- **Queries**

1. **Information about all the couriers sent more than 500km away.**

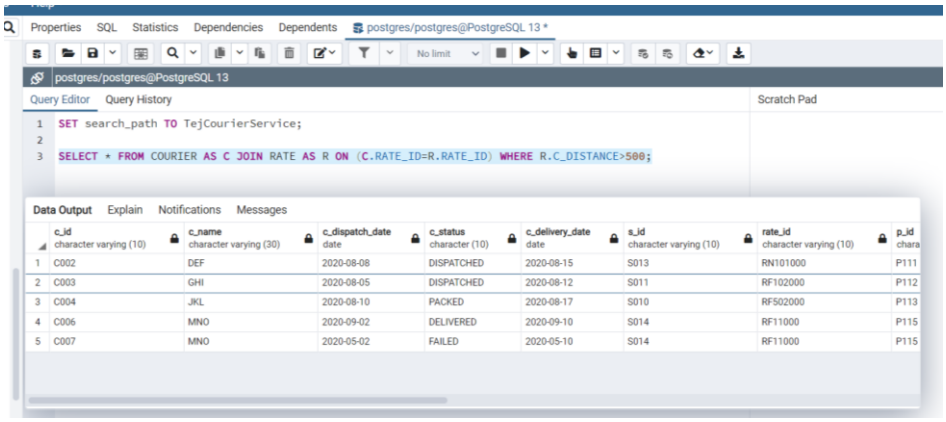
SQL Code:

```
SELECT * FROM COURIER AS C JOIN RATE AS R ON
(C.RATE_ID=R.RATE_ID) WHERE R.C_DISTANCE>500;
```

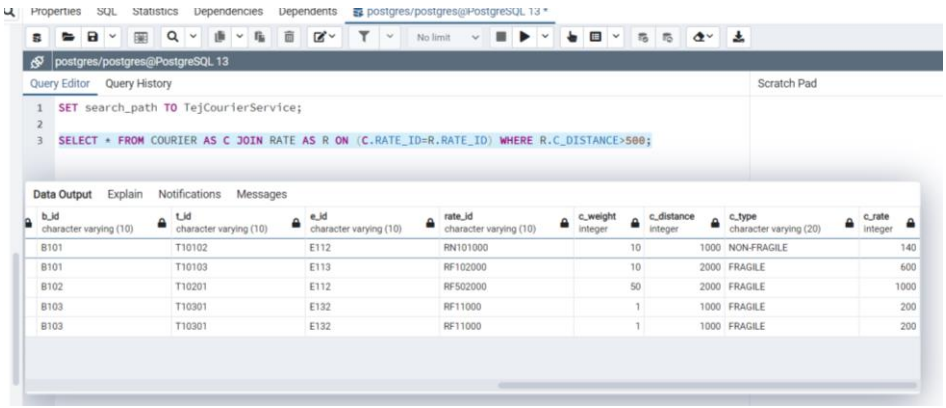
Relational Algebra:

$$\pi_{(c.c\_id)}(\rho(r,\sigma_{<r.c\_distance>500>}rate)\bowtie_{<c.rate\_id=d.rate\_id>}\rho(c,courier))$$

Output:



	c_id character varying (10)	c_name character varying (30)	c_dispatch_date date	c_status character (10)	c_delivery_date date	s_id character varying (10)	rate_id character varying (10)	p_id chara
1	C002	DEF	2020-08-08	DISPATCHED	2020-08-15	S013	RN101000	P111
2	C003	GHI	2020-08-05	DISPATCHED	2020-08-12	S011	RF102000	P112
3	C004	JKL	2020-08-10	PACKED	2020-08-17	S010	RF502000	P113
4	C006	MNO	2020-09-02	DELIVERED	2020-09-10	S014	RF11000	P115
5	C007	MNO	2020-05-02	FAILED	2020-05-10	S014	RF11000	P115



b_id character varying (10)	t_id character varying (10)	e_id character varying (10)	rate_id character varying (10)	c_weight integer	c_distance integer	c_type character varying (20)	c_rate integer
B101	T10102	E112	RN101000	10	1000	NON-FRAGILE	140
B101	T10103	E113	RF102000	10	2000	FRAGILE	600
B102	T10201	E112	RF502000	50	2000	FRAGILE	1000
B103	T10301	E132	RF11000	1	1000	FRAGILE	200
B103	T10301	E132	RF11000	1	1000	FRAGILE	200

2. **Name of the branch which sent the maximum number of couriers in month of August.**

SQL Code:

SELECT B_NAME FROM COURIER JOIN BRANCH ON COURIER.B_ID=BRANCH.B_ID WHERE C_DISPATCH_DATE=(SELECT MAX(C_DISPATCH_DATE) FROM COURIER WHERE C_DISPATCH_DATE BETWEEN '#2020-08-01#' AND '#2020-08-19#');

Relational Algebra:

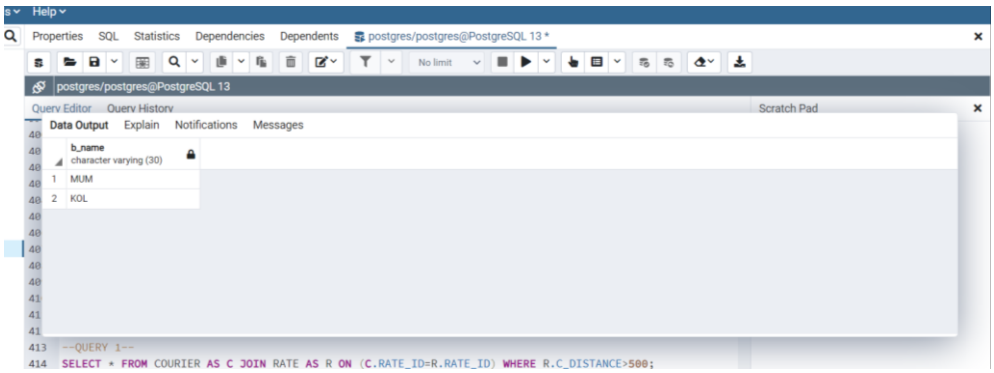
$r1 \leftarrow \sigma_{<c\_dispatch\_date > 2020\text{-}08\text{-}01 \text{ AND } c\_dispatch\_date < 2020\text{-}08\text{-}31>} (courier)$

$r2 \leftarrow \mathscr{F}_{max <c\_dispatch\_date>} (r1)$

$r3 \leftarrow \sigma_{<c\_dispatch\_date = r2>} Courier \bowtie_{<courier.B\_ID=Branch.B\_ID>} Branch$

$result \leftarrow \Pi_{B\_Name} (r3)$

Output:



3. **The average cost of courier in September.**

SQL Code:

SELECT AVG(C_RATE) FROM RATE AS R JOIN COURIER AS C ON (R.RATE_ID = C.RATE_ID) WHERE C.C_STATUS='DELIVERED' AND C_DELIVERY_DATE BETWEEN '#2020-09-01#' AND '#2020-09-30#';

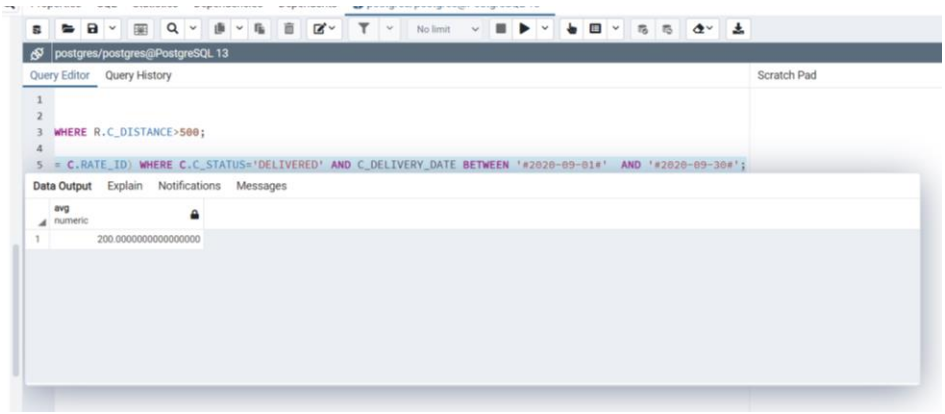
Relational Algebra:

$r1 \leftarrow \rho(c, \sigma_{<c\_delivery\_date>2020\text{-}09\text{-}01 \text{ AND } <c\_ delivery \_date<2020\text{-}09\text{-}31 \text{ AND } c\_status='deliverd'>} courier)$

$r2 \leftarrow \rho(r, rate)$

result ← $\mathscr{F}_{\text{AVG (C\_RATE)}}$ r1 ⋈ $_{\text{<c.rate\_id=r.rate\_id>}}$ r2

Output:



4. **To retrieve the status of couriers that are delivered.**

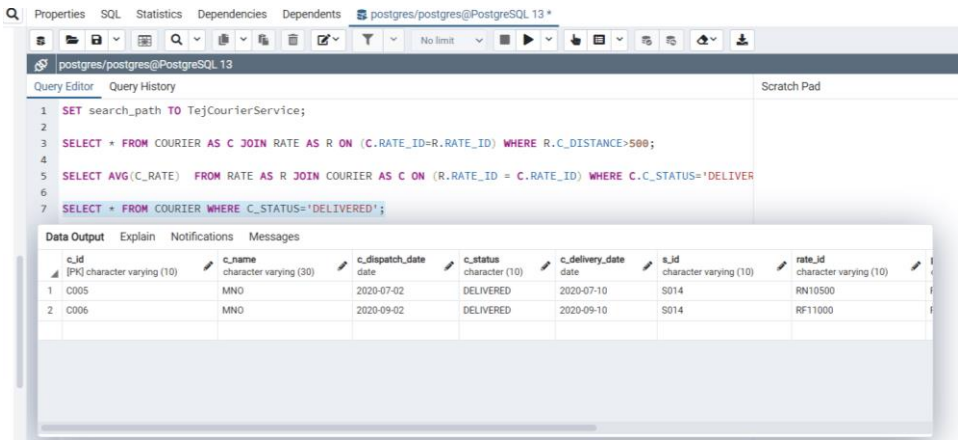
SQL Code:

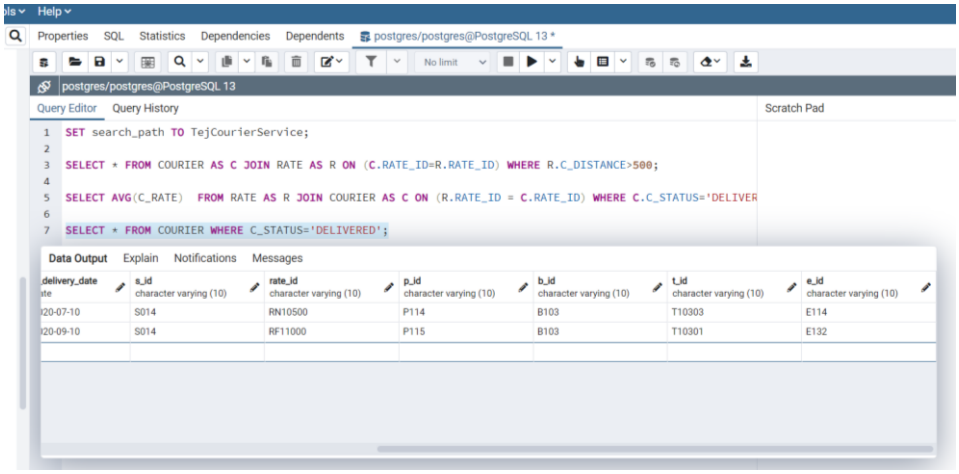
SELECT * FROM COURIER WHERE C_STATUS='DELIVERED';

Relational Algebra:

$\Pi_{\text{<c.c\_id>}}(\rho(c,\sigma_{\text{<c.c\_status="delivered">}}\text{courier}))$

Output:





5. **To retrieve the couriers which have been dispatched.**

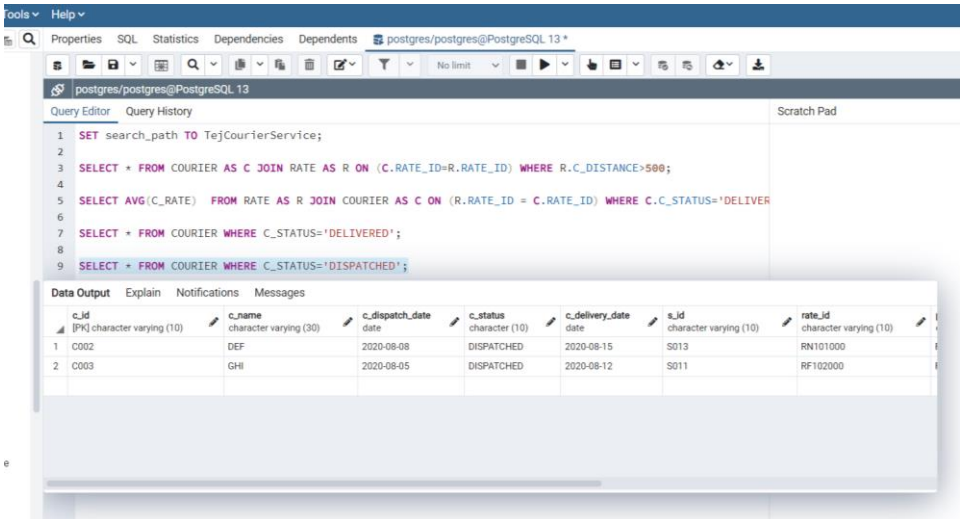
SQL Code:

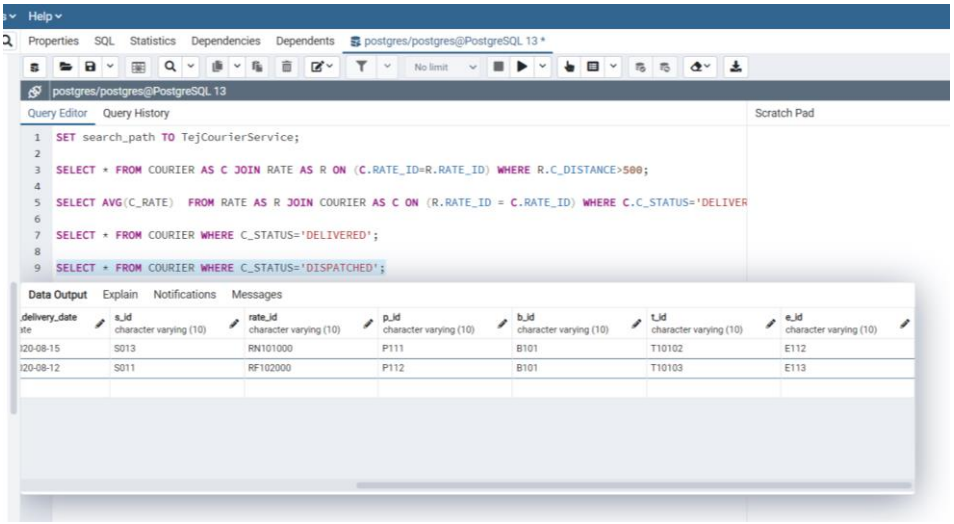
SELECT * FROM COURIER WHERE C_STATUS='DISPATCHED';

Relational Algebra:

$\Pi_{<c.c\_id>}(\rho(c,\sigma_{<c.c\_status="dispatched">}courier))$

Output:





6. **To retrieve the Employee's name and his/her Designation.**

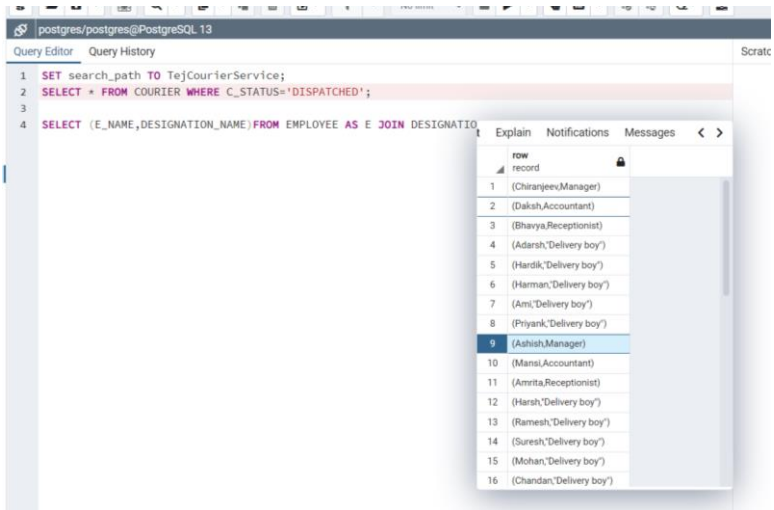
SQL Code:

SELECT (E_NAME, DESIGNATION_NAME) FROM EMPLOYEE AS E JOIN DESIGNATION D ON (E.E_ID=D.E_ID);

Relational Algebra:

$\Pi_{<e\_name,designation\_name>}((employee) \bowtie_{<employee.eid=designation.did>} (designation))$

Output:



7. **To Retrieve payment mode and Transaction Details.**

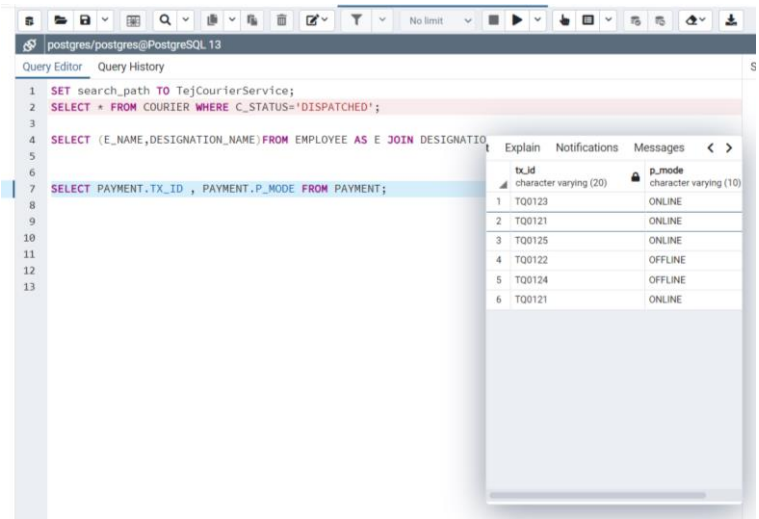
SQL Code:

SELECT PAYMENT.TX_ID , PAYMENT.P_MODE FROM PAYMENT;

Relational Algebra:

$\Pi_{<p\_mode,tx\_id>}(payment)$

Output:



8. **To retrieve info about courier id no C005 (Weight, Distance and type)**

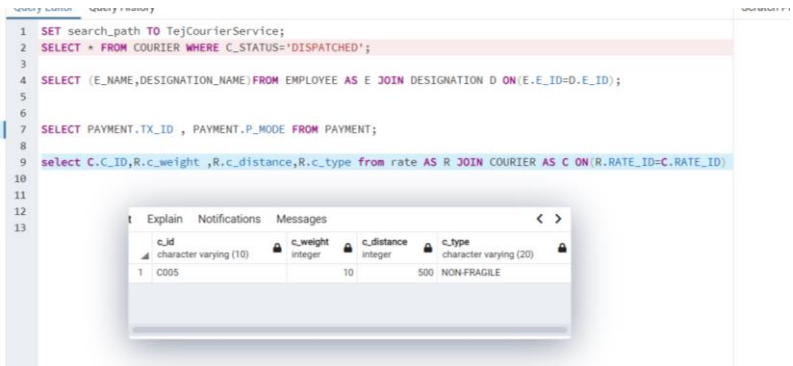
SQL Code:

select C.C_ID,R.c_weight ,R.c_distance,R.c_type from rate AS R JOIN COURIER AS C ON(R.RATE_ID=C.RATE_ID) WHERE C.C_ID='C005';

Relational Algebra:

$\Pi_{<c.c\_id,r.c\_weight,r.c\_didtance,r.c\_type>}\rho(c,(\sigma_{<c.c\_id=C005>}courier)\bowtie_{<r.rate\_id=c.rate\_id>} \rho\ (r,rate))$

Output:



9. Total no. of fragile products sent for each of the year from 2016 to 2020

SQL Code:

```
SELECT COUNT (*) FROM COURIER AS C JOIN RATE AS R ON
(C.RATE_ID=R.RATE_ID) WHERE R.C_TYPE='FRAGILE' AND
C.C_DISPATCH_DATE BETWEEN '#2016-01-01#' AND '#2020-12-01#';
```

Relational Algebra:

$r1 \leftarrow \rho(c, \sigma_{<c\_dispatch\_date>2016\text{-}01\text{-}01 \text{ AND } <c\_dispatch\_date<2016\text{-}12\text{-}31>} \text{curier})$

$r2 \leftarrow \rho(r, \sigma_{<c\_type = \text{'Fragile'}>} \text{rate})$

$\text{result} \leftarrow \mathscr{F}_{\text{COUNT }(*)}\, r1 \bowtie_{<c.rate\_id=r.rate\_id>} r2$

Output:

10. To find payable amount of courier which has 10 kg or less weight.

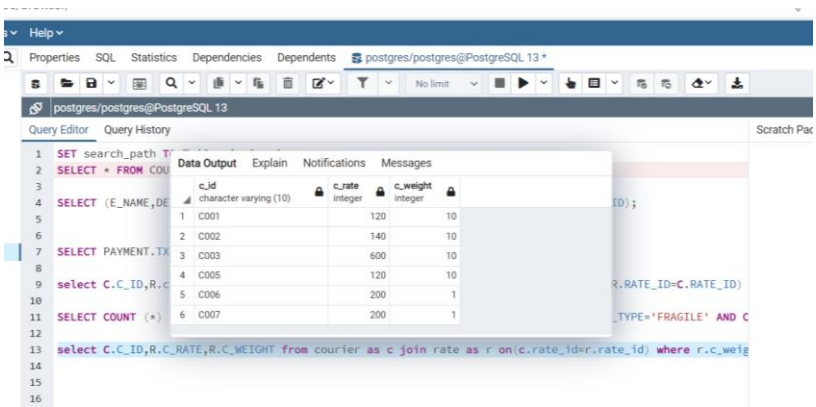
SQL Code:

```
SELECT C.C_ID,R.C_RATE,R.C_WEIGHT from courier as c join rate as r on
(c.rate_id=r.rate_id) where r.c_weight<= '10' ;
```

Relational Algebra:

$\Pi_{\text{c.c\_id, R.c\_rate, R.cweight}} ((\sigma_{<R.cweight <=10>} (\rho(c, \text{couriers})) \bowtie_{<c.rate\_id=r.rate\_id>} \rho(r,\text{rate}))$

Output:



11. Total no. of Employee working in Ahmedabad branch.

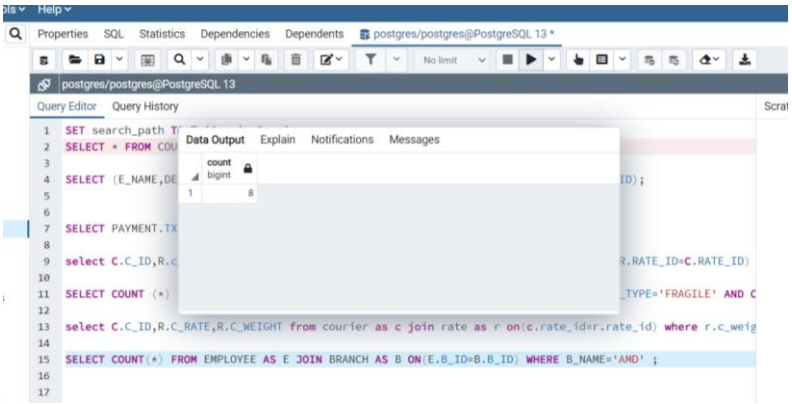
SQL Code:

```
SELECT COUNT(*) FROM EMPLOYEE AS E JOIN BRANCH AS B
ON(E.B_ID=B.B_ID) WHERE B_NAME='AMD' ;
```

Relational Algebra:

$\mathscr{F}_{\text{COUNT }(*)}$ ((σ $_{\text{<B\_name=AMD>}}$ (ρ(E, Employee)) ⋈ $_{\text{<E.b\_id=B.b\_id>}}$ ρ(B, Branch))

Output:



12. To retrieve name of employees who are male and working in branch Mumbai.

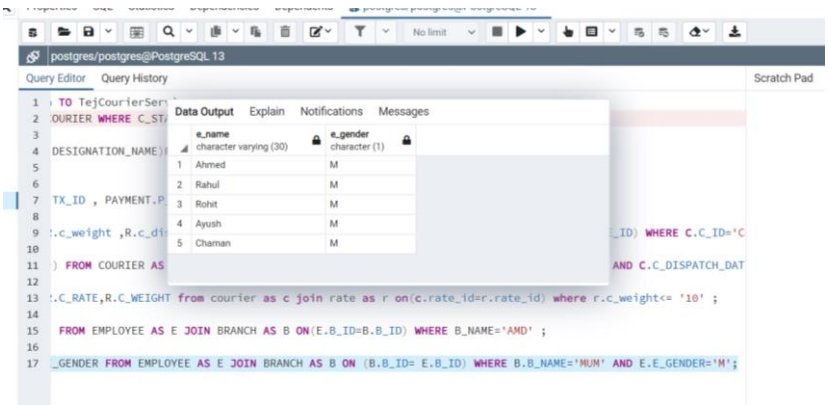
SQL Code:

```
SELECT E_NAME,E_GENDER FROM EMPLOYEE AS E JOIN BRANCH AS B
ON (B.B_ID= E.B_ID) WHERE B.B_NAME='MUM' AND E.E_GENDER='M';
```

Relational Algebra:

Π $_{\text{e\_name, e\_gender}}$ ((σ $_{\text{<B.b\_name=mum AND E.e\_gender=m>}}$ (ρ(e, employee)) ⋈ $_{\text{<B.bid=E.bid>}}$ ρ(B,Branch))

Output:



13. **To select name of customer who has sent courier between 1st August 2020 to 30 September 2020.**

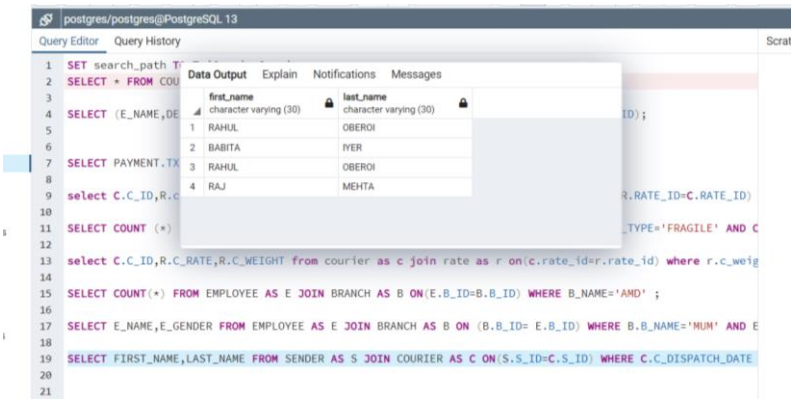
SQL Code:

SELECT FIRST_NAME, LAST_NAME FROM SENDER AS S JOIN COURIER AS C ON(S.S_ID=C.S_ID) WHERE C.C_DISPATCH_DATE BETWEEN '#2020-08-01#' AND '#2020-08-30#';

Relational Algebra:

$\Pi_{<first\_name,last\_name>}\rho(s,(\sigma_{<c.dispatch\_date>2020\text{-}08\text{-}01 \text{ and } c.dispatch\_date<2020\text{-}09\text{-}30>}sender)\bowtie_{<s.s\_id=c.s\_id>}\rho(c,courier)$

Output:



14. **To retrieve name of branch manager of Ahmedabad branch**

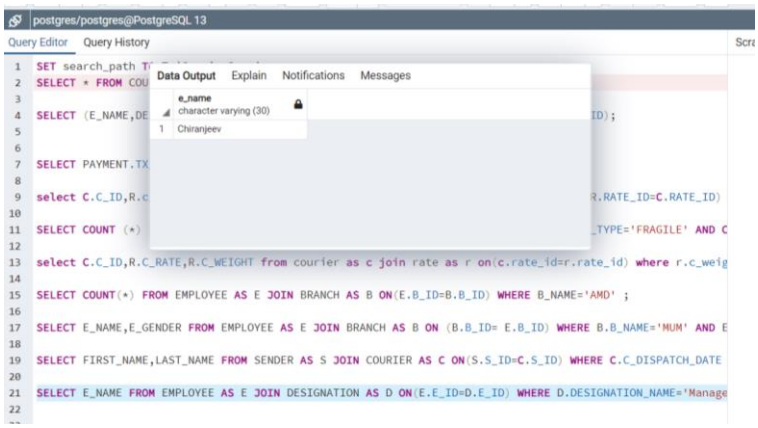
SQL Code:

SELECT E_NAME FROM EMPLOYEE AS E JOIN DESIGNATION AS D ON(E.E_ID=D.E_ID) WHERE D.DESIGNATION_NAME='Manager' AND E.B_ID='B101';

Relational Algebra:

$\Pi_{e\_name}((\sigma_{<D.designation\_name=Manager \text{ AND } E.bid=B101>}(\rho(e, employee)) \bowtie_{<EE.e\_id=D.e\_id>} \rho(D,Designation))$

Output:



15. **Retrieve the name of the employee who have salary more than 20,000 but less than 70,000**

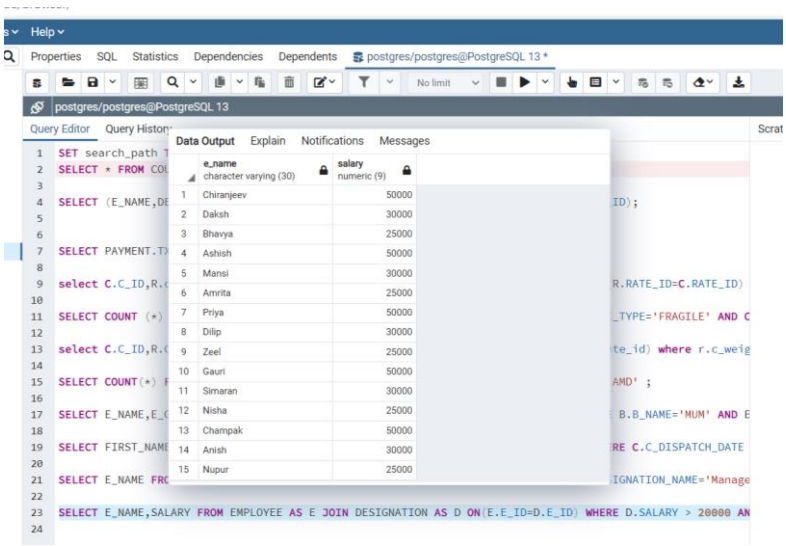
SQL Code:

SELECT E_NAME,SALARY FROM EMPLOYEE AS E JOIN DESIGNATION AS D ON(E.E_ID=D.E_ID) WHERE D.SALARY > 20000 AND D.SALARY < 70000;

Relational Algebra:

$\Pi_{\text{e\_name, salary}}((\sigma_{\text{<D.salary > 20000 AND D.salary < 70000>}}(\rho(\text{e, employee})) \bowtie_{\text{<E.E.e\_id=D.e\_id>}} \rho(\text{D,Designation}))$

Output:



16. Find the courier ID whose Delivery status is FAILED.

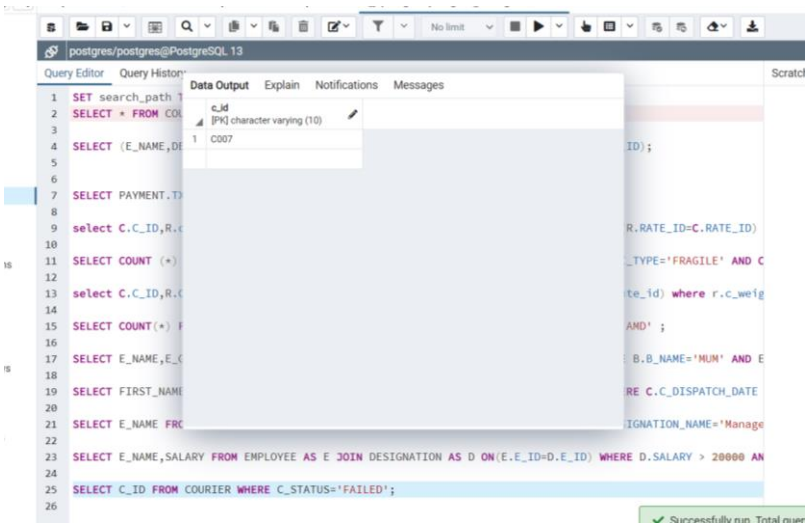
SQL Code:

```
SELECT C_ID FROM COURIER WHERE C_STATUS='FAILED';
```

Relational Algebra:

$\Pi_{<c\_id>}(\sigma_{<c\_status="failed">}(courier))$

Output:



17. Total number of couriers delivered in Maharashtra In Year 2020.

SQL Code:

```
SELECT COUNT(*) FROM COURIER AS C JOIN RECEIVER AS R
ON(R.R_ID=C.R_ID) WHERE R.R_STATE='MAHARASHTRA' AND
C.C_STATUS='DELIVERED' AND C_DELIVERY_DATE BETWEEN '#2020-01-
01#' AND '#2020-12-31#';
```

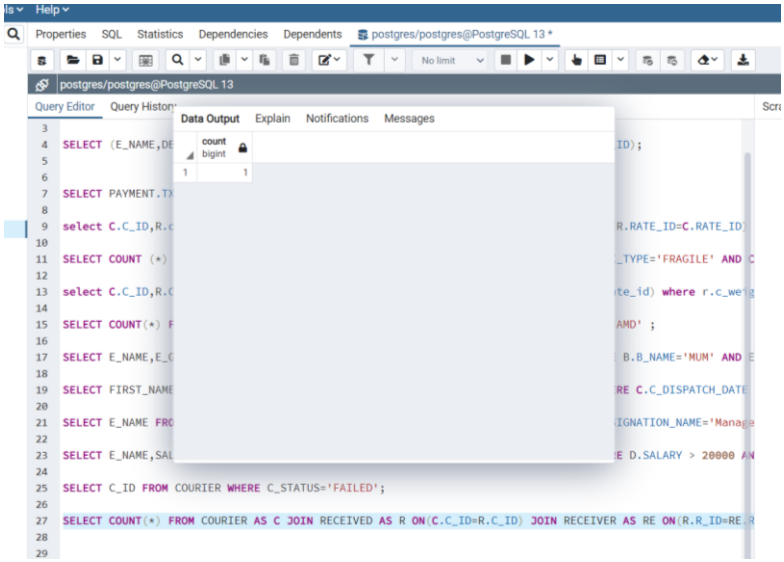
Relational Algebra:

$r1 \leftarrow \rho(c, \sigma_{<c\_delivery\_date>2020\text{-}01\text{-}01 \text{ AND } <c\_delivery\_date<2020\text{-}12\text{-}31>} curier)$

$r2 \leftarrow \rho(r, \sigma_{<r\_state = \text{'Maharashtra'}>} receiver)$

$result \leftarrow \mathscr{F}_{COUNT\ (*)}\ r1 \bowtie_{<c.c\_id=r.c\_id>} r2$

Output:



18. Retrieve courier's sender and receiver name whose STATUS is SUCCESSFUL.

SQL Code:

SELECT DISTINCT S.FIRST_NAME,S.LAST_NAME,R.FIRST_NAME,R.LAST_NAME FROM (SENDER AS S JOIN SENDTO AS SE ON(S.S_ID=SE.S_ID) JOIN RECEIVER AS R ON(SE.R_ID=R.R_ID) join courier as c on(c.s_id=s.s_id)) WHERE c.c_status='DELIVERED'

Relational Algebra:

r1 ← $\rho(c, \sigma_{<c.c\_status='delivered>}$ curier)

r2 ← ρ(r,receiver)

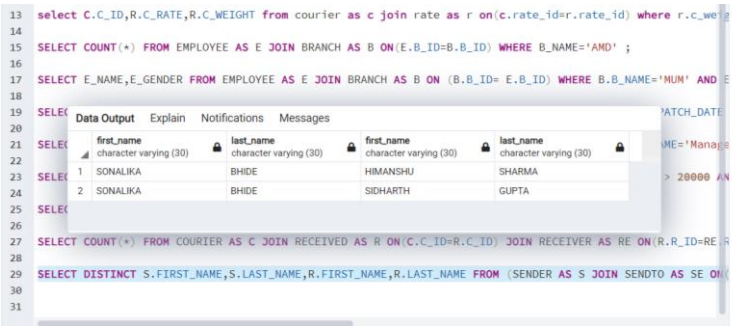
r3 ← ρ(se,sendto)

r4 ← ρ(s,sender)

result ← $\Pi_{<s.first\_name,\ s.last\_name,\ r.first\_name,\ r.last\_name>}$ r1 $\bowtie_{<c.c\_id=s.c\_id>}$ r4 $\bowtie_{<s.s\_id=se.s\_id>}$

r3 $\bowtie_{<se.r\_id=r.r\_id>}$ r2

Output:



19. **Select name of customers whose mode of payment is offline and courier status is in transit.**

SQL Code:

SELECT FIRST_NAME,LAST_NAME FROM SENDER AS S JOIN COURIER AS C ON(S.S_ID=C.S_ID) JOIN PAYMENT AS P ON(C.P_ID=P.P_ID) WHERE P.P_MODE='OFFLINE' AND C.C_STATUS='DISPATCHED';

Relational Algebra:

$\Pi_{<s.first\_name,\ s.last\_name>}\rho(s,(\sigma_{<p.p\_mode="offline",c.c\_status="dispatched">}sender)\bowtie_{<s.sid=c.sid>}\rho(c,courier)\bowtie_{<c.p\_id=p.p\_id>}\rho(p,payment))$

Output:

20. **Select all the courier whose cost is greater than Rs 200 and less than Rs 500 & mode of payment is online.**

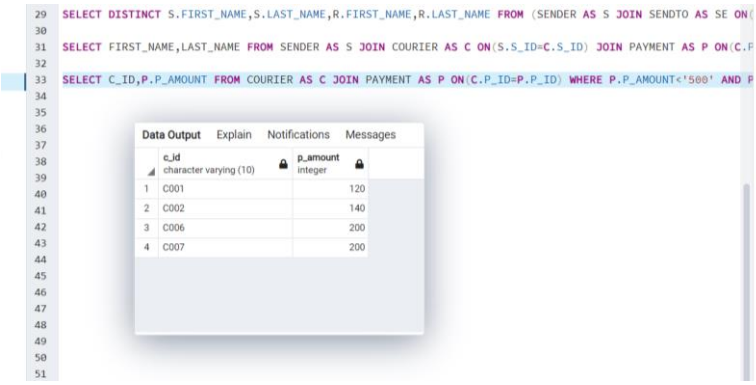
SQL Code:

SELECT C_ID, P.P_AMOUNT FROM COURIER AS C JOIN PAYMENT AS P ON(C.P_ID=P.P_ID) WHERE P.P_AMOUNT<'500' AND P.P_AMOUNT >'100' AND P.P_MODE='ONLINE';

Relational Algebra:

$\Pi_{<.c\_id,p.p\_id>}(p,(\sigma_{<p.p\_amount>200 \text{ AND } p.p\_amount<500}$
$p.p\_mode="online">payment)\bowtie_{<c.p\_id=p.p\_id>}(c,courier))$

Output:



```
SELECT DISTINCT S.FIRST_NAME,S.LAST_NAME,R.FIRST_NAME,R.LAST_NAME FROM (SENDER AS S JOIN SENDTO AS SE ON(

SELECT FIRST_NAME,LAST_NAME FROM SENDER AS S JOIN COURIER AS C ON(S.S_ID=C.S_ID) JOIN PAYMENT AS P ON(C.P

SELECT C_ID,P.P_AMOUNT FROM COURIER AS C JOIN PAYMENT AS P ON(C.P_ID=P.P_ID) WHERE P.P_AMOUNT<'500' AND P
```

Data Output | Explain | Notifications | Messages

	c_id character varying (10)	p_amount integer
1	C001	120
2	C002	140
3	C006	200
4	C007	200