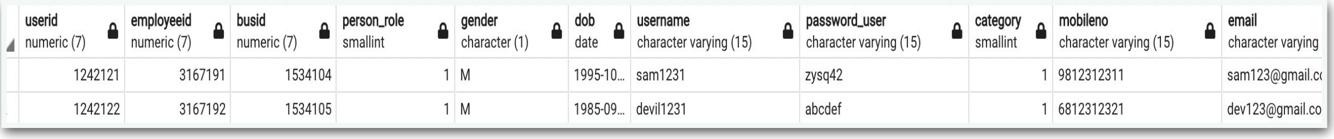
$$\sigma_{Person\_Role=1}(Employee \bowtie User\_login)$$

Here, natural join is used.

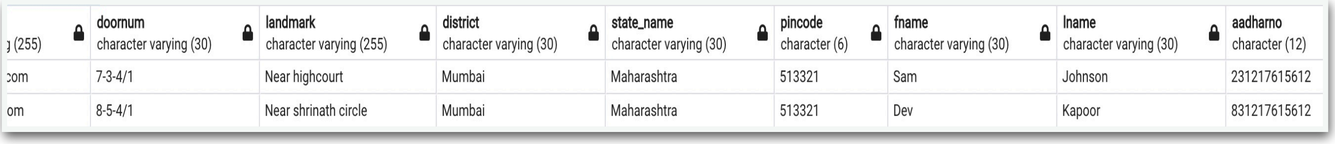
**SQL Statements:**

```
SELECT * FROM Employee NATURAL JOIN User_login
WHERE Person_Role = 1;
```

**Output:**



| userid<br>numeric (7) | employeeid<br>numeric (7) | busid<br>numeric (7) | person_role<br>smallint | gender<br>character (1) | dob<br>date | username<br>character varying (15) | password_user<br>character varying (15) | category<br>smallint | mobileno<br>character varying (15) | email<br>character varying |
|---|---|---|---|---|---|---|---|---|---|---|
| 1242121 | 3167191 | 1534104 | 1 | M | 1995-10... | sam1231 | zysq42 | 1 | 9812312311 | sam123@gmail.c |
| 1242122 | 3167192 | 1534105 | 1 | M | 1985-09... | devil1231 | abcdef | 1 | 6812312321 | dev123@gmail.co |



| ı (255) | doornum<br>character varying (30) | landmark<br>character varying (255) | district<br>character varying (30) | state_name<br>character varying (30) | pincode<br>character (6) | fname<br>character varying (30) | lname<br>character varying (30) | aadharno<br>character (12) |
|---|---|---|---|---|---|---|---|---|
| :om | 7-3-4/1 | Near highcourt | Mumbai | Maharashtra | 513321 | Sam | Johnson | 231217615612 |
| om | 8-5-4/1 | Near shrinath circle | Mumbai | Maharashtra | 513321 | Dev | Kapoor | 831217615612 |

**6.** Count the number of bookings done on each day.

**Ans:**

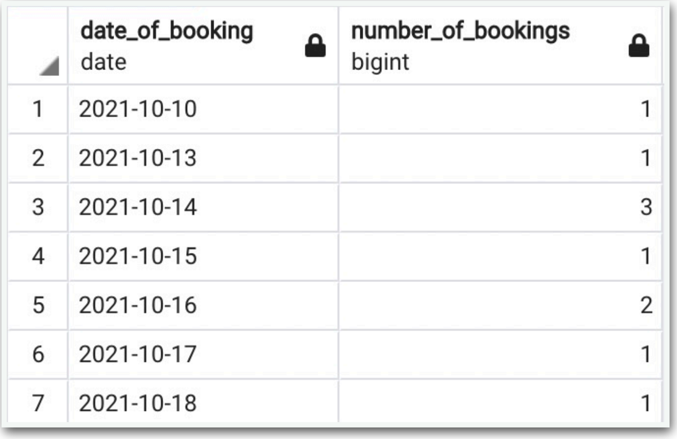
**Relation algebra Expression:**

$(date\_of\_booking)\mathscr{F}_{COUNT(*)}(Booking)$

**SQL Statements:**

```
SELECT date_of_booking, COUNT(*) AS number_of_bookings FROM
Booking

GROUP BY date_of_booking

ORDER BY date_of_booking
```

**Output:**



| | date_of_booking<br>date | number_of_bookings<br>bigint |
|---|---|---|
| 1 | 2021-10-10 | 1 |
| 2 | 2021-10-13 | 1 |
| 3 | 2021-10-14 | 3 |
| 4 | 2021-10-15 | 1 |
| 5 | 2021-10-16 | 2 |
| 6 | 2021-10-17 | 1 |
| 7 | 2021-10-18 | 1 |

**7.** Find the details of route which covers the minimum distance between 2 stations(more than 1 route should be present).

**Ans:**

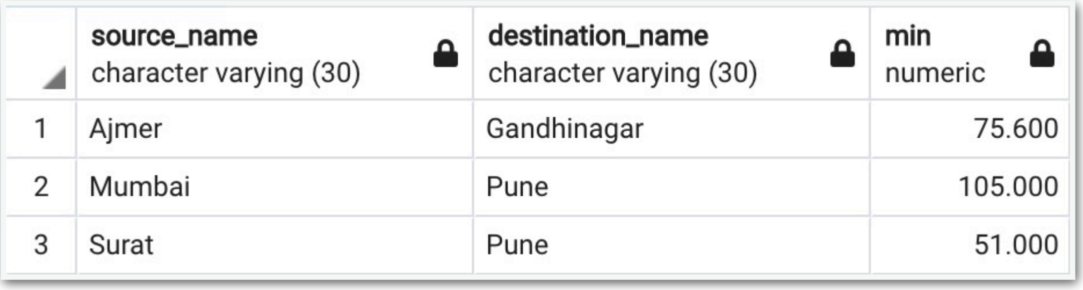
**Relation algebra Expression:**

$\sigma_{COUNT(*)>1}(source\_name, destination\_name)\mathscr{F}_{MIN(Distance)}Route)$

**SQL Statements:**

```
SELECT source_name,destination_name,MIN(distance) FROM route

GROUP BY source_name, destination_name

HAVING COUNT(*) > 1;
```

**Output:**



| | source_name<br>character varying (30) | destination_name<br>character varying (30) | min<br>numeric |
|---|---|---|---|
| 1 | Ajmer | Gandhinagar | 75.600 |
| 2 | Mumbai | Pune | 105.000 |
| 3 | Surat | Pune | 51.000 |

8. List the details of parcels which are to be travelled with a given Bus. (BusID is given).

**Ans:**

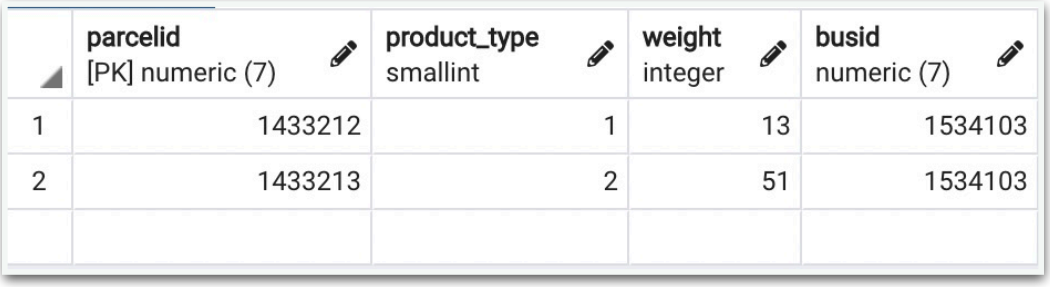
**Relation algebra Expression:**

$\sigma_{(busid=1534103)}(Parcels)$

**SQL Statements:**

```
SELECT * from Parcels

WHERE busid = 1534103;
```

**Output:**



| | parcelid<br>[PK] numeric (7) | product_type<br>smallint | weight<br>integer | busid<br>numeric (7) |
|---|---|---|---|---|
| 1 | 1433212 | 1 | 13 | 1534103 |
| 2 | 1433213 | 2 | 51 | 1534103 |
| | | | | |

**9.** Find the details of all bookings whose payment is done offline.

**Ans:**

**Relation algebra Expression:**

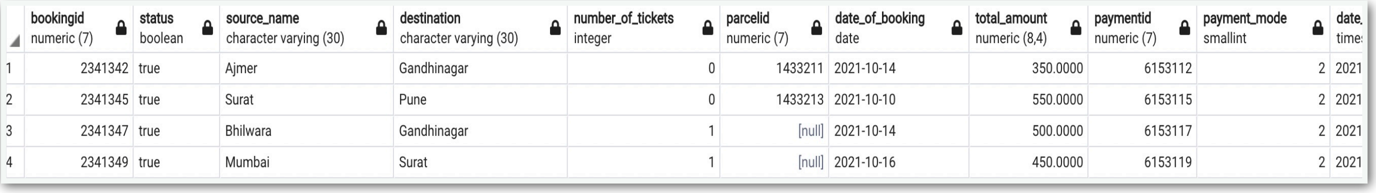
$$\sigma_{payment\_mode=2}(Booking \bowtie Payment)$$

Here, natural join is used.

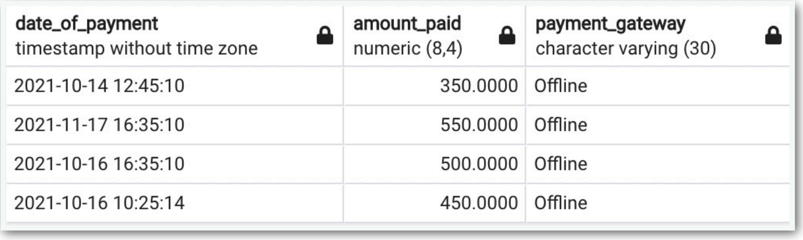
**SQL Statements:**

```
SELECT * from Booking NATURAL JOIN Payment

WHERE payment_mode = 2;
```

**Output:**



| | bookingid<br>numeric (7) | status<br>boolean | source_name<br>character varying (30) | destination<br>character varying (30) | number_of_tickets<br>integer | parcelid<br>numeric (7) | date_of_booking<br>date | total_amount<br>numeric (8,4) | paymentid<br>numeric (7) | payment_mode<br>smallint | date<br>time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2341342 | true | Ajmer | Gandhinagar | 0 | 1433211 | 2021-10-14 | 350.0000 | 6153112 | 2 | 2021 |
| 2 | 2341345 | true | Surat | Pune | 0 | 1433213 | 2021-10-10 | 550.0000 | 6153115 | 2 | 2021 |
| 3 | 2341347 | true | Bhilwara | Gandhinagar | 1 | [null] | 2021-10-14 | 500.0000 | 6153117 | 2 | 2021 |
| 4 | 2341349 | true | Mumbai | Surat | 1 | [null] | 2021-10-16 | 450.0000 | 6153119 | 2 | 2021 |



| date_of_payment<br>timestamp without time zone | amount_paid<br>numeric (8,4) | payment_gateway<br>character varying (30) |
|---|---|---|
| 2021-10-14 12:45:10 | 350.0000 | Offline |
| 2021-11-17 16:35:10 | 550.0000 | Offline |
| 2021-10-16 16:35:10 | 500.0000 | Offline |
| 2021-10-16 10:25:14 | 450.0000 | Offline |

**10.** Fine the details of Drivers and Conductors who are not assigned any Bus.

**Ans:**

**Relation algebra Expression:**

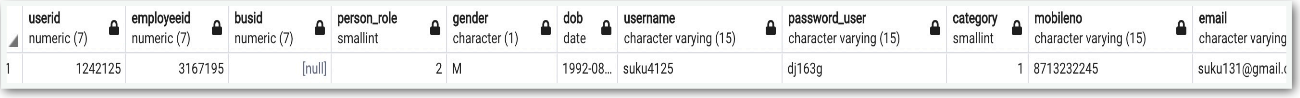
$$\sigma_{(busid=NULL\ AND(person\_role=1\ OR\ person\_role=2))}(Employee \bowtie User\_login)$$

Here, natural join is used.

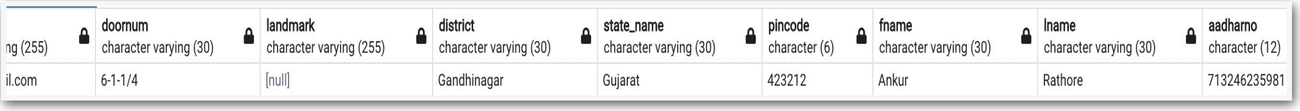
**SQL Statements:**

```
SELECT * FROM Employee NATURAL JOIN User_login

WHERE busid IS null AND (person_role = 1 OR person_role = 2);
```

**Output:**



|  | userid<br>numeric (7) | employeeid<br>numeric (7) | busid<br>numeric (7) | person_role<br>smallint | gender<br>character (1) | dob<br>date | username<br>character varying (15) | password_user<br>character varying (15) | category<br>smallint | mobileno<br>character varying (15) | email<br>character varying |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1242125 | 3167195 | [null] | 2 | M | 1992-08... | suku4125 | dj163g | 1 | 8713232245 | suku131@gmail.( |



| ng (255) | doornum<br>character varying (30) | landmark<br>character varying (255) | district<br>character varying (30) | state_name<br>character varying (30) | pincode<br>character (6) | fname<br>character varying (30) | lname<br>character varying (30) | aadharno<br>character (12) |
|---|---|---|---|---|---|---|---|---|
| il.com | 6-1-1/4 | [null] | Gandhinagar | Gujarat | 423212 | Ankur | Rathore | 713246235981 |

**11.** Find the details of all routes whose starting point is “Mumbai”.

**Ans:**

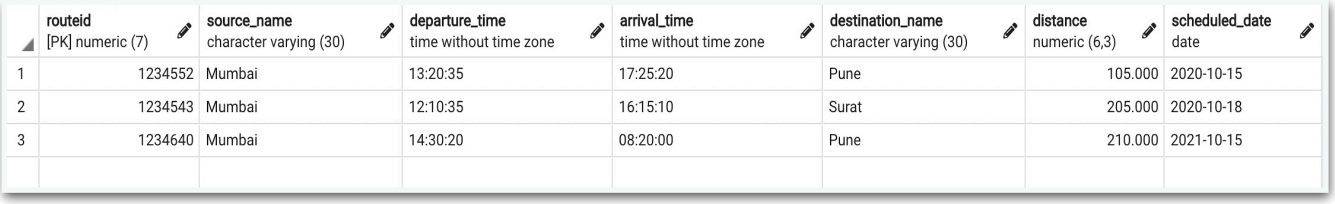
**Relation algebra Expression:**

$$\sigma_{(source\_name='Mumbai')}(Route)$$

**SQL Statements:**

```
SELECT * FROM Route

WHERE source_name = 'Mumbai'
```

**Output:**



| | routeid | source_name | departure_time | arrival_time | destination_name | distance | scheduled_date |
|---|---|---|---|---|---|---|---|
| | [PK] numeric (7) | character varying (30) | time without time zone | time without time zone | character varying (30) | numeric (6,3) | date |
| 1 | 1234552 | Mumbai | 13:20:35 | 17:25:20 | Pune | 105.000 | 2020-10-15 |
| 2 | 1234543 | Mumbai | 12:10:35 | 16:15:10 | Surat | 205.000 | 2020-10-18 |
| 3 | 1234640 | Mumbai | 14:30:20 | 08:20:00 | Pune | 210.000 | 2021-10-15 |

**12.** Count the number of bookings done for each bus.

**Ans:**

**Relation algebra Expression:**

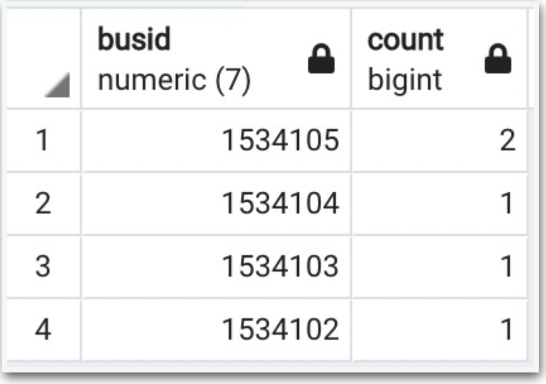
$$(busid)\mathscr{F}_{COUNT(*)}(\sigma_{(bookingid \neq NULL)}(Booking))$$

**SQL Statements:**

```
SELECT busid, COUNT(*) FROM Seat
```

```
WHERE bookingid is not null

GROUP BY busid
```

**Output:**



|  | busid<br>numeric (7) | count<br>bigint |
|---|---|---|
| 1 | 1534105 | 2 |
| 2 | 1534104 | 1 |
| 3 | 1534103 | 1 |
| 4 | 1534102 | 1 |

**13.** Find the details of customers who paid more money than the average money paid by all the customers.

**Ans:**

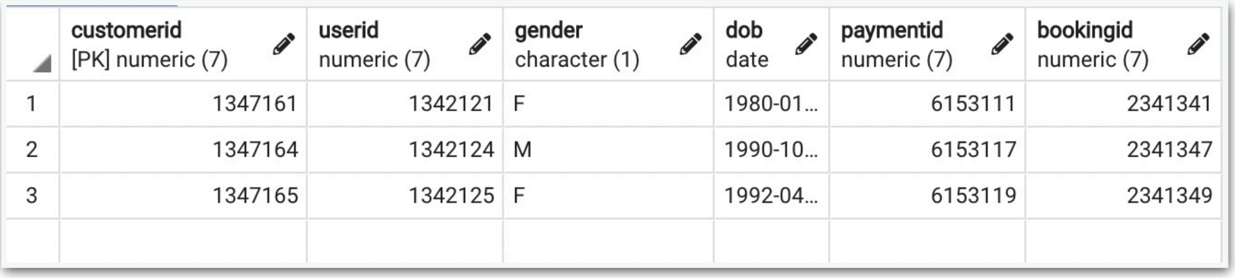
**Relation algebra Expression:**

$$(Customer)\ SEMI-JOIN_{customer.bookingID=b.bookingID}(\sigma_{b.total\_amount>\mathscr{F}_{AVG(total\_amount)}(Booking)}\rho(b, Booking))$$

**SQL Statements:**

```
SELECT * FROM Customer WHERE bookingID IN (

     SELECT bookingid FROM Booking AS b

     WHERE b.total_amount > (SELECT AVG(total_amount) FROM 
Booking)

);
```

**Output:**



| | customerid<br>[PK] numeric (7) | userid<br>numeric (7) | gender<br>character (1) | dob<br>date | paymentid<br>numeric (7) | bookingid<br>numeric (7) |
|---|---|---|---|---|---|---|
| 1 | 1347161 | 1342121 | F | 1980-01... | 6153111 | 2341341 |
| 2 | 1347164 | 1342124 | M | 1990-10... | 6153117 | 2341347 |
| 3 | 1347165 | 1342125 | F | 1992-04... | 6153119 | 2341349 |

**14.** Find the details of buses for which either a driver and conductor are not assigned.

**Ans:**

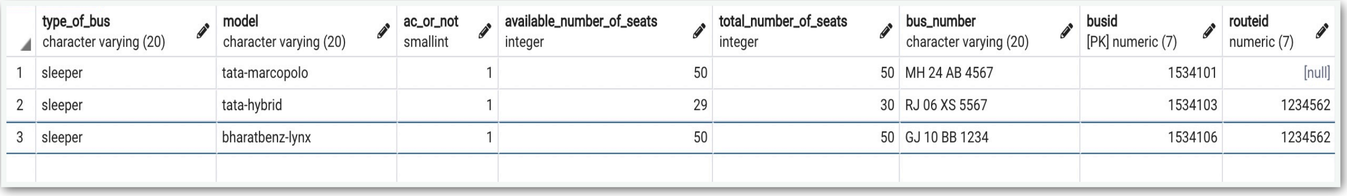
**Relation algebra Expression:**

$Bus - (Bus\ SEMI - JOIN_{(Bus.busid=Employee.busid)}(\pi_{busid}(\sigma_{(busid\ != \ NULL)}Employee)))$

**SQL Statements:**

```
SELECT * FROM bus

WHERE busid

NOT IN (SELECT busid FROM employee WHERE busid IS NOT null);
```

**Output:**



| | type_of_bus<br>character varying (20) | model<br>character varying (20) | ac_or_not<br>smallint | available_number_of_seats<br>integer | total_number_of_seats<br>integer | bus_number<br>character varying (20) | busid<br>[PK] numeric (7) | routeid<br>numeric (7) |
|---|---|---|---|---|---|---|---|---|
| 1 | sleeper | tata-marcopolo | 1 | 50 | 50 | MH 24 AB 4567 | 1534101 | [null] |
| 2 | sleeper | tata-hybrid | 1 | 29 | 30 | RJ 06 XS 5567 | 1534103 | 1234562 |
| 3 | sleeper | bharatbenz-lynx | 1 | 50 | 50 | GJ 10 BB 1234 | 1534106 | 1234562 |

**15.** Find the average age of employees of each employees of each category with age greater than 25.

**Ans:**

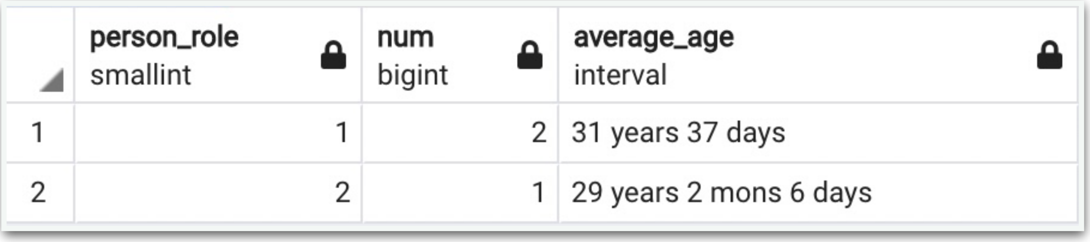
**Relation algebra Expression:**

$$(person\_role)\mathscr{F}_{\rho(num,COUNT(*)),\rho(average\_age,age(dob))}(\sigma_{age(dob>25)}(Employee))$$

**SQL Statements:**

```
SELECT person_role, COUNT(*) AS num, AVG(age(dob)) as
average_age from employee

WHERE age(dob) > '25 years'

GROUP BY person_role

ORDER BY person_role
```

**Output:**



| | person_role<br>smallint | num<br>bigint | average_age<br>interval |
|---|---|---|---|
| 1 | 1 | 2 | 31 years 37 days |
| 2 | 2 | 1 | 29 years 2 mons 6 days |

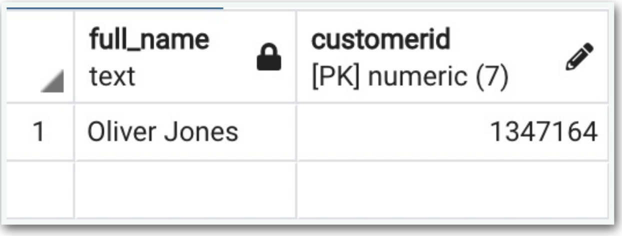
**16.** Find the names of customers with the same gender as the driver of the bus

**Ans:**

**SQL Statements:**

```
SELECT fname || ' ' || lname AS
full_name ,customer_log.customerid FROM user_login JOIN

(SELECT customerid,userid

FROM Customer AS cus

WHERE cus.bookingid IN (

     SELECT bookingid FROM Seat AS s

     WHERE cus.bookingid = s.bookingid AND busid IN(

     SELECT busid FROM Employee AS e WHERE busid IS NOT null
AND cus.gender = e.gender AND person_role = 1)

)) AS customer_log

ON(customer_log.userid = user_login.userid);
```

**Output:**



|   | full_name<br>text | customerid<br>[PK] numeric (7) |
|---|---|---|
| 1 | Oliver Jones | 1347164 |

**17.** Find average number of parcels per each bus.

**Ans:**

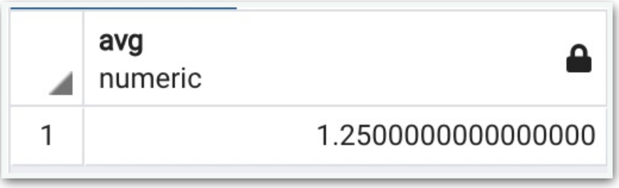
**Relation algebra Expression:**

$\mathscr{F}_{AVG(num)}(b,(busid)\mathscr{F}_{\rho(num,COUNT(*))}(Parcels))$

**SQL Statements:**

```
SELECT AVG(num) FROM

(SELECT COUNT(*) AS num,busid FROM parcels GROUP BY busid) AS
b
```

**Output:**



|   | avg<br>numeric |
|---|---|
| 1 | 1.2500000000000000 |

**18.** Find the details of routes which are not assigned to any buses and distance is greater than 100.

**Ans:**

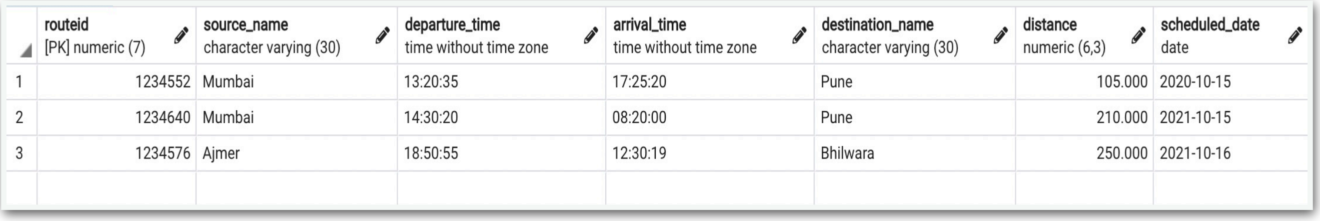
**Relation algebra Expression:**

$\sigma_{distance>100\ AND\ routeid\ != \ (\pi_{routeid}(\sigma_{routeid!=NULL})(Bus))}(Route)$

**SQL Statements:**

```
SELECT * FROM route

WHERE distance > 100 AND routeid

NOT IN (SELECT routeid FROM Bus WHERE routeid IS NOT null);
```

**Output:**



| | routeid<br>[PK] numeric (7) | source_name<br>character varying (30) | departure_time<br>time without time zone | arrival_time<br>time without time zone | destination_name<br>character varying (30) | distance<br>numeric (6,3) | scheduled_date<br>date |
|---|---|---|---|---|---|---|---|
| 1 | 1234552 | Mumbai | 13:20:35 | 17:25:20 | Pune | 105.000 | 2020-10-15 |
| 2 | 1234640 | Mumbai | 14:30:20 | 08:20:00 | Pune | 210.000 | 2021-10-15 |
| 3 | 1234576 | Ajmer | 18:50:55 | 12:30:19 | Bhilwara | 250.000 | 2021-10-16 |

**19.** Count number of damageable and non damageable parcels with weight greater than 15kg.

**Ans:**

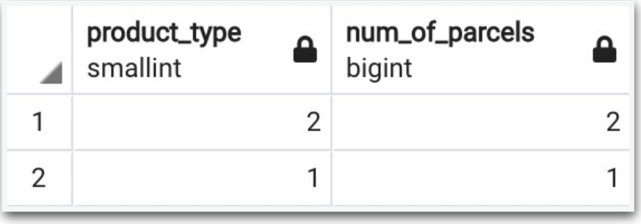
**Relation algebra Expression:**

$$(product\_type)\mathscr{F}_{\rho(num\_of\_parcels,COUNT(*))}(\sigma_{weight>15}(Parcels))$$

**SQL Statements:**

```
SELECT product_type, count(*) as num_of_parcels

FROM parcels

WHERE weight>15

GROUP BY product_type;
```

**Output:**



| | product_type<br>smallint | num_of_parcels<br>bigint |
|---|---|---|
| 1 | 2 | 2 |
| 2 | 1 | 1 |

**20.** List the username and category, phone number of users whose phone number starts with 98.

**Ans:**

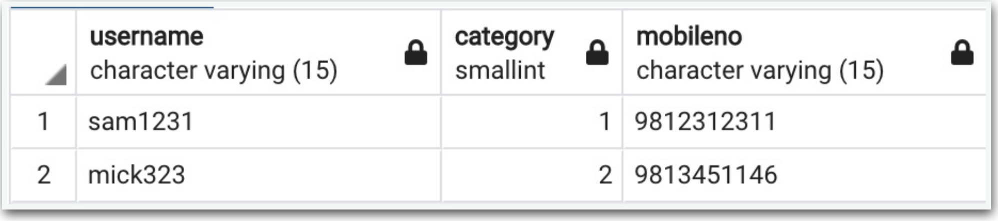
**Relation algebra Expression:**

$\pi_{username,category,MobileNo}(\sigma_{(MobileNo\ LIKE'98\%')}(User\_login))$

**SQL Statements:**

```
SELECT username,category,MobileNo

FROM User_Login

WHERE MobileNo LIKE '98%';
```

**Output:**



| | username<br>character varying (15) | category<br>smallint | mobileno<br>character varying (15) |
|---|---|---|---|
| 1 | sam1231 | 1 | 9812312311 |
| 2 | mick323 | 2 | 9813451146 |

**21.** Find the details of buses which are models of 'tata' company

**Ans:**

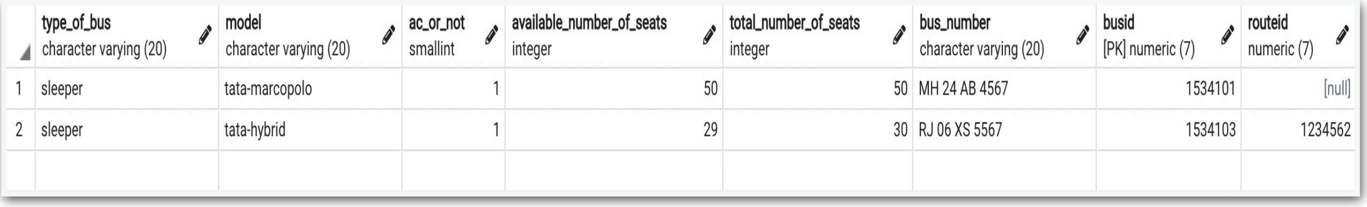
**Relation algebra Expression:**

$\sigma_{(Model\ LIKE'\%tata\%')}(Bus)$

**SQL Statements:**

```
SELECT * FROM Bus

WHERE Model LIKE ‘%tata%';
```

**Output:**



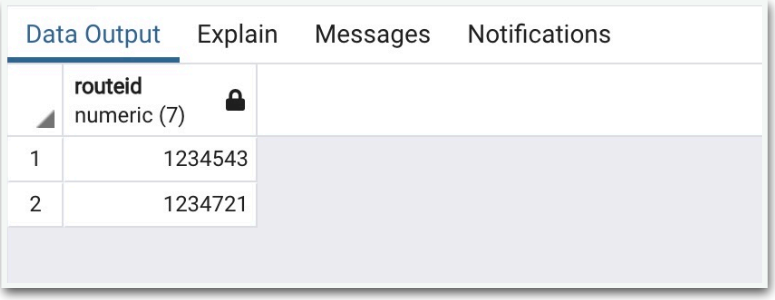
| | type_of_bus<br>character varying (20) | model<br>character varying (20) | ac_or_not<br>smallint | available_number_of_seats<br>integer | total_number_of_seats<br>integer | bus_number<br>character varying (20) | busid<br>[PK] numeric (7) | routeid<br>numeric (7) |
|---|---|---|---|---|---|---|---|---|
| 1 | sleeper | tata-marcopolo | 1 | 50 | 50 | MH 24 AB 4567 | 1534101 | [null] |
| 2 | sleeper | tata-hybrid | 1 | 29 | 30 | RJ 06 XS 5567 | 1534103 | 1234562 |

**22.** Find the routes which pass through all the stations (Division Query)

**SQL Statements:**

```
SELECT distinct routeid FROM connects

WHERE routeid not in ( SELECT routeid FROM (

(SELECT  routeid  ,  stationid  FROM  (SELECT  stationid  FROM
station ) AS P CROSS JOIN

(SELECT DISTINCT routeid FROM connects) AS sp)

EXCEPT

(SELECT routeid , stationid FROM connects) ) AS r );
```

**Output:**



|  | routeid<br>numeric (7) |
|---|---|
| 1 | 1234543 |
| 2 | 1234721 |